

The Language List - Version 2.4, January 23, 1995

Collected information on about 2350 computer languages, past and present.

Available online as:

[http://cui\\_www.unige.ch/langlist](http://cui_www.unige.ch/langlist)

<ftp://wuarchive.wustl.edu/doc/misc/lang-list.txt>

Maintained by:

Bill Kinnersley  
Computer Science Department  
University of Kansas  
Lawrence, KS 66045

[billk@cs.ukans.edu](mailto:billk@cs.ukans.edu)

Started Mar 7, 1991 by Tom Rombouts <[tomr@ashtate.A-T.com](mailto:tomr@ashtate.A-T.com)>

This document is intended to become one of the longest lists of computer programming languages ever assembled (or compiled). Its purpose is not to be a definitive scholarly work, but rather to collect and provide the best information that we can in a timely fashion. Its accuracy and completeness depends on the readers of Usenet, so if you know about something that should be added, please help us out. Hundreds of netters have already contributed to this effort. We hope that this list will continue to evolve as a useful resource available to everyone on the net with an interest in programming languages.

"YOU LEFT OUT LANGUAGE \_\_\_\_!"

If you have information about a language that is not on this list, please e-mail the relevant details to the current maintainer, as shown above. If you can cite a published reference to the language, that will help in determining authenticity.

#### What Languages Should Be Included

The "Published" Rule - A language should be "published" to be included in this list. There is no precise criterion here, but for example a language devised solely for the compiler course you're taking doesn't count. Even a language that is the topic of a PhD thesis might not necessarily be included. But if material on that language was published in a technical journal or report, or if it formed the basis for additional research, the language belongs in this list. A language does NOT have to be implemented (actually running on at least one computer) to be included. Many languages appearing in the ACM SIGPLAN Notices fall into this category.

In general when there's any doubt, an entry will be included. Making the list as complete as possible necessarily means there will be a large number of obscure entries. To compensate for this "clutter" effect, more widespread languages such as C or FORTRAN should have longer entries.

For historical completeness roughly 200 early pre-1959 "automatic

programming systems" were included, based on a list from CACM 2(5):16, May 1959. It can be argued that many of these are not really programming languages as the term is used today. We've also included some formalisms which are clearly not meant to be used as a source language for writing programs: metalanguages such as BNF, intermediate languages such as P-Code, and computational models such as Linda.

## Dialects, Variants, Versions and Implementations

Computer languages evolve, and are related to one another in rather complex ways. Almost every language can be regarded as an improved version of something else. Sometimes it's hard to know where to draw the line and say "this is a separate language". Taking LISP as an example, what started out as a single language has evolved into a large family. Dialects (such as Scheme and Common LISP) have major differences and are certainly considered by their users to be distinct languages. Variants (such as Kyoto Common LISP and Allegro CL) are primarily intended to be the same, but have certain features which make them incompatible. Implementations are designed to run on particular machines or operating systems and will usually have special features added. Inevitably a series of revisions will be issued, causing further small changes in the language. It has even been suggested that if command line options are present, each choice of options could be considered a distinct language!

A language's name by itself is not always an accurate guide to its identity. Sometimes a language will undergo significant evolution without any official change in name (e.g. SETL2 has done this). Sometimes just the name will change (IAL to ALGOL to ALGOL 58). And occasionally a name has been used for several distinct languages (e.g. Vulcan).

It may also be debatable what is "in" a language and what is not. For example SML is defined in stages: a "core syntax" surrounded by a standard set of extensions. Technically that makes it two separate languages. Other languages have purposely omitted essential features like I/O from their definition because they were never intended to be used without a standard library (C) or interface (Smalltalk-80), or because they constitute the command language for a particular product or system. Still other languages are by their very nature extensible, and the large number of macro packages built on TeX for example could be called an endless list of separate "languages".

Brand names - You might wonder why we do include a number of commercial products such as Turbo Pascal. Usually these items offer non-standard extensions to the base language. This has been particularly true in the varieties of BASIC and Prolog. But also one could argue that in a strict sense Microsoft C and Turbo C for example are distinct languages.

Another reason for including entries of this type is that many languages are proprietary, appearing only in a certain product. Such languages may be distinctive and interesting and deserve to be here. On the other hand we don't want the list to become a catalog of commercial programming products.

## What Each Entry Should Contain

Name: An explanation of the language name, which in perhaps 80% of the

cases is some form of acronym or abbreviation.

**Date of origin:** The year when a language first appeared. Since the design, implementation and distribution of a language can take place over a period of several years, such dates are often approximate. Any language that has an ANSI, ISO or BSI standard should include the date approved. For specific brands such as Turbo Pascal the release dates of each version can be listed.

**Reference:** At least one reference work on the language, as definitive or as official as possible.

**Availability:** ftp site, commercial source or publisher, contacts for further information.

"See also:" Related languages or terms that may also be of interest.

Any material marked with brackets "[]" is doubtful and may be considered a request for further information.

**Editorial Comments -** What constitutes a good language has often become the subject of intense debate. We've tried to avoid adding to this by avoiding any remarks that are subjective, such as calling a language "powerful". Nevertheless some comments might still be construed this way. For instance saying that Pascal is "ALGOL-like" could offend both some ALGOL and some Pascal users. Also, some questions of historical origin are not universally agreed upon.

**Classification -** It's been suggested that the languages in this list should be arranged into categories, but to do so would be extremely difficult. For every classification scheme there will be a large proportion of languages that do not fit. The languages are therefore listed alphabetically, and in fact we think that this is the most useful organization. You'll find that the following categories have been referred to in the list, but we must emphasize that most languages are not purely one or the other, and we are really categorizing language features.

#### Imperative language

A language which operates by a sequence of commands that change the value of data elements. Typified by assignments and iteration.

#### Declarative language

A language which operates by making descriptive statements about data and relations between data. The algorithm is hidden in the semantics of the language. This category encompasses both applicative and logic languages. Examples of declarative features are set comprehensions and pattern-matching statements.

#### Procedural language

A language which states how to compute the result of a given problem. Encompasses both imperative and functional languages.

#### Applicative language

A language that operates by application of functions to values, with no side effects. A functional language in the broad sense.

#### Functional language

In the narrow sense, a functional language is one that operates by use of higher-order functions, building operators that manipulate functions directly without ever appearing to manipulate data. Example: FP.

#### Definitional language

An applicative language containing assignments interpreted as definitions. Example: Lucid.

#### Single Assignment language

An applicative language using assignments with the convention that a variable may appear on the left side of an assignment only once within the portion of the program in which it is active.

#### Dataflow language

A language suitable for use on a dataflow architecture. Necessary properties include freedom from side effects, and the equivalence of scheduling constraints with data dependencies. Examples: Val, Id, SISAL, Lucid.

#### Logic language

A logic language deals with predicates or relationships  $p(X,Y)$ . A program consists of a set of Horn clauses which may be:

facts -  $p(X,Y)$  is true

rules -  $p$  is true if  $q_1$  and  $q_2$  and ... $q_n$  are true

queries - is  $g_1$  and  $g_2$  and ... $g_n$  true? ( $g_i$ 's are the goals.)

Further clauses are inferred using resolution. One clause is selected containing  $p$  as an assumption, another containing  $p$  as a consequence, and  $p$  is eliminated between them. If the two  $p$ 's have different arguments they must be unified, using the substitution with the fewest constraints that makes them the same.

Logic languages try alternative resolutions for each goal in succession, backtracking in a search for a common solution. OR-parallel languages try alternative resolutions in parallel, while AND-parallel languages try to satisfy several goals in parallel.

#### Constraint language

A language in which a problem is specified and solved by a series of constraining relationships.

#### Object-Oriented language

A language in which data and the functions which access it are treated as a unit.

#### Concurrent language

A concurrent language describes programs that may be executed in parallel. This may be either  
multiprogramming: sharing one processor  
multiprocessing: separate processors sharing one memory  
distributed

Concurrent languages differ in the way that processes are created:  
coroutines - control is explicitly transferred - Simula I, SL5, BLISS, Modula-2.  
fork/join - PL/I, Mesa  
cobegin/coend - ALGOL 68, CSP, Edison, Argus  
process declarations - DP, SR, Concurrent Pascal, Modula, PLITS, Ada

and the ways in which processes interact:  
semaphores - ALGOL 68  
conditional critical regions - Edison, DP, Argus  
monitors - Concurrent Pascal, Modula  
message passing - CSP, PLITS, Gypsy, Actors  
remote procedure calls - DP, \*Mod  
rendezvous - Ada, SR  
atomic transactions - Argus

Fourth generation language (4GL's)

A very high-level language. May use natural English or visual constructs. Algorithms or data structures may be chosen by the compiler.

Query language

An interface to a database.

Specification language

A formalism for expressing a hardware or software design.

Assembly language

A symbolic representation of the machine language of a specific computer.

Intermediate language

A language used as an intermediate stage in compilation. May be either text or binary.

Metalanguage

A language used for formal description of another language.

\* \* \* \* \*

2.PAK - AI language with coroutines. "The 2.PAK Language: Goals and Description", L.F. Melli, Proc IJCAI 1975.

20-GATE - Carnegie, ca 1965. Algebraic language for the G-20.

3-LISP - Brian Smith. A procedurally reflective dialect of LISP which uses an infinite tower of interpreters. "The Implementation of Procedurally Reflective Languages", J. des Rivi et al, ACM J Lisp and Functional Programming, pp.331-347 (1984).

473L Query - English-like query language for Air Force 473L system. Sammet 1969, p.665. "Headquarters USAF Command and Control System Query Language", Info Sys Sci, Proc 2nd Congress, Spartan Books 1965, pp.57-76.

9PAC - 709 PACkage. 1959. Report generator for IBM 7090. Sammet 1969, p.314. "IBM 7090 Prog Sys, SHARE 7090 9PAC Part I: Intro and Gen Princs", IBM J28-6166, White Plains, 1961.

\*LISP - ("StarLISP") Cliff Lasser, Jeff Mincy, J.P. Massar, Thinking Machines Corp. A data-parallel extension of Common LISP for the Connection Machine, uses 'pvars'. "The Essential \*LISP Manual", TM Corp 1986.  
ftp://think.com/public/cm/starlisp/\*  
info: customer-support@think.com  
documentation-order@think.com

\*MOD - ("StarMOD") Concurrent language combining the modules of Modula and the communications of Distributed Processes. "\*MOD - A Language for Distributed Programming", R.P. Cook, IEEE Trans Soft Eng SE-6(6):563-571 (Nov 1980).

A# - Object-oriented and functional, a separable component of Version 2 of the AXIOM computer algebra system. Both types and functions are first class values. Designed for compilation to efficient machine code.

A+ - Dialect of APL used at Morgan-Stanley.

A0 or A-0 - Grace Hopper's team at Remington Rand, 1952, for the UNIVAC I or II. Possibly the first compiler ever. Later internal versions: A-1, A-2 ("The A-2 Compiler System", Rem Rand, 1955), A-3, AT-3. AT-3 was released as MATH-MATIC. Sammet 1969, p.12.

AACC - Language for building finite state automata. [?]

AADL - Axiomatic Architecture Description Language. "AADL: A Net-Based Specification Method for Computer Architecture Design", W. Damm et al in Languages for Parallel Architectures, J.W. deBakker ed, Wiley 1989.

ABC -

1. Leo Geurts, Lambert Meertens, Steven Pemberton. Simple interactive language designed for quick easy programming. Includes a programming environment with syntax-directed editing, suggestions, persistent variables and multiple workspaces and infinite precision arithmetic. "An Alternative Simple Language and Environment for PC's", S. Pemberton, IEEE Software 4(1):56-64 (Jan 1987). "The ABC Programmer's Manual", Leo Geurts et al, P-H 1989.

ftp://ftp.uu.net/languages/abc/\*  
info: abc@cwil.nl  
list: abc-list@cwil.nl maintained by Steven Pemberton <abc-list-request@cwil.nl>.

2. (A="argument",B="basic value",C=?). Intermediate language for the ABC abstract machine for implementation of functional languages, similar to the spineless tagless G-machine. "Compiling Clean to Abstract ABC-Machine Code", J.E.W. Smetsers, TR 89-20, U Nijmegen 1989. "The ABC-Machine: A Sequential Stack-Based Abstract Machine for Graph Rewriting", P. Koopman et al, TR 88-1, U Nijmegen 1988. Functional Programming and Parallel Graph Rewriting, Rinus Plasmeijer et al, A-W 1993, ISBN 0201416638.

ABC ALGOL - An extension of ALGOL 60 with arbitrary data structures and user-defined operators, for symbolic math. "ABC Algol, A Portable Language for Formula Manipulation Systems", R.P. van de Riet, Amsterdam Math Centrum 1973.

ABCL/1 - An Object-Based Concurrent Language. Yonezawa, U Tokyo 1986. Language for the ABCL concurrent (MIMD) system. Asynchronous message passing to objects. Implementations in KCL and Symbolics LISP available from the author. "ABCL: An Object-Oriented Concurrent System", A. Yonezawa ed, MIT Press 1990.

ftp://camille.is.s.u-tokyo.ac.jp/pub/acbl1/\*  
info: matsu@is.s.u-tokyo.ac.jp

ABCL/c+ - Concurrent object-oriented language, an extension of ABCL/1 based on C. "An Implementation of An Operating System Kernel using Concurrent Object Oriented Language ABCL/c+", N. Doi et al in ECOOP '88, S. Gjessing et al eds, LNCS 322, Springer 1988.

ABCL/R - Yonezawa, Tokyo Inst Tech 1988. A reflective subset of ABCL/1, written in ABCL/1. "Reflection in an Object-Oriented Concurrent Language", T. Watanabe et al, SIGPLAN Notices 23(11):306-315 (Nov 1988).  
ftp://camille.is.s.u-tokyo.ac.jp/pub/abclr/\*

ABCL/R2 - Yonezawa et al, Tokyo Inst Tech 1992. A reflective concurrent object-oriented language, based on Hybrid Group Architecture. Provides almost all the functionality of ABCL/1. Written in Common LISP.  
ftp://camille.is.s.u-tokyo.ac.jp/pub/abclr2/\*

Abel - HP Labs. Strongly-typed object-oriented language with contravariant semantics. Inherited interfaces are not required to be subtypes.  
info: Walter Hill <whill@netcom.com>

ABLE - Simple language for accountants. "ABLE, The Accounting Language, Programming and Reference Manual," Evansville Data Proc Center, Evansville, IN, Mar 1975. Listed in SIGPLAN Notices 13(11):56 (Nov 1978).

ABSET - U Aberdeen. Early declarative language. "ABSET: A Programming Language Based on Sets", E.W. Elcock et al, Mach Intell 4, Edinburgh U Press, 1969, pp.467-492.

ABSYS 1 - U Aberdeen. Early declarative language, anticipated a number of features of Prolog. "ABSYS 1: An Incremental Compiler for Assertions", J.M. Foster et al, Mach Intell 4, Edinburgh U Press, 1969, pp.423-429.

Accent - Very high level interpreted language with strings, tables, etc. Strongly typed, remote function calls. CaseWare Inc.

Access - English-like query language used in the Pick OS.

ACL - A Coroutine Language. A Pascal-based implementation of coroutines. "Coroutines", C.D. Marlin, LNCS 95, Springer 1980.

ACOM - Early system on IBM 705. Listed in CACM 2(5):16 (May 1959).

ACOS - BBS language for PRODOS 8 on Apple ][. MacOS is a hacked version of ACOS.

ACP - Algebra of Communicating Processes. "Algebra of Communicating Processes with Abstraction", J.A. Bergstra & J.W. Klop, Theor Comp Sci 37(1):77-121 (1985). (compare CCS).

ACT++ - Concurrent extension of C++ based on actors. "ACT++: Building a Concurrent C++ With Actors", D.G. Kafura TR89-18, VPI, 1989.

ACT ONE - Specification language. "An Algebraic Specification Language with Two Levels of Semantics", H. Ehrig et al, Tech U Berlin 83-03 Feb 1983.

Act1 - An actor language, descendant of Plasma. "Concurrent Object Oriented Programming in Act1", H. Lieberman in Object Oriented Concurrent Programming, A. Yonezawa et al eds, MIT Press 1987.

Act2 - An actor language. "Issues in the Design of Act2", D. Theriault, TR728, MIT AI Lab, June 1983.

Act3 - High-level actor language, descendant of Act2. Provides support for automatic generation of customers and for delegation and inheritance. "Linguistic Support of Receptionists for Shared Resources", C. Hewitt et al in Seminar on Concurrency, S.D. Brookes et al eds, LNCS 197, Springer 1985, pp. 330-359.

Actalk - Briot, 1989. Smalltalk-based actor language. "Actalk: A Testbed for Classifying and Designing Actor Languages in the Smalltalk-80 Environment", J-P. Briot, Proc ECOOP '89, pp.109-129.

Active Language I - Early interactive math, for XDS 930 at UC Berkeley. "Active Language I", R. de Vogelaere in ACM Symposium on Interactive Systems for Experimental Applied Mathematics, M. Klarer et al eds, A-P 1968.

Actor - Charles Duff, Whitewater Group, ca 1986. Object-oriented language for Microsoft Windows. Pascal/C-like syntax. Uses a token-threaded interpreter. Early binding is an option. "Actor Does More than Windows", E.R. Tello, Dr Dobb's J 13(1):114-125 (Jan 1988). Version 4. Now supported by Genesis Development Systems, (800) OKACTOR.  
list:ACTOR-L@hearn.nic.surfnet.nl  
<http://www.cs.rulimburg.nl/~plugge/actor-1>  
<ftp://bommel.cs.rulimburg.nl:/pub/actor-1>

Actors - C. Hewitt. A model for concurrency. "Laws for Communicating Parallel Processes", C. Hewitt et al, IFIP 77, pp. 987-992, N-H 1977. "ACTORS: A Model of Concurrent Computation in Distributed Systems", Gul A. Agha <agha@cs.uiuc.edu>, Cambridge Press, MA, 1986.

Actra - An exemplar-based Smalltalk. LaLonde et al, OOPSLA '86.

Actus - Pascal with parallel extensions, similar to the earlier Glypnir. Parallel constants, index sets. Descendants include Parallel Pascal, Vector C, and CMU's recent language PIE. "A Language for Array and Vector Processors," R.H. Perrott, ACM TOPLAS 1(2):177-195 (Oct 1979).

Ada - (named for Ada Lovelace (1811-1852), arguably the world's first computer programmer.) Jean Ichbiah's team at CII Honeywell, for the U.S. Department of Defense, 1979. Ada is a large, complex block-structured language aimed primarily at embedded computer applications. It has facilities for real-time response, concurrency, hardware access, and reliable run-time error handling. In support of large-scale software engineering, it emphasizes strong typing, data abstraction and encapsulation. The type system uses name equivalence and includes both subtypes and derived types. Both fixed and floating point numerical types are supported.

Control flow is fully bracketed: if-then-elsif-end if, case-is-when-end case, loop-exit-end loop, goto. Subprogram parameters are in, out, or inout. Variables imported from other packages may be hidden or directly visible. Operators may be overloaded, and so may enumeration literals. There are user-defined exceptions and exception handlers.

An Ada program consists of a set of packages encapsulating data objects and their related operations. A package has a separately compilable body and interface. Ada permits generic packages and subroutines, possibly parametrized.

Ada programming places a heavy emphasis on multitasking. Tasks are synchronized by the rendezvous, in which a task waits for one of its subroutines to be executed by another. The conditional entry makes it possible for a task to test whether an entry is ready. The selective wait waits for either of two entries or waits for a limited time.

"Reference Manual for the Ada Programming Language", ANSI/MIL STD 1815A, U.S. DoD (Jan 1983). Earlier draft versions appeared in July 1980 and July 1982. ANSI 9183, ISO 1987. Russian: GOST 27831-88.

info: adainfo@ajpo.sei.cmu.edu

ftp repository: wsmr-simtel20.army.mil

ftp info: ajpo.sei.cmu.edu

ftp interpreters: stars.rosslyn.unisys.com:pub/ACE\_8.0, for SunOS

AdaEd compiler/interpreter for Unix, MS-DOS, Atari ST, Amiga

wuarchive.wustl.edu:amiga/languages/ada/AdaEd1.11.0a.bin.lzh for Amiga  
cs.nyu.edu:pub/adaed

Ada-83 - The original Ada, as opposed to Ada 9X.

Ada 9X - Revision and extension of Ada begun in 1988, currently under development. Additions include object-orientation (tagged types, abstract types and class-wide types), hierarchical libraries, and synchronization with shared data (protected types) similar to Orca. Lacks multiple inheritance. "Introducing Ada 9X", J.G.P. Barnes, Feb 1993.

ftp://ajpo.sei.cmu.edu/public/ada9x/\*

mailing list: Chris Anderson <anderson@uv4.eglin.af.mil> (Ada 9X Project Manager)

Ada++ - Object-oriented extension to Ada, implemented as an Ada preprocessor.

Ada' - ORA. Subset of Ada used by the Penelope verification system. Omits tasking, generics, fixed and floating point. "Formal Verification of Ada

Programs", D. Guaspari et al, IEEE Trans Soft Eng 16(9):1058-1075 (Sept 1990).

ADAM - A DATA Management system.

Ada-0 - U Karlsruhe, 1979. Ada subset used for compiler bootstrapping. Lacks overloading, derived types, real numbers, tasks and generics. "Revised Ada-0 Reference Manual", G. Persch et al, U Karlsruhe, Inst fur Infor II, Bericht Nr 9/81.

Adaplex - An extension of Ada for functional databases. "Adaplex: Rationale and Reference Manual 2nd ed", J.M. Smith et al, Computer Corp America, Cambridge MA, 1983.

ADAPT - Subset of APT. Sammet 1969, p.606.

AdaTran - Name given informally to an Ada subset and coding style reminiscent of the worst examples of Fortran, incomprehensible and full of GOTO's. The ENCORE Project at GE Corporate Research used this term for the output of their Fortran-to-Ada translator. ENCORE (ENVIRONMENT for CODE RE-engineering) was a system for turning AdaTran into readable Ada.

ADD 1 TO COBOL GIVING COBOL - Bruce Clement. Tongue-in-cheek suggestion for an object-oriented COBOL. SIGPLAN Notices 27(4):90-91 (Apr 1992).

ADELE - Language for specification of attribute grammars, used by the MUG2 compiler compiler. "An Overview of the Attribute Definition Language ADELE", H. Ganziger in GI3, Fachsprach "Compiler-Compiler", W. Henhpl ed, Munchen Mar 1982, pp.22-53.

ADES - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959). Version: ADES II.

ADL -

1. Adventure Definition Language. Ross Cunniff <cunniff@fc.hp.com> & Tim Brengle, 1987. An adventure language, semi-object-oriented with LISP-like syntax. A superset of DDL. Available for Unix, MS-DOS, Amiga and Acorn.  
ftp://ftp.uu.net/usenet/comp.sources.games/volume2  
//wuarchive.wustl.edu/systems/amiga/fish/fish/f0/ff091

2. Ada Development Language. R.A. Lees, 1989.

3. Assertion Definition Language. Sun Labs, for the Japanese Ministry of Trade. Language for automated generation of interface tests.  
ftp://ftp.uu.net/vendor/adl/release

AdLog - Adds a Prolog layer to Ada. "AdLog, An Ada Components Set to Add Logic to Ada", G. Pitette, Proc Ada-Europe Intl Conf Munich, June 1988.

ADM - Picture query language, extension of Sequel2. "An Image-Oriented Database System", Y. Takao et al, in Database Techniques for Pictorial Applications, A. Blaser ed, pp.527-538.

ADS - Expert system.

ADVSYs - David Betz, 1986. An adventure language, object-oriented and

LISP-like.

ftp://ftp.uu.net/usenet/comp.sources.games/volume2

AE - Application Executive. Brian Bliss <bliss@sp64.csr.d.uiuc.edu> An embeddable language, written as a C interpreter.  
ftp://sp2.csr.d.uiuc.edu/pub/bliss/ae.tex.Z

AED - Automated Engineering Design (aka ALGOL Extended for Design). MIT System Laboratory ca 1965 by a team led by Douglas T. Ross (now at Softech). Systems language for IBM 7090 and 360, an extension of ALGOL-60 with records ("plexes"), pointers, and dynamic allocation. DYNAMO II was written in AED, as was the first BCPL compiler. "The Automated Engineering Design (AED) Approach to Generalized Computer-Aided Design", D.T. Ross, Proc ACM 22nd Natl Conf, 1967. Sammet 1969 and 1978. Versions: AED-0, AED-1, AED-JR.

Aeolus - Concurrent language with atomic transactions. "Rationale for the Design of Aeolus", C. Wilkes et al, Proc IEEE 1986 Intl Conf Comp Lang, IEEE 1986, pp.107-122.

AESOP - An Evolutionary System for On-line Programming. Early interactive query system with light pen for IBM 1800. "AESOP: A Final Report: A Prototype Interactive Information Control System", J.K. Summers et al, in Information System Science and Technology, D. Walker ed, 1967. Sammet 1969, p.703.

AFAC - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

AGORA - Distributed object-oriented language.[?]

AHDL - Analog VHDL. US Air Force, under development. Mentioned in Electronic Times or Electronic Engineering Design[?] IEEE 1076.1

AHPL - A Hardware Programming Language. Hill & Peterson. A register-level language, some of whose operators resemble APL. "Digital Systems: Hardware Organization and Design", F. Hill et al, Wiley 1987. HPSIM2: a function-level simulator, available from Engrg Expt Sta, U Arizona.

AID - Algebraic Interpretive Dialogue. Version of Joss II for the PDP-10. "AID (Algebraic Interpretive Dialogue)", DEC manual, 1968.

AIDA -

1. M. Gfeller. A functional dialect of Dictionary APL. "APL Arrays and Their Editor", M. Gfeller, SIGPLAN Notices 21(6):18-27 (June 1986) and SIGAPL Conf Proc [?]

2. Karlsruhe, 1980. An intermediate representation language for Ada, was merged with TCOL.Ada to form Diana. "AIDA Introduction and User Manual", M. Dausmann et al, U Karlsruhe, Inst fur Inform II, TR Nr 38/80. "AIDA Reference Manual", ibid, TR Nr 39/80, Nov 1980.

AIMACO - AIr MATERIAL COMmand compiler. Modification of FLOW-MATIC. Supplanted by COBOL. Sammet 1969, p.378.

AGP-L - Language for natural language recognition. [?]

AKCL - Austin Kyoto Common LISP. Wm Schelter <wfs@math.utexas.edu>, U Texas, 1987-1994. Enhancements to KCL. In 1994, AKCL was renamed Gnu CL. <ftp://rascal.ics.utexas.edu/pub/akcl-1-625.tar.Z>

AKL - Andorra Kernel Language. Successor of KAP. "Programming Paradigms of the Andorra Kernel Language", S. Janson <sverker@sics.se> et al in Logic Programming: Proc 1991 Intl Symp, MIT Press 1991. Prototype implementation available from the author.

AL - Assembly Language. Stanford U, 1970's. Language for industrial robots. "The AL Language for an Intelligent Robot", T. Binford in Languages et Methods de Programation des Robots Industriels, pp.73-88, IRIA Press 1979. "AL User's Manual", M.S. Mujtaba et al, Stanford AI Lab, Memo AIM-323 (Jan 1979).

ALADIN -

1. A Language for Attributed DefINitions. A language for formal specification of attributed grammars. Input language for the GAG compiler generator. Applicative, strongly typed. "GAG: A Practical Compiler Generator", Uwe Kastens <uwe@uni-paderborn.de> et al, LNCS 141, Springer 1982.

2. Interactive math for IBM 360. "A Conversational System for Engineering Assistance: ALADIN", Y. Siret, Proc Second Symp Symb Algebraic Math, ACM Mar 1971.

ALAM - Atlas LISP Algebraic Manipulation. Symbolic math, especially for General Relativity. "ALAM Programmer's Manual", Ray D'Inverno, 1970. (See CLAM).

A-language. An early Algol-like surface syntax for Lisp. "An Auxiliary Language for More Natural Expression--The A-language", W. Henneman in The Programming Language LISP, E.C. Berkeley et al eds, MIT Press 1964, pp.239-248.

ALC - Assembly Language Compiler. Alternative name for IBM 360 assembly language. (cf. BAL).

Alcool-90 - An object-oriented extension of ML with runtime overloading and a type-based notion of modules, functors and inheritance. Built on CAML Light.

<ftp://ftp.inria.fr/INRIA/lang/alcool/>  
info: Francois Rouaix <rouaix@inria.fr>

ALCOR - Subset of ALGOL. Sammet 1969, p.180.

Aldat - Database language, based on extended algebra. Listed by M.P. Atkinson & J.W. Schmidt in a tutorial in Zurich, 1989. [?]

ALDES - ALgorithm DEScription. "The Algorithm Description Language ALDES", R.G.K. Loos, SIGSAM Bull 14(1):15-39 (Jan 1976).

ALDiSP - Applicative Language for Digital Signal Processing. 1989, TU Berlin. Functional language with special features for real-time I/O and numerical processing. "An Applicative Real-Time Language for DSP-Programming Supporting Asynchronous Data-Flow Concepts", M. Freericks

<mfxf@cs.tu-berlin.de> in Microprocessing and Microprogramming 32, N-H 1991.

ALEC - A Language with an Extensible Compiler. Implemented using RCC on an ICL 1906A. "ALEC - A User Extensible Scientific Programming Language", R.B.E. Napper et al, Computer J 19(1):25-31.

ALEF - Concurrent language for systems programming. C-like syntax, but a different type system. Exception handling, process management and synchronization primitives, both shared variable and message passing. Used in Plan 9 OS.

ftp://research.att.com/dist/plan9man/05alef.ps.Z

info: Phil Winterbottom <philw@reserch.att.com>

ALEPH -

1. A Language Encouraging Program Hierarchy. ca 1975. "On the Design of ALEPH", D. Grune, CWI, Netherlands 1986.

2. Peter Henderson ca. 1970. Formal semantics. CACM 15(11):967-973 (Nov 1972).

Alex -

1. Stephen Crawley <sxc@itd.dtso.oz.au>, Defence Science & Tech Org, Australia. Under development. Polymorphic with ADT's, type inference, inheritance.

2. ISWIM-like language with exception handling. "An Exception Handling Construct for Functional Languages", M. Brez et al, in Proc ESOP88, LNCS 300, Springer 1988.

Alexis - Alex Input Specification. Input language for the scanner generator Alex. "Alex: A Simple and Efficient Scanner Generator", H. Mossenbock, SIGPLAN Notices 21(5), May 1986.

ALF - Algebraic Logic Functional language. WAM-based language with narrowing/rewriting. Horn clauses with equality. Any functional expression can be used in a goal. "The Implementation of the Functional-Logic Language ALF", M. Hanus and A. Schwab.

ftp://ftp.germany.eu.net/pub/programming/languages/LogicFunctional/alf\*

info: Rudolf Opalla <opalla@julien.informatik.uni-dortmund.de>

Alfl - Paul Hudak <hudak-paul@cs.yale.edu>, Yale 1983. Functional, weakly typed, lazy. Implemented as a Scheme preprocessor for the Orbit compiler, by transforming laziness into force-and-delay. "Alfl Reference Manual and Programmer's Guide", P. Hudak, YALEU/DCS/RR322, Yale U, Oct 1984. (See ParAlfl).

ALGEBRAIC - Early system on MIT's Whirlwind. Listed in CACM 2(5):16 (May 1959).

ALGOL 58 - See IAL.

ALGOL 60 - ALGORithmic Language. Designed as a portable language for scientific computations. ALGOL 60 was small and elegant. It was block-structured, nested, recursive, and free form. It was also the first language to be described in BNF. There were three lexical representations:

hardware, reference, and publication. The only structured data types were arrays, but they were permitted to have lower bounds and could be dynamic. Keywords. Conditional expression. Introduced :=, if-then-else, very general 'for' loops. Switch declaration (an array of statement labels generalizing FORTRAN's computed goto). Parameters were call-by-name and call-by-value. Static local 'own' variables. Lacked user-defined types, character manipulation and standard I/O. "Report on the Algorithmic Language ALGOL 60", Peter Naur ed, CACM 3(5):299-314 (May 1960).

ALGOL 60 Modified - "A Supplement to the ALGOL 60 Revised Report", R.M. DeMorgan et al, Computer J 19(4):364 and SIGPLAN Notices 12(1) 1977. Erratum in Computer J 21(3):282 (Aug 1978) applies to both.

ALGOL 60 Revised - Still lacked standard I/O. "Revised Report on the Algorithmic Language ALGOL 60", Peter Naur ed, CACM 6(1):1-17 (Jan 1963). <ftp://locke.ccil.org/pub/retro/algol60-0.17.tar.gz> NASE A60 Interpreter

ALGOL 68 - Adriaan van Wijngaarden et al. Discussed from 1963 by Working Group 2.1 of IFIP. Definition accepted Dec 1968. ALGOL 68 was complex, and posed difficulties for both implementors and users. Structural equivalence. Automatic type conversion, including dereferencing. Flexible arrays. Generalized loops (for-from-by-to-while-do-od), if-then-elif-fi, integer case statement with 'out' clause, skip statement, goto. Blocks, procedures and user-defined operators. Procedure parameters. Concurrent execution (cobegin/coend) and semaphores. Generators heap and loc for dynamic allocation. No abstract data types, no separate compilation.

ALGOL 68-R - April, 1970. Royal Signals and Radar Establishment, Malvern, Worcs UK, 1970. A restriction of ALGOL 68 permitting one-pass compilation: identifiers, modes and operators must be declared before use, no automatic proceduring, no concurrency. Implemented in ALGOL 60 under GEORGE 3 on an ICL 1907F. "ALGOL 68-R, Its Implementation and Use", I.F. Currie et al, Proc IFIP Congress 1971, N-H 1971, pp.360-363.

ALGOL 68 Revised - Significantly simplified the language. "Revised Report on the Algorithmic Language ALGOL 68," A. Van Wijngaarden et al, Acta Informatica 5:1-236 (1975), also Springer 1976, and SIGPLAN Notices 12(5):1-70 (May 1977).

ALGOL 68C - S. Bourne and Mike Guy, Cambridge U 1975. Variant of ALGOL 68, allowing two-pass compilation. Used as the implementation language for the CHAOS OS for the CAP capability computer. Ported to IBM 360, VAX/VMS, several others.

ALGOL 68RS - Royal Signals and Radar Establishment, Malvern UK. An extension of ALGOL 68 supporting function closures. Has been ported to Multics and VAX/VMS.

ALGOL 68S - A subset of ALGOL 68 allowing simpler compilation. Intended mainly for numerical computation. "A Sublanguage of ALGOL 68", P.G. Hibbard, SIGPLAN Notices 12(5) (May 1977). Rewritten in BLISS for the PDP-11, and later in Pascal. Available as shareware from Charles Lindsey <chl@cs.man.ac.uk>, Version 2.3 for Sun3's under OS4.x and Atari under GEMDOS (or potentially other machines supported by the Amsterdam Compiler Kit).

ALGOL C - Clive Feather, Cambridge U, ca. 1981. Variant of ALGOL 60; added

structures and exception handling. Designed for beginning students.

ALGOL D - "A Proposal for Definitions in ALGOL", B.A. Galler et al, CACM 10:204-219 (1967).

ALGOL N - Yoneda. Proposed successor to ALGOL 60.

ALGOL W - Derivative of ALGOL 60. Introduced double precision, complex numbers, bit strings and dynamic data structures. Parsed entirely by operator precedence. Used call-by-value-result. "A Contribution to the Development of Algol", N. Wirth, CACM 9(6):413-431 (June 1966). "ALGOL W Implementation", H. Bauer et al, TR CS98, Stanford U, 1968.

ALGOL X - Proposed successor to ALGOL 60, a "short-term solution to existing difficulties". The three designs proposed were by Wirth, Seegmuller and van Wijngaarden. Sammet 1969, p.194.

ALGOL Y - Proposed successor to ALGOL 60, a "radical reconstruction". Originally a language that could manipulate its own programs at runtime, it became a collection of features that were not accepted for ALGOL X.

ALGY - Early language for symbolic math. Sammet 1969, p.520.

ALIAS - ALgorIthmic ASsembly language. Machine oriented language, a variant of BLISS. Implemented in BCPL for the PDP9. "ALIAS", H.E. Barreveld, Int Rep, Math Dept, Delft U Tech, Netherlands (1973).

ALJABR - An implementation of MACSYMA for the Mac. Fort Pond Research. info: aljabr@fpr.com

ALLOY - Combines functional, object-oriented and logic programming ideas, suitable for massively parallel systems. "The Design and Implementation of ALLOY, a Parallel Higher Level Programming Language", Thanasis Mitsolidis <mitsolid@cs2.nyu.edu>, PhD Thesis NYU 1990. Version: ALLOY 2.0  
ftp://cs.nyu.edu/pub/local/alloy/\*

ALM - Assembly Language for Multics. Language on the GE645. Critical portions of the Multics kernel were written in ALM.

ALP - List-processing extension of Mercury Autocode. "ALP, An Autocode List-Processing Language", D.C. Cooper et al, Computer J 5:28-31 (1962).

ALPAK - Subroutine package used by ALTRAN. "The ALPAK System for Nonnumerical Algebra on a Digital Computer", W.S. Brown, Bell Sys Tech J 42:2081 (1963). Sammet 1969, p.502.

ALPHA - A.P. Ershov, Novosibirsk, 1961. Also known as "Input". Extension of ALGOL 60 for the M-20 computer, including matrix operations, slices, complex arithmetic. "The Alpha Automatic Programming System", A.P. "Yershov" ed., A-P 1971.

Alphard - (named for the brightest star in Hydra). Wulf, Shaw and London, CMU 1974. Pascal-like. Data abstraction using the 'form', which combines a specification and an implementation. "Abstraction and Verification in Alphard: Defining and Specifying Iteration and Generators", Mary Shaw, CACM 20(8):553-563 (Aug 1977).

ALPS -

1. Richard V. Andree, U Oklahoma. Early interpreted algebraic language for Bendix G15, said to have preceded and influenced development of BASIC.

2. Parallel logic language. "Synchronization and Scheduling in ALPS Objects", P. Vishnubhotia, Proc 8th Intl Conf Distrib Com Sys, IEEE 1988, pp.256-264.

ALTAC - An extended FORTRAN II for Philco 2000, built on TAC. Sammet 1969, p.146.

ALTRAN - W.S. Brown, Bell Labs, ca. 1968. A FORTRAN extension for rational algebra. "The ALTRAN System for Rational Function Manipulation - A Survey", A.D. Hall, CACM 14(8):517-521 (Aug 1971).

Amber -

1. Adds CSP-like concurrency to ML. Similar to Galileo. Concurrency, multiple inheritance, persistence. Programs must be written in two type faces, roman and italics! Both static and dynamic types. "Amber", L. Cardelli, TR Bell Labs 1984. Implementation for Mac.

2. U Washington, late 80's. An object-oriented distributed language based on a subset of C++.

AMBIT - Algebraic Manipulation by Identity Translation (also claimed: "Acronym May Be Ignored Totally"). C. Christensen, Massachusetts Computer Assocs, 1964. An early pattern-matching language aimed at algebraic manipulation. Sammet 1969, pp.454-457.

AMBIT/G - (G for graphs). "An Example of the Manipulation of Directed Graphs in the AMBIT/G Programming Language", C. Christensen, in Interactive Systems for Experimental Applied Mathematics, M. Klerer et al, eds, Academic Press 1968, pp.423-435.

AMBIT/L - (L for lists). List handling, allows pattern matching rules based on two-dimensional diagrams. "An Introduction to AMBIT/L, A Diagrammatic Language for List Processing", Carlos Christensen, Proc 2nd ACM Symp Symb and Alg Manip (Mar 1971).

AMBIT/S - (S for strings).

AMBUSH - Language for linear programming problems in a materials-processing/transportation network. "AMBUSH - An Advanced Model Builder for Linear Programming", T.R. White et al, National Petroleum Refiners Assoc Comp Conf (Nov 1971).

AML - IBM, 1980's. High-level language for industrial robots. "AML: A Manufacturing Language", R.H. Taylor et al, Inst J Robot Res 1(3):19-43.

AML/E - AML Entry. Simple version of AML, implemented on PC, with graphic display of the robot position.

AMP - Algebraic Manipulation Package. Symbolic math, written in Modula-2, seen on CompuServe.

AMPL - "AMPL: Design, Implementation and Evaluation of a Multiprocessing Language", R. Dannenberg, CMU 1981. "Loglan Implementation of the AMPL Message Passing System", J. Milewski SIGPLAN Notices 19(9):21-29 (Sept 1984).

AMPLE - Hybrid Technologies, Cambridge England, mid 80's. FORTH-like language for programming the 500/5000 series of add-on music synthesizers for the BBC micro. Many listings published in Acorn User magazine.

AMPPL-II - Associative Memory Parallel Processing Language. Early 70's.

AMTRAN - Automatic Mathematical TRANslation. NASA Huntsville, 1966. For IBM 1620, based on Culler-Fried System, requires special terminal. "AMTRAN: An Interactive Computing System", J. Reinfelds, Proc FJCC 37:537-542, AFIPS (Fall 1970).

ANCP - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

ANDF - Architecture Neutral Distribution Format. OSF's request for a universal intermediate language, allowing software to be developed and distributed in a single version, then installed on a variety of hardware. "Architecture Neutral Distribution Format: A White Paper", Open Software Foundation, Nov 1990. (See UNCOL).  
list: andf-tech@osf.org

Andorra-I - The OR parallelism of Aurora plus the AND parallelism of Parlog. "Andorra-I: A Parallel Prolog System that Transparently Exploits both And- and Or-Parallelism", V.S. Costa et al, SIGPLAN Notices 26(7):83-93 (July 1991).

Andorra-Prolog - "Andorra-Prolog: An Integration of Prolog and Committed Choice Languages", S. Haridi et al, Intl Conf Fifth Gen Comp Sys 1988, ICOT 1988.

Animus - "Constraint-Based Animation: The Implementation of Temporal Constraints in the Animus System", R. Duisberg, PhD Thesis U Washington 1986.

Anna - ANNotated Ada. ca. 1980. Adds semantic assertions to Ada as formal comments. Based on first-order logic. Includes generalized type constraints, virtual checking functions, and behavior specification. "ANNA - A Language for Annotating Ada Programs", David Luckham et al, Springer 1987.  
ftp://anna.stanford.edu/pub/anna/\*

ANTLR - ANOther Tool for Language Recognition. Parser generator, part of PCCTS (Purdue Compiler-Construction Tool Set).  
ftp://ecn.purdue.edu

APAL - Array Processor Assembly Language. For the DAP parallel machine.

APAREL - A Parse REquest Language. PL/I extension to provide BNF parsing routines, for IBM 360. "APAREL: A Parse Request Language", R.W. Balzer et al, CACM 12(11) (Nov 1969).

APDL - Algorithmic Processor Description Language. ALGOL-60-like language

for describing computer design, for CDC G-21. "The Description, Simulation, and Automatic Implementation of Digital Computer Processors", J.A. Darringer, Ph.D Thesis EE Dept, CMU May 1969.

APESE - The language of the APE100 SIMD machine. (See TAO.)  
<http://slacvm.slac.stanford.edu:5080/FIND/NAME/APESE/FULL>

APL - A Programming Language. Ken Iverson Harvard U 1957-1960. Designed originally as a notation for the concise expression of mathematical algorithms. Went unnamed and unimplemented for many years. Finally a subset APL\360 was implemented in 1964. APL is an interactive array-oriented language with many innovative features, written using a non-standard character set. It is dynamically typed with dynamic scope. All operations are either dyadic infix or monadic prefix, and all expressions are evaluated from right to left. The only control structure is branch. APL introduced several functional forms but is not purely functional. "A Programming Language", Kenneth E. Iverson, Wiley, 1962. Versions: APL\360, APL SV, VS APL, Sharp APL, Sharp APL/PC, APL\*PLUS, APL\*PLUS/PC, APL\*PLUS/PC II, MCM APL, Honeyapple, and DEC APL.  
(See Iverson's Language).

APL2 - IBM. An APL extension with nested arrays. "APL2 Programming: Language Reference", IBM Aug 1984. Order No. SH20-9227-0.

APLGOL - H-P? An APL with ALGOL-like control structure.

APPLE - Revision of APL for the Illiac IV.

AppleScript - An object-oriented shell language for the Macintosh, approximately a superset of HyperTalk.

Applesoft BASIC - Version of BASIC on Apple computers.

APPROG - Unifies logic and functional programming. "The APPROG Language", S. Cohen in Logic Programming, deGroot et al eds, P-H 1986, pp.39-276.

APT - Automatically Programmed Tools. For numerically controlled machine tools. The first language to be an ANSI standard: ANSI X3.37. "APT Part Programming", McGraw-Hill. Versions: APT II (IBM 704, 1958), APT III (IBM 7090, 1961). Sammet 1969, p.605.

APX III - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

AQL - Picture query language, extension of APL. "AQL: A Relational Database Management System and Its Geographical Applications", F. Antonacci et al, in Database Techniques for Pictorial Applications, A. Blaser ed, pp.569-599.

ARCHI - A microarchitecture description language with C-like syntax, intended for input to a one-pass firmware tool generator. "A Microarchitecture Description Language for Retargeting Firmware Tools", J.F. Nixon et al, Proc 19th Ann Workshop Microprogramming (MICRO-19), 1986, pp.34-43.

Arctic - Real-time functional language, used for music synthesis. "Arctic: A Functional Language for Real-Time Control", R.B. Dannenberg, Conf Record

1984 ACM Symp on LISP and Functional Prog, ACM.

ARES - Pictorial query language. "A Query Manipulation System for Image Data Retrieval", T. Ichikawa et al, Proc IEEE Workshop Picture Data Description and Management, Aug 1980, pp.61-67.

Ariel - Array-oriented language for CDC 6400. "Ariel Reference Manual", P. Devel, TR 22, CC UC Berkeley, Apr 1968.

Argus - LCS, MIT. A successor to CLU. Supports distributed programming through guardians (like monitors, but can be dynamically created) and atomic actions (indivisible activity). cobegin/coend. "Argus Reference Manual", B. Liskov et al., TR-400, MIT/LCS, 1987. "Guardians and Actions: Linguistic Support for Robust, Distributed Programs", B. Liskov <liskov@lcs.mit.edu> et al, TOPLAS 5(3):381-404 (1983).

Ariel - An array-oriented language. "A New Survey of the Ariel Programming Language", P. Deuel, TR 4, Ariel Consortium, UC Berkeley (June 1972).

ARITH-MATIC - Alternate name for A-3.

ART - Real-time functional language, timestamps each data value when it was created. "Applicative Real-Time Programming", M. Broy, PROC IFIP 1983, N-H.

ARTSPEAK - Early simple language for plotter graphics. "The Art of Programming, ARTSPEAK", Henry Mullish, Courant Inst (Nov 1974).

ASDIMPL - ASDO Implementation Language. A C-like language, run on Burroughs' mainframe computers in the early 80's, and cross-compiled to x86-based embedded processors.

ASDL - "ASDL - An Object-Oriented Specification Language for Syntax-Directed Environments", M.L. Christ-Neumann et al, European Software Eng Conf, Strasbourg, Sept 1987, pp.77-85.

ASF - An algebraic specification language. "Algebraic Specification", J.A. Bergstra et al, A-W 1989.

Ashmedai - Michael Levine <levine@cpwsca.psc.edu> Symbolic math package. Had an influence on SMP and FORM. Versions for Univac 1108 and VAX/VMS.

ASIS - Ada Semantic Interface Specification. A layered, vendor-independent architecture providing an interface to the Ada program library.  
info: //ajpo.sei.cmu.edu/public/asis/\*

ASF - Algebraic Specification Formalism. CWI. Language for equational specification of abstract data types. "Algebraic Specification", J.A. Bergstra et al eds, A-W 1989.

ASL - Algebraic Specification Language. "Structured Algebraic Specifications: A Kernel Language", M. Wirsing, Theor Comput Sci 42, pp.123-249, Elsevier 1986.

ASM - Assembly language on CP/M machines (and a lot of others).

ASN.1 - Abstract Syntax Notation. Data description language, designed for

the exchange of structured data over networks. Derived from the 1984 standard CCITT X.408 used to describe the syntax of messages in the X.400 mail system. Used by the Natl Center for Biotechnology Information. CCITT, ITU TS X.208 (1988), ISO 8824. "An Overview of ASN.1", G. Neufeld et al, Computer Networks and ISDN Systems, 23(5):393-415 (Feb 1992). Available from Logica, UK. (See BER).  
ftp://cs.ubc.ca/pub/local/src/snacc/snacc1.1.tar.Z [?]

ASP - Query language? Sammet 1969, p.702.

ASpect - Algebraic Specification of abstract data Types. Strict functional language that compiles to C. Versions for Sun, Ultrix, NeXT, Mac, OS2/2.0, linux, RS6000, Atari, Amiga.  
ftp://wowbagger.uni-bremen.de/pub/programming/lanugages/ASpect/\*

ASPOL - A Simulation Process-Oriented Language. An ALGOL-like language for computer simulation. "Process and Event Control in ASPOL", M.H. MacDougall, Proc Symp on Simulation of Computer Systems, NBS (Aug 1975).

ASPEN - Toy language for teaching compiler construction. "ASPEN Language Specifications", T.R. Wilcox, SIGPLAN Notices 12(11):70-87 (Nov 1977).

ASPIK - Multiple-style specification language. "Algebraic Specifications in an Integrated Software Development and Verification System", A. Voss, Diss, U Kaiserslautern, 1985.

Aspirin - MITRE Corp. A language for the description of neural networks. For use with the MIGRAINES neural network simulator. Version: 6.0  
ftp://ftp.cognet.ucla.edu/alexis/am6\*

ASPLE - Toy language. "A Sampler of Formal Definitions", M. Marcotty et al, Computing Surveys 8(2):191-276 (Feb 1976).

ASSEMBLY - Early system on IBM 702. Listed in CACM 2(5):16 (May 1959).

ASTAP - Advanced STatistical Analysis Program. Analyzing electronic circuits and other networks. "Advanced Statistical Analysis Program (ASTAP) Program Reference Manual", SH-20-1118, IBM, 1973.

Astral - Based on Pascal, never implemented. "ASTRAL: A Structured and Unified Approach to Database Design and Manipulation", T. Amble et al, in Proc of the Database Architecure Conf, Venice, June 1979.

AT-3 - Original name of MATH-MATIC. Sammet 1969, p.135.

ATLAS - Abbreviated Test Language for Avionics Systems. MIL-spec language for automatic testing of avionics equipment. Replaced Gaelic and several other test languages. "IEEE Standard ATLAS Test Language", IEEE Std 416-1976 and 416-1984.

Atlas Autocode - Autocode for the Ferranti Atlas, which may have been the first commercial machine with hardware-paged virtual memory. Whereas other autocodes were basically symbolic assembly languges, Atlas Autocode was high-level and block-structured, resembling a cross between FORTRAN and ALGOL 60. It had call-by value, loops, declarations, complex numbers, pointers, heap and stack storqage generators, dynamic arrays, extensible syntax, etc.

Atlas Commercial Language - [?]

ATOLL - Acceptance, Test Or Launch Language. Language used for automating the checkout and launch of Saturn rockets. "SLCC ATOLL User's Manual", IBM 70-F11-0001, Huntsville AL Dec 1970.

A'UM - K. Yoshida and T. Chikayama <chik@icot.or.jp>. Built on top of KL1. "A'UM - A Stream-based Concurrent Logic Object-Oriented Language", K. Yoshida et al, Proc 3rd Intl Conf Fifth Gen Comp Sys, Springer 1988, pp.638-649.

Aurora - "The Aurora Or-Parallel Prolog System", E. Lusk et al, Proc 3rd Intl Conf on Fifth Generation Comp Systems, pp. 819-830, ICOT, A-W 1988.

Autocode - Alick E. Glennie, 1952. AUTOCODER was possibly the first primitive compiler, it translated symbolic statements into machine language for the Manchester Mark I computer. Autocoding came to be a generic term for symbolic assembly language programming, and versions of Autocode were developed for many machines: Ferranti Atlas, Titan, Mercury and Pegasus, and IBM 702 and 705.

AUTOGRAF - Describing bar charts. "User's Manual for AUTOGRAF", Cambridge Computer Assoc (Dec 1972).

AUTOGRP - AUTOMated GROuPing system. Interactive statistical analysis. An extension of CML. "AUTOGRP: An Interactive Computer System for the Analysis of Health Care Data", R.E. Mills et al, Medical Care 14(7) (Jul 1976).

Autolisp - Dialect of LISP used by the Autocad CAD package, Autodesk, Sausalito, CA.

AUTOMATH - Eindhoven, Netherlands. A very high level language for writing proofs. "The Mathematical Language AUTOMATH, Its Usage and Some of its Extensions", N.G. deBruijn, in Symp on Automatic Demonstration, LNM 125, Springer 1970.

Autopass - "Autopass: An Automatic Programming System for Computer-Controlled Mechanical Assembly", L.I. Lieberman et al, IBM J Res Dev 21(4):321-333 (1979).

AUTO-PROMPT - Numerical control language from IBM for 3-D milling. Sammet 1969, p.606.

Autostat - "Autostat: A Language for Statistical Programming", A.S. Douglas et al, Computer J 3:61 (1960).

AVA - A Verifiable Ada. Michael Smith. A formally defined subset of Ada, under development. "The AVA Reference Manual", M. Smith, TR64, Computational Logic, Austin TX (June 1990).

Avalon/C++ - 1986. Fault-tolerant distributed systems, influenced by Argus. A concurrent extension of C++ with servers and transactions. "Camelot and Avalon: A Distributed Transaction Facility", J.L. Eppinger et al, Morgan Kaufmann 1990.

Avalon/Common LISP - Prototype only. "Reliable Distributed Computing with Avalon/Common LISP", S.M. Clamen et al, CMU-CS-89-186 and Proc Intl Conf on Computer Languages, Mar 1990.

Avon - Dataflow language. "AVON: A Dataflow Language", A. Deb, ICS 87, Second Intl Conf on Supercomputing, v.3, pp.9-19 (ISI 1987).

AXIOM - IBM. Commercially available subset of Scratchpad. "Axiom - The Scientific Computing System", R. Jenks et al, Springer 1992.

AXIS - H-P. Algebraic language with user-definable syntax. [?]

AXLE - An early string processing language. Program consists of an assertion table which specifies patterns, and an imperative table which specifies replacements. "AXLE: An Axiomatic Language for String Transformations", K. Cohen et al, CACM 8(11):657-661 (Nov 1965).

AWK - Aho Weinberger Kernighan. 1978. Text processing/macro language. "The AWK Programming Language" A. Aho, B. Kernighan, P. Weinberger, A-W 1988. (See Bawk, Gawk, Mawk, Nawk, Tawk.)  
ftp://netlib.att.com/research/awk\*

B -

1. Thompson, 1970. A systems language written for Unix on the PDP-11. Derived from BCPL, and very similar to it except for syntax. B was the predecessor of C. Used as the systems language on Honeywell's GCOS-3. "The Programming Language B", S.C. Johnson & B.W. Kernighan, CS TR 8, Bell Labs (Jan 1973).

2. L. Meertens & S. Pemberton. Simple interactive programming language, the predecessor of ABC[1]. "Draft Proposal for the B Language", Lambert Meertens, CWI, Amsterdam, 1981.  
ftp://ftp.uni-kl.de/pub/languages/B.tar.Z

3. Jean-Raymond Abrial. Specification language similar to Z, but also supports development of C code from specifications. B Core UK, Magdalen Centre, Oxford Science Park, Oxford OX4 4GA.  
info: Ib.Sorensen@comlab.ox.ac.uk

B-0 - Original name of FLOW-MATIC, Remington Rand. UNIVAC I or II ca. 1958.

Babbage - GEC Marconi Ltd. Named after "the first programmer to slip schedule and go over budget". Low-level language, used on their OS4000 operating system. The British videotext system Prestel is programmed in Babbage. Article in Datamation, ca Oct, 1980[?]

BABEL -

1. A subset of ALGOL 60, with many ALGOL W extensions. "BABEL, A New Programming Language", R.S. Scowen, Natl Phys Lab UK, Report CCU7, 1969.

2. Mentioned in The Psychology of Computer Programming, G.M. Weinberg, Van Nostrand 1971, p.241.

3. Higher-order functional plus first-order logic language. "Graph-Based

Implementation of a Functional Logic Language", H. Kuchen et al, Proc ESOP 90, LNCS 432, Springer 1990, pp.271-290. "Logic Programming with Functions and Predicates: The Language BABEL", Moreno-Navarro et al, J Logic Prog 12(3) (Feb 1992).

BABYLON - Development environment for expert systems.  
ftp://gmdzi.gmd.de/gmd/ai-research/Software/\*

BACAIC - Boeing Airplane Company Algebraic Interpreter Coding system.  
Pre-FORTRAN system on the IBM 701, IBM 650.

BAL - Basic Assembly Language. What most people called IBM 360 assembly language. (See ALC).

BALGOL - ALGOL on Burroughs 220. Sammet 1969, p.174.

BALITAC - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

BALM - Block And List Manipulation. Harrison, 1970. Extensible language with LISP-like features and ALGOL-like syntax, for CDC 6600. "The Balm Programming Language", Malcolm Harrison, Courant Inst (May 1973).

BAP - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

Baroque - Boyer & Moore, 1972. Early logic programming language.  
"Computational Logic: Structure Sharing and Proof of program Properties", J. Moore, DCL Memo 67, U Edinburgh 1974.

BASCMP - A modification of STAGE2, used to implement the Basic Wisp translator. Implementing Software for Non-numeric Applications, W. M. Waite, P-H 1973.

bash - Bourne Again SHell. GNU's command shell for Unix.  
ftp://prep.ai.mit.edu/pub/gnu/bash-1.10.tar.Z

BASIC - Beginner's All-purpose Symbolic Instruction Code. John G. Kemeny & Thomas E. Kurtz, Dartmouth College, designed 1963, first ran on an IBM 704 on May 1, 1964. Quick and easy programming by students and beginners. BASIC exists in many dialects, and is popular on microcomputers with sound and graphics support. Most micro versions are interactive and interpreted, but the original Dartmouth BASIC was compiled. ANSI Minimal BASIC, ANS X3.60-1978.

list: basic@ireq.hydro.qc.ca

BASIC AUTOCODER - Early system on IBM 7070. Listed in CACM 2(5):16 (May 1959).

Basic COBOL - Subset of COBOL from COBOL-60 standards. Sammet 1969, p.339.

Basic FORTRAN - Subset of FORTRAN. Sammet 1969, p.150.

Basic JOVIAL - Subset of JOVIAL, ca. 1965. Sammet 1969, p.529.

bawk - Bob Brodt. AWK-like pattern-matching language, distributed with Minix.

bc - [Belinda's Calculator?] An interactive mini-language for numerical

calculation. Part of the Unix toolkit since V7. Originally implemented by Belinda Cherry as a preprocessor for dc, supporting infix notation. The GNU toolkit contains a clone of it.  
ftp://prep.ai.mit.edu/pub/gnu

BC NELIAC - Version of NELIAC, post 1962. Sammet 1969, p.197.

BCL - Successor to Atlas Commercial Language. "The Provisional BCL Manual", D. Hendry, U London 1966.

BCPL - Basic CPL. Richards 1969. British systems language, a descendant of CPL and the inspiration for B and C. BCPL is low-level, typeless and block-structured, and provides only one-dimensional arrays. Case is not significant, but conventionally reserved words begin with a capital.

Flow control: If-Then, Test-Then-Else, Unless-Do, While-Do, Until-Do, Repeat, Repeatwhile, Repeatuntil, For-to-By-Do, Loop, Break and Switchon-Into-Case-Default-Endcase. BCPL has conditional expressions, pointers, and manifest constants. BCPL had both procedures: 'Let foo(bar) Be command' and functions: 'Let foo(bar) = expression'. 'Valof \$(..Resultis..\$)' causes a compound command to produce a value. Parameters are call-by-value.

Program segments communicate via the global vector where system and user variables are stored in fixed numerical locations in a single array. BCPL was used to implement the TRIPOS OS, which was subsequently reincarnated as AmigaDOS. "BCPL - The Language and its Compiler", Martin Richards & Colin Whitby-Stevens, Cambridge U Press 1979. (See OCODE, INTCODE).

Oxford BCPL differed slightly: Test-Ifso-Ifnot, and section brackets in place of \$( \$).

BDL - Block Diagram Compiler. A block-diagram simulation tool, with associated language. "A Software Environment for Digital Signal-Processing Simulations," D.H. Johnson & R.E. Vaughan, Circuits Systems and Signal Processing 6(1):31-43, (1987).

BeBOP - Combines sequential and parallel logic programming, object-oriented and meta-level programming. Both "don't know" nondeterminism and stream AND-parallelism. Prolog theories are first order entities and may be updated or passed in messages. Implemented by translation to NU-Prolog and PNU-Prolog.

ftp://munnari.oz.au/pub/bebop.tar.Z  
info: Andrew Davidson <ad@cs.mu.oz.au>

B EGL - Back End Generator Language. A code generator description language. The input language for the back end generator BEG. "BEG - A Generator for Efficient Back Ends", H. Emmelmann et al, SIGPLAN Notices 24(7):227-237 (Jul 1989). "BEG - A Back End Generator - User Manual", H. Emmelmann <emmel@karlsruhe.gmd.dbp.de>, GMD, U Karlsruhe 1990.  
ftp://iraun1.ira.uka.de/pub/programming/cocktail/\*

BELL - Early system on IBM 650 and Datatron 200 series. [Is Datatron version the same?] Listed in CACM 2(5):16 (May 1959). Versions: BELL L2, BELL L3.

BER - Basic Encoding Rules. Provides a universal (contiguous) representation of data values. Used with ASN.1.

Bertrand - (named for the British mathematician Bertrand Russell (1872-1970)). Wm. Leler. Rule-based specification language based on augmented term rewriting. Used to implement constraint languages. The user must explicitly specify the tree-search and the constraint propagation. "Constraint Programming Languages - Their Specification and Generation", W. Leler, A-W 1988, ISBN 0-201-06243-7.  
ftp://nexus.yorku.ca/pub/scheme/scm/bevan.shar

BETA - Kristensen, Madsen<olm@det.eng.sun.com>, Moller-Pedersen & Nygaard, 1983. Object-oriented language with block structure, coroutines, concurrency, strong typing, part objects, separate objects and classless objects. Central feature is a single abstraction mechanism called "patterns", a generalization of classes, providing instantiation and hierarchical inheritance for all objects including procedures and processes. "Object-Oriented Programming in the BETA Programming Language", Ole Lehrmann et al, A-W June 1993, ISBN 0-201-62430-3. Mjolner Informatics ApS, implementations for Mac, Sun, HP, Apollo.  
info: info@mjolner.dk  
list: usergroup@mjolner.dk

BIOR - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

BLAZE - Single assignment language for parallel processing. "The BLAZE Language: A Parallel Language for Scientific Programming", P. Mehrotra <mehrotra@csrd.uiuc.edu> et al, J Parallel Comp 5(3):339-361 (Nov 1987).

BLAZE 2 - Object-oriented successor to BLAZE. "Concurrent Object Access in BLAZE 2", P. Mehrotra et al, SIGPLAN Notices 24(4):40-42 (Apr 1989).

Blazon - "From Blazon to Postscript", Daniel V. Klein, LoneWolf Systems, USENIX Symp on Very High Level Languages, Oct 1994.

B-LINE - Early CAD language. "B-LINE, Bell Line Drawing Language", A.J. Frank, Proc Fall JCC 33 1968.

BLISS - Basic Language for Implementation of System Software (or allegedly, "System Software Implementation Language, Backwards"). W.A. Wulf, CMU ca. 1969. An expression language, block-structured, and typeless, with exception handling facilities, coroutines, a macro system, and a highly optimizing compiler. One of the first non-assembly languages for OS implementation. Gained fame for its lack of a goto. Also lacks implicit dereferencing: all symbols stand for addresses, not values. "BLISS: A Language for Systems Programming", W.A. Wulf et al, CACM 14(12):780-790 (Dec 1971). Versions: CMU BLISS-10 for the PDP-10. CMU BLISS-11, a cross compiler for PDP-11 running on PDP-10, to support the C.mmp/Hydra project. DEC BLISS-32 for VAX/VMS.

Bloop - Douglas Hofstadter, 1979. Imperative language, designed for pedagogical purposes. Mechanizes primitive-recursive functions. Douglas Hofstadter, "Godel, Escher, and Bach: An Eternal Golden Braid", Basic Books, Chap 13. ISBN 0-465-02685-0. Implementation in Perl by John Cowan <cowan@locke.ccil.org>, 1994. (See Floop).  
ftp://locke.ccil.org/pub/retro/bloop.\*

Blosim - Block-Diagram Simulator. A block-diagram simulator. "A Tool for Structured Functional Simulation", D.G. Messerschmitt, IEEE J on Selected Areas in Comm, SAC-2(1):137-147, 1984.

BLOX - A visual language.

Blue - Softech. A language proposed to meet the DoD Ironman requirements which led to Ada. "On the BLUE Language Submitted to the DoD", E.W. Dijkstra, SIGPLAN Notices 13(10):10-15 (Oct 1978).

BMASF - Basic Module Algebra Specification Language? "Design of a Specification Language by Abstract Syntax Engineering", J.C.M. Baeten et al, in LNCS 490, pp.363-394.

BMDP - BioMedical Package. UCB, 1961. Statistical language, first implemented in FORTRAN for the IBM 7090.

BMF - Bird-Meertens Formalism. A calculus for derivation of a functional program from a given specification. "A Calculus of Functions for Program Derivation", R.S. Bird, in Res Topics in Fnl Prog, D. Turner ed, A-W 1990. Also known as Squiggol. "The Squiggolist", ed Johan Jeuring, published irregularly by CWI Amsterdam.

BNF - Backus Normal Form, later renamed Backus-Naur Form at the suggestion of Donald Knuth. A formalism to express the productions of context-free grammars. First used in the specification of Algol-58.

BNR Pascal - "Remote Rendezvous", N. Gammage et al, Soft Prac & Exp 17(10):741-755 (Oct 1987).

BNR Prolog - Constraint logic.

Bob - David Betz. A tiny object-oriented language. Dr Dobbs J, Sep 1991, p.26.  
<ftp://ftp.mv.com/pub/ddj/packages/bob15.arc>

BOEING - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May 1959).

Booster - Data parallel language. "The Booster Language", E. Paalvast, TR PL 89-ITI-B-18, Inst voor Toegepaste Informatica TNO, Delft, 1989.

BOPL - Basic Object Programming Language. Minimal object-based language for teaching. "Object-Oriented Syte Systems", J. Palsberg et al, Wiley, 1993.

BOSS - Bridgport Operating System Software. Derivative of the ISO 1054 numerical machine control language for milling, etc.

Boxer - Hal Abelson and Andy diSessa, Berkeley. A visual language, claims to be the successor to Logo. Boxes used to represent scope.

BRAVE - ?

BRIDGE - Component of ICES for civil engineers. Sammet 1969, p.616.

Bridgetalk - A visual language.

Brilliant - One of five pedagogical languages based on Markov algorithms, used in "Nonpareil, a Machine Level Machine Independent Language for the

Study of Semantics", B. Higman, ULICS Intl Report No ICSI 170, U London (1968). (cf. Diamond, Nonpareil, Pearl[3], Ruby[2]).

BRUIN - Brown University Interactive Language. Simple interactive language with PL/I-like syntax, for IBM 360. "Meeting the Computational Requirements of the University, Brown University Interactive Language", R.G. Munck, Proc 24th ACM Conf, 1969.

bs - A BASIC-like interactive language, really a sort of super-extended calculator utility, shipped with some early System V Unixes.

BSL -

1. Variant of IBM's PL/S systems language. Versions: BSL1, BSL2.

2. Backtracking Specification Language. A logic programming language fundamentally different from Prolog. A nondeterministic Algol-like language where variables cannot be assigned more than once except in controlled contexts. Each BSL program corresponds to an assertion in first order logic, and executing the program amounts to proving the assertion. Used to write an expert system CHORAL for harmonization of Bach-style chorales. "Report on the CHORAL Project: An Expert System for Chorale Harmonization", K. Ebcioğlu, RC 12628, IBM TJWRC, 1987.

BUGSYS - Pattern recognition and preparing animated movies, for IBM 7094 and IBM 360. "BUGSYS: A Programming System for Picture Processing - Not for Debugging", R.A. Ledley et al, CACM 9(2) (Feb 1966).

Burge's Language - Unnamed functional language based on lambda-calculus. Recursive Programming techniques", W.H. Burge, A-W 1975.

Butterfly Common LISP - Parallel version of Common LISP for the BBN Butterfly machine.

Butterfly Scheme - Parallel version of Scheme for the BBN Butterfly.

byacc - See yacc.

C - Dennis Ritchie, Bell Labs, ca. 1972. Originally a systems language for Unix on the PDP-11, briefly named NB. Influenced by BCPL through Thompson's B. Terse, low-level and permissive. Preprocessor. Partly due to its distribution with Unix, C became the language most widely used for software implementation.

K&R C - C as originally described. "The C Programming Language", Brian Kernighan & Dennis Ritchie, P-H 1978.

ANSI C - Revision of C, adding function prototypes, structure passing and assignment, and standardized library functions. ANSI X3.159-1989.

<ftp://ftp.cs.princeton.edu/pub/lcc/>\*

<http://www.cs.princeton.edu/software/lcc>

GNU C - Many extensions: compound statement within an expression, pointers to labels, local labels, nested functions, typeof operator, compound and conditional expressions and casts allowed as lvalues, long long ints, arrays of variable length macros with variable number of arguments, nonconstant initializers, constructor expressions, labeled elements in initializers, case ranges, variable attributes. "Using and Porting GNU CC", R.M. Stallman, 16 Dec 1992.

<ftp://prep.ai.mit.edu/pub/gnu/gcc-2.3.3.tar.Z>

C\* - Thinking Machines, 1987. Superset of ANSI C, object-oriented, data-parallel with synchronous semantics, for the Connection Machine. Adds a data type, the 'domain', and a selection statement for parallel execution in domains. J.R. Rose et al, "C\*: An Extended C Language for Data Parallel Programming", in Proc Second Intl Conf on Supercomputing, L.P. Kartashev et al eds, May 1987, pp.2-16. "C\* Programming Manual", Thinking Machines Corp, 1986. Version: 6.x  
info: customer-support@think.com  
documentation-order@think.com

C++ - Stroustrup <bs@alice.att.com>. An object-oriented superset of C. In C++ a class is a user-defined type, syntactically a struct with member functions. Constructors and destructors are member functions called to create or destroy instances. A friend is a nonmember function that is allowed to access the private portion of a class. C++ allows implicit type conversion, function inlining, overloading of operators and function names, default function arguments, and pass by reference. It has streams for I/O. "The C++ Programming Language", Bjarne Stroustrup, A-W 1986 (1st edition), 1991 (2nd edition).

ftp://grape.ecs.clarkson.edu/pub/msdos/djgpp/djgpp.zip for MS-DOS

ftp://prep.ai.mit.edu/pub/gnu/g++-1.39.0.tar.Z for Unix

draft ANSI C++ - X3J16 committee. Exceptions.

C++ release 2.0 - May 1989. Added multiple inheritance, type-safe linkage, pointers to members, abstract classes. "C++ 2.0 Draft Reference Manual"

C++ release 2.1 - Added nested types. "The Annotated C++ Reference Manual", M. Ellis et al, A-W 1990.

C++ release 3.0 - Added templates.

C++Linda - "The AUC C++Linda System", C. Callsen et al, U Aalborg, in Linda-Like Systems and Their Implementation, G. Wilson ed, U Edinburgh TR 91-13, 1991.

C+@ - (formerly Calico). Bell Labs. Object-oriented language, uniformly represents all data as a pointer to a self-described object. Provides multiple inheritance with delegation, with control over which methods come from which delegated object. Default methodologies. Simple syntax, with emphasis on graphics. Originally used for prototyping of telecommunication services. "A Dynamic C-Based Object-Oriented System for Unix", S. Engelstad et al, IEEE Software 8(3):73-85 (May 1991). "The C+@ Programming Language", J. Fleming, Dr Dobbs J, Oct 1993, pp.24-32. Implementation for SunOS, compiles to Vcode. Unir Tech, (800)222-8647.  
info: Jim Vandendorpe <jimvan@iexist.att.com>

C-10 - Improved version of COLINGO. Sammet 1969, p.702.

C with Classes - Short-lived predecessor to C++. "Classes: An Abstract Data Type Facility for the C Language", B. Stroustrup, CSTR-84 Bell Labs, Apr 1980. Also in SIGPLAN Notices (Jan 1982).

CADET - Computer Aided Design Experimental Translator. Sammet 1969, p.683.

CAFE - "Job Control Languages: MAXIMOP and CAFE", J. Brandon, Proc BCS Symp on Job Control Languages--Past Present and Future, NCC, Manchester, England 1974.

CAGE - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

CAJOLE - Dataflow language. "The Data Flow Programming Language CAJOLE: An Informal Introduction", C.L. Hankin et al, SIGPLAN Notices 16(7):35-44 (Jul 1981).

CAL - Course Author Language. CAI language for IBM 360. "Design of a Programming Language for Computer Assisted Learning", F.M. Tonge, Proc IFIP Congress 1968, v2.

Caliban - Kelly, Imperial College. Declarative annotation language, controlling the partitioning and placement of the evaluation of expressions in a distributed functional language. "Functional Programming for Loosely-coupled Multiprocessors", P. Kelly <phjk@doc.ic.ac.uk>, Pitman/MIT Press, 1989.

Calico - See C+@.

CAMAL - CAMbridge ALgebra system. Symbolic math used in Celestial Mechanics and General Relativity. Implemented in BCPL on Titan. "CAMAL User's Manual", John P. Fitch, Cambridge U, England (1975). "The Design of the Cambridge Algebra System", S.R. Bourne et al, Proc 2nd Symp of Symb & Alg Manip, SIGSAM 1971.

Camelot Library - "The Camelot Library", J. Bloch, in Guide to the Camelot Distributed Transaction Facility: Release I, A.Z. Spector et al eds, CMU 1988, pp.29-62.

CAMIL - Computer Assisted/Managed Instructional Language. Used for CAI at Lowry AFB, CO. "The CAMIL Programming Language", David Pflasterer, SIGPLAN Notices 13(11):43 (Nov 1978).

CAML -

1. Categorical Abstract Machine Language. G. Huet and G. Cousineau. A dialect of ML intermediate between LCF ML and SML. Lazy data structures. Built on the Categorical Abstract Machine. "The CAML Reference Manual", P. Weis et al, TR INRIA-ENS, 1989.

ftp://ftp.inria.fr/INRIA/caml/V3.1  
info: caml-light@margaux.inria.fr  
list:caml-list@margaux.inria.fr

2. Language for preparation of animated movies, listed [?] 1976.

CAML Light - Xavier Leroy. CAML subset. A small portable implementation, uses a bytecode interpreter written in C. Runs on Unix, MS-DOS, and Mac. Version: 0.6

ftp://ftp.inria.fr/lang/caml-light/\*  
info: caml-light@margaux.inria.fr

Candle - Language used in Scorpion environment development system. Related to IDL?

ftp://cs.arizona.edu/scorpion/\*  
info: scorpion-project@cs.arizona.edu

Cantor - Object-oriented language with fine-grained concurrency. Athas, Caltech 1987. "Multicomputers: Message Passing Concurrent Computers", W.

Athas et al, Computer 21(8):9-24 (Aug 1988).

CASE SOAP III - Version of SOAP assembly language for IBM 650. Listed in CACM 2(5):16 (May 1959).

CAT - Common Abstract Tree Language. R. Voeller & Uwe Schmidt, U Kiel, Germany 1983. Universal intermediate language, used by Norsk Data in their family of compilers. "A Multi-Language Compiler System with Automatically Generated Codegenerators, U. Schmidt et al, SIGPLAN Notices 19(6):202-2121 (June 1984).

CATO - FORTRAN-like CAI language for PLATO system on CDC 1604. "CSL PLATO System Manual", L.A. Fillman, U Illinois, June 1966.

C/ATLAS - DoD test language, variant of ATLAS.

CAYLEY - Symbolic math system for group theory. John Cannon, U Sydney, Australia, 1976. "An Introduction to the Group Theory Language CAYLEY", J. Cannon, Computational Group Theory, M.D. Atkinson ed, Academic Press 1984, pp.148-183. Current version: V3.7 for Sun, Apollo, VAX/VMS.  
info: cayley@maths.su.oz.au

CBASIC - Gordon Eubanks, now at Symantec. A BASIC compiler. Evolved from/into EBASIC.

cc - Concurrent Constraints. A family of languages generalizing CLP, including concurrency, atomic tell and blocking ask. "Concurrent Constraint Programming", V. Saraswat, MIT Press 1993.

CC++ - Compositional C++. Extensions to C++ for compositional parallel programming.

ftp://csvax.cs.caltech.edu/pub/comp  
info: Carl Kesselman <carl@vlsi.cs.caltech.edu>

CCalc - Symbolic math for MS-DOS, available from Simtel.

CCL -

1. Coral Common LISP.

2. Computer Control Language. English-like query language based on COLINGO, for IBM 1401 and IBM 1410.

CCLU - Cambridge CLU. G. Hamilton et al, CUCL. CLU extended to support concurrency, distributed programming, remote procedure calls.  
contact: Jean Bacon <jmb@cl.cam.ac.uk>

CCP - Concurrent Constraint Programming. Not a language, but a general approach.

CCS - Calculus of Communicating Systems. "A Calculus of Communicating Systems", LNCS 92, Springer 1980. "Communication and Concurrency", R. Milner, P-H 1989.

CCSP - Based on CSP. "Contextually Communicating Sequential Processes - A Software Engineering Approach", M. Hull et al, Software Prac & Exp 16(9):845-864 (Sept 1986).

CDIF - CASE Data Interchange Format. Used by Cadre and other CASE tool vendors.

CDL -

1. Computer Definition [Design?] Language. A hardware description language. "Computer Organization and Microprogramming", Yaohan Chu, P-H 1970.

2. Command Definition Language. Portion of ICES used to implement commands. Sammet 1969, p.618-620.

3. Compiler Description Language. C.H.A. Koster, 1969. Intended for implementation of the rules of an affix grammar by recursive procedures. A procedure may be a set of tree-structured alternatives, each alternative is executed until one successfully exits. Used in a portable COBOL-74 compiler from MPB, mprolog system from SzKI, and the Mephisto chess computer. "CDL: A Compiler Implementation Language", in Methods of Algorithmic Language Implementation, C.H.A. Koster, LNCS 47, Springer 1977, pp.341-351. "Using the CDL Compiler Compiler", C.H.A. Koster, 1974. Versions: CDL2 (used in an Algol 68 compiler at TU Berlin), CDLM used at Manchester.

4. Common Design Language. "Common Design Language", IBM, Software Engineering Inst, Sept 1983.

5. Control Definition Language. Ideas which contributed to Smalltalk. "Control Structures for Programming Languages", David A. Fisher, PhD Thesis, CMU 1970.

Cecil - Object-oriented language combining multi-methods with a classless object model, object-based encapsulation, and optional static type checking. Distinguishes between subtyping and code inheritance. Includes both explicit and implicit parameterization of objects, types, and methods. "The Cecil Language: Specification and Rationale", C. Chambers, TR 93-03-05, U Wash (Mar 1993).  
ftp://cs.uwashington.edu/pub/chambers/cecil-spec.ps.Z

Cedar - Xerox PARC. Superset of Mesa, adding garbage collection, dynamic types and a universal pointer type (REF ANY). A large complex language designed for custom Xerox hardware and the Cedar OS/environment. Data types: atoms, lists, ropes ("industrial strength" strings), conditions. Multiprocessing features include threads, monitors, signals and catch phrases. Used to develop the Cedar integrated programming environment. "A Description of the Cedar Language", Butler Lampson, Xerox PARC, CSL-83-15 (Dec 1983). "The Structure of Cedar", D. Swinehart et al, SIGPLAN Notices 20(7):230-244 (July 1985).

CEEMAC+ - Graphics language for DOS 3.3 on Apple [].

CELIP - A cellular language for image processing. "CELIP: A cellular Language for Image Processing", W. Hasselbring <willi@informatik.uni-essen.de>, Parallel Computing 14:99-109 (1990).

CELLAS - CELLular ASsemblies. A concurrent block-structured language. Mentioned in Attribute Grammars, LNCS 323, Springer, p.97.

CELLSIM - Modeling populations of biological cells. "CELLSIM II User's Manual", C.E. Donaghey, U Houston (Sep 1975).

CELP - Computationally Extended Logic Programming. "Computationally Extended Logic Programming", M.C. Rubenstein et al, Comp Langs 12(1):1-7 (1987).

CESP - Common ESP. AI Language Inst, Mitsubishi - Object-oriented extension of Prolog, a Unix-based version of ESP[3].  
info: cesp-request@air.co.jp

CESSL - CELL Space Simulation Language. Simulating cellular space models. "The CESSL Programming Language", D.R. Frantz, 012520-6-T, CS Dept, U Michigan (Sept 1971).

CFD - Computational Fluid Dynamics. FORTRAN-based parallel language for the Illiac IV.

CFP - Communicating Functional Processes. "Communicating Functional Processes", M.C. van Eekelen et al, TR 89-3, U Nijmegen, Netherlands, 1989.

CGGL - ("seagull") Code-Generator Generator Language. A machine-description language based on modeling the computer as a finite state machine. "A Code Generator Generator Language", M.K. Donegan et al, SIGPLAN Notices 14(8):58-64 (Aug 1979).

CGOL - V.R. Pratt, 1977. A package providing ALGOL-like surface syntax for MACLISP. "CGOL - An Alternative External Representation for LISP Users", V. Pratt, MIT AI Lab, Working Paper 89, 1976.  
ftp://mc.lcs.mit.edu/its/ai/lisp/cgol.fasl

CHAMIL - Sperry Univac. A Pascal-like microprogramming language. "CHAMIL - A Case Study in Microprogramming Design", T.G. Weidner, SIGPLAN Notices 15(1):156-166 (Jan 1980).

CHARITY - Cockett, Spencer, Fukushima, 1990-1991. Functional language based purely on category theory. "About Charity", J.R.B. Cockett <cockett@cpcs.ucalgary.ca> et al. Version for Sun4 available from Tom Fukushima <fukushim@cpsc.ucalgary.ca>.

CHARM -

1. An explicitly parallel programming language based on C, for both shared and nonshared MIMD machines. "The CHARM(3.2) Programming Language Manual", UIUC (Dec 1992)  
ftp://a.cs.uiuc.edu/pub/CHARM  
list:charm@cs.uiuc.edu

2. Peter Nowosad, 1990. Block-structured imperative language, strongly typed, with ideas borrowed from Pascal, C, RTL2 and ARM Assembler. PD compiler for the Acorn. Review in Archive magazine?

CHARM++ - An object-oriented parallel programming system, similar to CHARM but based on C++. TR 1796, UIUC.  
ftp://a.cs.uiuc.edu/pub/CHARM/Charm++  
info: Sanjeev Krishnan <sanjeev@cs.uiuc.edu>

Charme - Bull, 1989. A language with discrete combinatorial constraint logic aimed at industrial problems such as planning and scheduling. Implemented in C. An outgrowth of ideas from CHIP. Semantically nondeterministic, with choice and backtracking, similar to Prolog. "Charme Reference Manual", AI Development Centre, Bull, France 1990.  
info: cras@bull.fr

CHARYBDIS - LISP program to display math expressions. Related to MATHLAB. Sammet 1969, p.522.

CHASM - CHear ASseMbler. Shareware assembler for MS-DOS.

CHI - A wide spectrum language, the forerunner of Refine. "Research on Knowledge-Based Software Environments at Kestrel Institute", D.R. Smith et al, IEEE Trans Soft Eng, SE-11(11) (1985).

CHILI - D.L. Abt. Language for systems programming, based on ALGOL 60 with extensions for structure and type declarations. "CHILI, An Algorithmic Language for Systems Programming", CHI-1014, Chi Corp (Sep 1975).

CHILL - CCITT HIGH-Level Language. ca. 1980. Real-time language widely used in European telecommunications. "An Analytical Description of CHILL, the CCITT High Level Language", P. Branquart, LNCS 128, Springer 1982. "CHILL User's Manual", ITU, 1986, ISBN 92-61-02601-X. CCITT/ISO/IEC International Standard ISO/IEC 9496, Recommendation Z.200, ISBN 82-61-03801-8 ("The Blue Book", 1988?). Versions: CHILL-80, CHILL-84, CHILL-88. Forthcoming compiler by Cygnus, based on gcc.

CHIP -

1. Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May 1959).

2. Constraint Handling In Prolog. M. Dincbas, ECRC Munich 1985. Constraint logic language, includes boolean unification and a symbolic simplex-like algorithm. Constraints over integers, rationals and booleans. Symbolic constraints, cumulative constraints, and update demons. Introduced the domain-variable model. "The Constraint Logic Programming Language CHIP", M. Dincbas et al, Proc 2nd Intl Conf on Fifth Generation Computer Sys, Tokyo (Nov 1988), pp.249-264. "Constraint Satisfaction in Logic Programming", Van Hentenryck. V4 available from COSYTEC, 4 rue Jean Rostand, F91893 Orsay, France.  
info: <cosytec@cosytec.fr>

CHIP-48 - Reimplementation of CHIP-8 for the HP-48 calculator. Andreas Gustafson <gson@niksula.hut.fi>, comp.sys.handhelds, Sep 1990.  
ftp://vega.hut.fi/pub/misc/hp48sx/asap/\*

CHIP-8 - RCA, Late 70's. Low-level language (really a high-level machine code) for video games on computers using RCA's CDP1802 processor: COSMAC VIP, DREAM 6800 and ETI-660. Now there's an interpreter for the Amiga.  
ftp://ftp.cso.uiuc.edu/pub/amiga/fish/f5/ff537/CHIP8.lzh

CHISEL - An extension of C for VLSI design, implemented as a C preprocessor. It produces CIF as output. "CHISEL - An Extension to the Programming language C for VLSI Layout", K. Karplus, PHD Thesis, Stanford U, 1982.

CHOCs - Generalization of CCS. "A Calculus of Higher-Order Communicating Systems", B. Thomsen, 16th POPL pp.143-154 (1989).

CIAL - Interval constraint logic language. Contains a linear Gauss-Seidel constraint solver, in addition to the interval narrowing solver. Implemented as an extension to CLP(R). "Towards Practical Interval Constraint Solving in Logic Programming", C.K. Chiu et al, TR, Chinese U Hong Kong, 1994. Version 1.0 (beta)  
info: Jimmy Lee <cial@cs.cuhk.hk>

CIEL - Object-oriented Prolog-like language. "CIEL: Classes et Instances En Logique", M. Gandriau, Thesis ENSEEIHT (1988).

CIF - Caltech Intermediate Form. Geometry language for VLSI design, in which the primitives are colored rectangles. Mead & Conway, "Introduction to VLSI Systems", A-W 1980, Section 4.5.

Cigale - A parser generator language with extensible syntax. "CIGALE: A Tool for Interactive Grammar Construction and Expression Parsing", F. Voisin, Sci Comp Prog 7:61-86 (1986).

CIL - Common Intermediate Language. "Construction of a Transportable, Multi-Pass Compiler for Extended Pascal", G.J. Hansen et al, SIGPLAN Notices 14(8):117-126 (Aug 1979).

CIMS PL/I - Courant Institute of Mathematical Sciences PL/I. A PL/I subset. "CIMS PL/I", P.W. Abrahams, Courant Inst.

CIP-L - CIP Language. (CIP stands for Computer-aided Intuition-guided Programming.) Wide-spectrum language for incremental program transformation. There are ALGOL-like and Pascal-like variants. "The Munich Project CIP, v.I: The Wide Spectrum Language CIP-L", LNCS 183, Springer 1984. Version: CIP85.

CIRCAL - "CIRCAL and the Representation of Communication, Concurrency and Time", G.J. Mitre, ACM TOPLAS 7(2):270-298 (1985).

CITRAN - Caltech's answer to MIT's JOSS. Sammet 1969, p.217.

CL - Control Language. Batch language for the IBM RPG/38, used in conjunction with RPG III. (See OCL).

CLAM -

1. Symbolic math, especially General Relativity. Implemented in ATLAS assembly language first, LISP later. "CLAM Programmer's Manual", Ray d'Inverno & Russell-Clark, King's College London, 1971. (See ALAM).

2. Constraint Language Abstract Machine. The underlying abstract machine in the implementation of CLP(R). Based on the WAM. "An Abstract Machine for CLP(R)", J. Joffar et al, Proc ACM SIGPLAN 92 Conf on Prog Lang and Impl, 1992, pp.128-139.

Clarion - MS-DOS 4GL.

CLASP - Computer Language for Aeronautics and Programming. NASA. Real-

time language with focus on fixed-point math. Near subset of SPL[2], with some ideas from PL/I. "Flight Computer and Language Processor Study", Raymond J. Rubey, Management Information Services, Detroit, 1971.

Classic-Ada - Object-oriented extension to Ada, said to be Smalltalk-like. Implemented as an Ada preprocessor.

Clean - Subset of Lean. Experimental lazy higher-order functional language with no syntactic sugaring (not even infix expressions or complex lists.) Also used as an intermediate language. Implemented via graph rewriting on the ABC abstract machine. "Clean - A Language for Functional Graph Rewriting", T. Brus et al, IR 95, U Nijmegen, Feb 1987. (See Concurrent Clean).

CLEAR - Specification language based on initial algebras. "An Informal Introduction to Specification Using CLEAR", R.M. Burstall in The Correctness Problem in Computer Science, R.S. Boyer et al eds, A-P 1981, pp.185-213.

CLEO - Clear Language for Expressing Orders. ICL, 1960's. Used until early 1972 on Leo III mainframes.

C-Linda - The most widely used variant of Linda, with C as the base language. Available from Sci Comp Assocs <linda@sca.com>.

CLIP -

1. Compiler Language for Information Processing. 1958-1959. Based on IAL, led to JOVIAL. One of the first languages used to write its own compiler. Sammet 1969, p.635.

2. Common LISP in Parallel. Allegro. Version for the Sequent Symmetry.

Clipper - Compiled dBASE dialect from Nantucket Corp, LA. Versions: Winter 85, Spring 86, Autumn 86, Summer 87, 4.5 (Japanese Kanji), 5.0.

CLIPS - C Language Integrated Production System. NASA JSC. A language for developing expert systems, with the inferencing and representation capabilities of OPS5, and support for three paradigms: forward chaining rule-based, object-oriented and procedural. LISP-like syntax. Available for MS-DOS, comes with source code in C. COSMIC, U Georgia, (404) 542-3265. Austin Code Works <info@acw.com> (512)258-0785. "Expert Systems: Principles and Programming", Joseph Giarratano and Gary Riley, PWS Publ 1994, ISBN 0-534-93744-6. Versions: CLIPS 5.1, CLIPS/Ada 4.3, CLIPS6.0 (See PCLIPS).

info: service@cossack.cosmic.uga.edu

telnet: cosline@cosmic.uga.edu

ftp://earth.rs.itd.umich.edu/mac.bin/etc/compsci/Clips/CLIPS 4.20

//ftp.ensmp.fr/pub/clips/clips-5.1/dos/clips

//ftp.ensmp.fr/pub/clips/clips-5.0/doc/mac-clips-50

list: CLIPS-LIST@UGA.BITNET

CLISP - Conversational LISP. A mixed English-like, Algol-like surface syntax for Interlisp. "CLISP: Conversational LISP", W. Teitelman, in Proc Third Intl Joint Conf on AI, Stanford, Aug 1973, pp.686-690.

CLIX - "Overview of a Parallel Object-Oriented Language CLIX", J. Hur et

al, in ECOOP '87, LNCS 276, Springer 1987, pp.265-273.

Clock - ? Mentioned in the documentation for TXL.

CLOS - Common LISP Object System. Object-oriented extension to Common LISP, based on generic functions, multiple inheritance, declarative method combination and a meta-object protocol. A descendant of CommonLoops. "Common LISP Object System Specification X3J13 Document 88-002R", D.G. Bobrow et al, SIGPLAN Notices 23 (Sep 1988). (See PCL[2]).  
ftp://parcftp.xerox.com/pcl/\*

CLP -

1. Cornell List Processor. List processing language, an extension of CORC, used for simulation. Sammet 1969, p.461.

2. Constraint Logic Programming. A programming framework based (as Prolog) on LUSH (or SLD) resolution, but in which unification has been replaced by a constraint solver. A CLP interpreter contains a Prolog-like inference engine and an incremental constraint solver. The engine sends constraints to the solver one at a time. If the new constraint is consistent with the collected constraints it will be added to the set. If it was inconsistent, it will cause the engine to backtrack. "Constraint Logic Programming", J. Jaffar et al, 14th POPL, ACM 1987.

CLP(R) - Constraint Logic Programming (Real). Joxan Jaffar, TJWRC & S. Michaylov, Monash U, 1986. A constraint-logic programming language with real-arithmetic constraints. A superset of Prolog. "The CLP(R) Language and System", J. Jaffar et al, IBM RR RC16292 (#72336) (Nov 1990). Version: 1.2 for Unix, MS-DOS and OS/2, available from the author.  
info: <joxan@watson.ibm.com>

CLP\* - Derivative of CLP. "CLP\* and Constraint Abstraction", T. Hickey, 16th POPL, ACM 1989, pp.125-133.

CLP(sigma\*) - "CLP(sigma\*): Constraint Logic Programming with Regular Sets", C. Walinsky, Proc ICLP, 1989, pp.181-190.

CLU - CLUster. 1974-1975. CLU is an object-oriented language of the Pascal family designed to support data abstraction, similar to Alphard. Introduced the iterator: a coroutine yielding the elements of a data object, to be used as the sequence of values in a 'for' loop.

A CLU program consists of separately compilable procedures, clusters and iterators, no nesting. A cluster is a module naming an abstract type and its operations, its internal representation and implementation. Clusters and iterators may be generic. Supplying actual constant values for the parameters instantiates the module.

There are no implicit type conversions. In a cluster, the explicit type conversions 'up' and 'down' change between the abstract type and the representation. There is a universal type 'any', and a procedure force[] to check that an object is a certain type. Objects may be mutable or immutable. Garbage collection is built in.

Exceptions are raised using 'signal' and handled with 'except'. Assignment is by sharing, similar to the sharing of data objects in LISP. Arguments are passed by call-by-sharing, similar to call by value, except that the arguments are objects and can be changed only if they are mutable. CLU has own variables and multiple assignment.

TED (a text editor), R (a document formatter), SWIFT (an operating system), and lp (a proof tool used for formal specification) have been written in CLU.

"Abstraction and Specification in Program Development", Barbara Liskov and John Guttag, McGraw-Hill, 1986. "CLU Reference Manual", Barbara Liskov et al, LNCS 114, Springer 1981.

contact: Paul R. Johnson <prj@pm-prj.lcs.mit.edu>

ftp://pion.lcs.mit.edu/pub/clu/\* - versions for Sun, VAX/VMS.

//ftp.lcs.mit.edu/pub/pclu/\* - portable version, compiles to C

//ftp.is.titech.ac.jp/pub/clu2c/\* - compiles to C

info: clu2c-report@is.titech.ac.jp

Cluster 86 - Shang, Nanjing U ca1986. Distributed object-oriented language. A cluster is a metatype. "Cluster: An Informal Report", L. Shang <lshang@inf.ethz.ch>, SIGPLAN Notices 26(1):57-76 (Jan 1991). Versions for MS-DOS, Unix.

CMAY - "A Microkernel for Distributed Applications", R. Bagrodia et al, Proc 5th Intl Conf Distrib Comp Sys IEEE 1985, pp.140-149.

CML -

1. A query language. "Towards a Knowledge Description Language", A. Borgida et al, in On Knowledge Base Management Systems, J. Mylopoulos et al eds, Springer 1986.

2. Concurrent ML. J. Reppy, Cornell 1990. A concurrent extension of SML/NJ, supporting dynamic thread creation and synchronous message passing on typed channels. Threads are implemented using first-class continuations. "CML: A Higher-Order Concurrent Language", John H. Reppy, SIGPLAN Notices 26(6):293-305 (June 1991).

ftp://ftp.cs.cornell.edu/pub/CML-0.9.8.tar.Z

info: cml-bugs@cs.cornell.edu

Cmm - C Minus Minus. Scripting language.

ftp://ftp.std.com/vendors/CEnvi-Cmm/share

CMS-2 - General purpose language used for command and control applications in the US Navy. Variants: CMS-2M and CMS-2Y. "CMS-2Y Programmers Reference Manual", M-5049, PDCSSA, San Diego CA (Oct 1976).

CO2 - (a blend of C and O2). Object-oriented database language. GIP Altair, Versailles, France. Francois Bancilon et al, in Advances in Object-Oriented Database Systems, K.R. Dittrich ed, LNCS 334, Springer 1988.

COALA - "COALA: The Object Code of the Compiler Producing System", S. Kruszewski et al, MERA, Warsaw 1974.

COBOL - COmmon Business Oriented Language. 1960. CODASYL Committee, Apr 1960. Simple computations on large amounts of data. The most widely used programming language today. The natural language style is intended to be largely self-documenting. Introduced the record structure. "Initial Specifications for a Common Business Oriented Language" DoD, US GPO, Apr 1960. Major revisions in 1968 (ANS X3.23-1968), 1974 (ANS X3.23-1974), and 1985. A new ISO standard is expected in 1997.

COBOL-1961 Extended - Short-lived separation of COBOL specifications.  
Sammet 1969, p.339.

CoCoA - [Symbolic math? On a Radio Shack CoCo??? I have no idea.]

Cocol - Coco Language. A language for writing left-attributed LL(1) grammars. Syntactic resemblance to Modula-2. Used as the input language for the Coco LL(1) parser generator, which produces Modula-2 output. "A Compiler Generator for Microcomputers", P. Rechenberg et al, P-H 1989. Version: Cocol-2 for the Coco-2 generator.  
ftp://neptune.inf.ethz.ch/pub/Coco (Oberon and modula-2 versions)  
//cs.ru.ac.za/pub/coco (C version)

Code 2.0 - Large-grain dataflow language. Has a graphical interface for users to draw communication structure. "The CODE 2.0 Parallel Programming Language", P. Newton et al, Proc ACM Intl Conf on Supercomput, Jul 1992.  
info: Peter Newton <newton@cs.utexas.edu>

CODIL - COntext Dependent Information Language. Early language for non-numerical business problems. "CODIL, Part1. The Importance of Flexibility", C.F. Reynolds et al, Computer J 14(3):217-220 (May 1971).

COFF - Common Object File Format. Binary file format used by Unix System V Release 3.

COGENT - COmpiler and GENeralized Translator. Compiler writing language with pattern-directed string and list processing features, for CDC 3600 and CDC 3800. A program consists of productions defining a context-free language, plus analysis and synthesis function generators, "COGENT Programming Manual", J.C. Reynolds, ANL-7022, Argonne, Mar 1965. Sammet 1969, p.638. "An Introduction to the COGENT System", J.C. Reynolds, Proc ACM 20th Natl Conf, 1965.

COGO - Co-ordinate geometry problems in Civil Engineering. A subsystem of ICES. "Engineer's Guide to ICES COGO I", R67-46, CE Dept MIT (Aug 1967).

Coherent Parallel C - Data parallel language. "Coherent Parallel C", E. Felten et al in Third Conf on Hypercube Concurrent Computers and Appls, ACM, 1988, pp.440-450.

COIF - FORTRAN with interactive graphic extensions for circuit design, on UNIVAC 1108. "An Interactive Software System for Computer-Aided Design: An Application to Circuit Projects", CACM 9(13) (Sep 1970).

COLASL - Early system for numerical problems on IBM 7030. Special character set for input of natural math expressions. Sammet 1969, pp.265-271.

COLD - A sugared version of COLD-K.

COLD-K - Formal design kernel language for describing (sequential) software systems in intermediate stages of their design. "An Introduction to COLD-K", H.B.M. Jonkers in Algebraic Methods: Theory, Tools and Applications, M. Wirsing et al eds, LNCS 394, Springer 1989, pp.139-205.

COLINGO - Compile On-LINE and GO. MITRE Corp. English-like query system for IBM 1401. "The COLINGO System Design Philosophy", Information System

Sciences, Proc Second Congress, 1965. Sammet 1969, p.664.

COMAL - COMMon Algorithmic Language. Benedict Loefstedt & Borge Christensen, 1973. A language for beginners, popular in Europe and Scandinavia. Pascal-like structure added to BASIC. COMAL-80 has been adopted as an introductory language in Denmark. "Beginning COMAL", B. Christensen, Ellis Harwood 1982. COMAL User's Group, 5501 Groveland Terr, Madison WI 53716. Version for Amiga.

COMIT - Victor H. Yngve, MIT, 1957-8. The first string-handling and pattern-matching language, designed for applications in natural language translation. The user has a workspace organized into shelves. Strings are made of constituents (words), accessed by subscript. A program is a set of rules, each of which has a pattern, a replacement and goto another rule. Implemented on IBM 7090. "COMIT Programmer's Reference Manual", V.H. Yngve, MIT Press 1961. Sammet 1969, pp.416-436.

COMIT II - "Computer Programming with COMIT II", Victor H. Yngve, MIT Press, 1963.

Comma - COMputable MATHematics. Esprit project at KU Nijmegen.

COMMEN -L.J. Cohen. Proc SJCC 30:671-676, AFIPS (Spring 1967).

Commercial Translator - English-like pre-COBOL language for business data processing. Sammet 1969, p.378.

Common LISP - An effort begun in 1981 to provide a common dialect of LISP. The result is a large and complex language, fairly close to a superset of MacLisp. Lexical binding, data structures using defstruct and setf, closures, multiple values, types using declare, a variety of numerical types. Function calls allow optional, keyword and &rest arguments. Generic sequence can either be a list or an array. Formatted printing using escape characters. Common LISP now includes CLOS, an extended LOOP macro, condition system, pretty printing, logical pathnames. "Common LISP: The Language", Guy L. Steele, Digital Press 1984, ISBN 0-932376-41-X. "Common LISP: The Language, 2nd Edition", Guy L. Steele, Digital Press 1990, ISBN 1-55558-041-6.

(See AKCL, CCL, DCL, KCL, MCL)

list: common-lisp@ai.sri.com.

ftp://lisp-rtl.slisp.cs.cmu.edu/pub/16e-\* CMU Common LISP Version 16e

//ftp.think.com/public/think/lisp:public-review.text

Draft proposed ANS Common Lisp

CommonLoops - Xerox. An object-oriented LISP. Led to CLOS. "CommonLoops: Merging Lisp and Object-Oriented Programming", D.G. Bobrow et al, SIGPLAN Notices 21(11):17-29 (Nov 1986). (See CLOS, PCL).

ftp://arisia.xerox.com/pub/pcl/September-16-92-PCL-c.tar.Z - Pcl (Portable CommonLoops)

info: CommonLoops@xerox.com

Common Objects - H-P. An object-oriented LISP. "Inheritance and the Development of Encapsulated Software Components", A. Snyder, Proc 20th Hawaii Conf on Sys Sci, pp.227-238 (1987).

Compact COBOL - Subset of COBOL defined, but not published, ca. 1961. Sammet 1969, p.339.

Compas Pascal - Predecessor of Turbo Pascal, by POLY Data of Denmark. Later renamed POLY Pascal, and afterwards sold to Borland.

COMPASS - COMPrehensive ASSEMBler. Assembly language on CDC machines.

Compel - COMpute Parallel. The first single-assignment language. "A Language Design for Concurrent Processes", L.G. Tesler et al, Proc SJCC 32:403-408, AFIPS (Spring 1968).

Compiler-Compiler - Early compiler generator for the Atlas, with its own distinctive input language. "The Compiler-Compiler", R.A. Brooker et al, Ann Rev Automatic Programming 3:229-275, Pergamon 1963.

COMPL - "The COMPL Language and Operating System", A.G. Fraser et al, Computer J 9(2):144-156 (1966).

COMPREHENSIVE - Early system on MIT's Whirlwind. Listed in CACM 2(5):16 (May 1959).

COMPROSL - COMpound PROcedural Scientific Language. Language for scientists or engineers. Sammet 1969, p.299-300.

Computer Animation Movie Language. "A Computer Animation Movie Language for Educational Motion Pictures", D.D. Weiner et al, Proc FJCC 33(2), AFIPS (Fall 1968).

Computer Compiler - Proposed language for compiler design. Sammet 1969, p.695.

Computer Design Language - ALGOL-like language for computer design. "An ALGOL-like Computer Design Language", Y. Chu, CACM 8(10) (Oct 1965).

COMSL - COMMunication System Simulation Language. "COMSL - A Communication System Simulation Language", R.L. Granger, Proc FJCC 37 (1970).

COMTRAN - "Communications Computer Language COMTRAN", D.W. Clark et al, RADC-TR-69-190, Rose Air Development Center, Griffiss AFB, NY (July 1969). Sammet 1969, p.324, 331.

ConC - Concurrent extension of C based on DPN (decomposed Petri nets), using 'handshake' and 'unit' constructs. "ConC: A Language for Distributed Real-Time Programming", V.K. Garg et al, Computer Langs 16(1):5-18 (1991).

Concert/C - IBM TJWRC, July 1993. A parallel extension of ANSI C with asynchronous message passing.

<ftp://software.watson.ibm.com/pub/concert/cncrt.B1.tar.Z>  
info: concert-c@watson.ibm.com

CONCUR - "CONCUR, A Language for Continuous Concurrent Processes", R.M. Salter et al, Comp Langs 5(3):163-189 (1981).

Concurrent Aggregates (CA) - 1990. Concurrent object-oriented language based on the Actor model plus RPC. Pure object oriented, single inheritance, with first class selectors, continuations and messages. "Concurrent Aggregates: Supporting Modularity in Massively Parallel Programs", Andrew A. Chien. Compiler for CM5 and workstations.

definition:<http://www-csag.cs.uiuc.edu/papers/ca-report.ps>  
implementation:<http://www-csag.cs.uiuc.edu/projects/concert/release.html>  
info:<http://www-csag.cs.uiuc.edu>

#### Concurrent C -

1. Extension of C with rendezvous-based concurrency. "Concurrent C", N.H. Gehani et al, Soft Prac & Exp 16(9):821-844 (1986). "The Concurrent C Programming Language", N. Gehani et al, Silicon Press 1989. Versions for most Unix systems available commercially from AT&T.

2. Extension of C with asynchronous message passing. [NOT the same as above] "Concurrent C: A Language for Distributed Systems", Y. Tsujino et al, Soft Prac & Exp 14(11):1061-1078 (Nov 1984).

Concurrent C++ - "Concurrent C++: Concurrent Programming with Class(es)", N. Gehani, Bell labs 1986.

Concurrent Clean - An implementation of CFP. A version of Clean for loosely coupled parallel architectures. Lazy, purely functional. Strongly typed (Milner/Mycroft), modules, functional I/O (including windows and mouse). Compiles to the PABC machine, based on graph rewriting. "Concurrent Clean", M.C. van Eekelen et al, TR 89-18, U Nijmegen, Netherlands, 1989. Version: 0.8.1, October 1992.  
[ftp://ftp.cs.kun.nl/pub/Clean/\\*](ftp://ftp.cs.kun.nl/pub/Clean/*) - simulator for Mac, Sun3, Sun4  
info: <[clean@cs.kun.nl](mailto:clean@cs.kun.nl)>

Concurrent CLU - Hamilton, Cambridge U, 1984. "Preserving Abstraction in Concurrent Programming", R. Cooper et al, IEEE Trans Soft Eng SE-14(2):258-263 (Feb 1988).

Concurrent Euclid - J.R. Cordy & R.C. Holt, U Toronto, 1980. Subset of Euclid ("Simple Euclid") with concurrent extensions. Separate compilation, modules, processes and monitors, signal and wait on condition variables. 'Converters' to defeat strong type checking, absolute addresses. All procedures and functions are re-entrant. TUNIS (a Unix-like OS) is written in Concurrent Euclid. "Specification of Concurrent Euclid", J.R. Cordy & R.C. Holt, Reports CSRI-115 & CSRI-133, CSRI, U Toronto, Jul 1980, rev. Aug 1981. "Concurrent Euclid, The Unix System, and Tunis," R.C. Holt, A-W, 1983.

Concurrent LISP - "A Multi-Processor System for Concurrent Lisp", S. Sugimoto et al, Proc 1983 Intl Conf parallel Proc, 1983 pp.135-143.

Concurrent Oberon - not a separate language, but rather a modification of the Oberon system.

Concurrent Pascal - Brinch Hansen, 1972-75. Extension of a Pascal subset, Sequential Pascal. The first language to support monitors. Access to hardware devices through monitor calls. Also processes and classes. "The Programming Language Concurrent Pascal", Per Brinch Hansen, IEEE Trans Soft Eng 1(2):199-207 (Jun 1975).

Concurrent Prolog - Ehud "Udi" Shapiro, Yale <[shapiro-ehud@yale.edu](mailto:shapiro-ehud@yale.edu)>. Guarded clauses and committed-choice [= dont-care] nondeterminism. A subset's been implemented, but not the full language. "Concurrent Prolog: Collected Papers", E. Shapiro, V.1-2, MIT Press 1987. (See Mandala).

Concurrent Scheme - M. Swanson <swanson%teewinot@cs.utah.edu>. A parallel Lisp, for the Mayfly. "Concurrent Scheme", R.R. Kessler et al, in Parallel Lisp: Languages and Systems, T. Ito et al eds, LNCS 441, Springer 1989.

ConcurrentSmalltalk - Concurrent variant of Smalltalk (what did you expect). "Concurrent Programming in ConcurrentSmalltalk", Y. Yokote et al in Object-Oriented Concurrent Programming, A. Yonezawa et al eds, MIT Press 1987, pp.129-158.

condela - Connection Definition Language. A language for defining neural nets. Procedural and parallel.  
ftp://tut.cis.ohio-state.edu/pub/condela

CONIC - "Dynamic Configuration for Distributed Systems", J. Kramer et al, IEEE Trans Soft Eng SE-11(4):424-436 (Apr 1985).

Connection Machine LISP - LISP with a parallel data structure, the 'xapping', an array of values assigned to an array of sites. G.L. Steele et al, "Connection Machine LISP: Fine-Grained Parallel Symbolic Processing", in Proc 1986 ACM Conf on LISP and Functional Prog, Aug 1986, pp.279-297. "Connection Machine LISP Reference Manual", Thinking Machines Corp, Feb 1987.

CONNIVER - AI language for automatic theorem proving. An outgrowth of PLANNER, based on coroutines rather than backtracking. Allowed multiple database contexts with hypothetical assertions. "The CONNIVER Reference Manual", D. McDermott & G.J. Sussman <gjs@zurich.ai.mit.edu>, AI Memo 259, MIT AI Lab, 1973.

ConstraintLisp - Object-oriented constraint language based on CSP. An extension of Common Lisp and CLOS. "ConstraintLisp: An Object-Oriented Constraint Programming Language", Bing Liu <bing@iti.gov.sg> et al, SIGPLAN Notices 27(11):17-26 (Nov 1992).

CONSTRAINTS - Constraints using value inference. "CONSTRAINTS: A Language for Expressing Almost-Hierarchical Descriptions", G.J. Sussman et al, Artif Intell 14(1):1-39 (Aug 1980).

Consul - Constraint-based [future-based?] language with LISP-like syntax. "Consul: A Parallel Constraint Language", D. Baldwin, IEEE Software 6(4):62-71.

CONVERT -

1. String processing language, combined the pattern matching and transformation operations of COMIT with the recursive data structures of Lisp. "Convert", A. Guzman et al, CACM 9(8):604-615 (Aug 1966).

2. Early language to convert programs and data from one language to another. "CONVERT Manual", OLI Systems Inc (Oct 1976).

cooC - Concurrent Object-Oriented C. Toshiba. Concurrent object execution with synchronous or asynchronous message passing. Implemented for SunOS. SIGPLAN Notices 28(2) [?]  
ftp://tsbgw.isl.rdc.toshiba.co.jp/pub/toshiba/cooc-beta.1.1.tar.Z  
info:cooc@isl.rdc.toshiba.co.jp

COOL -

1. Concurrent Object-Oriented Language. An extension of C++ with task-level parallelism for shared-memory multiprocessors. "COOL: A Language for Parallel Programming", R. Chandra et al in Languages and Compilers for Parallel Computing, D. Gelernter et al eds, MIT Press 1990, pp.126-148.

info: Rohit Chandra <rohit@cool.stanford.edu>

2. CLIPS Object-Oriented Language?

CORAL -

1. Class Oriented Ring Associated Language. L.G. Roberts, MIT 1964. Graphical display and systems programming on the TX-2. Used "rings" (circular lists) from Sketchpad. "Graphical Communication and Control Languages", L.B. Roberts, Information System Sciences: Proc Second Congress, 1965. Sammet 1969, p.462.

2. U Wisconsin Madison. Language for deductive database. Prolog-like syntax with SQL-like extensions. Many evaluation techniques are supported. Implemented in C++.

3. Royal Signals and Radar Establishment, Malvern UK. Real-time system programming language, a derivative of JOVIAL and ALGOL-60. Strongly associated with MASCOT, a design technique for real time systems. Adopted as the British military standard from 1970 until the arrival of Ada. Versions: CORAL 64, CORAL 66. "Official Definition of CORAL 66", P.M. Woodward et al, HMSO, London, 1970, ISBN 0-11-470221-7.

CORBIE - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

CORC - CORnell Compiler. Simple language for student math problems. "The Cornell Computing Language", R.W. Conway et al, CACM 6(6):317-320 (Jun 1963) Sammet 1969, p.294-296.

Coroutine Pascal - "Control Separation in Programming languages", Lemon et al, ACM Ann Conf 1977.

CORREGATE - Based on IT. Sammet 1969, p.139.

Correlatives and Conversions -Data description language used in the Pick OS. "Exploring the Pick Operating System", J.E. Sisk et al, Hayden 1986.

CORTL - Carl McConnell. Intermediate language, a form of RTL?  
info: mcconnell@cs.uiuc.edu

Coursewriter III - ca. 1976. Simple CAI language. "Coursewriter III, Version 3 Author's Guide", SH20-1009, IBM.

COWSEL - COntrolled Working SpacE Language. Burstall and Popplestone, U Edinburgh, 1964-66. LISP-like semantics with FORTH-like stack, and reverse Polish syntax. Forerunner of POP. EPU-R-12, U Edinburgh (Apr 1966).

CP - A concurrent Prolog. "The Concurrent Logic Programming Language CP": Definition and Operational Semantics", V. Saraswat, 14th POPL, ACM 1987, pp.49-62.

CParaOps5 - Anurag Acharya, <acharya@cs.cmu.edu>. Parallel version of OPS5, written in C and compiling to C. Available for Unix, Mach, Encore Multimaxen, and Sequent. Version: 5.4.  
ftp://dravido.soar.cs.cmu.edu//usr/nemo/cparaops5/CParaOPS5-5.4.tar.Z

CPL -

1. Combined Programming Language. U Cambridge and U London. A very complex language, syntactically based on ALGOL-60, with a pure functional subset. Provides the ..where.. form of local definitions. Strongly typed but has a "general" type enabling a weak form of polymorphism. Functions may be defined as either normal or applicative order. Typed array and polymorphic list structures. List selection is through structure matching. Partially implemented on the Titan (Atlas 2) computer at Cambridge. Led to the much simpler BCPL. "The Main Features of CPL", D.W. Barron et al, Computer J 6(2):134-143 (Jul 1963).

2. Conversational Programming Language. DEC. Language similar to PL/I, for DEC-10 and DEC-20. Manual: DEC-10-LCPLA-B-D.

CPS -

1. Conversational Programming System. Allen-Babcock Corp, 1965. Interactive extended subset of PL/I. "Conversational Programming System under TSO (PBPO), Terminal User's Manual", SH20-1197, IBM. Sammet 1969, p.232-240.

2. Continuation Passing Style. A semantically clean language with continuations, has been used as an intermediate language for Scheme and the SML/NJ compiler. "Rabbit: A Compiler for Scheme", G.L. Steele, AI-TR-474, MIT (May 1978). "Compiling With Continuations", A. Appel, Cambridge U Press 1992.

C-Refine - Lutz Prechelt <prechelt@ira.uka.de> Symbolic naming of code fragments to redistribute complexity and provide running commentary. Implemented as a C preprocessor.  
ftp://ftp.uu.net/usenet/comp.sources.reviewed/volume02/crefine

CRISP - Jeff Barnett, SDC, Santa Monica CA, early 70's. A LISP-like language and compiler for the IBM 370. Differences from LISP included a generalization of 2-part cons nodes to n-part.

CRL - Carnegie Representation Language. (c)Carnegie Group Inc. Frame language derived from SRL. Written in Common LISP. Used in the product Knowledge Craft.

CROSSTABS - Simple language for statistical analysis of tabular data. "User's Manual for the CROSSTABS System", Cambridge Computer Assoc (Feb 1977).

Crystal - Concurrent Representation of Your Space-Time ALgorithms. A recursion equation parallel language. "A Parallel Language and its Compilation to Multiprocessor Machines or VLSI", M.C. Chen, 13th POPL, ACM 1986 pp.131-139.

CS-4 - "CS-4 Language Reference Manual and Operating System Interface", Ben

M. Brosgol et al, Report IR-130-2, Intermetrics, Cambridge MA, Oct 1975.

CS-Prolog - Distributed logic language. "CS-Prolog on Multi-Transputer Systems", I. Futo et al, Microprocessors & Microsystems, March 1989.

C-Scheme - Joe Bartlett at DEC/WRL? Dialect of Scheme implemented in and embeddable in C.

csh - C-Shell. William Joy. Command shell interpreter and script language for Unix.

CSL -

1. Computer Structure Language. A computer hardware description language, written in BCPL. "Computer Structure Language (CSL)", Proc 1975 Symp on Comp Hardware Description Languages and their Appl, ACM (Sep 1975).

2. Control and Simulation Language. Esso and IBM. Language for industrial simulations. "Control and Simulation Language", J.N. Buxton et al, Computer J 5(3):194-199 (Oct 1962). Version: CSL 2 (1966 for IBM 7094).

CSM - "CSM - A Distributed Programming Language", S. Zhongxiu et al, IEEE Trans Soft Eng SE-13(4):497-500 (Apr 1987).

CSMP - Continuous System Modeling Program. Simulation of dynamics of continuous systems. Similar to CSSL. "A Guide to Using CSMP - The Continuous System Modeling Program", Frank H. Speckhart et al, P-H 1976.

CSP - Communicating Sequential Processes. 1978. A notation for concurrency based on synchronous message passing and selective communications. cobegin/coend. "Communicating Sequential Processes", A.R. Hoare, P-H 1985.

CSP/80 - Based on CSP. "CSP/80: A Language for Communicating Processes", M. Jazayeri et al, Proc Fall COMPCON80, IEEE pp.736-740 (Sept 1980).

CS/PCode - Used at Microsoft.

CSP/k - Concurrent SP/k. A PL/I-like concurrent language. "Structured Concurrent Programming with Operating System Applications", R.C. Holt et al, A-W 1978.

CSP-S - "Implementation of CSP-S for Description of Distributed Algorithms", L. Patnaik et al, Comput Lang 9(3):193-202 (1984).

CSPS - "Toward Comprehensive Specification of Distributed Systems", G. Roman et al, Proc 7th Intl Conf on Distrib Comp Sys, IEEE 1987, pp.282-289.

CS/QCode - Used at Microsoft.

CSS/II - Computer System Simulator II. Like GPSS, for IBM 360. "Computer System Simulator II (CSS II) Program Description and Operations Manual", SH20-0875, IBM.

CSSA - Object-oriented language. "Key Concepts in the INCAS Multicomputer Project", J. Nehmer et al, IEEE Trans Soft Eng SE-13(8):913-923 (Aug 1987).

CSSL - Continuous System Simulation Language. Versions include ACSL, HYTRAN, SL-I, S/360 and CSMP. "The SCi Continuous System Simulation Language (CSSL)", Simulation, 9(6) (Dec 1967).

CSTools - Concurrency through message-passing to named message queues.

CTL -

1. Checkout Test language. "Checkout Test Language: An Interpretive Language Designed for Aerospace Checkout Tasks", G.S. Metsker, Proc FJCC 33(2) (1968).

2. Compiler Target Language. Intermediate language used by the ALICE parallel machine. "The Design and Implementation of ALICE: A Parallel Graph Reduction Machine", M.D. Cripps et al, Proc Workshop on Graph Reduction, Springer 1987.

Cube - Three-dimensional visual language for higher-order logic. "The Cube Language", M. Najork et al, 1991 IEEE Workshop on Visual Langs, Oct 1991, pp.218-224.

CUCH - CUrry-CHurch. Lambda-calculus. "A Type-Theoretical Alternative to CUCH, ISWIM, OWHY", Dana Scott, Oxford U 1969. "Introduction to the CUCH", C. Bohm et al, in Automata Theory, E.R. Caianiello ed, A-P 1966, pp.35-65.

Culler-Fried System - System for interactive mathematics. Sammet 1969, p.253-255.

CUPID - Graphic query language. "CUPID: A Graphic Oriented Facility for Support of Nonprogrammer Interactions with a Database", N. McDonald, PhD Thesis, CS Dept, UC Berkeley 1975.

CuPit - Parallel language for neural networks. "CuPit - A Parallel Language for Neural Algorithms: Language Reference and Tutorial", Lutz Prechelt, TR, U Karlsruhe, 1993.  
<ftp://ftp.ira.uka.de/pub/uni-karlsruhe/papers/cupit.ps.gz>

CUPL - Cornell University Programming Language. Simple math problems, based on CORC, with PL/I-like syntax. "An Instruction Language for CUPL", R.J. Walker, Cornell U, Jul 1967,  
<ftp://locke.ccil.org/pub/retro>

CWIC - Compiler for Writing and Implementing Compilers. Val Schorre. One of the early metacompilers. (cf. Meta-II).

CYBIL - Control Data's system programming language in the 80's. Major parts of CDC systems written in this.

CYCL - Frame language. "Building Large Knowledge-Based Systems", D.B. Lenat et al, A-W 1990.

CypherText - Interactive language for text formatting and typesetting. "CypherText: An Extensible Composing and Typesetting Language", C.G. Moore et al, Proc FJCC 37, AFIPS (Fall 1970).

D -

1. "The Data Language." MS-DOS 4GL.
2. A Haskell-like language, with type classes.  
info: polar@cs.syr.edu
3. Scripting language in the Teleuse Motif GUI builder.

DACAPO - Broad-range hardware specification language. "Mixed Level Modelling and Simulation of VLSI Systems", F.J. Rammig in Logic Design and Simulation, E. Horbst ed, N-H 1986.

DACTL - Declarative Alvey Compiler Target Language. U East Anglia. An intermediate language. "DACTL - A Computational Model and Compiler Target Language Based on Graph Reduction", J.R.W. Glauert et al, ICL Tech J 5(3) (1987). Version: Dactl0. (See Lean, Parallel SML).

DAD - Declarative Ada Dialect. Dialect of Ada intended to aid rapid prototyping of Ada programs. Adds many features including manipulation of first-order functions, lazy evaluation, and streams. Implemented as a pre-processor to Ada. "DAD Defined", P.A. Bailes et al, TR-229, CS Dept, U Queensland.  
info: Dan Johnston <dan@cs.uq.oz.au>

Daisy - Functional. "Daisy Programming Manual", S.D. Johnson, CS Dept TR, Indiana U, 1988.

DAISY 201 - Early system on G-15. Listed in CACM 2(5):16 (May 1959).

DAP-16 - assembly language for the Honeywell 2600 test station.

DAP Fortran - "Efficient High Speed Computing with the Distributed Array Processor", P.M. Flanders et al, pp.113-127 (1977). [same as Fortran-Plus?]

DAPLEX - "The Functional Data Model and the Data Language DAPLEX", D.W. Shipman, ACM Trans Database Sys, 6(1):140-173 (Mar 1981).

DARE - Differential Analyzer REplacement. A family of simulation languages for continuous systems. "Digital Continuous System Simulation", G.A. Korn et al, P-H 1978.

Darms - Music language. "The Darms Project: A Status Report", R.F. Erickson, Computers and the Humanities 9(6):291-298 (June 1975).

Dartmouth BASIC - Term for the original BASIC by Kemeny & Kurtz.

DAS - Digital Analog Simulator. Represents analog computer design.

DASL - Datapoint's Advanced System Language. Gene Hughes. A cross between C and Pascal with custom features for Datapoint hardware (no stack), used internally by Datapoint.

Data/BASIC - Also known as Pick BASIC. A BASIC-like language with database capabilities, the main programming language on the Pick OS. "The Data/BASIC Language - A Data Processing Language for Non-Professional Programmers", P.C. Dressen, Proc SJCC 36, AFIPS (Spring 1970).

DATABUS - DATApoint BUSiness Language. Like an interpreted assembly language, used for custom applications on Datapoint machines.

DATA CODE I - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

Dataparallel-C - Hatcher & Quinn, U New Hampshire. C with parallel extensions, based on an early version of C\*. For Intel iPSC-2 and nCube.

Data Parallel Haskell - Adds PODs and POD comprehensions to Haskell. (POD=Parallel Object with arbitrary Dimension) "Data Parallel Haskell: Mixing Old and New Glue", J. Hill.  
ftp://redstar.dcs.qmw.ac.uk/cpc/jon\_hill/dpGlue.ps.Z

Data Structures Language - MAD dialect with extensions for lists and graphics, on Philco 212. "A Compiler Language for Data Structures", N. Laurance, Proc ACM 23rd Natl Conf 36 (1968).

DATA-TEXT - Harvard. Numerical computations in the Social Sciences. "DATA-TEXT Primer", D.J. Armor, Free Press 1972.

DataVis - Dataflow language for scientific visualization. "Data Flow Visual Programming Languages", D. Hils, J Vis Langs and Comput, Dec 1991.

dBASE - Language used by the dBASE system. First release was dBASE II, ca 1980. (There never was a "dBASE I") Later versions: dBASE III, dBASE III+, and dBASE IV.

DBC - Data-parallel Bit-serial C. SRC, Bowie MD. Based on MPL.  
info: maya@super.org

dBFAST - dBASE dialect for MS-DOS, MS-Windows.

DBPL - Procedural language with relational database constructs. A successor to Pascal/R and Modula/R. "DBPL Report", J.W. Schmidt et al, DBPL-Memo 111-88, Fachbereich Informatik, Johann Wolfgang Goethe-Universitaet, Frankfurt, Germany, 1988.

dBXL - dBASE-like interpreter/language for MS-DOS from WordTech, Orinda, CA.

dc - Desk Calculator. A stack-based mini-language and its interpreter, shipped with every Unix since V7.

DCALGOL - Data Communications ALGOL. A superset of Burroughs Extended ALGOL used for writing Message Control Systems.

DCDL - Digital Control Design Language. A language for simulating computer systems. "DCDS Digital Simulating System", H. Potash et al, Proc FJCC 35, AFIPS (Fall 1969).

DCG - A variant of BNF.

DCL -

1. DIGITAL Command Language. The interactive command and scripting

language for VAX/VMS.

2. Delphi Common LISP. An implementation of Common LISP that has been used as a basis for CLOS.

DDL -

1. "A Digital System Design Language" (DDL)", J.R. Duley, IEEE Trans on Computers c-17(9):850-861 (Sep 1968).

2. M. Urban, C. Kostanick et al, UCLA Computer Club. An adventure language, the forerunner of ADL.

3. Data Definition Language. Specification language for a database based on the entity-relationship model. Used in the Eli compiler-compiler to manage type definitions. "DDL Reference Manual", ECE Dept U Colorado, 1991.

DDM - Dataflow language. "The Architecture and System Method of DDM-1: A Recursively Structured Data Driven Machine", A. Davis, Proc 5th Ann Symp Comp Arch, IEEE 1978.

DEACON - Direct English Access and CONtrol. English-like query system. Sammet 1969, p.668.

Delirium - An embedding coordinate language for parallel programming, implemented on Sequent Symmetry, Cray, BBN Butterfly. "Parallel Programming with Coordination Structures", S. Lucco et al, 18th POPL, pp.197-208 (1991).

Delta -

1. J.C. Cleaveland, 1978. Expression-based. [?]

2. Tandem. A string-processing language with single-character commands.

3. Language for system specification of simulation execution. "System Description and the DELTA Language", E. Holback-Hansen et al, DELTA Proj Rep 4, Norweg Comput Ctr, Feb 1977.

Delta-Prolog - Prolog extension with AND-parallelism, don't-know nondeterminism and interprocess communication using synchronous event goals. Distributed backtracking. "Delta-Prolog: A Distributed Logic Programming Language", L.M. Pereira et al, Intl Conf 5th Gen Comp Sys, Nov 1984.

DEMON - Program generator for differential equation problems. N.W. Bennett, Australian AEC Research Establishment, AAEC/E142, Aug 1965.

Design System language - J. Gaffney, Evans & Sutherland 1976. Interpretive FORTH-like language for 3-D graphics databases. Earliest forerunner of both Interpress and PostScript. Mentioned in PostScript Language Reference Manual, Adobe Systems, A-W 1985.

DETAB - DEcision TABLE. A. Chapman, 1964. Decision table COBOL preprocessor. Sammet 1969, p.315. Versions: DETAB 65, DETAB X.

DETOL - Directly Executable Test Oriented Language. Simple language to control a specific type of test equipment. "Improved DETOL Programming Manual for the Series 5500 Automatic Test System, Pub. 5500-31-0-1, AAI Corp. (Sep 1973).

Deva - Functional. "The Generic Development Language Deva: Presentation and Case Studies", M. Weber et al, LNCS 738, Springer 1993.

DEX - W. van Oortmerssen. A cross between Modula-2 and C.  
ftp://ftp.cso.uiuc.edu/pub/amiga/fish/f7/ff743/TurboDEX.lzh Version 1.2 for Amiga

DFC - Dataflow language. "Data Flow Language DFC: Design and Implementation", S. Toshio et al, Systems and Computers in Japan, 20(6):1-10 (Jun 1989).

DG/L -

1. Descriptive Geometry Language. Early CAD/CAE language, used light pen. "Interactive Graphic Consoles - Environment and Software", R.L. Beckermeyer, Proc FJCC 37 (1970).

2. Data General ca 1973-1974. Derivative of ALGOL 60, developed from DG's Algol-5, used as the systems language under AOS and RDOS for the DG Eclipse family of computers. Replaced by PL/I in the early 80's. Data General manual 093-000229-01.

DIALECT - High-level language for LALR grammars. Part of Software Refinery from Reasoning Systems.  
info: help@reasoning.com

DIALOG - Illinois Inst Tech, 1966. Interactive math using graphics tablet. "DIALOG: A Conversational Programming System with a Graphical Orientation", S.H. Cameron et al, CACM 10:349-357 (1967). Sammet 1969, p.255-258.

DIAMAG - An interactive extension of ALGOL. Sammet 1969, p.195.

Diamond - One of five pedagogical languages based on Markov algorithms, used in "Nonpareil, a Machine Level Machine Independent Language for the Study of Semantics", B. Higman, ULICS Intl Report No ICSI 170, U London (1968). (cf. Brilliant, Nonpareil, Pearl[3], Ruby[2]).

DIANA - Descriptive Intermediate Attributed Notation for Ada. Goos & Wulf, CMU, Jan 1981. A de facto standard intermediate language for Ada programs. An attributed tree representation, with an abstract interface defined in IDL. Resulted from a merger of AIDA and TCOL.Ada. "DIANA - An Intermediate Language for Ada", G.T. Goos et al, LNCS 161, Springer 1983.

DIBOL - Digital Interactive Business Oriented Language. DEC, 1970. FORTRAN syntax with BCD arithmetic. Versions for PDP-8 and RT-11. ANSI X3.165-1988.

Dictionary APL - nickname for Sharp APL.

Dijkstra's guarded command language - Edsger Dijkstra, ca. 1974. Introduced the concept of guards and committed choice [=don't care] nondeterminism. Described and used (but never named) in A Discipline of

Programming, E. Dijkstra, P-H 1976. Dijkstra later (1972) referred to this language as DOVPA (Dijkstra's Own Version of Pidgin Algol).

DIMATE - Depot Installed Maintenance Automatic Test Equipment. For automatic equipment tests, on RCA 301. "A Simple User-Oriented Source Language for Programming Automatic Test Equipment", B.H. Scheff, CACM 9(4) (Apr 1966). Sammet 1969, p.647.

DinnerBell - Object-oriented dataflow language with single assignment. "Object-Oriented Load Distribution in DinnerBell", S. Kono <kono@csl.sony.co.jp> et al, in TOOLS Pacific 90.

DINO - Data parallel language, a superset of C. "The DINO Parallel Programming Language", M. Rosing et al, J Parallel Dist Comp 13(9):30-42 (Sep 1991). Implemented using ALADIN. "DINO Parallel Programming Language", M. Rosing et al, CU-CS-457-90, U Colorado, April 1990. ftp://ftp.cs.colorado.edu/pub/cs/distrib/dino/\*

Disiple - DSP language. "A Compiler that Easily Retargets High Level Language Programs for Different Signal Processing Architectures", J.E. Peters & S.M. Dunn, Proc ICASSP 89, pp.1103-1106, (May 1989).

Dislang - "Dislang: A Distributed Programming Language/System", C. Li et al, Proc 2nd Intl Conf Distrib Comp Sys, IEEE 1981, pp.162-172.

Distributed Eiffel - "Distributed Eiffel: A Language for Programming Multi-Granular Distributed Objects on the Clouds Operating System", L. Gunaseelan et al, IEEE Conf Comp Langs, 1992.

Distributed Processes - (Also "DP"). First concurrent language based on remote procedure calls. "Distributed Processes: A Concurrent Programming Concept", P. Brinch Hansen CACM 21(11):934-940 (Nov 1978).

Distributed Smalltalk - "The Design and Implementation of DIstributed Smalltalk", J. Bennett, SIGPLAN Notices 22(12):318-330 (Dec 1980).

DL/1 - Query language, linear keyword.

DLG - DFA-based Lexical analyzer Generator. Part of PCCTS (Purdue Compiler-Construction Tool Set). ftp://ecn.purdue.edu

DLP - Logic programming similar to Prolog, combined with parallel object orientation similar to POOL. Supports distributed backtracking over the results of a rendezvous between objects. Multi-threaded objects have autonomous activity and may simultaneously evaluate method calls. "DLP: A Language for Distributed Logic Programming", A. Eliens, Wiley 1992.

DLX - Assembly language. "Computer Architecture: A Quantitative Approach", J.L. Hennessy and Patterson.

DMAD - Diagnostic Machine Aid-Digital. Functional testing of digital devices. "DMAD M/MM Manual", BR-8392, Raytheon Co. (Oct 1973).

DMALGOL - ALGOL with extensions to interface to DMS II, the Burroughs database.

DML -

1. Data Management Language. Early ALGOL-like language with lists, graphics, on Honeywell 635. "DML: A Data Management Language", D.W. Bray et al, GE, Syracuse NY.

2. "DML: A Meta-language and System for the Generation of Practical and Efficient Compilers from Denotational Specifications", M. Pettersson et al, IEEE Conf Comp Langs, 1992.

Doc - Directed Oc. "Programming Language Doc and Its Self-Description, or 'X=X Is Considered Harmful'", M. Hirata, Proc 3rd Conf Japan Soc Soft Sci Tech, pp.69-72 (1986).

DOCUS - Display Oriented Computer Usage System. Interactive system using push buttons. Sammet 1969, p.678

DoD-1 - Unofficial name of the language that became Ada.

DOUGLAS - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

DOL - Display Oriented Language. Subsystem of DOCUS. Sammet 1969, p.678.

DOVPA - Dijkstra's Own Version of Pidgin Algol. See "Dijkstra's guarded command language".

DOW COMPILER - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

DOWL - Distributed OWL. B. Achauer, U Karlsruhe. An extension of Trellis supporting transparently distributed objects. "The DOWL Distributed Object-Oriented Language", B. Achauer, CACM 36(9) (Sep 1993).

d-Prolog - Prolog extended with defeasible reasoning.  
ftp://aisun1.ai.uga.edu/ai.prolog/dprolog\* for MS-DOS and Unix

DPL - DECmmp Parallel Language. C-like parallel language for the DECmmp machine.

DPL-82 - "DPL-82: A Language for Distributed Processing", L. Ericson, Proc 3rd Intl Conf Distrib Comp Sys, IEEE 1982, pp.526-531.

DPS - Real-time language with direct expression of timing requests. "Language Constructs for Distributed Real-Time Programming", I. Lee et al Proc IEEE Real-Time Sys Symp pp.57-66 (Dec 1985).

dpSather - Data-parallel Sather. Fine-grained deterministic parallelism  
info: hws@csis.dit.csiro.au  
ftp://lynx.csis.dit.csiro.au/pub/ather/dpsather.papers

draco - Chris Gray, 1987. A blend of Pascal, C and ALGOL 68. Implemented for CP/M-80 and Amiga.

DRAGON - Implementation language used by BTI Computer Systems.  
info: Pat Helland <helland@hal.com>

DRAGOON - Colin Atkinson, Imperial College 1989. (current address:

<atkinson@cl.uh.edu>, U Houston-Clear Lake). Ada-based language, bringing object-oriented programming to embeddable systems. Presently implemented as an Ada preprocessor. "Object-Oriented Reuse, Concurrency and Distribution: An Ada-Based Approach", C. Atkinson, A-W 1991, ISBN 0-2015-6-5277.

info: adimaio@vms.eurokom.ie

DROOL - Dave's Recycled Object-Oriented Language. Language for writing adventure games. An updated implementation of AdvSys. Multiple inheritance, garbage collection. "Dave's Recycled OO Language", David Betz, Dr Dobbs J, Oct 1993, pp.74-78.

DRUCO I - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

DSL -

1. Digital Simulation Language. Extensions to FORTRAN to simulate analog computer functions. "DSL/90 - A Digital Simulation Program for Continuous System Modelling", Proc SJCC 28, AFIPS (Spring 1966). Version: DSL/90 for the IBM 7090. Sammet 1969, p.632.

2. Denotational Semantics Language. Specification language used by the SIS compiler generator. "SIS - Semantics Implementation System", P.D. Mosses, TR DAIMI MD-30, Aarhus U, Denmark.

DSM -

1. Data Structure Manager. J.E. Rumbaugh and M.E. Loomis, GE. Object-oriented language similar to C++, used in implementation of CAD/CAE software. DSM is written in itself and C, and produces C as output. "DSM: An Object-Relationship Modeling Language", A. Shah et al, SIGPLAN Notices 24(10):191-202 (OOPSLA '89) (Oct 1989).

2. DIGITAL Standard MUMPS. DEC. (See MUMPS).

DSP/C - Numerical extension to C, for DSP applications. "DSP/C: A Standard High Level Language for DSP and Numeric Processing", K. Leary & W. Waddington, Proc ICASSP 90, Apr 1990, pp.1065-1068.

DSP32 Assembly Language - A high-level assembly language for the DSP32 Programmable DSP Chip.

DSPL: Digital Signal Processing Language. A C-derived DSP language. "The Programming Language DSPL", A. Schwarte & H. Hanselmann, Proc PCIM 90, 1990.

DTALGOL - Decision Table Algol. Victoria U, Wellington. An ALGOL superset that added decision tables. On Burroughs Large System.

DUAL-607 - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

Dual FCP - [?]

DuoTalk - Smalltalk-like language with separate inheritance hierarchies for types and classes. "Separation of Hierarchies in DuoTalk", C. Lunau, J Object-Oriented Programming 2(2):20-26, 1989.

Durra - Description language for coarse-grained concurrency on heterogeneous processors. "Durra: A Task-level Description Language", M.R. Barbacci et al, CMU/SEI-86-TR-3, CMU 1986.

DYANA - DYnamics ANALyzer. Early specialized language for vibrational and other dynamic systems. Sammet 1969, p.628.

Dylan - DYnamic LANguage. Advanced Technology Group East, Apple Computer. A simple object-oriented LISP dialect, most closely resembles CLOS and Scheme. "Dylan, an Object-Oriented Dynamic Language", Apple 1992.  
ftp://crl.dec.com/pub/DEC/Thomas - compiler implemented in Scheme  
info: dylan-manual-request@cambridge.apple.com  
list:info-thomas@crl.dec.com

Dynace - DYnamic C language Extension. Blake McBride, 1993. Extension of C, adds classes, multiple inheritance, dynamic binding, garbage collection and threads. Implemented as a C preprocessor and library. Available from Algorithms Corp, (615)791-7736.  
info: Blake McBride <blake@edge.ercnet.com>

DYNAMO - DYnamic MODels. Phyllis Fox & A.L. Pugh, 1959. Continuous simulation including economic, industrial and social systems. Versions: DYNAMO II, DYNAMO II/370, DYNAMO II/F, DYNAMO III and Gaming DYNAMO. "DYNAMO User's Manual", A.L. Pugh, MIT Press 1976.

DYSAC - Digital Simulated Analog Computer. Sammet 1969, p.629.

DYSTAT - DYnamic STorage ALlocation. Adds lists, strings, sorting, statistics and matrix operations to FORTRAN. Sammet 1969, p.388. "DYSTAT: Dynamic Storage Allocation Language in FORTRAN", J.M. Sakoda, in Symbol Manipulation Languages and Techniques, D.G. Bobrow ed, N-H 1971, pp.302-311.

E -

1. An extension of C++ with database types and persistent objects. Used in the Exodus database system. "Persistence in the E Language: Issues and Implementation", J.E. Richardson et al, Soft Prac & Exp 19(12):1115-1150 (Dec 1989).  
ftp://ftp.cs.wisc.edu/exodus/E/gnu\_E\*

2. Wouter van Oortmerssen <wouter@mars.let.uva.nl>. A procedural language with semantics similar to C. Lists, low-level polymorphism, exception handling, quoted expressions, pattern matching, object inheritance. Version for the Amiga.  
ftp://wuarchive.wustl.edu/pub/aminet/dev/e/AmigaE21b.lha  
list: amigae@bookhouse.cts.com

Eagle - dBASE-like dialect bundled with Emerald Bay, sold by Migent from 1986-1988, later renamed Vulcan when Wayne Ratliff reacquired the product.

Ease - General purpose parallel programming language, combining the process constructs of CSP and the distributed data structures of Linda. "Programming with Ease: Semiotic Definition of the Language", S.E. Zenith, <zenith-stein@yale.edu> Yale U TR-809, Jul 1990.

EASE II - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

EASIAC - Early system on Midac computer. Listed in CACM 2(5):16 (May 1959).

EASY FOX - Early system on JOHNNIAC computer. Listed in CACM 2(5):16 (May 1959).

EBASIC - Gordon Eubanks, now at Symantec. Form of BASIC that led to CBASIC.

EBNF - Extended BNF. Backus-Naur Form with one or more added constructs, usually regular expressions. CACM, 1977?

ECAP II - Electronic Circuit Analysis Program. Simple language for analyzing electrical networks. "Introduction to Computer Analysis: ECAP for Electronics Technicians and Engineers", H. Levin, P-H 1970.

Echidna - Constraint logic programming embedded in an object-oriented language. The syntax is an extension of Edinburgh Prolog. "Hierarchical Arc Consistency Applied to Numeric Processing in Constraint Logic Programming", G. Sidebottom et al, TR-91-06, CSS-IS, Simon Fraser U, and Comp Intell 8(4) (1992).

ftp://cs.sfu.edu/pub/ecl/papers  
info: expert@cs.sfu.edu

ECL - Extensible CL. Wegbreit, ca 1970. "The ECL Programming System", B. Wegbreit, Proc FJCC 39:253-261, AFIPS (Fall 1971). "ECL Programmer's Manual", B. Wegbreit, TR 23-74, Harvard U (Dec 1974).

ECMA - Subset of ALGOL. Sammet 1969, p.180.

ECP - Extended Concurrent Prolog. Concurrent Prolog with OR parallelism, set abstraction and meta-inference features. "AND-OR Queuing in Extended Concurrent Prolog", J. Tanaka et al, Proc Logic Prog Conf '85, LNCS 193, Springer 1985. [???

ECRC-Prolog - Evidently Prolog with coroutine extensions. "ECRC-Prolog User's Manual Version 1.0", K. Estenfeld, TR-LP-08 ECRC, Feb 1986. (See SEPIA).

ECSL - Extended CSL. Discrete simulation language, successor to CSL. "Extended Control and Simulation Language", A.T. Clementson, Comp J 9(3):215-220 (1966).

ECSP - An extension to CSP, supporting dynamic communication channels and nested processes. "Static Type Checking of Interprocess Communication in ECSP", F. Baiardi et al, SIGPLAN Notices 19(6):290-299 (June 1984).

ECSS II - Extendable Computer System Simulator. An extension of SIMSCRIPT II. "The ECSS II Language for Simulating Computer Systems", D.W. Kosy, R-1895-GSA, Rand Corp.

ECSSL - Formerly APSE. Equation-oriented specifications for continuous simulations. The compiler outputs HYTRAN, which must be run on an analog processor.

Eden - Concurrent object-oriented language with both synchronous and

asynchronous message passing. "The Eden System: A Technical Review", G. Almes et al, IEEE Trans Soft Eng SE-11(1):43-59 (Jan 1985).

EDIF - Electronic Design Interchange Format. Not a programming language, but a format to simplify data transfer between CAD/CAE systems. LISP-like syntax. "Designer's Guide to EDIF", E. Marx et al, EDN 1987. "EDIF Electronic Design Interchange Format Version 200", ANSI/EIA Standard 548. info: edif-support@cs.man.ac.uk ftp://edif.cs.man.ac.uk/pub/edif

Edinburgh Prolog - Prolog dialect which eventually developed into the standard, as opposed to Marseille Prolog. (The difference is largely syntax.) Clocksin & Mellish describe Edinburgh Prolog. Version: C-Prolog.

Edison -

1. (named for the American inventor Thomas Edison (1847-1931)) Brinch Hansen. A simplified Pascal, with modules and concurrency (cobegin/coend). "Edison - A Multiprocessor Language", P. Brinch Hansen, CS Dept, USC, Sep 1980. "Programming a Personal Computer", Brinch Hansen, P-H 1977.

2. Adds an OPS5-like system to C. Implemented as a C preprocessor. "Edison, A Unix and C Friendly Rete Based Production System", B. Thirion, SIGPLAN Notices 27(1):75-84 (Jan 1992).

EDL -

1. Experiment Description Language. J.S. Jenkins. "A Programmable System for Acquisition and Reduction of Respiratory Physiological Data", J.S. Jenkins et al, Ann Biomed Eng, 17:93-108 (1989).

2. Event Description Language. "EDL: A Basis for Distributed System Debugging Tools", P.C. Bates et al, in Proc Hawaii Intl Conf on Sys Sci, Jan 1982, pp.86-93.

3. Event Driven Language. Language for input to the EDX (Event Driven Executive) for the IBM Series/1. The output was machine code for IBM Series/3 or Series/7, and interpreted on Series/1 by an emulator.

EFL - Extended FORTRAN Language. FORTRAN preprocessor to provide structured programming much like C. A descendant of RATFOR, written in C. "An Informal Description of EFL", S.I. Feldman.

Eh - "A". Software Portability Group, U Waterloo. A typeless language derived from (and similar to) B. Provides guaranteed order of evaluation for side effects in expressions. Also character indexing operators. "Eh Reference Manual", R.S.C. Braga, RR CS-76-45, U Waterloo, Nov 1976. (See Zed.)

Eiffel - Bertrand Meyer <bertrand@eiffel.com> ca. 1986. An object-oriented language. Classes with multiple and repeated inheritance, deferred classes (like Smalltalk's abstract class), and clusters of classes. Objects can have both static and dynamic types. The dynamic type must be a descendant of the static (declared) type. Dynamic binding resolves clashes from the multiple inheritance. Flattened forms of classes, in which all of the inherited features are added at the same level. Generic classes parametrized by type. Persistent objects, garbage collection, exception

handling, interface to routines written in other languages. Implemented as a C preprocessor. "Eiffel: The Language", Bertrand Meyer, P-H 1992. Interactive Software Eng, Goleta CA, (805) 685-1006. Version 2.3.4. (See Sather).  
info: queries@eiffel.com

Eiffel 3 - Latest version of the Eiffel language. Available as Eiffel/S from SiG Computer GmbH, Germany.

EL1 - Extensible Language One. B. Wegbreit, Harvard ca 1974. An extensible language, internally somewhat LISP-like, but fully typed with records and pointers. The external syntax is Algol-like and extensible, supporting user-defined data structures, control structures and operations. The parser is table-driven, with a modifiable set of productions. Used as the basis for the ECL operating system. "Studies in Extensible Programming Languages", B. Wegbreit, Garland Pub 1980.

el(alpha) - Aims to be a high-level language that knows about real hardware, for systems programming. "Essential Language el(alpha) - A Reduced Expression Set Language for Systems Programming", T. Watanabe et al, SIGPLAN Notices 26(1):85-98.

Elan - "Top-down Programming with Elan", C.H.A. Koster, Ellis Horwood 1987.

ELF - Binary format used by System V Release 4 Unix.

ELI -

1. Early system on IBM 705, IBM 650. Listed in CACM 2(5):16 (May 1959).
2. Embedded Lisp Interpreter. Bob Glickstein <bobg@andrew.cmu.edu> Small Common Lisp-like interpreter embedded in the Andrew mail system.

ELISP - Chuck Hedrick, Rutgers. Implemented originally for DEC-20's, later used as the LISP in EMACS.

Elk - Extension Language Kit. Oliver Laumann <net@tub.cs.tu-berlin.de>, Tech U Berlin. A Scheme interpreter, easily extendable with new types and primitive procedures. First-class environments, dynamic-wind, fluid-let, macros, dynamic loading of object files, autoloading, and a dump.  
ftp://tub.cs.tu-berlin.de/pub/elk-2.1.tar.Z

ELLA - Defence Research Agency, Malvern UK, 1979. First prototype 1982. Hardware design language.

ELLA 2000 - Version of ELLA with more powerful generics and user-defined attributes. "ELLA 2000: A Language for Electronic System Design", J.D. Morison and A.S. Clarke, McGraw-Hill 1993, ISBN 0-07-707821-7. Implemented in Algol68-RS.  
info: ella@dra.hmg.gb  
ftp://src.doc.ic.ac.uk/packages/ELLA for SPARC

Ellie - Object-oriented language with fine-grained parallelism for distributed computing. Based on BETA, Smalltalk, and others. Parallelism by unbounded RPC and 'future' objects. Synchronization by 'dynamic interfaces'. Classes, methods, blocks and objects all modeled by first-class 'Ellie objects'. Genericity, polymorphism and

delegation/inheritance. "Ellie Language Definition Report", Birger Andersen <birger@diku.dk>, SIGPLAN Notices 25(11):45-65 (Nov 1990).  
doc: diku.dk:ellie/papers

ELLIS - EuLisp LInda System. An object-oriented Linda system written for EuLisp. "Using Object-Oriented Mechanisms to Describe Linda", P. Broadbery <pab@maths.bath.ac.uk> et al, in Linda-Like Systems and Their Implementation, G. Wilson ed, U Edinburgh TR 91-13, 1991.

ELMAGUIDE - Tallinn Poly Inst, 1978. Metalanguage used for interpretation of user actions in the ELMA compiler writer.

ELMAMETA - Tallinn Poly Inst, 1978. A FORTRAN extension used for lexical, syntactic and semantic sepecification in the ELMA compiler writer. This system was widely used in the Soviet Union, and produced an Ada to Diana compiler.

ELP -

1. English Language Programs. Language for testing avionics equipment, on Varian 620/i. "Multiband Automatic test Equipment - A Computer Controlled Checkout System", T. Kuroda et al, Proc SJCC, 38 (1971).

2. Equational Logic Programming. O'Donnell <odonnell@cs.uchicago.edu>. Semantically pure, fully lazy. "Equational Logic as a Programming Language", M.J. O'Donnell, MIT Press 1985. Current version: 4.2.  
ftp://gargoyle.uchicago.edu/pub/equations/eq4.2.tar.Z for Sun and DEC.

ELSIE - A distributed version of ELLIS. "Using Object-Oriented Mechanisms to Describe Linda", P. Broadbery <pab@maths.bath.ac.uk> et al, in Linda-Like Systems and Their Implementation, G. Wilson ed, U Edinburgh TR 91-13, 1991.

EM-1 - Experimental Machine. An intermediate language, the assembly language for a stack-based machine, used by the Amsterdam Compiler Kit. "Using Peephole Optimization on Intermediate Code", A.S. Tanenbaum et al, ACM TOPLAS 4(1):21-36 (1982).  
info: Andrew Tanenbaum <ast@cs.vu.nl>, Vrije U, Amsterdam.

EMA - Extended Mercury Autocode. (See Autocode).

EMACS LISP - Richard Stallman. Variant of LISP used by the EMACS editor. (This is the "offical" name, based on the EMACS FAQ file. See ELISP.)

Emerald - U Washington, early 80's. The successor of EPL[3]. A polymorphic object-oriented distributed programming language/environment. Strongly typed, uses signatures and prototypes rather than classes and inheritance. "Distribution and Abstract Types in Emerald", A. Black et al, IEEE Trans Soft Eng SE-13(1):65-76 (Jan 1987). "Emerald: An Object-Based Language for Distributed Programming", Norman C. Hutchinson, PhD Thesis, U Wash, Jan 1987, TR 87-01-01.

EML - Extended ML. A language for formally specifying SML programs. "Formal Program Development in Extended ML for the Working Programmer", D. Sannella, Proc 3rd BCS/FACS Workshop on Refinement", Springer 1990.

EMPL - Extensible Microprogramming Language. An early object-oriented

microprogramming language with PL/I-like syntax. Extensions include datatype definitions and operators. Never fully implemented. "A Machine Independent Approach to the Production of Horizontal Microcode, D.J. DeWitt, PhD Thesis, U Mich 1976.

English - Database language used in the Pick OS. "Exploring the Pick Operating System", J.E. Sisk et al, Hayden 1986.

EOL - Expression Oriented Language. A low-level language for strings. "EOL - A Symbol Manipulation Language", L. Lukaszewicz, Computer J 10(1):53 (May 1967). Versions: EOL-1, EOL-2, EOL-3.

EPILOG -

1. Extended Programming In LOGic. PROLOG with several AND's having different time constraints. "Epilog: A Language for Extended Programming in Logic", A. Porto in Implementations of Prolog, J.A. Campbell ed, Ellis Horwood 1984.

2. A data-driven PROLOG, with both AND and OR parallelism. "EPILOG = PROLOG + Data Flow", M.J. Wise, SIGPLAN Notices 17:80-86 (1982).

EPL -

1. Early PL/I. McIlroy, Morris et al. PL/I subset dialect, the first running PL/I compiler. Used by Bell Labs and MIT to write Multics, EPL had extensions to handle the segment/offset nature of Multics pointers. "EPL Reference Manual", Project MAC, April 1966. Sammet 1969, p.542. (See REPL, TMG)

2. Experimental Programming Language. David May. Influenced occam. "EPL: An Experimental Language for Distributed Computing", D.C. May, in Trends and Applications 1978: Distributed Processing, NBS, pp.69-71.

3. Eden Programming Language. U Washington. Based on Concurrent Euclid and used with the Eden distributed OS. Influenced Emerald and Distributed Smalltalk. "EPL Programmer's Guide", A. Black et al, U Washington June 1984.

4. Equational Programming Language. Szymanski, RPI. Equational language for parallel scientific applications. "EPL - Parallel Programming with Recurrent Equations", B. Szymanski in Parallel Functional Languages and Compilers, B. Szymanski et al, A-W 1991.

EPROS - A specification/prototyping language. Implemented in Franz Lisp. "Software Prototyping, Formal Methods and VDM", Sharam Hekmatpour et al, A-W 1988.

<ftp://utsun.s.u-tokyo.jp/lang/epros>

EPSILON - P.A. Ershov, Novosibirsk, 1967. Macro language with high level features: strings, lists, etc. Used to implement ALGOL 68 on the M-220. "Application of the Machine-Oriented Language Epsilon to Software Development", I.V. Pottosin et al, in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, pp.417-434.

EPSIMONE - Concurrent simulation language derived from Simone. "EPSIMONE Manual", J. Beziin et al, Pub Int No 90, IRISA, Sept 1978.

EqL - An equational language. Bharat Jayaraman <bharat@cs.buffalo.edu>. "EqL: The Language and its Implementation", B. Jayaraman et al, IEEE Trans Soft Eng SE-15(6):771-780 (June 1989).

EQLog - OBJ2 plus logic programming based on Horn logic with equality. "EQLog: Equality, Types and Generic Modules for Logic Programming", J. Goguen et al in Functional and Logic Programming, D. DeGroot et al eds, pp.295-363, P-H 1986.

Eqn - Language for typesetting mathematics. "A System for Typesetting Mathematics", B.W. Kernighan and L.L. Cherry, CACM 18(3):151-157 (Mar 1975).

Equel - Embedded Quel. INGRES, Inc. Combines QUEL theories with C code.

Erlang - Armstrong, Williams & Virding, Ellemtel, Sweden. Concurrent functional programming language for large real-time systems. Untyped. Pattern matching syntax, recursion equations. Explicit concurrency, asynchronous message passing. Transparent cross-platform distribution. Primitives for detecting runtime errors. Real-time garbage collection. Modules, dynamic code replacement. Foreign language interface. "Concurrent Programming in Erlang", J. Armstrong et al, P-H 1993. Interpreter in SICStus Prolog, compilers in C and in Erlang, for several Unix platforms. Free version with no support, commercial version from Erlang Systems AB.  
info: erlang@erix.ericsson.se  
<http://www-cslab.ericsson.se:5000>

ERFPI - Early system on LGP-30 computer. Listed in CACM 2(5):16 (May 1959).

es -

1. Extensible Shell. Unix shell derived from rc, includes real functions, closures, exceptions, and the ability to redefine most internal shell operations. "Es - A Shell with Higher Order Functions", P. Haahr et al, Proc Winter 1993 Usenix Technical Conference.  
<ftp://ftp.sys.utoronto.ca/pub/es/es-0.84.tar.Z>

2. Expert System. Forward and backward chaining, and fuzzy set relations, for IBM PC. BYTE Oct 1990.  
<ftp://ftp.uu.net/pub/ai/expert-sys/summers.tar.Z>

ES-1 - Early text editing interpreter. Sammet 1969, p.684.

ESCAPE - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

ESI - Dialect of JOSS. Sammet 1969, p.217.

esim - A simulation language for VLSI, at the switch level. Primitives are nodes and transistors. C.M. Baker et al, "Tools for Verifying Integrated CIrcuit Design", Lambda 1(3):22-30 (1980).

ESP -

1. Extra Simple Pascal. Subset of Pascal.

2. Econometric Software Package. Statistical analysis of time series. "Econometric Software Package, User's Manual", J.P. Cooper, Graduate School of Business, U Chicago. Sammet 1978.

3. Extended Self-containing Prolog. Chikayama. An object-oriented extension of KL0. Backtracking-based control, unification-based parameter passing, object-oriented calling. An object in ESP is an axiom set. A class definition consists of nature definitions (inheritance), slot definitions (class variables) and clause definitions. Multiple inheritance similar to Flavors. Implemented for ICOT's PSI Sequential Inference machine. "Unique Features of ESP", T. Chikayama, Proc Intl Conf 5th Gen Comp Sys, ICOT 1984. (See CESP)  
info: k-hata@air.co.jp

4. Early symbolic math system. A. Rom, Celest Mech 3:331-345 (1971).

ESPOL - An ALGOL superset used to write the MCP (Master Control Program) OS on Burroughs 6700. Not available to users, it was Extended Algol with the added ability to access any CPU instruction as a procedure call. Superseded by NEWP. "The B6700 ESPOL Reference Manual", Burroughs, 1970.

Estelle - A Pascal extension for specification of computer network protocols. Protocols are described by modules which are communicating NFA's. Modules are arranged in a dynamic hierarchy and communicate at named interaction points. "The Formal Description Technique Estelle", M. Diaz et al eds, N-H 1989. Adopted by CCITT. ISO 9074 (1989).  
ftp://osi.ncsl.nist.gov/pub/osikit/estpc Compiles Estelle into C  
petdingo Translates Estelle into C++

Esterel - Distributed language for synchronous interaction of real-time systems with their environment. Uses explicit timing requests. Esterel programs are compiled into finite automata. "The ESTEREL Programming Language and its Mathematical Semantics", G. Berry & L. Cosserat, TR 327, INRIA, 1984.

ET - Bernd Gersdorf, U Bremen. An integration of functional and logic programming.

ET++ - Weinand, UBILAB Zurich. A smalltalk-like system for Suns, built on C++.  
ftp://iamsun.unibe.ch/C++/ET++/et2.2.tar.Z

ETC - ExTensible Compiler. FORTRAN-like, macro extendible. "ETC - An Extendible Macro-Based Compiler", B.N. Dickman, Proc SJCC 38 (1971).

ETHER - Concurrent object-oriented language?

Euclid - (named for the Greek geometer, fl ca 300 BC.) A Pascal descendant for development of verifiable system software. No goto, no side effects, no global assignments, no functional arguments, no nested procedures, no floats, no enumeration types. Pointers are treated as indices of special arrays called collections. To prevent aliasing, Euclid forbids any overlap in the list of actual parameters of a procedure. Each procedure gives an imports list, and the compiler determines the identifiers that are implicitly imported. Iterators. "Report on the Programming Language Euclid", B.W. Lampson et al, SIGPLAN Notices 12(2):1-79 (Feb 1977).

EULER -

1. (named for the Swiss mathematician Leonhard Euler (1707-1783))  
Revision of ALGOL. A small predecessor of Pascal. "EULER: A  
Generalization of ALGOL and Its Formal Definition", N. Wirth, CACM 9(1)  
(Jan 1966) and 9(2) (Feb 1966).

2. 1960's. Lisp with Algol syntax, used for graphics on IBM 360/30 at U  
Utah. CACM article. [?]

EuLisp - 1985-present. LISP dialect intended to be a common European  
standard, with influences from Common LISP, Le LISP, Scheme and T. First-  
class functions, classes and continuations, both static and dynamic scope,  
modules, support for parallelism. The class system (TELOS) incorporates  
ideas from CLOS, ObjVLisp and Oaklisp.  
info: eudist@maths.bath.ac.uk  
ftp://ftp.bath.ac.uk/pub/eulisp/feel-0.70.tar.Z (Free and Eventually  
Eulisp)

Euphoria - End User Programming with Hierarchical Objects for Robust  
Interpreted Applications. Interpreted language with dynamic storage and  
dynamic typing. Rapid Deployment Software, for MS-DOS.  
ftp://oak.oakland.edu/SimTel/msdos/misclang/euphor12.zip for 386  
info: robert.craig@canrem.com

Eurisko - Lenat 1978. Language for "opportunistic programming".  
Constructs its own methods and modifies its strategies as it tries to solve  
a problem. Mentioned by Alan Kay, SIGPLAN Notices 28(3) (March 1993), p.88

Eva -

1. A toy ALGOL-like language used in "Formal Specification of Programming  
Languages: A Panoramic Primer", F.G. Pagan, P-H 1981.

2. Explicit Vector Language. [?]

EXAPT - EXTended APT.

EXEC - Early batch language for IBM VM/CMS systems. SC19-6209 Virtual  
Machine/ System Product CMS Command and Macro Reference (Appendix F. CMS  
EXEC Control Statements)

EXEC2 - IBM, late 70's. SC24-5219 Virtual Machine/System Product EXEC 2  
Reference. Superseded by REXX.

expect - A script language for dealing with interactive programs. Written  
in Tcl. "expect: Scripts for Controlling Interactive Tasks", Don Libes,  
Comp Sys 4(2), U Cal Press Journals, Nov 1991.  
ftp://ftp.uu.net/languages/tcl/expect/\*

Express -

1. ParaSoft Corp. Concurrency through message-passing to named message  
queues. Available from ParaSoft Corp, (818)792-9941.  
ftp://ftp.parasoft.com/express/docs  
info: support@parasoft.com

2. Data definition language, meant to become an ISO standard for product data representation and exchange. TC 184/SC4 N83, ISO, 31 May 1991.  
info: smith@cme.nist.gov

Extended ALGOL - An extension of ALGOL 60, used to write the ESPOL compiler on Burroughs B5500, B6500, B6700. "Burroughs B6700 Extended ALGOL Language Information Manual", No. 5000128 (Jul 1971) Sammet 1969, p.196.

Extended C++ - G. Masotti <masotti@lipari.usc.edu> Extensions to C++ including preconditions, postconditions, class invariants, parametrized classes, exception handling and garbage collection. Implemented as a C++ preprocessor.  
ftp://ftp.uu.net/languages/misc/EC++.tar.Z

Extended ML - Don Sannella, Edinburgh. Algebraic specification meets functional programming. "Program Specification and Development in Standard ML", D. Sannella et al, 12th POPL, ACM 1985.

Extended Pascal - ISO, 1992. A superset of ANSI and ISO Pascal. Many enhancements, including modules, separate compilation, type schemata, variable-length strings, direct-access files, complex numbers, initial values, constant expressions. ANSI/IEEE770X3.160-1989 and ISO 10206.

EXTRA - Object-oriented, Pascal style, handles sets. "A Data Model and Query Language for EXODUS", M.J. Carey et al, SIGMOD 88 Conf Proc, pp.413-423, ACM SIGMOD Record 17:3 (Sept 1988).

EZ - High-level string-processing language derived from SNOBOL4, SL5 and Icon. "The EZ Reference Manual", C.W. Fraser et al, CS TR 84-1, U Arizona, 1984.

FAC - Functional Array Calculator. APL-like but purely functional and lazy, allowing infinite arrays. "FAC: A Functional APL Language", H.-C. Tu and A.J. Perlis, IEEE Trans Soft Eng 3(1):36-45 (Jan 1986).

Facile - SUNY Stony Brook, late 80's. Since 1991 at ECRC, Munich. Extends SML with a model of higher-order concurrent processes based on CCS. Suitable for loosely connected systems with distributed memory. "Facile: A Symmetric Integration of Concurrent and Functional Programming", A. Giacalone <ag@ecc.de> et al, Intl J Parallel Prog 18(2):121-160 (Apr 1989).

FACT - Fully Automated Compiling Technique. ca. 1959. Pre-COBOL English-like business DP language for Honeywell 800. (Aka Honeywell-800 Business Compiler.) Sammet 1969, p.327.

FAD - "FAD, A Simple and Powerful Database Language", F. Bancilon et al, Proc 13th Intl Conf on VLDB, Brighton England, Sept 1987.

FAIR - Early system on IBM 705. Listed in CACM 2(5):16 (May 1959).

FALSE - W. van Oortmerssen. A small compiled extensible language with lambda abstractions. Stack-based, like FORTH, and not very readable -- FALSE code has been said to "resemble TECO".  
ftp://ftp.cso.uiuc.edu/pub/amiga/fish/ff885 for Amiga

FAP - Assembly language for Sperry-Rand 1103 and 1103A. Listed in CACM

2(5):16 (May 1959).

FAS - General purpose language sponsored by the Finnish government in the 70's and 80's.

FASBOL - "FASBOL. A SNOBOL4 Compiler", P.J. Santos, Memo ERL-M134, UC Berkeley 1971. (See SNOBOL, SPITBOL).

FASE - Fundamentally Analyzable Simplified English. L.E. McMahon, Bell Labs. Sammet 1969, p.720.

FAST - FORTRAN Automatic Symbol Translator. Assembly language on IBM 650 by MITRE Corp. Listed in CACM 2(5):16 (May 1959). Sammet 1969, p.526.

FC - Functional language. "FC Manual", L. Augustsson, Memo 13, Programming Methodology Group, Chalmers U, Sweden 1982.

F-code - Code for the FPM abstract machine, an optimized SECD machine. "FP/M Abstract Syntax Description", R. Bailey, Dept Computing, Imperial College, U London, 1985.

FCP - Flat Concurrent Prolog. "Design and Implementation of Flat Concurrent Prolog", C. Mierowsky, TR CS84-21 Weizmann Inst, Dec 1984.

Feel - Free and Eventually EuLisp. An initial implementation of EuLisp. <ftp://ftp.bath.ac.uk/pub/eulisp/feel-0.70.tar.Z>

FEL - Function Equation Language. Programs are sets of definitions. Sequences are lists stored in consecutive memory. "FEL Programmer's Guide", R. M. Keller, AMPS TR 7, U Utah, March 1982.

FFP - Formal FP. Similar to FP, but with regular sugarless syntax, for machine execution. "Can Programming be Liberated From the von Neumann Style? A Functional Style and Its Algebra of Programs", John Backus, 1977 Turing Award Lecture, CACM 21(8):165-180 (Aug 1978). (See FP, FL).

FGHC - Flat GHC. A variant of GHC in which guard calls can be only to primitives. (See KL1).

FGL -

1. Flow Graph Lisp. A distributed dataflow language for AMPS (Applicative Multi-Processing System). "A Loosely-Coupled Applicative Multi-Processing System", R. Keller et al, NCC, AFIPS June 1979, pp.613-622.

2. Function Graph Language. Related to FEL.

FGL+LV - "Functional Programming and the Logical Variable", G. Lindstrom, POPL 1985, pp.266-280.

FGRAAL - FORTRAN extended GRAPh Algorithmic Language. A FORTRAN extension for handling sets and graphs. "On a Programming Language for Graph Algorithms", W.C. Rheinboldt et al, BIT 12(2) 1972.

FIDIL - Based on "maps", generalized arrays whose index sets ("domains") are arbitrary d-dimensional sets. Domains are first-class objects and may

be constructed by union, intersection, etc. "Fidil: A Language for Scientific Programming", P.N. Hilfinger et al, TR UCRL-98057, LLNL Jan 1988.

FIDO - FInite DOmains. A constraint language implemented on top of Prolog. <ftp://ftp.uni-kl.de/pub1/unix/languages/fido/>\*

Fifth - An enhanced version of FORTH. M.S. Dissertation, Cliff Click <cliff@cs.rice.edu>, Texas A&M, 1985. Available from the Software Construction Co, (409)696-5432.

File Composition - Typesetting language. "File Composition System Reference Manual", No. 90388, Information Intl.

F+L - Equational clauses within function definitions to solve for logical variable bindings. "Functions plus Logic in Theory and Practice", R.B. Kieburtz, Feb 1987, unpublished.

FL - Function Level. John Backus, ca. 1985. Successor to FP. Dynamically typed. Adds higher-order functions, exceptions, user-defined types, and other features. "FL Language Manual, Parts 1 & 2", J. Backus et al, IBM Research Report RJ 7100 (1989). FL compiler: IBM Almaden, 1992. (See FP, FFP).  
info: John Williams <williams@almaden.ibm.com>

FLAIR - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

FLAP - Symbolic math, for IBM 360. "FLAP Programmer's Manual", A.H. Morris Jr., TR-2558 (1971) U.S. Naval Weapons Lab. Sammet 1969, p.506.

Flavors - D. Weinreb & D.A. Moon <moon@cambridge.apple.com> 1980. LISP with object-oriented features. "Object-Oriented Programming with Flavors", D.A. Moon, SIGPLAN Notices 21(11):1-8 (OOPSLA '86) (Nov 1986).

Fleng - Parallel logic language. "Massively Parallel Implementation of Flat GHC on the Connection Machine", M. Nilsson, Proc Intl Conf on 5th Gen Comp Sys, 1988, pp.1031-1040.

FLEX -

1. Faster LEX. A reimplement of Lex. "The FLEX Scanner Generator", Vern Paxson <vern@ee.lbl.gov>, Systems Engineering, LBL, CA. (See Lex).  
<ftp://ftp.ee.lbl.gov/pub/flex-2.4.7.tar.Z>  
//liege.ics.uci.edu/pub/irus/aflex-ayacc\_1.2a.tar.Z with Ada output  
//ftp.th-darmstadt.de/pub/programming/languages/C++/tools/  
flex++-3.0.tar.gz with C++ output

2. Real-time language for dynamic environments. "FLEX: Towards Flexible Real-Time Programs", K. Lin et al, Computer Langs 16(1):65-79 (Jan 1991).

3. Alan Kay, ca 1967. Early object-oriented language for the FLEX machine. A simplification of Simula, and a predecessor of Smalltalk.

Flex 2 - ca. 1980. A preprocessor designed to make FORTRAN look more like Pascal. DECUS?

FLIC - Functional Language Intermediate Code. Intermediate language used

in the Chalmers LML compiler. "FLIC - A Functional Language Intermediate Code", S. Peyton Jones <simonpj@cs.ucl.ac.uk> et al, RR 148, U Warwick, Sep 1989.

FLIP -

1. Early assembly language on G-15. Listed in CACM 2(5):16 (May 1959).
2. "FLIP User's Manual", G. Kahn, TR 5, INRIA 1981.
3. Formal LIst Processor. Early language for pattern-matching on LISP structures. Similar to CONVERT. "FLIP, A Format List Processor", W. Teitelman, Memo MAC-M-263, MIT 1966.
4. J. B. Hext, U Sydney. A simple functional list processor, a simplified version of Lisp "M expressions", no functional arguments, no free variables. Implemented in KDF9 assembly language.

FLIP-SPUR - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May 1959).

F-Logic - "F-Logic: A Higher-Order Language for Reasoning about Objects, Inheritance and Scheme", ACM SIGMOD May 1989, pp.134-146.

FLOP - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

Floop - Douglas Hofstadter, 1979. Imperative language, designed for pedagogical purposes. Mechanizes functions. Douglas Hofstadter, "Godel, Escher, and Bach: An Eternal Golden Braid", Basic Books, Chap 13. ISBN 0-465-02685-0. Implementation in Perl by John Cowan <cowan@locke.ccil.org>, 1994. (See Bloop).  
ftp://locke.ccil.org:/pub/retro/bloop.\*

FLOW-MATIC or FLOWMATIC - (originally B-0). Remington Rand, 1958. Possibly the first English-like DP language. For UNIVAC I. Sammet 1969, pp.316-324.

FLPL - FORTRAN List Processing Language. Rochester, Gelernter, and Gerberich, ca 1960. Package of FORTRAN subroutines for handling lists. Weizenbaum's program Eliza was first implemented in FLPL. Sammet 1969, p.388.

FLUB - First Level Under Bootstrap. Language for an abstract machine, designed for the implementation of a system-independent macro processor. Its output is in SIMCMP. Used to implement STAGE2. Implementing Software for Non-numeric Applications, W. M. Waite, P-H 1973.

FMPL - Frobozz Magic Programming Language. Experimental Computing Facility, UC Berkeley. A prototype-based object-oriented language with functional features. Event-driven.  
ftp://xcf.berkeley.edu/src/local/fmpl  
list: fmpl@xcf.berkeley.edu

FOCAL -

1. FOrmula CALculator. Rick Merrill, DEC, 1969. Interactive, for PDP-5/PDP-8's, a descendant of AID/JOSS. Versions: FOCAL-69, FOCAL-1971,

FOCAL-11 (for PDP-11 under RT-11).  
ftp://locke.ccil.org/pub/retro/focal-81.shar.gz      Interpreter

2. Forty-One CALculator Language. Programming language of the HP-41 calculator line.

FOCL - Expert system shell, a backward chaining rule interpreter for Mac.  
ftp://ics.uci.edu/pub/machine-learning-programs/KR-FOCL-ES.cpt.hqx  
info: pazzani@ics.uci.edu

FOCUS - Hierarchical database language. Information Builders Inc.

FOIL - File Oriented Interpretive Language. CAI language. "FOIL - A File Oriented Interpretive Language", J.C. Hesselbart, Proc ACM 23rd National Conf (1968).

foogol - Per Lindberg. A tiny ALGOL-like language based on the VALGOL I compiler, G.A. Edgar, DDJ May 1985.  
ftp:/comp.sources.unix/V8      Per Lindberg's original, generates VAX MACRO  
    //locke.ccil.org/pub/retro/cfoogol.shar.gz      by John Cowan, generates C  
    //wuarhive.wustl.edu/systems/amiga/fish/fish/ff066

FOOL - Fool's Lisp. A small Scheme interpreter.  
ftp://scam.berkeley.edu/src/local/fools.tar.Z

FOOP - OBJ2 plus object-orientation. "Extensions and Foundations for Object-Oriented Programming", J. Goguen et al, in Research Directions in Object-Oriented Programming, B. Shriver et al eds, MIT Press 1987.

FORC - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

Force - dBASE dialect for MS-DOS.

The Force - Data parallel language, providing extensions to Fortran for shared memory multiprocessors. Parallel 'case' statements and critical sections. "The Force", H. Jordan in The Characteristics of Parallel Algorithms, L. Jamieson et al eds, MIT Press 1987, pp.395-436.  
info: Harry Jordan <harry@boulder.colorado.edu>

ForceOne - Andrew K. Wright. "Polymorphism in the Compiled Language ForceOne", G.V. Cormack et al, Proc 20th Annual Hawaii Intl Conf on System Sciences, 1987, pp.284-292. "Design of the Programming Language ForceOne", A.K. Wright, MS Thesis, U Waterloo 1987.

ForceTwo - Andrew K. Wright. An unofficial successor to ForceOne.

FORM - Jos Vermaseren <t68@nikhefh.nikhef.nl> 1989. Designed for speedy handling of very large-scale symbolic math calculations. A descendant of Schoonschip. Available for many PC's and workstations.  
ftp://acm.princeton.edu, //nikhefh.nikhef.nl  
info: form@can.nl

FORMAC - FORMula MANipulation Compiler. J. Sammet & Tobey, IBM Boston APD, 1962. Extension of FORTRAN for symbolic math. "Introduction to FORMAC", J.E. Sammet et al, IEEE Trans Elec Comp (Aug 1964). Versions: PL/I-FORMAC and FORMAC73. Sammet 1969, pp.474-491.

FORMAL -

1. FORMula MANipulation Language. An early FORTRAN extension for symbolic math. "FORMAL, A Formula Manipulation Language", C.K. Mesztenyi, Computer Note CN-1, CS Dept, U Maryland (Jan 1971).

2. IBM LASC. Data manipulation language for nonprogrammers. "FORMAL: A Forms-Oriented and Visual-Directed Application System", N.C. Shu, IEEE Computer 18(8):38-49 (1985).

FORMAT-FORTRAN - FORTRAN Matrix Abstraction Technique FORTRAN. Manipulation, printing and plotting of large matrices. "FORMAT-FORTRAN Matrix Abstraction Technique (Vol. V)" AFFDL-TR-66-207, Douglas Aircraft Co (Oct 1968).

Formes - Object-oriented language for music composition and synthesis, written in VLISP. "Formes: Composition and Scheduling of Processes", X. Rodet & P. Cointe, Computer Music J 8(3):32-50 (Fall 1984).

FORML - Formal Object Role Modeling Language. CASE language?

Formula -

1. FORTH Music Language. An extension of FORTH with concurrent note-playing processes. "Formula: A Programming Language for Expressive Computer Music", D.P. Anderson et al Computer 24(7):12 (Jul 1991). For Mac and Atari ST, with MIDI output.

2. Preprocessor language for the Acorn Archimedes, allowing inline high-level statements to be entered in an assembly program. Written in nawk.

Formula ALGOL - ALGOL extension for symbolic math, strings and lists. Carnegie, CDC G-20, 1962. A.J. Perlis & R. Iturriaga. Sammet 1969, p.583. "An Extension of ALGOL for Manipulating Formulae", A.J. Perlis et al, CACM 7(2):127-130 (Feb 1964).

Fornax - "Fornax: A General Purpose Programming Language", J. Storrs Hall, Rutgers U, USENIX Symp on Very High Level Languages, Oct 1994.

Forsythe - An ALGOL-like language. "Preliminary Design of the Programming Language Forsythe", J.C. Reynolds, CMU-CS-88-159, 1988.  
ftp://e.ergo.cs.cmu.edu

FORTH - Fourth. Charles H. Moore, 1960's. An interactive extensible language using postfix syntax and a data stack. A program is a set of functions ("words") which are compiled by an outer interpreter into bytecodes. FORTH is small and efficient, but programs can be difficult to read. Used first to guide the telescope at NRAO, Kitt Peak. Versions include FORTH 79 and FORTH 83. FORTH Interest Group, Box 1105, San Carlos CA 94070. ANSI standard soon to be adopted.

FORTRAN - FORMula TRANslator. The first and still the most widely used language for numerical calculations. Nonrecursive, efficient.

FORTRAN I - John Backus, IBM for the IBM 704. Design begun 1954, compiler released April 1957.

FORTRAN II - 1958. Added subroutines.

FORTRAN III - This was only distributed to ca. 20 sites. See Wexelblat.

FORTRAN IV - IBM 1962. For the IBM 7090/94. Many implementations went well beyond the original definition.

FORTRAN V - Preliminary work on adding character handling facilities by IBM ca. 1962. This name never really used.

FORTRAN VI - Internal IBM name for early PL/I work ca. 1963. Sammet 1969, p.540.

FORTRAN 66 - FORTRAN IV standardized. ASA X3.9-1966.

FORTRAN 77 - Block IF, PARAMETER, SAVE statements added, still no WHILE. Fixed-length character strings, format-free I/O, arrays with lower bounds. ANSI X3.9-1978.

<ftp://prep.ai.mit.edu/pub/gnu/g77>

[//ftp.cso.uiuc.edu/amiga/fish/ff470/BCF](ftp://ftp.cso.uiuc.edu/amiga/fish/ff470/BCF) for Amiga

Fortran 90 - Previously Fortran 8x and Fortran Extended. An extensive enlargement of FORTRAN 77. Derived types, assumed shape arrays, array sections, functions returning arrays, case statement, module subprograms and internal subprograms, optional and keyword subprogram arguments, recursion, and dynamic allocation. ISO 1539:1991, soon to be adopted by ANSI. "Fortran 90 Explained", M. Metcalf et al, Oxford University Press 1990.

Fortran D - Ken Kennedy, Rice U. A data-parallel Fortran. "Fortran D Language Specification", G. Fox et al, TR 90079, Rice U, March 1991.  
info: Theresa Chapman <tlc@cs.rice.edu>

Fortran-Linda - Scientific Computer Assocs <linda@sca.com>.

Fortran M - Parallel extensions to Fortran with processes and channels. "Fortran M: A Language for Modular Parallel Programming", I. Foster et al, MCS-P327-0992, ANL, 1992.  
info: Ian Foster <fortran-m@mcs.anl.gov>

FORTRAN-Plus - FORTRAN for the DAP parallel machine, implements many Fortran 90 features.

FORTRANSIT - FORTRAN Internal Translator. Subset of FORTRAN translated into IT on the IBM 650. Sammet 1969, p.141.

FORTRUNCIBLE - A cross between FORTRAN and RUNCIBLE for the IBM 650. Listed in CACM 2(5):16 (May 1959).

FOSIL - Fredette's Operating System Interface Language. A portable job control language, for IBM OS360, UNIVAC EXEC 8 and Honeywell GCOS. "Fredette's Operating System Interface Language (FOSIL)", G.N. Baird in Command Languages, C. Unger ed, N-H 1973.

FoxBASE+ - dBASE III+-like product from Fox Software, Perrysburg, OH.

FoxPRO - dBASE IV-like product from Fox Software, Perrysburg, OH.

FP - Functional Programming. Backus. Combinator based. "Can Programming be Liberated From the von Neumann Style? A Functional Style and Its Algebra of Programs", John Backus, 1977 Turing Award Lecture, CACM 21(8):165-180 (Aug 1978). (See FFP, FL, IFP).

ftp://apple.com/ArchiveVoll/unix\_lang

Berkeley FP - distributed with 4.2BSD

ftp: comp.sources.unix/Volume20: fpc translates FP programs to C.

comp.sources.unix/volume13: FP by Andy Valencia

FP2 - Functional Parallel Programming. Term rewrite rules used to specify algebraic data types and parallel processes. "Term Rewriting as a Basis for the Design of a Functional and Parallel Programming Language. A Case Study: The Language FP2", Ph. Jorrand in Fundamentals of Artificial Intelligence, LNCS 258, Springer 1986, pp.221-276.

FP/M - An intermediate language for functional languages, used to implement Hope. "The Compilation of FP/M Programs into Conventional Machine Code", A.J. Field, Imperial College, London, 1985. "Functional Programming", A.J. Field & M.C. Harrison, A-W 1988.

FQL - Functional database language. "An Implementation Technique for Database Query Languages", O.P. Buneman et al, ACM Trans Database Sys 7(2):164-186 (June 1982).

FrameKit - Frame language. "The FrameKit User's Guide", E. Nyberg, TR CMU-CMT-88-MEMO, CMU 1988.

FRANK - "Using BINS for Interprocess Communication", P.C.J. Graham, SIGPLAN Notices 20(2):32-41 (Feb 1985).

Franz Lisp - (named for the Hungarian composer Franz Liszt (1811-1886)) R. Fateman et al, UC Berkeley ca 1980. A MacLisp-like dialect of LISP, developed primarily for work in symbolic algebra. Written in C. "The FRANZ LISP Manual", J.K. Foderaro et al. UC Berkeley 1980. Version: Opus 38.22. Liszt (the compiler) Version 8.08.  
ftp://ted.cs.uidaho.edu/pub/hol/franz.tar.Z

FRED - Robert Carr. Language used by Framework, Ashton-Tate.

Fresco - Object-oriented specification language. "Refinement in Fresco", in Object Oriented Specification Case Studies, K. Lano et al eds, P-H 1993.

Fresh - "Fresh: A Higher-Order Language Based on Unification", G. Smolka, in Logic Programming: Functions, Relations and Equations", D. DeGroot et al, P-H 1986, pp.469-524.

FRINGE - C. Katz, GE, 1961. Subcomponent of GE-255 GECOM system. Sorting and merging of data, reports and file maintenance.

FRL - Frame Representation Language. MIT. "The FRL Manual", R. Roberts et al, AI Memo 409, MIT AI Lab, 1977.

FRMT-FTRN - Scientific language, listed [?] 1976.

FSL - Formal Semantics Language. Language for compiler writing. "A Formal Semantics for Computer Languages and its Application in a Compiler-

Compiler", J.A. Feldman, CACM 9(1) (Jan 1966). Sammet 1969, p.641.

FSMDL - Finite State Machine Description Language. [?]

Fugue - Music language, implemented in Xlisp. "Fugue: A Functional Language for Sound Synthesis", R.B. Dannenberg et al, Computer 24(7):36-41 (Jul 1991).

Fun - A typed lambda-calculus, similar to SOL[2]. "On Understanding Types, Data Abstractions and Polymorphism", L. Cardelli et al, ACM Comp Surveys 17(4) (Dec 1985).

FUNLOG - Functional programming plus unification. "Lazy" in the sense that expressions are reduced only if they are not unifiable. "FUNLOG: A Computational Model Integrating Logic Programming and Functional Programming", P.A. Subrahmanyam et al, in Logic Programming: Functions, Relations and Equations, D. DeGroot et al eds, P-H 1986.

FX-87 - Effects. A polymorphic language based on Scheme, allowing side effects and first-class functions. Attempts to integrate functional and imperative programming. Expressions have types, side effects (e.g. reading, writing or allocating) and regions (stating where the effects may occur). "The FX-87 Reference Manual", D.K. Gifford <gifford@lcs.mit.edu> et al, MIT/LCS/TR-407, Oct 1987. Version: FX-89.  
ftp://brokaw.lcs.mit.edu

FX-90 - Partial type and effect reconstruction and first-class modules.

G -

1. "G: A Functional Language with Generic Abstract Data Types", P.A.G. Bailes, Computer Langs 12(2):69-94 (1987).

2. Oregon State U 1988. Combines functional, object-oriented, relational, imperative and logic programming (you name it we got it). "The Multiparadigm Language G", J. Placer, Computer Langs 16:235-258(1991).

Gabriel - Graphical DSP language for simulation and real systems. "A Design Tool for Hardware and Software for Multiprocessor DSP Systems," E.A. Lee, E. Goei, J. Bier & S. Bhattacharya, DSP Systems, Proc ISCAS-89, 1989.

GADS - Picture retrieval language. "Integrated Geographical Databases: The GADS Experience", P.E. Mantey et al, in Database Techniques for Pictorial Applications, A. Blaser ed, pp.193-198.

Gaelic - For automated test programs. Used in military, essentially replaced by ATLAS.

Galaxy - An extensible language in the vein of EL/1 and RCC. "Introduction to the Galaxy Language", Anne F. Beetem et al, IEEE Software 6(3):55-62.

Galileo - "Galileo: A Strongly Typed Interactive Conceptual Language", A. Albano et al, ACM Trans Database Sys 10(2):230-260 (June 1985).

Gambit - A variant of Scheme R3.99 supporting the 'future' construct of Multilisp. Implementation includes compilers for Mac and 680x0 Unix systems. Version: 2.0

ftp://acorn.cs.brandeis.edu/dist  
//trex.iro.umontreal.ca/pub/gambit/gambit-2.0/\*  
//ftp.cso.uiuc.edu/pub/amiga/fish/f7/ff764/Gambit\_Terp  
info: gambit@trex.umontreal.ca

GAMMA -

1. language for matrices and generation of mathematical programming reports. "GAMMA 3.3 for MPS/MPSX, IBM System:/360", Bonnor & Moore Assocs (Mar 1975).

2. A high-level parallel language. Research Directions in High-Level Parallel Languages, LeMetayer ed, Springer 1992.

GAN - Generating and Analyzing Networks. "GAN - A System for Generating and Analyzing Activity Networks", A. Schurmann, CACM 11(10) (Oct 1968).

GAP - Groups Algorithms and Programming. Johannes Meier, Alice Niemeyer, Werner Nickel, Martin Schonert, Aachen 1988. Symbolic math for computational discrete algebra, especially group theory. "GAP 3.3 Manual, M. Schonert et al, Lehrstuhl D Math, RWTH Aachen, 1993. Designed 1986, implemented 1987, version 2.4 released 1988, version 3.1 released 1992. ftp://ftp.math.rwth-aachen.de/pub/gap, version for Sun.

GAPLog - General Amalgamated Programming with Logic. LOGPRO group, Linkoping Sweden. Restricted version of constraint logic programming, using S-unification but not restricted to a single domain. "Logic Programs with External Procedures", J. Maluszynski et al, in Logic Programming Languages, Constraints, Functions and Objects, K.R. Apt et al eds, MIT Press 1993.

Gargoyle - For compiler writing. J.V. Garwick, CACM 7(1):16-20, (Jan 1964).

GARP - Graphical language for concurrent programming. "Visual Concurrent Programming in GARP", S.K. Goering et al, PARLE '89 v.II, LNCS 366, pp.165-180.

GASP - Graph Algorithm and Software Package. PL/I extension for programming graph algorithms. "GASP - Graph Algorithm Software Package", S. Chase, TR CS Dept, U Illinois, Dec 1969.

GAT - Generalized Algebraic Translator. Improved version of IT. On IBM 650 RAMAC. Sammet 1969, p.142.

GATE - GAT Extended? Based on IT. Sammet 1969, p.139.

Gauss - Aptech Systems [?]

Gawk - GNU's implementation of a superset of POSIX awk, a pattern scanning and data manipulation language.

ftp://prep.ai.mit.edu/pub/gnu/gawk-2.15.4.tar.Z  
//archive.umich.edu/mac/utilities/developerhelps/macgawk2.11.cpt.hqx

GCL -

1. General Control Language. A portable job control language. "A

General Control Interface for Satellite Systems", R.J. Dakin in Command Languages, C. Unger ed, N-H 1973.

2. Gnu Common Lisp. 1994. Formerly AKCL. (See KCL and AKCL).  
ftp://math.utexas.edu/pub/gcl/

G-Code -

1. Johnsson & Augustsson, Chalmers Inst Tech. Intermediate language used by the G-machine, an implementation of graph reduction based on supercombinators. "Efficient Compilation of Lazy Evaluation", T. Johnsson, SIGPLAN Notices 19(6):58-69 (June 1984).

2. A machine-like language for the representation and interpretation of attributed grammars. Used as an intermediate language by the Coco compiler generator. "A Compiler Generator for Microcomputers", P. Rechenberg et al, P-H 1989.

GDPL - Generalized Distributed Programming Language. "GDPL - A Generalized Distributed Programming Language", K. Ng et al, Proc 4th Intl Conf Distrib Comp Sys, IEEE 1984, pp.69-78.

GEA - Graph Extended ALGOL. Extension of ALGOL-60 for graph manipulation, on UNIVAC 1108. "A Language for Treating Graphs", S. Crespi-Reghezzi et al, , CACM 13(5) (May 1970).

GECOM - For the GE-255. Somewhat akin to COBOL with some ALGOL features added. Comprised of ALGOL, COBOL, FRINGE and TABSOL. FRINGE and TABSOL may not have actually been implemented. Sammet 1969, p.329.

Gedanken - John Reynolds, 1970. "GEDANKEN - A Simple Typeless Language Based on the Principle of Completeness and the Reference Concept", J.C. Reynolds, CACM 13(5):308-319 (May 1970).

GEL - Scripting language used in the object-oriented development environment GainMomentum. Mentioned in Structured Rapid Prototyping, 1989 or Object-Oriented Rapid Prototyping, John L. Connell, Yourdon Press 1995.

General Purpose Graphic Language - "A General Purpose Graphic Language", H.E. Kulsrud, CACM 11(4) (Apr 1968).

Gentleman's Portable Coroutine System - Coroutine package in FORTRAN. "A Portable Coroutine System", W.M. Gentleman, Info Proc 71, C.V. Freiman ed, 1972.

GEORGE - Charles Hamblin, 1957. One of the earliest programming languages, stack-oriented, used reverse Polish notation. Implemented on the English Electric DEUCE. "GEORGE: A Semi-Translation Programming Scheme for the DEUCE, Programming and Operations Manual", C. L. Hamblin, U New S Wales (1958). "Computer Languages", C.L. Hamblin, Aust J Sci 20(5):135-139 (Dec 1957) and Aust Comp J 17(4):195-198 (Nov 1985).

GEPURS - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

Gerald - "Gerald: An Exceptional Lazy Functional Programming Language", A.C. Reeves et al, in Functional Programming, Glasgow 1989, K. Davis et al eds, Springer 1990.

GEST - Generic Expert System Tool. Expert system shell with frames, forward and backward chaining, fuzzy logic. Version 4.0. For Symbolics LISP machines only.

<ftp://ftp.gatech.edu/pub/ai/gest.tar.Z>

info: John Gilmore <John.Gilmore@gtri.gatech.edu>

GHC - Guarded Horn Clauses. K. Ueda. Parallel logic language similar to Parlog. Guards and committed-choice nondeterminism. (See FGHC, KL1).

Gia-2 - Gary's Ikonas Assembler. "Differences Between GIA-2 and C", G. Bishop, TR 82-010, U N Carolina (June 1982).

GIM-1 - Generalized Information Management Language. Nelson, Pick, Andrews. Proc SJCC 29:169-73, AFIPS (Fall 1966).

GIN5 - Special-purpose macro assembler used to build the GEORGE 3 operating system for ICL1900 series machines. A macro could examine the assembly process and possibly modify its further course.

Ginger - U Warwick. Simple functional language with parallel constructs.

GIP - General Interpretive Programme. 1956. An early interpreted language for the English Electric DEUCE, with array operations and an extensive library of numerical methods. "Interpretive and Brick Schemes, with Special Reference to Matrix Operations", English Electric Company, DEUCE News No. 10 (1956).

GIRL - Graph Information Retrieval Language. Handling directed graphs. "Graph Information Retrieval Language", S. Berkowitz, Report 76-0085, Naval Ship Res Dev Center, (Feb 1976).

GKS - Graphical Kernel System.

GL - Graphics Language. Silicon Graphics.

Glammar - A pattern transformation language for text-to-text translation, Used for compiler writing and linguistics.

<ftp://phoibos.cs.kun.nl/pub/GLASS/glammar.tar.Z>

GLASS - General LAnguage for System Semantics. Esprit project at KU Nijmegen.

<ftp://phoibos.cs.kun.nl/pub/GLASS>

Glenda - Seyfarth, Arumugham and Bickam, U South Mississippi. A realization of Linda based on PVM. (See PVM).

Glish - Vern Paxson <vern@ee.lbl.gov>. Language for building loosely coupled distributed systems from modular event-oriented programs. "Glish: A User-Level Software Bus for Loosely Coupled Distributed Systems", V. Paxson et al, Proc 1993 Winter USENIX Conf, Jan 1993.

<ftp://ftp.ee.lbl.gov/glish/glish-2.3.2.tar.Z>

Glisp - Generalized LISP. D.C. Smith, Aug 1990. A coordinated set of high-level syntaxes for Common LISP. Contains Mlisp, Plisp and ordinary LISP, with an extensible framework for adding others. Written in Plisp.

<ftp://bric-a-brac.apple.com/dts/mac/lisp>

GLOS - Graphics Language Object System. Dan Johnston dan@cs.uq.oz.au> and Brian Hicks <cs.uq.oz.au>, U Queensland, St. Lucia 1978. Graphics objects correspond to language statements (e.g. line, circle, polygon etc). New objects defined using procedures. 2-D Transformations are context dependent and may be nested.

GLOW - A POP-11 variant with lexical scope. Reviewed in Byte's UK edition, May 1992, p.84UK-8. Available from Andrew Arnblaster, Bollostraat 6, B-3140 Keerbergen, Belgium, for Mac or MS-DOS.

Glypnir - 1966. An ALGOL-like language with parallel extensions. Similar to Actus. "GLYPNIR - A Programming Language for the Illiac IV", D.H. Lawrie et al, CACM 18(3) (Mar 1975).

GMAP - GCOS Macro Assembler Program - Macro assembler for the GCOS-8 operating system on Honeywell/Bull DPS-8 machines. "GCOS8 OS GMAP User's Guide", Bull.

GMPL - A microprogramming language for an HP machine. "A Microprogramming Language Directed Architecture", R.M. Guffin, Proc 15th Ann Workshop Microprogramming (MICRO-15), 1982, pp.42-49.

Goedel - Declarative language for AI, based on many-sorted logic. Strongly typed, polymorphic, declarative, with a module system. Supports bignums and sets. "The Goedel Programming Language", P. M. Hill et al, MIT Press 1994, ISBN 0-262-08229-2. Goedel 1.4 - partial implementation in SICStus Prolog 2.1.

ftp://ftp.cs.bris.ac.uk/goedel  
info: goedel@compsci.bristol.ac.uk

Gofer - Mark Jones <jones-mark@cs.yale.edu>, Oxford 1991. Similar to Haskell 1.1. Lazy evaluation, higher order functions, pattern matching, and type classes. Lambda, case, conditional and let expressions, and wildcard, as and irrefutable patterns. Lacks modules, arrays, standard classes. "Introduction to Gofer 2.20", M.P. Jones. Interpreter in C. Unix Version 2.28a, Mac\_Gofer version 0.16 beta.  
ftp://ftp.dcs.glasgow.ac.uk/pub/haskell/gofer/\*

GOL - General Operating Language. Subsystem of DOCUS. Sammet 1969, p.678.

GOM - Good Old MAD. Don Boettner, U Mich. MAD for the IBM 360. Parts of the MTS Time-sharing system were written in GOM.

GOOD - Graph-Oriented Object Database. A graph manipulation language for use as a database query language. "A Graph-Oriented Object Database Model", M. Gyssens et al, Proc ACM Symp Princs of Database Sys, Mar 1990.

GOSPL - Graphics-Oriented Signal Processing Language. A graphical DSP language for simulation. "Graphic Oriented Signal Processing Language - GOSPL", C.D. Covington et al, Proc ICASSP-87, 1987.

GP - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

GPL -

1. General Purpose Language. ALGOL 60 variant with user-definable types

and operators. Sammet 1969, p.195. "The GPL Language", J.V. Garwick et al, TER-05, CDC, Palo Alto 1969.

2. "A Sample Management Application Program in a Graphical Data-driven Programming language", A.L. Davis et al, Digest of Papers, Compcon Spring 81, Feb 1981, pp.162-167.

3. Genken Programming Language. K. Asai, Japan Atomic Energy Res Inst. Variant of PL360. "Experience With GPL", K. Asai, in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, pp.371-376.

GPM - General Purpose Macro-generator. Early text-processing language similar to TRAC, implemented on Atlas 2. "A General Purpose Macrogenerator", C. Strachey, Computer J 8(3):225-241 (Oct 1965).

GPSS - General Purpose Systems Simulator. Geoffrey Gordon, 1960. Discrete simulations. "The Application of GPSS V to Discrete System Simulation", G. Gordon, P-H 1975. Versions include GPSS II (1963), GPSS III (1965), GPS/360 (1967), and GPSS V (1970).

GPX - Early system on UNIVAC II. Listed in CACM 2(5):16 (May 1959).

GRAAL - ("Grail") General Recursive Applicative and Algorithmic Language. FP with polyadic combinators. "Gaal: A Functional Programming System with Uncurryfied Combinators and its Reduction Machine", P. Bellot in ESOP 86, G. Goos ed, LNCS 213, Springer 1986.

GRAF - Graphic Additions to FORTRAN. FORTRAN plus graphic data types. "GRAF: Graphic Additions to FORTRAN", A. Hurwitz et al, Proc SJCC 30 (1967). Sammet 1969, p.674.

GRAIL - Graphical Input Language. Flowchart language entered on a graphics tablet. The graphical followon to JOSS. "The GRAIL Language and Operations", T.O. Ellis et al, RM-6001-ARPA, RAND, Sept 1969.

GRAIN - Pictorial query language. "Pictorial Information Systems", S.K. Chang et al eds, Springer 1980.

GRAM - An extension of BNF used by the SIS compiler generator. "SIS - Semantics Implementation System", P.D. Mosses, TR DAIMI MD-30, Aarhus U, Denmark.

Grapes - A Modula-like system description language. "GRAPES Language Description. Syntax, Semantics and Grammar of GRAPES-86", Siemens Nixdorf Inform, Berlin 1991, ISBN 3-8009-4112-0.  
info: peter@cadlab.cadlab.de

Graphic ALGOL - Generation of shaded perspective pictures in real time. "An Extended ALGOL-60 for Shaded Computer Graphics", B. Jones, Proc ACM Symp on Graphic Languages, Apr 1976.

Graphic Language - For specifying graphic operations. "A Problem Oriented Graphic Language", P.J. Schwinn, proc ACM 22nd Natl Conf, 1967. Sammet 1969, p.677.

GRAPPLE - GRAPh Processing Language. 1968. "A Directed Graph Representation for Computer Simulation of Belief Systems", L.G. Tesler et

al, Math Biosciences 2:19-40 (1968).

GRASP/Ada - Graphical Representation of Algorithms, Structures and Processes. "A Graphically Oriented Specification Language for Automatic Code Generation", J.H. Cross, Auburn U, NASA CR-183212, 1989.

Green - Cii Honeywell-Bull. A proposed language to meet the DoD Ironman requirements which led to Ada. This language was the winner, in 1979. "On the GREEN Language Submitted to the DoD", E.W. Dijkstra, SIGPLAN Notices 13(10):16-21 (Oct 1978).

GRG - Computer algebra system for differential geometry, gravitation and field theory. Version 3.1 works with PSL-based REDUCE 3.3 or 3.4.  
info: V.V. Zhytnikov <vvzhy@phy.ncu.edu.tw>

GRIND - GRaphical INterpretive Display. Graphical input language for PDP-9. "GRIND: A Language and Translator for Computer Graphics", A.P. Conn, Dartmouth, June 1969.

Groff - GNU's implementation of roff. (See nroff, troff, RUNOFF).

GSBL - "GSBL: An Algebraic Specification Language Based on Inheritance", S. Clerici et al in in ECOOP '88, S. Gjessing et al eds, LNCS 322, Springer 1988, pp.78-92.

GSL - Grenoble System Language. M. Berthaud, IBM, Grenoble. "GSL Language Reference Manual", M. Berthaud et al, March 1973. "A MOL-Based Software Construction System", M. Berthaud et al, in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, pp.151-157.

GSPL - Greenberg's System Programming Language. Bernard Greenberg.

GTML - ? Mentioned in the documentation for TXL.

GVL - Graphical View Language. T.C.N. Graham & J.R. Cordy, Queen's U. Canada. A visual language for specifying interactive graphical output. "GVL: A Graphical, Functional Language for the Specification of Output in Programming Languages", J.R. Cordy & T.C.N. Graham, Proc IEEE Intl Conf on Comp Lang ICCL'90 (March 1990).

GW-BASIC - "Gee Whiz" BASIC. Microsoft's BASIC with graphic extensions.

Gypsy - Specification and verification of concurrent systems software. Message passing using named mailboxes. Separately compilable units: routine (procedure, function, or process), type and constant definition, each with a list of access rights. "Report on the Language Gypsy", A.L. Ambler et al, UT Austin ICSCS-CMP-1 Aug 1976.

GYVE - OS programming language, highly modular (similar to Modula?) "GYVE, A Programming Language for Protection and Control in a Concurrent Processing Environment", Phillip Shaw, Courant Inst, NYU, 1978.

HAL/S - Real-time language used by NASA for onboard shuttle software. "Two-Dimensional Characteristics of HAL, A Language for Spaceflight Applications", J.S. Miller, SIGPLAN Notices 7(10) (Oct 1972).

HALGOL - Hewlett-Packard. A simple language for communicating with devices

such as modems and X.25 PADs.

HALMAT - Intermediate language used by HAL/S.

Haskell - (named for the logician Haskell B. Curry). April 1990. Designed by a committee from the functional programming community. A lazy purely functional language largely derived from Miranda. Static polymorphic typing, higher-order functions, user-defined algebraic data types, and pattern-matching list comprehensions. Innovations include a class system, operator overloading, functional I/O system, functional arrays, and separate compilation. "Report on the Programming Language Haskell Version 1.1", Paul Hudak & P. Wadler eds, CS Depts, U Glasgow and Yale U. (Aug 1991). Version 1.2: SIGPLAN Notices 27(5) (Apr 1992).

list: haskell-request@cs.yale.edu, HASKLD-L@YALEVM.BITNET.

Yale Haskell - Version 2.0.6, Haskell 1.2 built on Common Lisp

ftp://nebula.cs.yale.edu/pub/haskell/yale/\*

info: haskell-request@cs.yale.edu

Glasgow Haskell - Version 0.20, generates C output

ftp://ftp.dcs.glasgow.ac.uk/pub/haskell/glasgow/ghc\*

info: glasgow-haskell-request@dcs.glasgow.ac.uk

Haskell-B - Haskell 1.2 implemented in LML, generates native code

ftp://ftp.cs.chalmers.se/pub/haskell/chalmers/lml-0.999.3.\*

info: hbc@cs.chalmers.se

HASL - SASL plus conditional unification. "A Prological Definition of HASL, A Purely Functional Language with Unification Based Conditional Binding Expressions", H. Abramson in Logic Programming: Functions, Relations and Equations, D. DeGroot et al eds, P-H 1986.

HCLP - Hierarchical CLP. "Constraint Hierarchies and Logic Programming", A. Borning et al, in Proc Sixth Intl Logic Prog Conf, June 1989, pp. 149-164.

H CPRVR - "H CPRVR: An Interpreter for Logic Programs", D. Chester in Proc First Natl Conf on AI, Stanford, 1980.

HDFL - Single assignment language. "Methods for Handling Structures in Data-Flow Systems", J.L. Gaudiot, Proc 12th Intl Symp Comp Arch, June 1985.

HDM - See SPECIAL.

HELP -

1. DEA. Language for industrial robots.

2. A lazy Scheme.

3. An implementation of FLIP in Wisp. Implementing Software for Non-numeric Applications, W. M. Waite, P-H 1973.

HEQS - E. Derman. Constraint language for financial modeling. Uses an extension of the equation solver in IDEAL. "A Simple Equation Solver and Its Application to Financial Modeling", E. Derman et al, Soft Prac & Exp 14(12):1169-1181 (Dec 1984).

HERAKLIT - A distributed object-oriented language. "Definition einer objektorientierten Programmiersprache mit hierarchischem Typkonzept", B.

Hindel, diss U Erlangen-Nuernberg, Dec 1987.

Hermes - IBM, June 1990. An imperative, strongly typed process-oriented language for complex distributed systems. A follow-on effort to NIL[2]. Threads, relational tables, typestate checking, capability-based access, dynamic configuration. "Hermes: A Language for Distributed Computing", R. Strom et al, P-H 1991, ISBN 0-13-389537-8.  
ftp://software.watson.ibm.com/pub/hermes 0.7alpha for Unix  
info: hermes-request@watson.ibm.com

HIBOL - A variant of DIBOL, used in Infotec computers.

High Performance Fortran - Proposed extension to Fortran 90 with additional support for data parallel programming. "High Performance Fortran: Status Report", G.L. Steele Jr <gls@think.com>, SIGPLAN Notices 28(1):1-4 (Jan 1993).

HiLog - W. Chen et al, Stony Brook, 1989. Logic programming in higher order logic. "HiLog as a Platform for Database Languages (Or Why Predicate Calculus is Not Enough)", W. Chen et al, 2nd Intl Workshop on Database Prog Langs, Morgan Kaufmann, 1989.  
ftp://sbcs.sunysb.edu/SB-hilog

HINT - Hierarchical Information NeTs. For CDC 3600. "HINT: A Graph Processing Language", R.D. Hart, Michigan State U, Apr 1970.

HLISP - "Monocopy and Associative Algorithms in an Extended Lisp", E. Goto, U Tokyo May 1974.

HLL - A machine-independent high level microprogramming language. "Automatic Microcode Generation for Horizontally Microprogrammed Processors", R.J. Sheraga et al, Proc 14th Ann Workshop Microprogramming (MICRO-14), 1981, pp.154-168.

HOL - Higher Order Logic. A proof-generating system for higher order logic based on LCF. "HOL: A Machine Oriented Formulation of Higher Order Logic", M.J.C. Gordon, Report 68, Comp Lab U Cambridge (1985). "Introduction to HOL", M.J.C. Gordon et al, Cambridge U Press 1993 ISBN 0-521-441897  
HOL-88 built on ML, from Mike Gordon <mjcg@cl.cam.ac.uk>  
ftp://ted.cs.uidaho.edu/pub/hol  
list: info-hol@ted.cs.uidaho.edu  
HOL-90 built on SML/NJ, from Brian Graham <graham@cpsc.ucalgary.ca>  
ftp://fsa.cpsc.ucalgary.ca/pub/hol90.tar.Z for Sun 4  
list: info-hol@clover.ucdavis.edu

Honeywell-800 Business Compiler - Another name for FACT. Sammet 1969, p.327.

HOOK - ? Object Oriented Kernel. Delphia. An object-oriented extension of Delphia Prolog.

Hope - ("springs eternal" and so forth.) R.M. Burstall, U Edinburgh 1978. Functional language with polymorphic types and lazy lists. First language to use call-by-pattern. "HOPE, an Experimental Applicative Language", R.M. Burstall et al, Conf Record 1980 LISP Conf, p.136-143 (Aug 1980), "A HOPE Tutorial", R. Bailey, BYTE Aug 1985, pp.235-258. "Functional Programming with Hope", R. Bailey, Ellis Horwood 1990.

<ftp://brolga.cc.uq.oz.au/pub/hope>. PC Hope, plus a lazy variant for Unix, Mac

Hope+ - Alvey Flagship project, Imperial College. An extension of Hope with real numbers, vectors, call-by-WHNF. "Hope+", N. Perry, Imperial College, IC/FPR/LANG/2.5.1/7, 1988.

Hope+C - Alvey Flagship project, Imperial College. Further evolution of Hope+ with continuation-based I/O, coroutines, and RFC's. For Sun-3's with Motorola FPU's. (See Massey Hope).  
info: John Darlington <jd@dic.ic.ac.uk>

HOS-STPL - Hospital Operating System - Structured Programming Language. A FORTRAN-like language with structured extensions. "HOS-STPL User Manual", Health Services Research, US Public Health Service (Jan 1975).

HPcode - Stack-based intermediate language used by HP in many of its compilers for RISC and stack-based architectures. Supports Fortran, Ada, Pascal, COBOL and C++. Descended from Stanford's U-code.

HPCode-Plus - Descendant of HPcode with data types, developed to be an ANDF language. "ANDF: Finally an UNCOL After 30 Years", M.E. Benitez, Jack Davidson <jwd@virginia.edu> et al, CS TR-91-05 U Virginia (Mar 1991). (See ANDF).

HPF - (see High Performance Fortran).

HP-GL - Hewlett-Packard Graphics Language. Vector graphics language used by HP plotters.

HP-GL/2 - "HP-GL/2 Programmer's Guide", No. 5959-9733, HP. (See PCL.)

HPL - Language used in HP9825A/S/T "Desktop Calculators", 1978(?) and ported to the early Series 200 family (9826 and 9836, MC68000). Fairly simple and standard, but with extensive I/O support for data acquisition and control (BCD, Serial, 16 bit custom and IEEE-488 interfaces), including interrupt handling. Currently owned by Structured Software Systems. "HPL Operating Manual for Series 200, Models 216, 226 and 235\6", HP 98614-90010, Jan 1984.

HSL-FX - Hierarchical Specification Language - Function Extension.

HTML - HyperText Markup Language. Markup language used by the World Wide Web. Plain text with tags enclosed in angle brackets. Defined in SGML.  
<http://www.ncsa.uiuc.edu/demoweb/html-primer.html>  
<http://info.cern.ch/hypertext/WWW/MarkUp/HTML.dtd.html>

HTML+ - Successor to HTML, will encode more structure. Under development.

HUGO - Geac. A bytecode-interpreted transaction handler.

Hybrid - Concurrent object-oriented language. "Active Objects in Hybrid", O.M. Nierstrasz, SIGPLAN Notices 22(12):243-253 (OOPSLA '87) (Dec 1987).

Hyper-C - HyperParallel Tech, France. Data parallel extension of C, for PVM, CM, Maspar. Available from Fortunel Systems <fortunel@vnet.net> (919) 319-1624.

info:hyperc-support@hyperparallel.polytechnique.fr

Hyperscript - Informix. The object-based programming language for Wingz, used for creating charts, graphs, graphics, and customized data entry.

HyperTalk - Bill Atkinson and Dan Winkler. A verbose semicompiled language with loose syntax and high readability. Relies on HyperCard as an object management system, development environment, and interface builder. Programs are organized into "stacks" of "cards", each of which may have "buttons" and "fields". All data storage is in zero-terminated strings in fields, local, or global variables; all data references are through "chunk expressions" of the form last item of background field "Name List" of card ID 34217'. Flow of control is event-driven and message-passing among scripts that are attached to stack, background, card, field and button objects. "Apple Macintosh HyperCard User Guide", Apple Computer 1987. "HyperTalk Language Reference Manual", A-W 1988. Available from Claris Corp.

HyTime - A hypermedia extension of SGML. "The HyTime Hypermedia/Time-based Document Structuring Language", S. Newcomb et al, CACM 34(11):67-83 (Nov 1991).

IAL - International Algebraic Language. Original name of ALGOL 58. "Preliminary report - International Algebraic Language", CACM 1(12):8 (1958).

IAM - Interactive Algebraic Manipulation. Interactive symbolic math for PDP-10. "IAM, A System for Interactive Algebraic Manipulation", C. Christensen et al, Proc Second Symp Symb Alg Manip, ACM Mar 1971.

IBEX - Command language for Honeywell's CP-6 OS.

ICES - Integrated Civil Engineering System. Subsystems include COGO, STRUDL, BRIDGE, LEASE, PROJECT, ROADS and TRANSET. Internal languages include ICETRAN and CDL. "An Integrated Computer System for Engineering Problem Solving", D. Roos, Proc SJCC 27(2), AFIPS (Spring 1965). Sammet 1969, pp.615-620.

ICETRAN - An extension of FORTRAN IV. Component of ICES. Sammet 1969, p.617.

ICI - Tim Long. Interactive CInterpreter? Interpreted language, syntax similar to C. Adds high-level garbage-collected associative data structures. Exception handling, sets, regular expressions, Dynamic arrays, database features, screen handling. Extensible.  
ftp://nexus.yorku.ca/pub/oz/ici.cpio.Z  
list: ici@research.canon.oz.au

Icon - Griswold, 1970's. A descendant of SNOBOL4 with Pascal-like syntax. Icon is a general-purpose language with special features for string scanning. Dynamic types. The central theme of Icon is the generator: when an expression is evaluated it may be suspended and later resumed, producing a result sequence of values until it fails. Resumption takes place implicitly in two contexts: iteration which is syntactically loop-like ('every-do'), and goal-directed evaluation in which a conditional expression automatically attempts to produce at least one result. Expressions that fail are used in lieu of Booleans. Data backtracking is

supported by a reversible assignment. Icon also has co-expressions, which can be explicitly resumed at any time. "The Icon Programming Language", Ralph & Marge Griswold, 2nd ed P-H 1990.

ftp://cs.arizona.edu

list: icon-group@arizona.edu

Iconicode - 1990-1992. Visual dataflow language, token-based with hierarchical, recursive and iterative constructs. Version: IDF with extensions for image processing. "IDF: A Graphical Data Flow Programming Language for Image Processing and Computer Vision", Neil Hunt, Proc IEEE Conf on Systems Man & Cybernetics, IEEE, Nov 1990. Available from Iconicon <icon@teleos.com>.

IC-Prolog - Clark & McCabe, Imperial College 1979. Logic language with coroutining. "IC-Prolog Language Features", K.L. Clark <klc@doc.ic.ac.uk> et al in Logic Programming, K.L. Clark et al eds, pp.253-266, Academic Press 1982.

IC Prolog ][ - Imperial College. A Prolog with multithreading, TCP primitives for interprocess communication, mailboxes, and an interface to Parlog. "IC Prolog ][: A Language for Implementing Multi-Agent Systems", Y. Cosmadopoulos et al, in Tutorial and Workshop on Cooperating Knowledge Based Systems, Keele U 1992.

info: Damien Chu <dac@doc.ic.ac.uk>

ftp://doc.ic.ac.uk/computing/programming/languages/icprolog/pd-ICP-0.94.tar.Z

Id - Irvine Dataflow. Arvind & Gostelow. Single assignment language, used on MIT's Tagged-Token Dataflow Architecture (and soon on Motorola's Monsoon). Incrementally compiled, non-strict. "An Asynchronous Programming Language for a Large Multiprocessor Machine", Arvind et al, TR114a, Dept ISC, UC Irvine, Dec 1978. "The U-Interpreter", Arvind et al, Computer 15(2):42-50 (1982). (See Id Nouveau).

IDAMS - Pictorial retrieval language, implemented in APL. "Concept of the Diagnostic Image Workstation", D. Meyer-Ebrecht, Proc 2nd Conf on Picture Archiving (PACS II), SPIE 418, pp.180-183 (1983).

IDEA - Interactive Data Entry/Access. Data General. A language in which you designed the screen first, and then wrote the program around the predefined fields. Precursor to the DG COBOL Screen Section.

IDEAL - Van Wyk, Stanford 1980. Numerical constraint language for typesetting graphics into documents. Inspired partly by Metafont. "A High-Level Language for Specifying Pictures", C.J. Van Wyk, ACM Trans Graphics 1(2):163-182 (Apr 1982). Distributed as part of Troff.

IDL -

1. Interactive Data analysis Language. Xerox. Built on Interlisp-D.

2. Interface Description Language. Nestor, Lamb & Wulf, CMU 1981. Description of data structures to be passed between the components of an application, to provide a language-independent intermediate representation. "The Interface Description Language", R. Snodgrass, Computer Science Press 1989. Also SIGPLAN Notices 22(11) (Nov 1987) special issue. Version by OMG (Object Management Group) for CORBA (Common Object Request Broker

Architecture).

list: info-idl@sei.cmu.edu

3. Interface Definition Language. Project DOE, SunSoft, Inc. Part of an effort to integrate distributed object technology into the Solaris OS. IDL provides the standard interface between objects, and is the base mechanism for object interaction.

info: <idl-cfe@sun.com> or Mache Creeger, SunSoft Inc (415)336-5884.

ftp://omg.org/pub/omg\_idl\_cfe.tar.Z

4. Interactive Data Language. Research Systems, 1977. Commercial array-oriented language with numerical analysis and display features, for Unix workstations, DOS Windows and VAX/VMS. Version 3.0.0

ftp://gateway.rsinc.com/pub/idl

info: info@rsinc.com

IDMS - Pictorial query language, an extension of Sequel2. "A Management System for an Integrated Database of Pictures and Alphanumeric Data", G.Y. Tang, Computer Graphics Image Processing 16:270-286 (1981).

Id Nouveau - Arvind <arvind@lcs.mit.edu> & Nikhil <nikhil@crl.dec.com>, LCS MIT, ca. 1986. Dataflow language, began as a functional language, added streams, resource managers and I-structures (mutable arrays). Loops are syntactic sugar for tail recursion. "Id Nouveau Reference Manual", R.S. Nikhil, CS TR, MIT, March 1988. "Id (Version 90.1) Reference Manual", R.S. Nikhil, CSG Memo 284-2, LCS MIT, July 15, 1991. (See Id).

IDOL - Icon-Derived Object Language. Object-oriented preprocessor for Icon. "Programming in Idol: An Object Primer", C.L. Jeffery, U Arizona CS TR #90-10.

ftp://src.doc.ic.ac.uk/pub/languages/icon/idol.tar.Z

IDS/I - Integrated Data Store. Extension to COBOL involving "chains" (circular lists), for GE computers. "A General Purpose Programming System for Random Access Memories", C.W. Bachman et al, Proc FJCC 26(1), AFIPS (Fall 1964). Sammet 1969, p.376.

IF1 - Graph language used as an intermediate language for dataflow hardware. Used by the OSC SISAL compiler. "The Manchester Prototype Dataflow Computer", J.R. Gurd et al, CACM 28(1):34-52 (Jan 1985).

IF2 - Graph language used by the OSC SISAL compiler, a superset of IF1. "IF2: An Applicative Language Intermediate Form with Explicit Memory Management", M. L. Welcome et al, UC-LLNL, Nov 1986.

IFIP - Subset of ALGOL. Sammet 1969, p.180.

IFP - Illinois FP. Arch Robinson. Variant of FP with Algol-like syntax. "The Illinois Functional Programming Interpreter", A.D. Robison, Proc 1987 SIGPLAN Conf on Interpreters and Interpretive Techniques (June 1987). "Illinois Functional Programming: A Tutorial", A.D. Robison, BYTE Feb 1987. ftp://a.cs.uiuc.edu/pub/ifp - Versions for Unix and MS-DOS

IFX - "Type Reconstruction with First-Class Polymorphic Values", J. O'Toole et al, SIGPLAN Notices 24(7):207-217 (Jul 1989).

IGL - Interactive Graphic Language. Used primarily by Physics Dept at

Brooklyn Poly, uses numerical methods on vectors to approximate continuous function problems that don't have closed form solutions. [Is this being confused with Tektronix's graphics library by the same name?]

IIS - Idealized Instruction Set. Assembly language for the Flagship parallel machine. "An Idealized Instruction Set for a Packet Rewrite Machine", J. Sargeant, Manchester U, 1988.

IITRAN - Simple PL/I-like language for students, on IBM 360. "The IITRAN Programming Language", R. Dewar et al, CACM 12(10):569-575 (Oct 1969).

ILIAD - Real time language. "On the Design of a Language for Programming Real-Time Concurrent Processes", H.A. Schutz, IEEE Trans Soft Eng SE-5(3):248-255 (May 1979).

ILLIAC - Assembly language for the ILLIAC computer. Listed in CACM 2(5):16, (May 1959) p.16.

ILOC - Rice U. Register-oriented intermediate language targeted to PC/RT. Source languages include FORTRAN and Russell.

IMP -

1. IMProved Mercury autocode. Used to code the Edinburgh Multi Access System (EMAS), one of the first OS's written in a high-level language, apparently predating Unix. Papers in J British Computer Soc.[?] Luis Damas' Prolog interpreter in IMP for EMAS, led to C-Prolog. (See Autocode).

2. Extensible dialect of ALGOL-60, on CDC 1604. Ned Irons, IDA, Princeton. "Experience with an Extensible Language", Edgar T. Irons, CACM 13(1):31-39 (Jan 1970).

3. Interpretive Menu Processor. Used to implement much of the user interface of the Alis office automation package from Applix, Inc.

4. IMPlementation language. W. Davidsen <davidsen@ariel.crd.ge.com>, 1970. An extension of B with floating point. For the GE 600, also cross-compiled to VAX and Intel 8080.

Ina Jo - [FDM?] "The Ina Jo Specification Language Reference Manual", J. Scheid et al, TR TM-(L)-6021/001/00, SDC Mar 1985.

Info BASIC - Variant of Pick BASIC used with PRIME's PRIMOS.

INFORM - Early database language, comparable to dBASE II. Intended for time-sharing use over telephone lines to teletypes. "INFORM Primer", March, 1974, CALL-A-COMPUTER, 30 Washington St, Wellesley Hills, MA 02181.

Information Algebra - Theoretical formalism for DP, never resulted in a language. Language Structure Group of CODASYL, ca. 1962. Sammet 1969, 709.

Inglish - English-like language used for Adventure games like "The Hobbit" (could distinguish between "take the rope and axe" and "take the money and run").

InnovAda - Object-oriented extension to Ada, said to be LISP-like.  
Implemented as an Ada preprocessor.

Input - See ALPHA.

INSIGHT - Simulation modeling language especially for health care problems.  
"Simulation Modeling with INSIGHT", S.D. Roberts Proc 1983 Winter Sim Conf,  
S.D. Roberts et al eds, pp.7-16.

INTCODE - A low-level interpreted language used in bootstrapping the BCPL  
compiler. The INTCODE machine has six control registers and eight  
functions. "INTCODE - An Interpretive Machine Code for BCPL", M. Richards,  
Computer Lab, U Cambridge 1972. "BCPL - The Language and its Compiler",  
Martin Richards & Colin Whitby-Stevens, Cambridge U Press 1979. (See  
OCODE).

INTELLECT - Larry Harris, 1977. A query language, close to natural  
English.

INTERACTIVE - Network simulation language. "Design and Implementation of a  
Pascal Based Interactive Network Simulation Language", R. Lakshmanan, PhD  
Thesis, Oakland U, Rochester MI 1983.

INTERCAL - (Allegedly stands for "Compiler Language With No Pronounceable  
Acronym"). Woods & Lyon, Princeton U, May 26, 1972. The most elaborate  
and extended joke in the history of language design; claims to have nothing  
in common with any other major programming language. "The INTERCAL  
Programming Language Reference Manual", Donald R. Woods & James M. Lyon.  
Implementations: INTERCAL-72 in SPITBOL for IBM/360; Atari version of  
unknown date; and C-INTERCAL, a retargetable compiler for Unix by Eric S.  
Raymond <esr@snark.thyrsus.com> in 1991, which uses C as an intermediate  
language. The latest C-INTERCAL version adds features like the COME FROM  
statement and multilingual support including Basque, Nahuatl, Sanskrit and  
Tagalog.  
<ftp://locke.ccil.org/pub/retro/intercal-0.10.tar.gz>

INTERCOM - Assembly language for the G-15. Listed in CACM 2(5):16 (May  
1959). Versions: INTERCOM 101, INTERCOM 1000.

Interlisp - Descendant of BBN-Lisp. Once Interlisp was one of two main  
branches of LISP (the other being MACLISP). In 1981 Common LISP was begun  
in an effort to combine the best features of both. Interlisp includes a  
LISP programming environment. Dynamically scoped. NLAMBDA functions do  
not evaluate their arguments. Any function could be called with optional  
arguments. "Interlisp Programming Manual", W. Teitelman, TR, Xerox Rec Ctr  
1975. Interlisp-10 used shallow binding, while Xerox's Interlisp-D used  
deep binding.

Intermediate Programming Language - Arthur W. Burks. A very early attempt  
to express machine language at a higher level of abstraction. Like  
Plankalkul, it used a right-handed style of assignment, in which the new  
value appears on the right.

Interpress - Xerox. Interpretive FORTH-like graphics language, possibly  
the first page description language, predating PostScript. Both are  
descendants of JaM. Used on Xerox printers. "Interpress, The Source  
Book", Steven Harrington et al, P-H 1988.

Iota - Specification language. "The Iota Programming System", R. Nakajima et al, Springer 1983.

IPL - Information Processing Language. Allen Newell, J.C. Shaw, H. Simon, Carnegie ca. 1956. Said to be the first list-processing language, also the first language to support recursion. Very low level. Sammet 1969, pp.388-400. "Information Processing Language-V Manual", A. Newell ed, P-H 1965. Versions: IPL-I (never implemented), IPL-II (1957 for JOHNNIAC), IPL-III (existed briefly), IPL-IV, IPL-V (1958, for IBM 650, 704, 7090, many others. Widely used), IPL-VI.

IPS - Threaded language. "IPS, An Unorthodox High Level Language", K. Meinzer, BYTE pp.146-159 (Jan 1979).

IQ - Pictorial query language, implemented in Ratfor. "Structured Implementation of an Image Query Language", Y.E. Lien et al, in Database Techniques for Pictorial Applications, A. Blaser ed, pp.416-430.

IRDATA - Industrial Robot DATA. A standardized robot control code. "IRDATA, Industrial Robot Data", DIN 66313, Beuth-Verlag 1991.

IRL - Industrial Robot Language. A high-level language for programming industrial robots. "IRL, Industrial Robot Language", DIN 66312, Beuth-Verlag 1992.

Ironman - HOLWG, DoD, Jan 1977, revised Jul 1977. Fourth of the series of DoD requirements that led to Ada. "Department of Defense Requirements for High Order Computer Programming Languages", SIGPLAN Notices 12(12):39-54 (Dec 1977). "Revised Ironman Requirements for High Order Computer Programming Languages", US Dept of Defense, Jul 1977. (See Strawman, Woodenman, Tinman, Steelman).

Isabelle-92 - A generic theorem prover, supporting a wide variety of logics. A system of type classes allows polymorphic object-logics with overloading and automatic type inference.  
<ftp://ftp.cl.cam.ac.uk/ml/92.tar.Z>

ISBL - Mathematical query language.

ISETL - Interactive SETL. Gary Levin <gary@clutx.clarkson.edu>, Clarkson U. "An Introduction to ISETL Version 1.9", G.M. Levin, Dept MCS, Clarkson U. Current version: 3.0.  
<ftp://sun.soe.clarkson.edu>, for MS-DOS, Mac, Unix, VAX/VMS, and source.

ISIS -

1. Dialect of JOSS. Sammet 1969, p.217.
2. Concurrent language?

ISL - Interface Specification Language. Xerox PARC. Interface description language used by the ILU (Inter-Language Unification) system. Includes descriptions of multiple inheritance, exceptions and garbage collection.  
info: Bill Janssen <janssen@parc.xerox.com>  
<ftp://parcftp.parc.xerox.com/pub/ilu/>

ISLisp - International Standard LISP, ISO WG 16, draft Dec 1992. An object-oriented Lisp intended as an international replacement for Common Lisp, EuLisp, Le-Lisp and scheme. Goals are object orientation, extensibility, efficiency, and suitability for non-academic use. <ftp://ma2s2.mathematik.uni-karlsruhe.de/pub/lisp/islisp/>\*

ISP - Instruction Set Processor. A family of languages for describing the instruction sets of computers. "Computer Structures: Readings and Examples", D.P. Siewiorek et al, McGraw-Hill 1982.

ISPL - Instruction Set Processor Language. ca 1971. Original ISP language, written in BLISS. "Computer Structures: Readings and Examples", D.P. Siewiorek et al, McGraw-Hill 1982.

ISPS - Barbacci, Carnegie-Mellon 1979. Instruction Set Processor Specifications. Operational hardware specification language. Successor to ISPL. "Instruction Set Processor Specifications", M.R. Barbacci et al, IEEE Trans Computers, C-30(1):24-80 (Jan 1981). [Bell, Newell, Siewiorek, Barbacci 1982?]

ISWIM - If You See What I Mean. Landin 1966. ISWIM is purely functional, a sugaring of lambda calculus, and the ancestor of most modern applicative languages. An ISWIM program is a single expression qualified by 'where' clauses (auxiliary definitions including equations among variables), conditional expressions and function definitions. ISWIM was the first language to use lazy evaluation, and introduced the offside rule for indentation. "The Next 700 Programming Languages", P.J. Landin, CACM 9(3):157-166 (Mar 1966).

IT - Internal Translator. A.J. Perlis et al, Carnegie Tech ca 1957. Early compiler for math originally for Burroughs 205, then IBM 650. Forerunner of RUNCIBLE, GATE, CORRELATE and GAT. IT source code was converted to PIT, thence to SPIT. Sammet 1969, pp.139-141. Versions: IT-2 produced machine language directly, IT-3 developed at Carnegie added double-precision floating point. CACM 1(5):22 (1958).

Ivan - A Diana-like language making up part of VHDL. "VHDL - The Designer Environment", A. Gilman, IEEE Design & Test 3, (Apr 1986).

Iverson's Language - APL, which went unnamed for many years. Sammet 1969, p.770.

IVTRAN - 1966. Parallel FORTRAN for the Illiac IV.

J - Derivative and redesign of APL. Purely functional with lexical scope and more conventional control structures, plus several new concepts such as function rank and function arrays. "APL\?", Roger K.W. Hui et al, APL90 Conf Proc, Quote Quad 20(4):192-200. Version 4.1 for MS-DOS, Sun, Mac, Archimedes. Source available in C from Iverson Software, (416)925-6096. <ftp://watserv1.waterloo.edu/languages/apl/j>

J3 - Dialect of JOVIAL. "Military Standard JOVIAL (J3)", MIL-STD-1588 (USAF), June 1976.

J73 - Yep, another JOVIAL dialect. "Military Standard JOVIAL (J73)", MIL-STD-1589 (USAF), Feb 1977.

JACAL - Jaffer's CAnonical ALgebra. A. Jaffer<jaffer@zurich.ai.mit.edu>. Symbolic math program, written in Scheme.  
ftp://altdorf.ai.mit.edu/archive/scm/jacalla0.tar.Z

Jade -

1. U Washington, late 80's. A strongly-typed language, object-oriented but without classes. For type research. The compiler output is Smalltalk. [Submitter claimed that Jade has exactly one user!]

2. Implicit coarse-grained concurrency. The constructs 'with', 'withonly' and 'without' create tasks with specified side effects to shared data objects. Implemented as a C preprocessor. "Coarse-Grain Parallel Programming in Jade", M.S. Lam et al, SIGPLAN Notices 26(7):94-105 (Jul 1991).

JaM - John and Martin. J. Warnock & M. Newell, PARC 1978. Interpretive FORTH-like graphics language, forerunner of both Interpress and PostScript. Mentioned in PostScript Language reference Manual, Adobe Systems, A-W 1985.

Janus -

1. Distributed language with an ask/tell constraint system. "Janus: A Step Towards Distributed Constraint Programming", V. Saraswat <saraswat@parc.xerox.com> et al in Logic Programming: Proc 1990 North Am Conf, S. Debray et al eds, MIT Press 1990.  
ftp://cs.arizona.edu/janus/qdjanus-1.2, a sequential implementation built on SICStus Prolog.

2. W. M. Waite, U Colorado. Intermediate language, claimed to be an implementation of UNCOL. Used on CDC 6600. "Experience with the Universal Intermediate Language Janus", B.K. Haddon et al, Soft Prac & Exp 8(5):601-616 (Sep 1978).

JAZ - Early system on LGP-30. Listed in CACM 2(5):16 (May 1959).

JCL - Job Control Language. Batch language on IBM OS/360 systems. Notoriously difficult to program in.  
ftp://locke.ccil.org/pub/retro/jcl.shar.gz Emulator

JCS-13 - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

JEAN - A dialect of JOSS.

JOSS - JOHNNIAC Open Shop System. Charles L. Baker, RAND 1964. An early simple interactive calculator language. "JOSS Users' Reference Manual", R.L. Clark, Report F-1535/9, RAND Corp (Jan 1975) Sammet 1969, pp.217-226. Versions: JOSS I and JOSS II.

Jossle - [?] Type checked language with separate compilation using a program library. Mentioned in "Rationale for the Design of Ada", J. Ichbiah, Cambridge U Press, 1986, p.192.

JOVIAL - Jule's Own Version of IAL. Jules I. Schwartz 1959-1960. Based on ALGOL 58, with extensions for large scale real-time programming. The data elements are items, entries (records) and tables. Extensive use by the US Air Force. Most of the software for AWACS is in JOVIAL, running on IBM's

AOCP (360 compatible). CACM 6(12):721 (Dec 1960)[?]. Versions include JOVIAL I (IBM 709, 1960), JOVIAL II (IBM 7090, 1961) and JOVIAL 3 (1965). Dialects: J3, JOVIAL J73, JS, JTS.  
info: Ada/Jovial Newsletter, Dale Lange (513)255-4472

Joyce - Brinch Hansen. Distributed language based on Pascal and CSP. "Joyce - A Programming Language for Distributed Systems", Per Brinch Hansen, Soft Prac & Exp 17(1):29-50 (Jan 1987).

JPL - JAM Programming Language. Imperative string-based language, part of the JAM tool for developing screen (non-window) applications. JYACC Corp.

JPLDIS - Jet Propulsion Laboratory Display Information System. Jack Hatfield, George Masters, W. Van Snyder, Jeb Long et al, JPL. Query system for UNIVAC 1108 [or PDP's?] written in FORTRAN, based on Tymshare's "Retrieve". Indirectly led to Vulcan[1] which led to dBASE II.

JS - Dialect of JOVIAL. Sammet 1969, p.639.

JTS - Simple dialect of JOVIAL. Sammet 1969, p.528.

Juno - Numerical constraint-oriented language for graphics applications. Solves its constraints using Newton-Raphson relaxation. Inspired partly by Metafont. "Juno, a Constraint-Based Graphics System", G. Nelson in SIGGRAPH '85 Conf Readings, B.A. Barsky ed, Jul 1985, pp.235-243.

Jym - Patrick Bellot, France. A predecessor to Graal.

K5 - Early system on Larc computer. Listed in CACM 2(5):16 (May 1959).

Kaleidoscope - Freeman-Benson <bnfb@cs.uvic.ca>, U Washington and Universite de Nantes, 1989; U Victoria, 1992. Object-oriented language which mixes imperative and constraint-oriented features. Similar to Siri. Vaguely related to Prose[2]. "Kaleidoscope: Mixing Objects, Constraints and Imperative Programming", B.N. Freeman-Benson, SIGPLAN Notices 25(10):77-88 (OOPSLA/ECOOP '90) (Oct 1990). "Constraint Imperative Programming", B.N. Freeman-Benson, Ph.D. Thesis, TR 91-07-02, U Wash (1991). "Constraint Imperative Programming", Freeman-Benson et al, IEEE Conf on Comp Lang, Apr 1992. Versions: Kaleidoscope'90, Kaleidoscope'91, Kaleidoscope'93.

Kali - Data parallel language. "Supporting Shared Data Structures on Distributed Memory Architectures", C. Koelbel et al in Second ACM SIGPLAN Symp on Princ and Prac of Parallel Programming, pp.177-186, Mar 1990.

KAP - Kernel Andorra Prolog. "Kernel Andorra Prolog and its Computation Model", S. Haridi <seif@sics.se> et al, in Logic Programming: Proc 7th Intl Conf, MIT Press 1990. Predecessor to AKL.

Karel - Language featured in Karel the Robot: A Gentle Introduction to Computer Programming, Richard E. Pattis, Wiley 1981.  
ftp://wuarchive.wustl.edu/mirrors/unix-c/languages/pascal/karel.tar-z

KBMS - Expert system.

KCL - Kyoto Common LISP. Taiichi Yuasa and Masami Hagiya, 1984. Compiles to ANSI C. "Design and Implementation of Kyoto Common Lisp", T. Yuasa

<yuasa@tutics.tut.ac.jp>, J Info Proc 13(3):284-295 (1990). "Kyoto Common Lisp Report", T. Yuasa & M. Hagiya. (See AKCL and GCL).  
ftp://rascal.ics.utexas.edu/pub/kcl.tar.Z  
list: kcl@rascal.ics.utexas.edu

K-code. Language recognized by the K-machine, a virtual machine with an incremental constraint solver and a constraint-based data store. Used to implement Kaleidoscope. "Implementing Constraint Imperative Programming Languages: The Kaleidoscope'93 Virtual Machine", G. Lopez et al, SIGPLAN Notices 29(10):259-271 (Oct 1994).

KEE - Knowledge Engineering Environment. Frame-based expert system. Supports dynamic inheritance, multiple inheritance, polymorphism. Classes, meta-classes and objects are all treated alike. A class is an instance of a meta-class. Can control rules for merging of each field when multiple inheritance takes place. Methods are written in LISP. Actions may be triggered when fields are accessed or modified. Extensive GUI integrates with objects. Can easily make object updates to be reflected on display or display selections to update fields. This can in turn trigger other methods or inference rules which may then update other parts of the display. Intellicorp, for TI Explorer. "The Role of Frame-Based Representation in Reasoning", R. Fikes et al, CACM 28(9):904-920 (Sept 1985).

Kernel Parlog - Modeless intermediate language for Parlog compilation. "Notes on the Implementation of Parlog", K.L. Clark et al, J Logic Prog 2(1):17-42 (1985).

Kevo - A. Taivalsaari <antero@cs.uta.fi>. Prototype-based object-oriented system built around a threaded code interpreter. Semantically resembles Self and Omega. Syntactically resembles Forth. TR DCS-197-19, U Victoria, June 1992.  
info:kevo-interest@ursamajor.uvic.ca  
ftp://cs.uta.fi/pub/kevo

KFX - Kernel language of FX-87. "Polymorphic Effect Systems", J.M. Lucassen et al, Proc 15th Ann ACM Conf POPL, ACM 1988, pp.47-57.

Kid - Kernel language for Id. A refinement of P-TAC, used as an intermediate language for Id. Lambda-calculus with first-class let-blocks, plus I-structures. "A Syntactic Approach to Program Transformations", Z. Ariola et al, SIGPLAN Notices 26(9):116-129 (Sept 1991).

KISS - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

KL0 - Sequential logic language based on Prolog, for the ICOT project, Japan.

KL1 - Kernel Language 1. An experimental AND-parallel version of KL0 for the ICOT project, Japan. An implementation of FGHC. "Design of the Kernel Language for the Parallel Inference Machine", U. Kazunori et al, Computer J (Dec 1990).

Klerer-May System - Columbia U. Early system with special math symbols. Its reference manual was two pages long! "Further Advances in Two-Dimensional Input-Output by Typewriter Terminals", M. Klerer et al, Proc FJCC 31 (1967). Sammet 1969, pp.284-294.

KL-ONE - Frame language. "An Overview of the KL-ONE Knowledge Representation System", R.J. Brachman and J. Schmolze, Cognitive Sci 9(2), 1985.

KLS - Knotted List Structures. List-processing language, a predecessor of SLIP. "Knotted List Structures", J. Weizenbaum, CACM 5:161 (Mar 1962).

KMODEL - An ancestor of Model-K. "Preliminary Results on the BEHAVIOR Specifications Language KMODEL-0", BEHAVIOR Memo 5-91, 1991, GMD, Sankt Augustin, Germany

KOMPILER - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959). Versions: KOMPILER 2 for IBM 701, KOMPILER 3 for IBM 704.

KRC - Kent Recursive Calculator. Turner 1981. Lazy functional language based on SASL, with pattern matching, ZF expressions. "Functional Programming and its Applications", David A. Turner, Cambridge U Press 1982.

KRL - Knowledge Representation Language. A frame-based language. "An Overview of KRL, a Knowledge Representation Language", D.G. Bobrow and T. Winograd, Cognitive Sci 1:1 (1977).

KRS - Frame-based language built on Common LISP.

KRYPTON - Frame language. "An Essential Hybrid Reasoning System: Knowledge and Symbol Level Accounts of of KRYPTON", R.J. Brachman et al, Proc IJCAI-85, 1985.

ksh - Korn Shell command interpreter for Unix.

L0 - Tech U Munich. Low level language, typed and with usual flow control, but only 3-address expressions. Higher levels L1 and L2 were planned. "Brief Survey of L0", H. Scheidig, in Machine Oriented Higher Level Languages, W. van der Poel ed, N-H 1974, pp.239-247.

L6 - Bell Telephone Laboratories Low-Level Linked List Language. Ken Knowlton, 1965. List processing language, typeless. "A Programmer's Description of L6, Bell Telephone Laboratories' Low-Level Linked List Language", K. Knowlton CACM 9(8):616-625 (Aug 1966). Sammet 1969, pp.400-405.

Lace - Language for Assembling Classes in Eiffel. Specifies how to assemble an Eiffel system : in which directories to find the clusters, which class to use as the root, permits class renaming to avoid name clashes. "Eiffel: The Language", Bertrand Meyer, P-H 1992.

LADE - Compiler-compiler language?

LADY - "Key Concepts in the INCAS Multicomputer Project", J. Nehmer et al IEEE Trans Soft Eng SE-13(8):913-923 (Aug 1987).

Lakota - Scripting language, extends existing OS commands.  
info: Richard Harter <rh@smds.UUCP> SMDS Inc.

LAMBDA - A version of typed lambda calculus, used to describe semantic domains. "Outline of a Mathematical Theory of Computation", D.S. Scott, TM

PRG-2, PRG, Oxford U, 1971.

lambda-Prolog - An extension of standard Prolog, in which terms are typed lambda-terms. Prolog/Mali compiler uses the MALI abstract memory system.  
ftp://ftp.irisa.fr/pm/pm.tar.Z  
info: pm@irisa.fr  
list: prolog-mali@irisa.fr

LAMINA - Concurrent object-oriented language. "Experiments with a Knowledge-based System on a Multiprocessor", Third Intl Conf Supercomputing Proc, 1988. "ELINT in LAMINA, Application of a Concurrent Object language", Delagi et al, KSL-88-3, Knowledge Sys Labs, Stanford U.

Language H - NCR. Early business-oriented language.

Laning and Zierler - J.H. Laning Jr and N. Zierler, 1953-1954. Possibly the first true working algebraic compiler. On MIT's Whirlwind computer. Sammet 1969, pp.131-132.

LAP - LISP Assembly Program. Assembly language embedded into early LISP. Sammet 1969, p.597. Also used by the Liar compiler for MIT Scheme, Emacs LISP, and MACLISP.

LAP4 - Early assembly language for Linc-8 machine.

LAPSE - Single assignment language for the Manchester dataflow machine. "A Single Assignment Language for Data Flow Computing", J.R.W. Glauert, M.Sc Diss, Victoria U Manchester, 1978.

Larch - John Guttag and Jim Horning <horning@src.dec.com>. The Larch Project develops aids for formal specifications. Each Larch specification has two components: an interface containing predicates written in the LIL (Larch Interface Language) designed for the target language and a 'trait' containing assertions about the predicates written in LSL, the Larch Shared Language common to all. "The Larch Family of Specification Languages", J. Guttag et al, IEEE Trans Soft Eng 2(5):24-365 (Sep 1985). "Larch: Languages and Tools for Formal Specification", Guttag and Horning, Springer 1993.

Larch/Ada - Used in the Penelope verification system, to provide semantics for Ada' types. Notationally similar to Anna.

Larch/CLU - Larch specification language for CLU. Used in Abstraction and Specification in Program Development, B. Liskov & J. Guttag, MIT Press 1986.

LaTeX - see TeX.

LAU - Langage a Assigination Unique. Single assignment language for the LAU dataflow machine, Toulouse. "Pipelining, Parallelism and Asynchronism in the LAU System", J.C. Syre et al, Proc 1977 Intl Conf Parallel Proc, pp.87-92.

LAURE - A language for knowledge representation combining object orientation and logic programming. Set operations. Object-oriented exception handling and a polymorphic type system. "An Object-Oriented Language for Advanced Applications", in Proc TOOLS 5, Santa Barbara 1991,

P-H.

info: Yves Caseau <caseau@france.bellcore.com>

LAVA - A language for VLSI that deals with "sticks", i.e. wires represented as lines with thickness. R.J. Matthews et al, "A Target Language for Silicon Compilers", IEEE COMPCON, 1982, pp.349-353.

LAX - LAnguage eXample. Toy language used to illustrate problems in compiler design. "Compiler Construction", W. M. Waite et al, Springer 1984.

LCC - Language for Conversational Computing. CMU 1960's. Similar to JOSS, with declarations, pointers and block structure from ALGOL-60. Implemented for IBM 360/370 under TSS. "LCC Reference Manual", H.R. Van Zoeren, CMU 1969.

LCL -

1. The Larch interface language for ANSI standard C. J.V. Guttag et al, TR 74, DEC SRC, Palo Alto CA, 1991.

2. Liga Control Language. Controls the attribute evaluator generator LIGA, part of the Eli compiler-compiler. "LCL: Liga Control Language", U. Kastens <uwe@uni-paderborn.de> , U Paderborn.

LCS - Language for Communicating Systems. Bernard Barthomieu. A concurrent SML with behaviors and processes, based upon higher order CCS. Implemented as a bytecode interpreter. Version 3.1  
ftp://laas.laas.fr/pub/lcs for Sun 3, Sun 4  
info: Bernard Berthomieu <bernard@laas.laas.fr>  
list: lcs@laas.laas.fr

LDL - "LDL: A Logic-Based Data-Language", S. Tsur et al, Proc VLDB 1986, Kyoto Japan, Aug 1986, pp.33-41.

LDL1 - Successor of LDL. "Sets and Negation in a Logic Database Language", C. Beeri et al, in Proc 6th Ann ACM Symp Princs Database Sys (1987), pp.21-37.

LDT - Logic Design Translator. Computer system design analysis. Sammet 1969, p.621.

LE/1 - Langage External. "An Evaluation of the LE/1 Network Command Language Designed for the SOC Network", J. du Masle, in Command Languages, C. Unger ed, N-H 1973.

LEAF -

1. LISP Extended Algebraic Facility. "An Algebraic Extension to LISP", P.H. Knowlton, Proc FJCC 35 (1969).

2. "LEAF: A Language which Integrates Logic, Equations and Functions", R. Barbuti et al in Logic Programming, Functions Relations and Equations, D. DeGroot et al eds, P-H 1986, pp.201-238.

Lean - U Nijmegen and U East Anglia. An experimental language based on graph rewriting, useful as an intermediate language. Descendant of Dact10.

"Towards an Intermediate Language Based on Graph Rewriting", H.P. Barendregt et al in PARLE: Parallel Architectures and Languages Europe, G. Goos ed, LNCS 259, Springer 1987, pp.159-175. (See Clean).

LEAP - Language for the Expression of Associative Procedures. ALGOL-based formalism for sets and associative retrieval, for TX-2. Became part of SAIL. "An ALGOL-based Associative Language", J.A. Feldman et al, CACM 12(8):439-449 (Aug 1969).

LECOM - Version of COMIT on GE 225 ca. 1966. Sammet 1969, p.419.

Leda - Tim Budd <budd@cs.orst.edu>, Oregon State U, 1990-1993. Multiparadigm language (imperative, declarative, procedural, applicative, functional, logic, and object-oriented!) "Blending Imperative and Relational Programming", Tim Budd, IEEE Software 8(1):58-65 (Jan 1991). Forthcoming book.  
ftp://cs.orst.edu/pub/budd/leda/\*

LeFun - MCC, Austin. Integration of logic and functional programming. "LeFun: Logic, Equations and Functions", H. Ait-Kaci et al, Proc 1987 Symp on Logic Programming, San Francisco.

Legion - Distributed language.  
<http://uvacs.cs.virginia.edu/~mentat/legion/legion.html>

LEGOL - "Application of MP/3 to the Design and Implementation of LEGOL, A Legally Oriented Language", S.H. Mandil et al, Intl Symp Programming, Paris 1974.

Le-Lisp - Jerome Chailloux and Emmanuel St James, INRIA, France. A LISP dialect close to Common Lisp, lexically scoped, with a CLOS-like object system. Uses both packages and modules. "le-lisp: A Portable and Efficient Lisp System", J. Chailloux et al, Proc 1984 ACM Symp on Lisp and Functional Programming, ACM. Version v.16, available from ILOG, France.

Leo - General-purpose systems language, syntactically like Pascal and Y, semantically like C. "The Leo Programming Language", G. Townsend, CS TR 84-7, U Arizona 1984.

Lex -

1. Input language to the Lex scanner generator. "Lex - A Lexical Analyzer Generator", M.E. Lesk, CS TR 39, Bell Labs (Oct 1975). (See Flex).

Implementation:

ML-lex - Implementation and output in SML/NJ.  
ftp://research.att.com/dist/ml/75.tools.tar.Z

2. Lexical specification language for COPS. "Metalanguages of the Compiler Production System COPS", J. Borowiec, in GI Fachgesprach "Compiler-Compiler", ed W. Henhapl, Tech Hochs Darmstadt 1978, pp.122-159.

LG - Simple language for analytic geometry, with graphic output. "LG: A Language for Analytic Geometry", J. Reymond, CACM 12(8) (Aug 1969). [???

LGDF - Large-Grain DataFlow. "A Large-grain Data Flow Scheduler for Parallel Processing on Cyberplus", R.G. Babb et al, Proc 1986 Intl Conf on

Paralllel Proc, Aug 1986.

LGEN - Bell Labs. A logic language for VLSI implementation. S.C. Johnson, "Code Generation for Silicon", Proc 10th POPL, 1983.

LGN - Linear Graph Notation. A linearized representation of TCOL trees. B.W. Leverett et al, "An Overview of the Production Quality Compiler-Compiler Projects", TR CMU-CS-79-105, Carnegie Mellon 1979. (See TCOL)

Liana - 1991. Similar to C++, aimed at Windows applications. No pointers, no multiple inheritance. Garbage collection. "The Liana Programming Language", R. Valdes, Dr Dobbs J Oct 1993, pp.50-52. Base Tech, 1320 Peral St, Boulder CO.

LIDO - Input language for the attribute evaluator generator LIGA (a successor of GAG and a subsystem of the Eli compiler-compiler). LIDO is derived from GAG's input language ALADIN. "LIDO: A Specification Language for Attribute Grammars", U. Kastens <uwe@uni-paderborn.de>, Fab Math-Inf, U Paderborn (Oct 1989).

LiE - Symbolic math aimed at Lie groups. "LiE, a Package for Lie Group Computations", M.A.A. van Leeuwen et al, in Computer Algebra Nederland, 1992 (ISBN 90-741160-02-7).

LIFE - Logic of Inheritance, Functions and Equations. Hassan Ait-Kacy <hak@prl.dec.com> et al, MCC, Austin, 1987. Object-oriented, functional, and constraint-based. Integration of ideas from LOGIN and LeFun. "Is There a Meaning to LIFE?", H. Ait-Kacy et al, Intl Conf on Logic Prog, 1991.

ftp://gatekeeper.dec.com/pub/plan/Life.tar.Z - Wild\_LIFE interpreter from Paradise project at DEC's Paris Research Lab  
list: life-users@prl.dec.com

Lila - Patrick Salle' <salle@geocub.greco-prog.fr>. A small assembly-like language used for implementation of Actor languages. [Plasma perhaps?]

LIMDEP - Linear programming language used by economists.

LIMP - "Messages in Typed Languages", J. Hunt et al, SIGPLAN Notices 14(1):27-45 (Jan 1979).

Linc - Burroughs/Unisys 4GL. Designed in New Zealand.

Lincoln Reckoner - ca 1965. Interactive math including matrix operations, on TX-2. "The Lincoln Reckonere: An Operation-Oriented On-line Facility with Distributed Control", A.N. Stowe et al, Proc FJCC 29 (1966). Sammet 1969, pp.245-247.

Linda - Yale. A "coordination language", providing a model for concurrency with communication via a shared tuple space. Usually implemented as a subroutine library for a specific base language. "Generative Communication in Linda", D. Gelernter <gelernter@cs.yale.edu> ACM TOPLAS 7(1):80-112 (1985). "Linda in Context", N. Carreiro <carreiro@cs.yale.edu> et al, CACM 32(4):444-458 (Apr 1989). (See C-Linda, Ease, Fortran-Linda, Glenda, LindaLISP, Lucinda, Melinda, Prolog-Linda).  
<http://www.cs.yale.edu/HTML/YALE/CS/Linda/linda/linda.html>

LindaLISP - Yep, you guessed it.

Lingo - An animation scripting language. MacroMind Director V3.0 Interactivity Manual, MacroMind 1991.

LINGOL - LINGuistics Oriented Language. Natural language processing. "A Linguistics Oriented Programming Language", V.R. Pratt, Third Intl Joint Conf on AI, 1973.

LIPL - Linear IPL. A linearized (i.e. horizontal format) version of IPL-V. Sammet 1969, p.394. R. Dupchak, "LIPL - Linear Information Processing Language", Rand Memo RM-4320-PR, Feb 1965.

LIS - Langage Implementation Systeme. Ichbiah, 1973. A predecessor of Ada, influenced by Pascal's data structures and Sue's control structures. A type declaration may have a low-level implementation specification. "The System Implementation Language LIS", J.D. Ichbiah et al, CII Honeywell-Bull, TR 4549 E/EN, Louveciennes France (Dec 1974). "The Two-Level Approach to Data Independent Programming in LIS", J.D. Ichbiah et al, in Machine Oriented Higher Level Languages, W. van der Poel ed, N-H 1974, pp.161-169.

LISA - Statistical data analysis. Similar to S.  
ftp://dolphin.mit.edu.

LISP - LISt Processing. John McCarthy <jmc@sail.stanford.edu> et al, MIT late 50's. Symbolic functional recursive language based on lambda-calculus, used especially for AI and symbolic math. Many dialects. Atoms and lists. Dynamic scope. Both programs and data are represented as list structures. Versions include LISP 1 (Original version), LISP 1.5 (MIT 1959), LISP 1.75, LISP 1.9.

LISP 2 - LISP 1.5 with an ALGOL60-like surface syntax. Also optional type declarations, new data types including integer-indexed arrays and character strings, partial-word extraction/insertion operators and macros. A pattern-matching facility similar to COMIT was proposed. Implemented for the Q-32 computer. "The LISP 2 Programming Language and System", P.W. Abrahams et al, Proc FJCC 29:661-676, AFIPS (Fall 1966).

LISP70 - LISP dialect, a descendant of MLISP and MLISP2. Also known as PLISP and VEL. Useful for parsing. Only the pattern-matching system was published and fully implemented. According to Alan Kay, LISP70 had an influence on Smalltalk-72. "The LISP70 Pattern Matching System, Larry Tesler et al, IJCAI 73.

LISP A - "LISP A: A LISP-like System for Incremental Computing", E.J. Sandewall, Proc SJCC 32 (1968).

Lispkit Lisp - Purely functional version of LISP. "Functional Programming, Application and Implementation", P. Henderson, P-H 1980.

Lisp-Linda - P. Dourish, U Edinburgh 1988.

LISP Machine LISP - An extension of Maclisp, now called Zetalisp.

Lisptalk - "Concurrent Programming Language Lisptalk", C. Li, SIGPLAN Notices 23(4):71-80 (Apr 1988).

LITHE - Object-oriented with extensible syntax. "LITHE: A Language Combining a Flexible Syntax and Classes", D. Sandberg, Conf Rec 9th Ann ACM Sym POPL, ACM 1982, pp.142-145.

LITTLE - Typeless language used to produce machine-independent software. LITTLE has been used to implement SETL. "Guide to the LITTLE Language", D. Shields, LITTLE Newsletter 33, Courant Inst (Aug 1977).

Little Smalltalk - A line-oriented near-subset of Smalltalk-80. "A Little Smalltalk", Timothy Budd, A-W 1987.  
ftp://ftp.cs.orst.edu/~budd/smallv3.tar, source in C.

LLM3 - J. Chailloux. Assembly language for a virtual machine, the implementation language for Le-Lisp.

LM3 - The Larch interface language for Modula-3. (See Larch). "LM3: A Larch/Modula-3 Interface Language", Kevin D. Jones, TR 72, DEC SRC, Palo Alto CA.

LML -

1. Chalmers U Tech, Gothenburg, Sweden. Lazy, completely functional variant of ML[2]. Implemented on the G-machine, and used to implement the Haskell B compiler.  
ftp://ftp.cs.chalmers.se/pub/haskell/chalmers/lml-0.999.1.\*

2. Logical ML. Adds to Lazy ML a data type of 'theories' whose objects represent logic programs. "Logic Programming within a Functional Framework", A. Brogi et al, in Programming Language Implementation and Logic Programming, P. Deransart et al eds, LNCS 456, Springer 1990.

LNF - "A Fully Lazy Higher Order Purely Functional Programming Language With Reduction Semantics", K.L. Greene, CASE Center TR 8503, Syracuse U 1985.

L&O - Logic and Objects. Implemented as a front end for IC Prolog. "Logic and Objects", Frank McCabe, P-H.  
info: Zacharias Bobolakis <zb@doc.ic.ac.uk>  
ftp://doc.ic.ac.uk/computing/programming/languages/pd-ICP-0.90.tar.Z

LO - Linear Objects. Concurrent logic programming language based on "linear logic", an extension of Horn logic with a new kind of OR-concurrency. "LO and Behold! Concurrent Structured Processes", J. Andreoli et al, SIGPLAN Notices 25(10):44-56 (OOPSLA/ECOOP '90) (Oct 1990).

{log} - "{log}: A Logic Programming Language with Finite Sets", A Dovier et al, Proc 8th Intl Conf Logic Prog, June 1991, pp.111-124.

LogC - C extension incorporating rule-oriented programming, for AI applications. Production rules are encapsulated into functional components called rulesets. Uses a search network algorithm similar to RETE. "LogC: A Language and Environment for Embedded Rule Based Systems", F. Yulin et al, SIGPLAN Notices 27(11):27-32 (Nov 1992). Version: LogC 1.6.

Logic Design Language - Language for computer design. "A System Description Language Using Parametric Text Generation", R.H. Williams, TR

02.487, IBM San Jose, Aug 1970.

LOGIN - Integration of logic programming and inheritance. "LOGIN: A Logic Programming Language with Built-In Inheritance", H. Ait-Kaci et al, J Logic Programming 3(3):185-215 (1986).

LOGLAN - Inst Informatics, Warsaw U. Object-oriented. Not to be confused with Charles Brown's 'Loglan', a high-precision artificial language designed for humans, intended to test the Sapir-Whorf Hypothesis and investigate the mechanisms of natural learning. "LOGLAN '88 - Report on the Programming Language", LNCS 414, Springer, ISBN 3-540-52325-1.

LOGLISP - Robertson & Sibert, Syracuse 1980. A Prolog-like language called LOGIC, embedded in LISP. "LOGLISP: An Alternative to Prolog", J. Alan Robinson et al in Machine Intelligence 10, D. Michie ed, Ellis Horwood 1982.

LOGO - Developed 1966-1968 by a group at Bolt, Beranek & Newman headed by Wally Fuerzeig <fuerzeig@bbn.com> and including Seymour Papert <seymour@media.mit.edu>. A LISP-like language aimed at children and other beginning programmers, noted for its "turtle graphics" used to draw geometric shapes. Logo interpreters for Mac, Unix, PC, X are available: <ftp://anarres.cs.berkeley.edu/pub/ucblog>

LOGOL - Strings are stored on cyclic lists or 'tapes', which are operated upon by finite automata. J. Mysior et al, "LOGOL, A String manipulation Language", in Symbol Manipulations Languages and Techniques, D.G. Bobrow ed, N-H 1968, pp.166-177.

LOLITA - Language for the On-Line Investigation and Transformation of Abstractions. Extension of Culler-Fried system for symbolic math. "An On-line Symbol Manipulation System", F.W. Blackwell, Proc ACM 22nd Natl Conf (1967). Sammet 1969, p.464.

Lolli - (named for the "lollipop" operator "-o") Based on linear logic, in which clauses can be used exactly once. All the operators of lambda-Prolog plus linear variations. Implemented in SML/NJ. Josh Hodas et al, "Logic Programming in a Fragment of Intuitionistic Linear Logic", Information and Computation, to appear. <ftp://ftp.cis.upenn.edu/pub/Lolli/Lolli-0.7.tar.Z>

LOM - Toulouse, early 1980's. Language for data processing.

LOOK - Specification language. "A Look at Algebraic Specifications", S.N. Zilles et al, IBM RR, 1982.

LOOKS -

1. "LOOKS: Knowledge-Representation System for Designing Expert Systems in a Logical Programming Framework", F. Mizoguchi, Proc Intl Conf 5th Gen Comp Sys, ICOT 1984.

2. Language for Object-Oriented Kinematics Specifications. E. Peeters <wsinerp@win.tue.nl>, 1993. Imperative, object-oriented language for specifying motion. "Design and Implementation of an Object-Oriented Interactive Animation System", Eric Peeters, in Technology of Object-Oriented Languages and Systems, TOOLS 12 & 9, C. Mingins et al eds, P-H

1993, pp.255-267.

LOOPN - U Tasmania. An object-oriented language for simulation of Petri nets.

ftp://ftp.utas.edu.au/departments/computer\_science/loopn.tar.Z

LOOPS - Lisp Object-Oriented Programming System. Xerox's object-oriented LISP extension, used in development of knowledge-based systems. "The LOOPS Manual", D.G. Bobrow & M. Stefik, Xerox Corp 1983. (See CommonLoops).

LOP - Language based on first-order logic. "SETHEO - A High-Perormance Theorem Prover for First-Order Logic", Reinhold Letz et al, J Automated Reasoning 8(2):183-212 (1992).

Lore -

1. Object-oriented language for knowledge representation. "Etude et Realisation d'un Language Objet: LORE", Y. Caseau, These, Paris-Sud, Nov 1987.

2. CGE, Marcoussis, France. Set-based language [same as 1?]  
info: Christophe Dony <chd.ibp.fr>

LOTIS - LLogic, TIming, Sequencing. Describes a computer via its data flow. Sammet 1969, p.620.

LOTOS - Specification language based on temporal ordering. "The Formal Description Technique LOTOS", P.H.J. van Eijk et al eds, N-H 1989. ISO 8807 (1990).

Lout - J. Kingson <jeff@cs.su.oz.au> Embedded language for the lout document preparation system. Procedural, with Scribe-like syntax.  
ftp://ftp.uu.net/tmp/lout.tar.Z

//wuarhive.wustl.edu/pub/aminet/text/dtp/loutBin203.lha for Amiga

Low-Ada - An intermediate language for Ada, intended for formal verification. Just a concept, not yet fully defined or implemented. "Low-Ada: an Ada Validation Tool", B.A. Wichmann, Ada User 11(1): pp.27-32.

LOWL - Abstract machine for bootstrapping ML/1. Mentioned in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, p.271.

LPC - ca 1988. Variant of C used to program the LP MUDs, programmable multi-user adventures.

LPG -

1. Linguaggio Procedure Grafiche (Italian for "Graphical Procedures Language"). dott. Gabriele Selmi. Roughly a cross between FORTRAN and APL, with graphical-oriented extensions and several peculiarities. Underlies the products of CAD.LAB Spa. "Graphical Procedure Language User's Guide and Reference Manual", CAD.LAB, Bologna, Italy, 1989, order code G089/9.

2. Langage de Programmation Generique. An applicative language, both specification and functional. Special emphasis on parametrized declarations. "Design and Implementation of a Generic, Logic and

Functional Programming Language", D. Bert et al, ESOP 86, LNCS 213, Springer 1986.

LPL - List Programming Language. LISP-like language with ALGOL-like syntax, for IBM 360. "LPL - LISP Programming Language", F.W. Blair et al, RC 3062, TJWRC, IBM, Sep 1970.

LPS - Sets with restricted universal quantifiers. "Logic Programming with Sets", G. Kuper, J Computer Sys Sci 41:44-64 (1990).

LRLTRAN - Lawrence Radiation Laboratory TRANslator. FORTRAN extension with vector arithmetic and dynamic storage, used for scientific work and systems programming, including the LTSS OS. "The LRLTRAN Compiler", S.F. Mendicino, CACM 11(11):747-775 (Nov 1969).

LSL -

1. Larch Shared Language. An assertion language. (See Larch).

2. Link and Selector Language. Graphic query language. "LSL: A Link and Selector Language", D.C. Tsichritzis, Proc Intl Conf Management of Data, ACM 1976, pp.123-134.

LSYD - Language for SYstems Development. PL/I-like language with data structure and character extensions. "Systems Programming Languages", R.D. Bergeron et al, in Advances in Computers 1971, A-P.

LT-2 - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

LTR - Langage Temps-Reel. A French predecessor to Ada, Modula-like with a set of special-purpose real-time constructs based on an event model. Mentioned in "An Overview of Ada", J.G.P. Barnes, Soft Prac & Exp 10:851-887 (1980).

LTR2 -

LTR3 - Parayre, France. Saw wide use by French military and avionics. "The LTR3 Reference Manual", A. Parayre, Delegation Generale pour l'Armement, France.

Lua - TeCGraf, Pontifical Cath U Rio de Janeiro (PUC/Rio), 1994. Pascal-like, functions can have variable number of arguments and return multiple values. Implemented in C, bytecode interpreted.  
ftp: ftp.icad.puc-rio.br:pub/lua/lua.1.1.tar.Z  
info: Luiz Henrique de Figueiredo <lhf@icad.puc-rio.br>

LUCID -

1. Early query language, ca. 1965, System Development Corp, Santa Monica, CA. Sammet 1969, p.701.

2. Ashcroft & Wadge <wwadge@csr.uvic.ca>, 1981. A dataflow language descended from ISWIM, lazy but first-order. Statements are regarded as equations defining a network of processors and communication lines, through which the data flows. Every data object is thought of as an infinite stream of simple values, every function as a filter. Lucid has no data constructors such as arrays or records. Iteration is simulated with 'is

current' and 'fby' (concatenation of sequences). "Lucid, the Dataflow Programming Language", W. Wadge, Academic Press 1985.

Lucinda - Combines Russell-like polymorphism with Linda-like concurrency. Implemented as a threaded interpreter written in C, for a Sun network and a Meiko Computing Surface. "Lucinda - An Overview", P. Butcher, U York <paubl@minster.york.ac.uk> et al, SIGPLAN Notices 26(8):90-100 (Aug 1991).

Lucy - Distributed constraint programming language. An actor subset of Janus. "Actors as a Special Case of Concurrent Constraint Programming", K. Kahn <kahn@parc.xerox.com> et al, SIGPLAN Notices 25(10):57-66 (OOPSLA/ ECOOP '90) (Oct 1990).

LUKKO - Heinanen, 1983. An object-oriented microprogramming language, influenced by Alphard and Modula. "A Data and Control Abstraction Approach to Microprogramming", J. Heinanen, Publ 18, Tampere U of Tech, Tampere, Finland, 1983.

LUSTRE - Real-time dataflow language for synchronous systems, especially automatic control and signal processing. A Lucid subset, plus timing operators and user-defined clocks. "Outline of a Real-Time Data-Flow Language", J.-L. Bergerand et al, Proc IEE-CS Real Time Systems Symp, San Diego, IEEE Dec 1985, pp.33-42. "LUSTRE: A Declarative Language for Programming Synchronous Systems", P. Caspi et al, Conf Rec 14th Ann ACM Symp on Princ Prog Langs, 1987.

LYaPAS - (Russian acronym for "Logical Language for the Representation of Synthesis Algorithms"). For the URAL-1 computer. Coded in octal! "LYaPAS: A Programming Language for Logic and Coding Algorithms", M.A. Gavrilov et al eds, Academic Press 1969.

LYNX - U Wisc 1984. Language for large distributed networks, using remote procedure calls. "The Lynx Distributed Programming Language: Motivation, Design and Experience", M.L. Scott, Computer Langs 16:209-233 (1991).

LYRIC - Language for Your Remote Instruction by Computer. CAI language implemented as a FORTRAN preprocessor. "Computer Assisted Instruction: Specification of Attributes for CAI Programs and Programmers", G.M. Silvern et al, Proc ACM 21st Natl Conf (1966).

M -

1. Alternative name for MUMPS.

2. Silicon Compiler Systems. A C-like language for multilevel hardware description. Currently available in the GDT package from Mentor Graphics.

M3 - Macro processor, forerunner of M4, for the AP-3 mini.

M4 - Macro processor for Unix and GCOS. "The M4 Macro Processor", Kernighan & Ritchie, Jul 1977.  
<ftp://prep.ai.mit.edu/pub/gnu/m4-1.0.tar.Z>

M5 - A. Dain, U Cincinnati, 1992. Macro processor, a generalization of M4. For Unix and DOS.

M6 - Yet another macro processor. Mentioned in Don Libes, "Life with

Unix".

MAC - Early system on Ferranti Mercury. Listed in CACM 2(5):16 (May 1959).

Mac-1 - Assembly language used in Structured Computer Organization, A.S. Tanenbaum, 3rd Edition, P-H 1989, Sect. 4.3. [See Mic-1]

MAC - ca 1959. Mercury Autocode 2. One of the first extensible languages, originally for the English Electric Mercury and later the ICT1300 series.

MAC-360 - ca. 1967. Solving numerical problems using equation-like input. "User's Guide to MAC-360", Charles Stark Draper Lab, Cambridge MA (Aug 1973) Sammet 1969, p.264.

Macaulay - Mike Stillman <mike@mssun7.msi.cornell.edu> and Dave Bayer <bayer@cunixa.columbia.edu> 1977. Symbolic math package for commutative algebra, algebraic geometry, cohomology. ftp://zariski.harvard.edu/Macaulay/\* Version 3 for Sun, Mac and Amiga, source in C

MACE - Concurrent object-oriented language.[?]

Machiavelli - Peter Buneman & Atsushi Ohori, U Pennsylvania, 1989. An extension of Standard ML based on orthogonal persistence. "Database Programming in Machiavelli: A Polymorphic Language with Static Type Inference", A. Ohori, Proc SIGMOD Conf, ACM, June 1989.

MACL - Macintosh Allegro CL. Former name of MCL. list: info-macl@cambridge.apple.com

MacLisp - MIT AI Lab, late 1960's. Later used by Project MAC, Mathlab, and Macsyma. Ran on the PDP-10. Introduced the LEXPR (a function with variable arity), macros, arrays, and CATCH/THROW. Was once one of two main branches of LISP (the other being Interlisp). In 1981 Common LISP was begun in an effort to combine the best features of both. "MACLISP Reference Manual", D.A. Moon <moon@cambridge.apple.com>, TR Project MAC, MIT 1974.

MACRO -

1. Assembly language for VAX/VMS.
2. PL/I-like language with extensions for string processing. "MACRO: A Programming Language", S.R. Greenwood, SIGPLAN Notices 14(9):80-91 (Sep 1979).

Macro SAP - Macro processing modification of SAP. D.E. Eastwood and D.M. McIlroy, unpublished memorandum, Bell Labs 1959. Led to TRAC.

MACSYMA - Project MAC's SYmbolic MANipulator. Joel Moses <moses@larch.lcs.mit.edu> MIT 1969, later Symbolics, Inc. The first comprehensive symbolic math system, written in LISP. "MACSYMA - The Fifth Year", J. Moses, SIGSAM Bulletin 8(3) (Aug 1974). Versions: Symbolics Macsyma, DOE Maxima (ANL), Vaxima. info: macsyma-service@symbolics.com ftp://rascal.ics.utexas.edu/pub/maxima-4-155.tar.Z DOE Maxima in Common LISP

MAD -

1. Michigan Algorithm Decoder. Developed at U Michigan by R. Graham, Bruce Arden <arden@hopper.ee.rochester.edu> and Bernard Galler <Bernard\_A\_Galler@um.cc.umich.edu>, 1959. Based on IAL. For the IBM 704, 709 and 7090, later ported to Philco, Univac and CDC machines. MAD was one of the first extensible languages: the user could define his own operators and data types. "Michigan Algorithm Decoder (The MAD Manual)", U Michigan Computing Center, 1966. Sammet 1969, p.205.

2. Dataflow language. "Implementation of Data Structures on a Data Flow Computer", D.L. Bowen, Ph.D. Thesis, Victoria U Manchester, Apr 1981.

Mad/1 - A later, much enhanced version of MAD, for the IBM 360. Michigan's answer to PL/I.

MADCAP - Math and set problems, for the Maniac II and CDC 6600. "MADCAP - A Scientific Compiler for a Displayed Formula Textbook Language", M.B. Wells, CACM 4(1):31-36 (Jan 1961). Sammet 1969, pp.271-281. Versions: Madcap 5 (1964), Madcap 6. "The Unified Data Structure Capability in Madcap 6", M.B. Wells et al, Intl J Comp Info Sci 1(3) (sep 1972).

MADTRAN - Early preprocessor that translated FORTRAN to MAD, for gain in speed.

MAGIC - Early system on Midac computer. Listed in CACM 2(5):16 (May 1959).

Magic Paper - Early interactive symbolic math system. Sammet 1969, p.510.

Magma2 - Language that allows programmability of the control environment, e.g. recursion, backtracking, coroutines, nondeterminism, etc. "Magma2: A Language Oriented Toward Experiments in Control", Franco Turini, ACM TOPLAS 6(4):468-486 (Oct 1984).

MagmaLISP - Predecessor of Magma2. "MagmaLISP: A Machine Language for Artificial Intelligence", C. Mantagero et al, Proc 4th Intl Joint Conf Artif Intell, 1975, pp.556-561.

MAGNUM - Tymshare Inc, late 70's. Database language for DEC-10's, used internally by Tymshare.

Magritte - J. Gosling. Constraint language for interactive graphical layout. Solves its constraints using algebraic transformations. "Algebraic Constraints", J. Gosling, PhD Thesis, TR CS-83-132, CMU, May 1983.

MAINSAIL - MACHine INdependent SAIL. From XIDAK, Palo Alto CA, (415) 855-9271.

Maisie - A C extension with concurrency via asynchronous typed message passing and lightweight processes.  
<ftp://cs.ucla.edu/pub/maisie.2.1.1.3.tar.Z>

Make - Language for the Unix file maintenance utility Make. "Make - A Program for Maintaining Computer Programs", A.I. Feldman, TR No 57, Bell Labs Apr 1977.

MAL - Micro Assembly Language - Microprogramming language with high-level syntax, used in Structured Computer Organization, A.S. Tanenbaum, 3rd ed, P-H 1989, Sect 4.4. [See Mic-1, Mac-1].

MALPAS IL - TA Consultancy Services. A strongly typed, block-structured intermediate language intended for static analysis and verification. Translators exist for Ada, C, Pascal, Fortran 77, Coral 66 and several assembly languages. "Code Verification with the Aid of MALPAS", N.J. Ward, Proc IEE Colloq of High Integrity Ada (Jan 1993).

Manchester Autocode - Predecessor of Mercury Autocode. "The Programming Strategy Used with the Manchester University Mark I Computer", R.A. Brooker, Proc IEE 103B Suppl:151-157, 1956.

Mandala - ICOT, Japan. A system based on Concurrent Prolog. "Mandala: A Logic Based Knowledge Programming System", K. Furukawa et al, Intl Conf 5th Gen Comp Sys 1984.

MAO - Early symbolic math system. A. Rom, Celest Mech 1:309-319 (1969).

MAP - Mathematical Analysis without Programming. On-line system under CTSS for math. Sammet 1969, p.240.

Maple - B. Char, K. Geddes, G. Gonnet, M. Monagan & S. Watt, U Waterloo, Canada 1980. Symbolic math system. Waterloo Maple Software. Current version: Maple V.  
info: wmsi@daisy.waterloo.edu  
list: glabahn@daisy.waterloo.edu

MARBLE - A Pascal-like microprogramming language. "MARBLE: A High Level Machine-Independent Language for Microprogramming", S. Davidson et al, in Firmware, Microprogramming and Restructurable Hardware, G. Chroust et al eds, N-H 1980, pp.253-263.

Maril - Machine description language used by the Marion code generator. "The Marion System for Retargetable Instruction Scheduling", D.G. Bradlee et al, SIGPLAN Notices 26(6):229-240 (June 1991).

Markov - [?]

Marseille Prolog - One of the two main dialects of Prolog, the other being Edinburgh Prolog. The difference is largely syntax. The original Marseille Interpreter (1973) was written in FORTRAN.

MARSYAS - MARshall SYstem for Aerospace Simulation. Simulation of large physical systems. "MARSYAS - A Software System for the Digital Simulation of Physical Systems", H. Trauboth et al, Proc SJCC, 36 (1970).

MARVIN - U Dortmund, 1984. Applicative language based on Modula-2, enhanced by signatures (grammars) terms (trees) and attribute couplings (functions on trees). Used for specification of language translators. "MARVIN - A Tool for Applicative and Modular Compiler Specification", H. Ganziger et al, Forsch 220, U Dortmund, Jul 1986.

Mary - Mark Rain. Machine-oriented language, a superset of ALGOL68, extensible. Hidden on the back cover of the manual: MARY HAD A LITTLE LAMB

- COERCION IMPOSSIBLE. "Mary Programmer's Reference Manual", M. Rain et al, R Unit, Trondheim Norway, 1974. "Operator Expressions in Mary", M. Rain, SIGPLAN Notices 8(1) (Jan 1973).

MAS - Modula-2 Algebra System. "Modula-2 Algebra System", H. Kredel, Proc DISCO 90 Capri, LNCS 429, Springer 1990, pp270-271.  
ftp://alice.fmi.uni-passau.de/pub/mas/\* for Atari, Amiga, MS-DOS, OS/2

MASM - Microsoft Assembler for MS-DOS.

Massey Hope - Massey U, NZ. Refinement of Hope+C with improved syntax, and no stream I/O.  
info: Nigel Perry <N.Perry@massey.ac.nz>

Matchmaker - A language for specifying and automating the generation of multi-lingual interprocess communication interfaces. MIG is an implementation of a subset of Matchmaker that generates C and C++ remote procedure call interfaces for interprocess communication between Mach tasks. "MIG - The Mach Interface Generator", R.P. Draves et al, CS CMU, (4 Aug 1989).

Mathcad - Symbolic math environment.

Mathematica - (name suggested by Steve Jobs). Wolfram Research, 1988. Symbolic math and graphics system. The language emphasizes rules and pattern-matching. "Mathematica: A System for Doing Mathematics by Computer", Stephen Wolfram, A-W 1988.  
ftp://otter.stanford.edu, //ftp.ncsa.uiuc.edu  
list: mathgroup-request@yoda.ncsa.uiuc.edu  
info: info@wri.com

MATHLAB - Symbolic math system, MITRE, 1964. Later version: MATHLAB 68 (PDP-6, 1967). "The Legacy of MATHLAB 68", C. Engelman, Proc 2nd Symp on Symbolic and Algebraic Manip, ACM (Mar 1971). Sammet 1969, p.498.

MATH-MATIC or MATHMATIC - Alternate name for AT-3. Early, pre-FORTRAN language for UNIVAC I or II. Sammet 1969.

Matrix Compiler - Early matrix computations on UNIVAC. Sammet 1969, p.642.

MATRIX MATH - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

mawk - Mike Brennan <brennan@bcsaic.boeing.com> 1991. An implementation of nawk, distributed under GNU license but distinct from GNU's gawk.  
ftp://oxy.edu/public/mawk

MAXIMOP - "Job Control Languages: MAXIMOP and CAFE", J. Brandon, Proc BCS Symp on Job Control Languages--Past Present and Future, NCC, Manchester, ENgland 1974.

MBASIC - Microsoft BASIC.

MC - Extension of C with modules. Symbols in other modules can be referenced using a dot notation. "Design and Implementation of a C-Based Language for Distributed Real-Time Systems", A. Rizk et al, SIGPLAN Notices 22(6):83-96 (June 1987).

McG360 - Interactive, similar to PAL[5], for IBM 360. "McG360 Programmer's Guide", RC 2693, IBM TJWRC, Nov 1969.

MCL - Macintosh Common LISP. (Previously MACL.)

M-Code -

- 1) Intermediate code produced by the original ETH Modula-2 compiler.
- 2) Intermediate language for an SECD-like machine, used by the Concert implementation of MultiLISP.

MCS - Meta Class System. A portable object-oriented extension of Common LISP.

<ftp://gmdzi.gmd.de/pub/lisp/mcs>

MDL - (originally "Muddle"). C. Reeve, C. Hewitt & G. Sussman, Dynamic Modeling Group, MIT ca. 1971. Intended as a successor to Lisp, and a possible base for Planner-70. Basically LISP 1.5 with data types and arrays. Many of its features were advanced at the time (I/O, interrupt handling and coroutining), and were incorporated into later LISP dialects ("optional", "rest" and "aux" markers). In the mid 80's there was an effort to use bytecoding to make the language portable. CLU was first implemented in MDL. Infocom wrote Zork in MDL, and used it as the basis for the ZIL interpreter. "The MDL Programming Language", S.W. Galley et al, Doc SYS.11.01, Project MAC, MIT (Nov 1975). Implementations exist for ITS, TOPS-20, BSD 4.3, Apollo Domain, SunOS and A/UX.

me too - Peter Henderson, 1984. Functional language for executable specifications. Like LispKit Lisp, but with sets, maps and sequences to describe the specification. "Functional Programming, Formal Specification and Rapid Prototyping", IEEE Trans Soft Eng, SE-12(2):241-250 (Feb 1986).

MELD - Concurrent, object-oriented, dataflow, modular and fault-tolerant! Comparable to SR. "MELDing Multiple Granularities of Parallelism", G. Kaiser et al, ECOOP '89, pp.147-166, Cambridge U Press 1989.

MeldC - Columbia U, 1990. A C-based concurrent object-oriented coordination language built on a reflective architecture. A redesign of MELD. Version 2.0 for Sun4's and DECstations.  
info: Gail Kaiser <[meldc@cs.columbia.edu](mailto:meldc@cs.columbia.edu)>

Melinda - "Melinda: Linda with Multiple Tuple Spaces", S. Hupfer, <[hupfer-susanne@yale.edu](mailto:hupfer-susanne@yale.edu)> YALEU/DCS/RR-766, Yale U Feb 1990.

Mentat - U Virginia. Object-oriented distributed language, an extension of C++, portable to a variety of MIMD architectures. "Mentat: An Object-Oriented Macro Data Flow System", A. Grimshaw <[grimshaw@cs.virginia.edu](mailto:grimshaw@cs.virginia.edu)> et al, SIGPLAN Notices 22(12):35-47 (Dec 1987) (OOPSLA '87). Available now for Sun 3 & 4 and iPSC/2, and soon Mach, iPSC860, RS/6000 and Iris.  
info: [mentat@uvacs.cs.virginia.edu](mailto:mentat@uvacs.cs.virginia.edu)

MENTOR - CAI language. "Computer Systems for Teaching Complex Concepts", Report 1742, BBN, Mar 1969.

MENYMA/S - "A Message Oriented Language for System Applications", A. Koch

et al, Proc 3rd Intl Conf Distrib Comp Sys, IEEE 1982, pp.824-832.

Mercury Autocode - Autocode for the Ferranti Mercury machine. (See Autocode).

MEROON - An object-oriented system built on Scheme.  
<ftp://nexus.yorku.ca/pub/scheme/new/>\*

Mesa - Xerox PARC, 1977. System and application programming for proprietary hardware: Alto, Dolphin, Dorado and Dandelion. Pascal-like syntax, Algol68-like semantics. An early version was weakly typed. Mesa's modules with separately compilable definition and implementation parts directly led to Wirth's design for Modula. Threads, coroutines (fork/join), exceptions, and monitors. Type checking may be disabled. Mesa was used internally by Xerox to develop ViewPoint, the Xerox Star, MDE, and the controller of a high-end copier. It was released to a few universities in 1985. Succeeded by Cedar. "Mesa Language Manual", J.G. Mitchell et al, Xerox PARC, CSL-79-3 (Apr 1979). "Early Experience with Mesa", Geschke et al, CACM 20(8):540-552 (Aug 1977).

META - CDC, ca 1977. Assembly language for the CYBER 200. CDC Pub 60256020.

META 5 - Early syntax-directed compiler-compiler, used for translating one high-level language to another. "META 5: A Tool to Manipulate Strings of Data", D.K. Oppenheim et al, Proc 21st Natl Conf, ACM 1966. Sammet 1969, p.638. Versions: META II, META-3.

Meta-II - An early compiler-compiler. "Meta-II: a Syntax Oriented Compiler Writing Language", V. Schorre, Proc 19th ACM Natl Conf 1964.

Meta-IV - See VDM-SL.

Meta-Crystal - A language for transformations of Crystal programs. Implemented in T. "Meta-Crystal- A Metalanguage for Parallel-Program Optimization", J.A. Yang et al, TR YALEU/DCS/TR-786, Yale Apr 1990. (See Crystal).

METAFONT - Knuth. A system for the design of raster-based alphabets. Companion to TeX. "The METAFONT Book," Donald Knuth, A-W 1986. Version 2.0, March 1990.

METAL -

1. Mega-Extensive Telecommunications Applications Language. BBS language for PRODOS 8 on Apple ][.

2. The syntax-definition formalism of the Mentor system. Metal specifications are compiled to specifications for a scanner/parser generator such as Lex/Yacc. "Metal: A Formalism to Specify Formalisms", G. Kahn et al, Sci Comp Prog 3:151-188 (1983).

Meta-Vlisp - E. St.James <esj@litp.ibp.fr> France. A Lisp dialect with many innovations.

Met-English - Metropolitan Life, early 60's. Fortran-like, with support for variable-length bit fields. Most MetLife DP in the 60's and 70's was

in Met-English. Originally for Honeywell machines, but many programs still run under IBM/MVS via a Honeywell emulator.

METEOR - A version of COMIT with Lisp-like syntax, written in MIT Lisp 1.5 for the IBM 7090. "METEOR - A List Interpreter for String Transformation", D.G. Bobrow in The Programming Language LISP and its Interpretation, E.D. and D.G. Bobrow eds, 1964.

Methods - Digitaltalk, ca 1985. Line-oriented Smalltalk for PC's, predecessor of Smalltalk/V.

MHDL -

1. MIMIC Hardware Description Language. Intermetrics. (703)827-2606

2. Microwave Hardware Description Language. Incorporates Haskell 1.2. Intermetrics.  
info: David Barton <dlb@hudson.wash.inmet.com>

Mic-1, Mic-2 - Microprogramming languages, used in Structured Computer Organization, A.S. Tanenbaum, 3rd ed, P-H 1989, Sect 4.4, 4.5. [See Mac-1].

microAPL - An APL-like microprogramming language. "High Level Microprogramming with APL Syntax", R.F. Hobson et al, Proc 14th Ann Workshop Microprogramming (MICRO-14), 1981, pp.131-139.

microPLANNER - G.J. Sussman et al, MIT. Subset of PLANNER, implemented in LISP. Superseded by Conniver. Important features: goal-oriented, pattern-directed procedure invocation, embedded knowledge base, automatic backtracking. "microPLANNER Reference Manual", G.J. Sussman et al, AI Memo 203, MIT AI Lab, 1970.

microTAL - A high level machine dependent microprogramming language based on TAL. Aim was to facilitate migration of TAL functions to microcode. "microTAL - A Machine-Dependent High-Level Microprogramming Language", J.F. Bartlett, Proc 14th Ann Workshop Microprogramming (MICRO-14), 1981, pp.109-114. (See TAL).

MIDAS - Digital simulation language. Sammet 1969, p.627.

MIDL - MicroInstruction Description Language. "MIDL - A Microinstruction Description Language", M. Sint, Proc 14th Ann Workshop Microprogramming (MICRO-14), 1981, pp.95-106.

MIIS - ("Meese"). Interpreted. One-letter keywords. Similar to MUMPS?

MIKE - Micro Interpreter for Knowledge Engineering. Expert system shell for teaching purposes, with forward and backward chaining and user-definable conflict resolution strategies. In Edinburgh Prolog. BYTE Oct 1990. Version 2.03  
ftp://hcrl.open.ac.uk/pub/software/src/MIKE-v2.03  
info: Marc Eisenstadt <M.Eisenstadt@hcrl.open.ac.uk>

MILITRAN - Sys Res Group, ONR 1964. Discrete simulation for military applications. Sammet 1969, p.657.

MIMIC - J.H. Andrews, NIH 1967. Early language for solving engineering problems such as differential equations that would otherwise have been done on an analog computer. "MIMIC, An Alternative Programming Language for Industrial Dynamics, N.D. Peterson, Socio-Econ Plan Sci. 6, Pergamon 1972.

MIMOLA - Operational hardware specification language. "A Retargetable Compiler for a High-Level Microprogramming Language", 17th Ann Workshop on Microprogramming, P. Marwedel, IEEE 1984, pp.267-274.

Mini-ML - "A Simple Applicative Language: Mini-ML", D. Clement et al, Proc 1986 ACM Conf on LISP and Functional Prog, (Aug 1986).

Mini PL/I - A commercial PL/I subset for the Olivetti Audit 7 minicomputer.

MINITAB II - Interactive solution of small statistical problems. "MINITAB Student Handbook", T.A. Ryan et al, Duxbury Press 1976.

MINT - Mint Is Not TRAC. Version of TRAC used as the extension language in the Freemacs editor.  
ftp://sun.soe.clarkson.edu/pub/freemacs

Miracula - Stefan Kahrs <smk@ed.ac.uk>, LFCS. An implementation of a subset of Miranda, no modules or files. Can be interactively switched between eager and lazy evaluation. Portable source in C from the author.

Miranda - (latin for "admirable", also the heroine of Shakespeare's Tempest). David A. Turner <dat@ukc.ac.uk>, U Kent early 1980's. Lazy, purely functional. A commercial descendant of SASL and KRC, with ML's type system. Terse syntax using the offside rule for indentation. Type declarations are optional. Nested pattern-matching, list comprehensions, modules. Sections rather than lambda abstractions. User types are algebraic, may be constrained by laws. Implemented by SKI reduction. The KAOS operating system is written entirely in Miranda. "Miranda: A Non Strict Functional Language with Polymorphic Types", D.A. Turner, in Functional Programming Languages and Computer Architecture, LNCS 201, Springer 1985. "Functional Programming with Miranda", Holyer, Pitman Press 0-273-03453-7. (See Miracula, Orwell).  
info: Research Software Ltd, or mira-request@ukc.ac.uk  
ftp://wombat.doc.ic.ac.uk/pub/mira2hs\* and mira2lml\* Translators from Miranda to Haskell and Miranda to LML

MIRFAC - Mathematics in Recognizable Form Automatically Compiled. Early interactive system resembling BASIC, typewriter output with special math symbols. Sammet 1969, pp.281-284.

MISHAP - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16, (May 1959).

MITILAC - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

MIXAL - MIX Assembly Language. Assembly language for Knuth's hypothetical MIX machine, used in The Art of Computer Programming v.1, Donald Knuth, A-W 1969.  
ftp://altdorf.ai.mit.edu/archive/adler/MIX/mix.tar.Z  
//locke.ccil.org/pub/retro/mixal-1.05.shar.gz

MJS - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

ML -

1. Manipulator Language. IBM language for handling robots.

2. Meta Language. R. Milner <rm@lfcs.edinburgh.ac.uk> et al, 1973. A strict higher-order functional language with statically-checked polymorphic types, garbage collection and a formal semantics. It began as the metalanguage for the Edinburgh LCF proof assistant. (LCF="Logic for Computable Functions") "A Metalanguage for Interactive Proof in LCF", M.J.C. Gordon et al, 5th POPL, ACM 1978. (See SML). LCF ML was implemented in Stanford LISP. Cardelli (1981) implemented ML in Pascal, using the FAM (Functional Abstract Machine).

ML-2000 - Dialect of ML, an extension and redesign of Standard ML. Under development.

MLAB - Modeling LABORatory. Interactive mathematical modeling. "MLAB, An On-Line Modeling Laboratory", NIH (Mar 1975).

ML/I - Early macro translating system. P.J. Brown, CACM 10(10):618-623, (Oct 1967).

MLISP -

1. M-expression LISP. J. McCarthy, 1962. The original "meta-language" syntax of LISP, intended for external use in place of the parenthesized S-expression syntax. "LISP 1.5 Programmer's Manual", J. McCarthy et al, MIT Press 1962.

2. Meta-LISP. D.C. Smith & H. Enea. LISP variant with ALGOL-like syntax. Not just a surface syntax, a full language. "MLISP", D.C. Smith, TR CS-179, CS Dept, Stanford (Oct 1970). Version: MLISP2.

3. A hybrid of M-expression LISP and Scheme. "M-LISP: Its Natural Semantics and Equational Logic", R. Muller, SIGPLAN Notices 26(9):234-242 (Sept 1991) (PEPM '91).

ML-Linda - U Edinburgh, under development.

ML Threads - Greg Morrisett <jgmmorris@cs.cmu.edu>. SML/NJ with mutual exclusion primitives similar to those in Modula-2+ and Mesa. Lightweight threads are created using 'fork'. They are pre-emptively scheduled, and communicate via shared memory which can be protected by a 'mutex' (monitor). "Adding Threads to Standard ML", E. Cooper et al, CMU-CS-90-186, CMU Dec 1990. Implementations for 68020, SPARC and MIPS, and also VAX- and MIPS-based multiprocessors.

Mma - R. Fateman, 1991. A fast Mathematica-like system, in Allegro CL. <ftp://peoplesparc.berkeley.edu/pub/mma.tar.Z>

MML - Human-Machine Language. CCITT. Language for telecommunications applications. Complex natural-language syntax. CCITT Recommendations Z.311-Z.318, Z-341, Nov 1984.

MOBSSL-UAF - Merritt and Miller's Own Block-Structured Simulation Language-Unpronounceable Acronym For. Interactive continuous simulations.

"MOBSSL - An Augmented Block Structured Continuous System Simulation Language for Digital and Hybrid Computers", M.J. Merritt et al, Proc FJCC 35, AFIPS (Fall 1969).

Mock Lisp - The LISP used by the Gosling Emacs editor.

MODCAL - Version of HP-PASCAL enhanced with system programming constructs, used internally by HP.

Mode - Object-oriented. "The Programming Language Mode: Language Definition and User Guide", J. Vihavainen, C-1987-50, U Helsinki, 1987.

MODEF - Pascal-like language with polymorphism and data abstraction. "Definition of the Programming Language MODEF", J. Steensgard-Madsen et al, SIGPLAN Notices 19(2):92-110 (Feb 1984).

MODEL - Pascal-like language with extensions for large-scale system programming and interface with FORTRAN applications. Generic procedures, and a "static" macro-like approach to data abstraction. Produced P-code. Used to implement DEMOS operating system on Cray-1. "A Manual for the MODEL Programming Language", J.B. Morris, Los Alamos 1976.

MODSIM II - 1986. Object-oriented modular language for discrete simulation, with multiple inheritance, strong typing, integrated 2D and 3D graphics. Compiles to C. CACI, La Jolla, (619) 457-9681.  
list: palmer@world.std.com

Modula - MODular LAnguage. Wirth, 1977. Predecessor of Modula-2, more oriented towards concurrent programming but otherwise quite similar. "Modula - A Language for Modular Multiprogramming", N. Wirth, Soft Prac & Exp 7(1):3-35 (Jan 1977).

Modula-2 - Wirth, ETH 1978. Developed as the system language for the Lilith workstation. The central concept is the module which may be used to encapsulate a set of related subprograms and data structures, and restrict their visibility from other portions of the program. Each module has a definition part giving the interface, and an implementation part. The language provides limited single-processor concurrency (monitors, coroutines and explicit transfer of control) and hardware access (absolute addresses and interrupts). Uses name equivalence. "Programming in Modula-2", N. Wirth, Springer 1985.  
ftp:gatekeeper.dec.com:pub/DEC/Modula-2/m2.tar.Z

Modula-2\* - M. Philippsen <philipp@ira.uka.de>, U Karlsruhe. Modula-2 extension. Uses a superset of data parallelism, allowing both synchronous and asynchronous programs, both SIMD and MIMD. Parallelism may be nested to any depth. Version for MasPar and simulator for SPARC. "Modula-2\*: An Extension of Modula-2 for Highly Parallel, Portable Programs", W. Tichy et al, TR 4/90, U Karlsruhe, Jan 1990.  
ftp://ftp.ira.uka.de/pub/programming/modula2star  
info: Ernst Heinz <heinz@ira.uka.de>

Modula-2+ - P. Rovner et al, DEC SRC, Palo Alto CA, 1984. Exceptions and threads. "Modula-2+ User's Manual", M-C van Leunen. "Extending Modula-2 to Build Large, Integrated Systems", P. Rovner, IEEE Software 3(6):46-57 (Nov 1986).

Modula-3 - L. Cardelli et al, DEC and Olivetti, 1988. A descendant of Modula-2+ and Cedar, designed for safety and simplicity. Objects, generics, threads, exceptions and garbage collection. Modules are explicitly safe or unsafe. As in Mesa, any set of variables can be monitored. No multiple inheritance, no operator overloading. Uses structural equivalence. "Modula-3 Report", Luca Cardelli et al, TR 52, DEC SRC, and Olivetti Research Center, Aug 1988 (revised Oct 1989). The changes are described in "System Programming with Modula-3", Greg Nelson ed, P-H 1991, ISBN 0-13-590464-1. "Modula-3", Sam Harbison, P-H 1992. Version: SRC Modula-3 V1.5.  
<ftp://gatekeeper.dec.com/pub/DEC/Modula-3/release/>\*

Modula-3\* - Incorporation of Modula-2\* ideas into Modula-3. "Modula-3\*: An Efficiently Compilable Extension of Modula-3 for Problem-Oriented Explicitly Parallel Programming", E. Heinz <heinze@ira.uka.de>, 1993.

Modula-3pi - Machine-independent intermediate language for compilation of Modula-3\*. "Modula-3pi Language Definition", E.A. Heinz, TR, U Karlsruhe 1993.

Modula-P - "Modula-P: A Language for Parallel Programming Definition and Implementation on a Transputer Network", R. Hoffart et al, IEEE Conf Comp Langs 1992.

Modula-Prolog - Adds a Prolog layer to Modula-2. "Modula-Prolog: A Software Development Tool", C. Muller IEEE Software pp.39-45 (Nov 1986).

Modula/R - Modula with relational database constructs added. LIDAS Group (J. Koch, M. Mall, P. Putfarken, M. Reimer, J.W. Schmidt, C.A. Zehnder) "Modula/R Report", LIDAS Memo 091-83, ETH Zurich, Sep 1983.

Modular C - Preprocessor-based extension to C allowing modules. Article by Stowe Boyd, Azrex Inc, SIGPLAN Notices, ca 1980.

Modular Prolog - An extension of SB-Prolog (version 3.1) extended with ML-style modules. For Sun-4. "A Calculus for the Construction of Modular Prolog Programs", D. Sannella et al, J Logic Prog 12:147-177 (1992)  
<ftp://ftp.dcs.ed.ac.uk/pub/dts/mod-prolog.tar.Z>

Modulex - Based on Modula-2. Mentioned by M.P. Atkinson & J.W. Schmidt in a tutorial in Zurich, 1989.

Mona - An experimental dialect of Oberon. Allows data types to be recursive. TR 102, ETH Zurich.

MooZ - Object-oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

MOPS - Michael Hore. A derivative of Neon. Multiple inheritance.  
<ftp://oddjop.uchicago.edu/pub/Yerk>

MORAL - Mentioned in "An Overview of Ada", J.G.P. Barnes, Soft Prac & Exp 10:851-887 (1980).

MORTRAN - A public domain FORTRAN preprocessor for structured programming.

Mouse - Peter Grogono, 1975. A mighty small macro language. "Mouse, A

Language for Microcomputers", P. Grogono <grogono@concour.cs.concordia.ca> Petrocelli Books, 1983.

Moxie - Language for real-time computer music synthesis, written in XPL. "Moxie: A Language for Computer Music Performance", D. Collinge, Proc Intl Computer Music Conf, Computer Music Assoc 1984, pp.217-220.

MP-1 - Assembly language for the MasPar machine.

MPGL - Micro-Program Generating Language. A retargetable register transfer language, in which the machine specification is included as part of the program. "A Microprogram-Generating System", T. Baba, in Information Processing 77, N-H 1977, pp. 739-744.

MPL -

1. Early possible name for PL/I. Sammet 1969, p.542.
2. MasPar. A data-parallel version of C.
3. Motorola Programming Language. A low-level PL/I-like language, similar to PL/M, but for the Motorola 6800.
4. MicroProgramming Language. The first high level microprogramming language. PL/I-like syntax. Data objects are declared as one- or two-dimensional arrays of bits, or as events. Statements on the same line represent register transfers caused by one microinstruction, and are executed in parallel. For vertical machines. "A High Level Microprogramming Language (MPL)", R.H. Eckhouse Jr, PhD Thesis, SUNY Buffalo, 1971.

MPL II - Burroughs VMS MPL II Language Reference Manual.

MPPL - Early possible name for PL/I. Sammet 1969, p.542.

M-Prolog -

1. Marseille Prolog.
2. An extension to Prolog involving modules. "The MProlog System", J. Bendl et al, Proc Logic Prog Workshop, 1980.

MPS III - Solving matrices and producing reports. "MPS III DATAFORM User Manual", Management Science Systems (1976).

MPSX - Mathematical Programming System Extended. Solution strategy for mathematical programming. "Mathematical Programming System Extended (MPSX) Control Language User's Manual", SH20-0932, IBM. Sammet 1978.

MRS - An integration of logic programming into LISP. "A Modifiable Representation System", M. Genesereth et al, HPP 80-22, CS Dept Stanford U 1980.

MSG.84 - "Analysis and Design in MSG.84: Formalizing Functional Specifications", V. Berzins et al IEEE Trans Soft Eng SE-11(8):657-670 (Aug 1985).

MUCAL - Language for playing music on PDP-8 [?]

Muddle - Original name of MDL.

muFP - Functional language for hardware design, predecessor to Ruby[1].

Mul-T - An implementation of Multilisp built on T, for the Encore Multimax. "Mul-T: A High-Performance Parallel Lisp", SIGPLAN Notices 24(7):81-90 (Jul 1989).

multiC - Wavetracer. A data-parallel version of C.

MultiLisp - Parallel extension of Scheme, with explicit concurrency. The form (future X) immediately returns a 'future', and creates a task to evaluate X. When the evaluation is complete, the future is resolved to be the value. "MultiLisp: A Language for Concurrent Symbolic Computation", R. Halstead, TOPLAS pp.501-538 (Oct 1985).

Multi-Pascal - Extension of Pascal-S with multiprocessing features. Used in "The Art of Parallel Programming", Bruce P. Lester, P-H 1993.

MultiScheme - An implementation of Multilisp built on MIT's C-Scheme, for the BBN Butterfly. "MultiScheme: A Paralled Processing System Based on MIT Scheme", J. Miller, TR-402, MIT LCS, Sept 1987.

MUMPS - Massachusetts General Hospital Utility Multi-Programming System. A database-oriented OS and the language that goes with it. Used originally for medical records. Only data type is the character string. Persistent associative arrays. Current versions for IBM RT and R6000, DSM (Digital Standard Mumps) for DEC, Datatree MUMPS for IBM PC, Unix MUMPS from PFCS <mumps@pfcs.com>. "MUMPS Language Standard", ANS X11.1-1977, ASN X11-1990. MUMPS User's Group, Box 208, Bedford MA 01730.  
list: MUMPS-L@UGA.BITNET.  
ftp:openmsql.intersys.com:/pub/dtstudent/\* for MS-DOS

MU-Prolog - L. Naish, U Melbourne 1982. Prolog with 'wait' declarations for corouting. "Negation and Control in Prolog", L. Naish, TR 85/12, U Melbourne (1985). (See NU-Prolog).

MuSimp - LISP variant used as the programming language for the PC symbolic math package MuMath.

Muse - OR-parallel logic programming.

Music - Bell Labs, 60's. A series of early languages for musical sound synthesis. Versions: Music I through Music V. "An Acoustical Compiler for Music and Psychological Stimuli", M.V. Mathews, Bell Sys Tech J 40 (1961).

MUSL - Manchester University Systems Language.

MYSTIC - Early system on IBM 704, IBM 650, IBM 1103 and 1103A. Listed in CACM 2(5):16 (May 1959).

NASTRAN - NAsa STress ANalysis program. Large stress analysis problems. "The NASTRAN User's Manual", SP-222(C3), NASA.

Napier - Atkinson & Morrison, St Andrews U; design began ca. 1985, first

implementation Napier88, 1988. Based on orthogonal persistence, permits definition and manipulation of namespaces. "The Napier88 Reference Manual", R. Morrison et al, CS Depts St Andrews U and U Glasgow, Persistent Programming Research Report PPRR-77-89, 1989.  
<http://www-fide.dcs.st-andrews.ac.uk:8080>

NAPLPS - North American Presentation-Level-Protocol Syntax. Format for sending text and graphics over communication lines. Used by videotex systems and (covertly) by Prodigy. See RIPscrip.

NAPSS - Numerical Analysis Problem Solving System. Purdue ca. 1965. "NAPSS - A Numerical Analysis Problem Solving System", J.R. Rice et al, Proc ACM 21st Natl Conf, 1966. Sammet 1969, p.299.

NASTRAN - Engineering language, listed [?] 1976.

NATURAL - Software AG, Germany. Integrated 4GL used by the database system ADABAS. Menu-driven version: SUPER/NATURAL. Also NATURAL 2?

Natural English - Used to mean programming in normal, spoken English. Sammet 1969, p.768.

Nawk - New AWK. AT&T. Pattern scanning and processing language. An enhanced version of AWK, with dynamic regular expressions, additional built-ins and operators, and user-defined functions.

NB - ("New B"?) Original name of C.

NDL- Network Definition Language. Used to program the DCP (Data Communications Processor) on Burroughs Large System. Version: NDL II.

Nebula - ICL. Early business-oriented language for Ferranti Orion computer. "NEBULA - A Programming Language for Data Processing", T.G. Braunholtz et al, Computer J 4(3):197-201 (1961).

NELIAC - Navy Electronics Laboratory International ALGOL Compiler. 1958-1959. Numeric and logical computations, based on IAL. "Neliac - A Dialect of Algol", H.D. Huskey et al, CACM 3(8):463-468 (Aug 1960). "Machine-Independent Computer Programming, Maurice Halstead, Spartan Books 1962. Version: BC NELIAC.

Neon - Charles Duff. An object-oriented extension of FORTH, for the Mac. Inheritance, SANE floating point, system classes and objects for Mac interfacing, overlays. Sold by Kriya Systems, 1985-1988. Modified, made PD and renamed Yerk.

NERECO - Network REMote COMMunications. CSP with extensions to allow asymmetrical and asynchronous communications and fault handling. Implemented on a network of Suns. "A Concurrent Programming Support for Distributed Systems", G. Spezzano et al, in Computing Systems vol 3, pp.423-447, U Cal Press, 1990.

NESL - Fine-grained, functional, data-parallel language with nested data structures and nested parallelism. Includes a built-in parallel data type and parallel operations on sequences. Loosely based on ML. Useful for parallel algorithms on sparse matrices and graphs. "NESL: A Nested Data-Parallel Language", Guy Blelloch, CMU-CS-93-129, April 1993. Version 2.6

for Cray Y-MP, CM-2, and Encore Multimax.  
ftp://nesl.scandal.cs.cmu.edu/pub/code/nesl/nesl/tar.Z

NETL - Semantic network language, for connectionist architectures. "NETL: A System for Representing and Using Real-World Data", S.E. Fahlman, MIT Press 1979.

New Flavors - Symbolics. An object-oriented LISP, successor to Flavors, led to CLOS. "Reference Guide to Symbolics-Lisp", Symbolics, March 1985.

NEWP - NEW Programming language. Replaced ESPOL on Burroughs Large System.

NewsClip - Looking Glass Software. Very high level language for writing netnews filters. Used by ClariNet Communications.  
info: newsclip@clarinet.com

Newspeak -

1. J.K. Foderaro. Inspired by Scratchpad. "The Design of a Language for Algebraic Computation", Ph.D. Thesis, UC Berkeley, 1983.

2. [?]

Newsqueak - Concurrent applicative language with synchronous channels. "Newsqueak: A Language for Communicating with Mice", R. Pike CSTR143, Bell Labs (March 1989). "The Implementation of Newsqueak", Rob Pike, Soft Prac & Exp 20(7):649-659 (July 1990).

Newton - (named after Isaac Newton (1642-1727)). Rapin et al, Swiss Federal Inst Tech, Lausanne 1981. General purpose expression language, syntactically ALGOL-like, with object-oriented and functional features and a rich set of primitives for concurrency. Used for undergraduate teaching at Lausanne (EPFL). "Procedural Objects in Newton", Ch. Rapin, SIGPLAN Notices 24(9) (Sep 1989). "The Newton Language", Ch. Rapin et al, SIGPLAN Notices 16(8):31-40 (Aug 1981). "Programming in Newton", Wuetrich and Menu, EPFL 1982. Versions: Newton 2.6 for VAX/VMS and Newton 1.2 for DEC-Alpha/OSF-1.  
info: J. Hulaas <hulaas@lcodecl.epfl.ch>  
ftp://ellc4.epfl.ch/pub/languages/Newton/\*

Nexpert Object - Expert system.

NFQL - "NFQL: The Natural Forms Query Language", D. Embley, Trans Database Sys 14(2):168-211 (June 1989).

NGL - Dialect of IGL.

NIAL - Nested Interactive Array Language. Queen's U, Canada. High-level array-oriented language, based on Array Theory as developed by Trenchard More Jr. (Papers on this subject are available from the IBM Cambridge Scientific Center, Cambridge MA.) "Programming Styles in NIAL", M.A. Jenkins et al, IEEE Software 3(1):46-55 (Jan 1986). (See Q'NIAL).

NICOL I -

1. Small subset of PL/I by (Massachusetts) Computer Assoc, ca. 1965. Version: NICOL II (1967). Sammet 1969, p.542.

2. ICL, 1968. [same as 1?]

NIKL - Frame language. "Recent Developments in NIKL", T.R. Kaczmarek et al, Proc AAAI-86, 1986.

NIL -

1. New Implementation of Lisp. Intended to be the successor of MacLisp. A large LISP, implemented mostly in VAX assembly language. A forerunner of Common LISP. "NIL: A Perspective", Jon L. White, MACSYMA Users' Conf Proc, 1979.

2. Network Implementation Language. Strom & Yemini, TJWRC, IBM. Implementation of complex networking protocols in a modular fashion. "NIL: An Integrated Language and System for Distributed Programming", R. Strom et al, SIGPLAN Notices 18(6):73-82 (June 1983).

NJCL - Network Job Control Language. "NJCL - A Network Job Control Language", J. du Masle et al, IFIP Congress 1974.

nML - Specification language for instruction sets, based on attribute grammars, for back-end generators. "The nML Machine Description Formalism", M. Freericks <mf@cs.tu-berlin.de> TR TU Berlin, FB20, Bericht 1991/15.

NODAL - Interpreted language implemented on Norsk Data's NORD-10 computers. Used by CERN and DESY high energy physics labs to control their accelerator hardware, PADAC and SEDAC. Included trackball input, graphics.

Noddy - A simple (hence the name) language to handle text and interaction on the Memotech home computer. Has died with the machine.

NOMAD - Database language. "NOMAD Reference Manual", Form 1004, National CSS Inc (Dec 1976). Version: NOMAD2, Must Software Intl.  
list: NOMAD2-L@TAMVM1.BITNET

Nonpareil - One of five pedagogical languages based on Markov algorithms, used in "Nonpareil, a Machine Level Machine Independent Language for the Study of Semantics", B. Higman, ULICS Intl Report No ICSI 170, U London (1968). (cf. Brilliant, Diamond, Pearl[3], Ruby[2]).

NORC COMPILER - Early system on NORC machine. Listed in CACM 2(5):16 (May 1959).

NORD PL - Intermediate language for Norsk Data computers. Sintran III (OS of the ND 10, late 70's) was written in NORD PL. "NORD PL User's Guide", ND-60.047.03.

Nother - Parallel symbolic math.  
info: karhu@cs.umu.se

NPL -

1. New Programming Language. IBM's original (temporary) name for PL/I, changed due to conflict with England's "National Physical Laboratory." MPL and MPPL were considered before settling on PL/I. Sammet 1969, p.542.

2. Burstall, 1977. A predecessor of HOPE. Pattern matching and set comprehensions.

3. NonProcedural Language. 1980. A relational database language. "An Introduction to Nonprocedural Languages Using NPL", T.D. Truitt et al, McGraw-Hill 1983. Versions for Apple ][, MS-DOS.

NPPL - Network Picture Processing Language. Interactive language for manipulation of digraphs. "A Graph Manipulator for On-line Network Picture Processing", H.A. DiGiulio, Proc FJCC 35 (1969).

N-Prolog - Prolog extended with explicit negation. Dov Gabbay, J Logic Programming.

Nqthm - Language[?] used in the Boyer-Moore theorem prover. "Proving Theorems About LISP Functions", R.S. Boyer et al JACM 22(1):129-144 (Jan 1975).

Nroff - Text formatting language/interpreter, based on Unix roff. (See Troff, Groff, RUNOFF.)

NUCLEOL - List processing language, influenced by EOL. J. Nievergelt, Computer J 13(3) (Aug 1970).

Nuprl - (pronounced "new pearl") Nearly Ultimate PRL. Interactive creation of formal mathematics, including definitions and proofs. An extremely rich type system, including dependent functions, products, sets, quotients and universes. Types are first-class citizens. Built on Franz Lisp and Edinburgh ML. "Implementing Mathematics in the Nuprl Proof Development System", R.L. Constable et al, P-H 1986.

NU-Prolog - L. Naish, U Melbourne. A Prolog with 'when' declarations, the successor to MU-Prolog. Type-checked. "NU-Prolog Reference Manual - Version 1.3", J.A. Thom et al eds, TR 86/10, U Melbourne (1988). Available (but not free). (See PNU-Prolog).  
info: jas@mulga.oz.au

NYAP - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

NYU OMNIFAX - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

O2 - ("Object-Oriented"). Object-oriented database language used in the Altair project. Implemented as an interpreter. GIP Altair, Versailles, France. Francois Bancilhon et al, in Advances in Object-Oriented Database Systems, K.R. Dittrich ed, LNCS 334, Springer 1988. (See CO2).

Oaklisp - K. Lang and B. Perlmutter. A portable object-oriented Scheme, syntactically a Scheme superset. Based on generic operations rather than functions. Anonymous classes. "Oaklisp: An Object-Oriented Scheme with First-Class Types", K. Lang et al, SIGPLAN Notices 21(11):30-37 (Nov 1986) (OOPSLA '86).

ftp://f.gp.cs.cmu.edu/usr/bap/oak/fttable  
ftp.cso.uiuc.edu/pub/amiga/fish/ff519 for Amiga

OBE - Office By Example. Moshe Zloof, IBM, early 1980's. Sequel to QBE,

descriptions published but apparently never implemented.

Oberon - Wirth, 1988. A descendant of Modula-2 eliminating many things: variant records, enumeration types, subranges, lower array indices and 'for' loops. Additions are extensible record types, multidimensional open arrays and garbage collection. "The Programming Language Oberon", N. Wirth, *Soft Prac & Exp* 18(7):671-690 (July 1988). "Programming in Oberon: Steps Beyond Pascal and Modula", M. Reiser & N. Wirth, A-W 1992.  
ftp://neptune.inf.ethz.ch for MacII, MS-DOS  
//wuarhive.wustl.edu/mirrors/msdos/pgmutl/oberonml1.zip for MS-DOS  
//ftp.cso.uiuc.edu/pub/amiga/fish/ff380 for Amiga

Oberon-2 - H. Moessenboeck, 1991. A superset of Oberon-1 to include object-orientation. A redesign of Object Oberon. Type-bound procedures (equivalent to methods), read-only export of variables and record fields, open array variables, and a 'with' statement with variants. The 'for' statement is reintroduced. Second Intl Modula-2 Conf, Sept 1991.  
ftp://neptune.inf.ethz.ch/pub/Oberon/\* Version 4  
doc: //neptune.inf.ethz.ch/pub/Oberon/Docu/Oberon2Report.ps.Z

Oberon-V - (formerly Seneca). R. Griesemer, 1990. Descendant of Oberon designed for numerical applications on supercomputers, especially vector or pipelined architectures. Includes array constructors and an ALL statement (like a parallel for loop). For the Cray Y-MP. "Seneca - A Language for Numerical Applications on Vectorcomputers", Proc CONPAR 90 - VAPP IV Conf. "A Programming Language for Vector Computers", R. Griesemer, Diss Nr. 10277, ETH Zurich, 1993.

OBJ - Joseph Goguen 1976. A family of declarative "ultra high level" languages. Abstract types, generic modules, subsorts (subtypes with multiple inheritance), pattern-matching modulo equations, E-strategies (user control over laziness), module expressions (for combining modules), theories and views (for describing module interfaces). For the massively parallel RRM (Rewrite Rule Machine). "Higher-Order Functions Considered Unnecessary for Higher-Order Programming", J.A. Goguen, in *Research Topics in Functional Programming*.

OBJ0 - Tardo. Based on unsorted equational logic.  
OBJT - Tardo. Error algebras plus an image construct.  
OBJ1 -

OBJ2 - Clear-like parametrized modules. A functional system based on equations. "Principles of OBJ2", K. Futatsugi et al, 12th POPL, ACM 1985, pp.52-66.

OBJ3 - SRI. Based on order-sorted rewriting. Agent-oriented.  
"Introducing OBJ3", J. Goguen et al, SRI-CSL-88-9, SRI Intl (1988). Runs on AKCL.  
info: obj3sys@crl.sri.com  
obj3dist@csl.sri.com

Object CHILL - "Object CHILL - An Object Oriented Language for Systems Implementation", J. Winkler et al, ACM Comp Sci Conf 1992, pp.139-147.

Object Lisp - LMI. An object-oriented Lisp. "ObjectLISP User Manual", G. Dreschere, LMI 1987.

ObjectLOGO - A variant of LOGO with object-oriented extensions. Lexical

scope. Version 2.6, for the Mac. Paradigm Software  
<paradigm@applelink.apple.com> (617)576-7675.

Object Oberon - H. Moessenboeck & J. Templ, 1989. Adds classes and methods to Oberon. "Object Oberon - An Object-Oriented Extension of Oberon", H. Moessenboeck et al, ETH TR 109 (Apr 1990). "Object Oberon - A Modest Object-Oriented Language", H. Moessenboeck & J. Templ, in Structured Programming 10(4), 1989. (See Oberon-2).

Object-CHILL - Proposed object-oriented extension of CHILL. G. Diesl et al, "Object-CHILL: The Road to Object Oriented Programming with CHILL", CHILL - CCITT High Level Language, Proceedings of the 5th CHILL Conference, Rio de Janeiro, Brazil, March 1990, pp.135-142.

Object-COBOL - Micro Focus. Largely compatible, but a subset of, the proposed object-oriented COBOL standard. "Object-COBOL: Object Orientation for Business", Raymond Obin, Micro Focus Press.

Object-Oriented Turing - R.C. Holt <holt@csri.toronto.edu>, U Toronto, 1991. Extension of Turing, and a replacement for Turing Plus. Imperative, object-oriented, concurrent. Modules, classes, single inheritance, processes, exception handling, optional machine-dependent programming. Integrated environment under X windows. Described in "A Conceptual Framework for Software Development", Mancoridis et al, eds, ACM SIGSCE Conference, Feb 1993, Indianapolis. Versions for Sun4, MIPS, RS-6000, etc. "Turing Reference Manual", 1992, ISBN 0-921598-15-7.  
ftp://128.100.1.192 (demo version)  
info: <ootinfo@turing.toronto.edu>

ObjectPAL - Object-oriented database language, part of Borland's MS-Windows version of Paradox.

Object Pascal - Developed jointly by Apple Computer and Niklaus Wirth. An object-oriented Pascal. "Object Pascal Report", Larry Tesler, Structured Language World 9(3):10-17 (1985).

Object-Z - U Queensland. "The Object-Z Specification Language: Version 1", TR 91-1, Software Verification Res Ctr, U Queensland, 1991. "Object-Z", G. Rose, in Object Orientation in Z, S. Stepney et al eds, Springer 1992.  
fp: ftp.cs.uq.oz.au:/pub/SVRC/techreports/tr91-1.ps.Z

Objective C - Brad Cox, Productivity Products. An object-oriented superset of ANSI C, incorporating many ideas from Smalltalk. Implemented as a preprocessor for C. No operator overloading, no multiple inheritance, no class variables. Does have run-time binding. Used as the system programming language on the NeXT. "Object-Oriented Programming: An Evolutionary Approach", Brad Cox, A-W 1986. Versions for MS-DOS, Macs, VAX/VMS and Unix workstations. Language versions by Stepstone, NeXT and GNU are slightly different. Stepstone Corp, (203) 426-1875.

Objlog - CNRS, Marseille. Frame-based language combining objects and Prolog II. "The Inheritance Processes in Prolog", C. Chouraki et al, GRTC/187bis/Mars 1987 (CNRS).  
info:somebody@grtc.cnrs-mrs.fr

ObjVlisp - 1984. An object-oriented extension of Vlisp. Reflective architecture. "Metaclasses are First Class: The ObjVlisp Model", P.

Cointe, SIGPLAN Notices 22(121):156-167 (Dec 1987) (OOPSLA '87).

ObjVProlog - Logic programming and object-orientation, an adaptation of the ObjVlisp model to Prolog. "ObjVProlog: Metaclasses in Logic", J. Malenfant, ECOOP '89, Cambridge U Press 1989, pp.257-269.

Obliq - Luca Cardelli, 1993. A distributed object-oriented scripting language. Small, statically scoped, untyped, higher order, and concurrent. State is local to an address space, while computation can migrate over the network. The distributed computation mechanism is based on Modula-3 network objects.

ftp://gatekeeper.dec.com/pub/DEC/Modula-3/contrib  
http://www.research.digital.com/SRC/Obliq/Obliq.html

Oblog - Object-oriented extension to Prolog. Small, portable.  
info: Margaret McDougall, EdCAAD, Dept Arch, U Edinburgh, EH1 1JZ.

OBSCURE - "A Formal Description of the Specification Language OBSCURE", J. Loeckx, TR A85/15, U Saarlandes, Saarbrucken, 1985.

Oc - ("Oh see!") Parallel logic language. "Self-Description of Oc and its Applications", M. Hirata, Proc 2nd Natl Conf Japan Soc Soft Sci Tech, pp.153-156 (1984).

OCAL - On-Line Cryptanalytic Aid Language. "OCAS: On-line Cryptanalytic Aid System", D.J. Edwards, MAC-TR-27, MIT Project MAC, May 1966. Sammet 1969, p.642.

occam - (named for the English philosopher William of Occam (1300-1349)) Now known as "occam 1". David May et al, 1982. Concurrent algorithms, based on CSP and EPL. Designed for the INMOS transputer and vice versa. Expressions are processes, which may be combined in serial and parallel. Processes communicate via named unidirectional channels. There is no operator precedence. "Occam", D. May, SIGPLAN Notices 18(4):69-79 (1983).  
ftp://watserv1.waterloo.edu simulator for VAX, Tahoe  
list: occam@sutcase.case.syr.edu

occam 2 - 1987. An extension of occaml. Occam 2 adds floating point, functions and a type system. "occam 2 Reference Manual", INMOS, P-H 1988, ISBN 0-13-629312-3.  
ftp://ftp.ecs.soton.ac.uk/pub/occam/spoc1.1 portable simulator for Unix

occam 3 - under development

OCL - Operator Control Language. Batch language for the IBM System/36, used specifically with the RPG II compiler. (See CL).

OCODE - Assembly language for a stack-based virtual machine, used as the intermediate language of the Cambridge BCPL compiler. "The Portability of the BCPL Compiler", M. Richards, Soft Prac & Exp 1(2) (1971).

Octave - High-level language primarily for numerical computations. Real and complex scalars and matrices, solution nonlinear algebraic equations, ordinary differential equations. Implemented in g++ and Fortran.  
ftp://ftp.che.utexas.edu/pub/octave/octave-1.0.tar.gz

odl - Fine-grained active object oriented design/programming language.

Part of the Diamonds project for software engineering on heterogeneous distributed systems.

[http://www.cat.syr.edu/daimonds\\_home.html](http://www.cat.syr.edu/daimonds_home.html)

<ftp://www.cat.syr.edu/pub/daimonds/release1.0.tar>

info: Gary L. Craig <craig@cat.syr.edu>

OIL -

1. "The Architecture of the FAIM-1 Symbolic Multiprocessing System", A. Davis et al, 9th Intl Joint Conf in Artif Intell, 1985, pp.32-38.

2. Operator Identification Language. Used for overload resolution by the Eli compiler-writing system.

OISC - One Instruction Set Computer. Assembly language for a machine based on the single instruction "subtract and jump if negative", constructed as a proof of principle by Ross Cuniff <cuniff@hpross.fc.hp.com>. Emulator with source code, includes a macro assembler and nontrivial example programs.  
<ftp://locke.ccil.org/pub/retro/oisc.shar.gz>

OLC - On-Line Computer system. UCSB ca. 1966. Predecessor of Culler-Fried System. Sammet 1969, p.253.

OLDAS - On-line Digital Analog Simulator. Interactive version of MIMIC, for IBM 360. "OLDAS: An On-line Continuous System Simulation Language", R.P. Cullen, in Interactive Systems for Experimental Applied Mathematics, A-P 1968.

OLGA - Ouf! un Langage pour les Grammaires Attribuees. Inria, 1985. Language for specification of attribute grammars, used as the input language of the compiler writing system FNC-2. Applicative, strongly typed, polymorphic, pattern-matching, modules.

Omega - Prototype-based object-oriented language. Austria. "Type-Safe Object-Oriented Programming with Prototypes - The Concept of Omega", G. Blaschek, Structured Programming 12:217-225 (1991).

OMNICODE - Thompson, 1956. Ran on IBM 650. Sammet 1969, p.5.

OMNIFAX - Alternate name for NYU OMNIFAX? Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

OMNITAB - Statistical analysis and desk calculator. "OMNITAB II User's Reference Manual", NBS Tech Note 552 (Oct 1971). Sammet 1969, pp.296-299. Version: OMNITAB II.

Ondine - "Concurrency Introduction to an Object-Oriented Language System Ondine", T. Ogihara et al, 3rd Natl Conf Record A-5-1, Japan Soc for Soft Sci Tech, Japan 1986.

Ontic - Object-oriented language for an inference system. LISP-like appearance, but based on set theory. "Ontic: A Knowledge Representation System for Mathematics", D.A. McAllester, MIT Press 1989.

OO-CHILL - Proposed object-oriented extension to CHILL. A. Scortese, "OO-CHILL: Integrating the Object Paradigm Into CHILL", CHILL - CCITT High Level Language, Proceedings of the 5th CHILL Conference, Rio de Janeiro, Brazil,

March 1990, pp.127-133.

OOF - Object-Oriented Fortran. Data items can be grouped into objects, which can be instantiated and executed in parallel. Available now for Suns, Iris, iPSC, soon for nCUBE.  
info: dreese@erc.msstate.edu

OOPS - "OOPS: A Knowledge Representation Language", D. Vermeir, Proc 19th Intl Hawaii Conf on System Sciences, IEEE (Jan 1986) pp.156-157.

OOT - Object-oriented Turing.

OOZE - Object oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

OPAL-0 - Predecessor of OPAL[5].

OPAL -

1. DSP language. "OPAL: A High Level Language and Environment for DSP boards on PC", J.P. Schwartz et al, Proc ICASSP-89, 1989.

2. Previous name of Smalltalk DB.

3. Simulation language with provision for stochastic variables. An extension of Autostat. "C-E-I-R OPAL", D. Pilling, Internal Report, C.E.I.R. Ltd (1963).

4. Language for compiler testing said to be used internally by DEC.

5. Technical University of Berlin. Strongly-typed, higher-order, strict applicative language, with algebraic specification. Uses parameterized structures rather than polymorphism. "OPAL: Design and Implementation of an Algebraic Programming Language", available by ftp.

info: opal@cs.tu-berlin.de

ftp://ftp.cs.tu-berlin.de/pub/local/uebb/ocs

http://www.cs.tu-berlin.de/~opal/

O-plan - Distributed language.

http://www.aiai.ed.ac.uk/~oplan/

OPS -

1. On-line Process Synthesizer. M. Greenberger, MIT ca. 1964. Discrete simulation under CTSS. Sammet 1969, p.660. Versions: OPS-3, OPS-4. "On-line Computation and Simulation: The OPS-3 System", M. Greenberger et al, MIT Press 1965.

2. Official Production System. CMU, 1970. The first production-system (i.e. rule-based) programming language, used for building expert systems. Written originally in Franz Lisp, later ported to other LISP dialects.

OPS5 - Charles L. Forgy. 1977 version of OPS[2], publicly available from the author <forgy@cs.cmu.edu>. First implemented in Lisp, later in BLISS. "Programming Expert Systems in OPS5", L. Brownston et al, A-W 1985. "An OPS5 Primer", Sherman et al, comes with OPS5 for DOS. Other versions: OPS4, OPS5+, OPS83. Inference Engine Tech, Cambridge MA.

ftp://wuarchive.wustl.edu/mirrors/unix-c/languages/ops5 an OPS5 interpreter in Common LISP.

C5 - An OPS5 implementation in C. "Rule-Based Programming in the Unix System", G.T. Vesonder, AT&T Tech J 67(1), 1988.

OPTRAN - R. Wilhelm, U Saarlandes, early 1980's. Specification language for attributed tree transformation. "POPSY and OPTRAN Manual", ESPRIT PROSPECTRA Project Item S.1.6-R.3.0, U Saarlandes (Mar 1986).

Orca - Vrije Universiteit, Amsterdam, 1986. Similar to Modula-2, but with support for distributed programming using shared data objects, like Linda. A 'graph' data type removes the need for pointers. Version for the Amoeba OS, comes with Amoeba. "Orca: A Language for Distributed Processing", H.E. Bal <bal@cs.vu.nl> et al, SIGPLAN Notices 25(5):17-24 (May 1990).

OREGANO - "On the Design and Specification of the Programming Language OREGANO", D.M. Berry. UCLA-ENG-7388, 1973.

Orient84/K - Y. Ishikawa, Keio U, Yokohama. "A Concurrent Object-Oriented Knowledge Representation Language Orient84/K", Y. Ishikawa et al, SIGPLAN Notices 21(11):232-241 (OOPSLA '86) (Nov 1986).

ORTHOCARTAN - A. Krasinski, Warsaw, early 80's. Symbolic math, especially General Relativity.

Orwell - Lazy functional language, Miranda-like. List comprehensions and pattern matching. "Introduction to Orwell 5.00", P.L. Wadler et al, Programming Research Group, Oxford U, 1988.

OSCAR -

1. Oregon State Conversational Aid to Research. Interactive numerical calculations, vectors, matrices, complex arithmetic, string operations, for CDC 3300. "OSCAR: A User's Manual with Examples", J.A. Baughman et al, CC, Oregon State U.

2. Object-oriented language used in the COMANDOS Project. "OSCAR: Programming Language Manual", TR, COMANDOS Project, Nov 1988.

O'small - Small object-oriented language intended for teaching.  
ftp:cs.uni-sb.de:/pub/osmall/machine/\*

OSQL - Object-oriented Structured Query Language. Functional language, superset of SQL, used in Hewlett-Packard's OpenODB database system.  
info: boronkay@cup.hp.com

OSSL - Operating Systems Simulation Language. "OSSL - A Specialized Language for Simulating Computer Systems", P.B. Dewan et al, Proc SJCC 40, AFIPS (Spring 1972).

Ottawa Euclid - Variant of Euclid.

OWHY - Early functional language? "A Type-Theoretical Alternative to CUCH, ISWIM, OWHY", Dana Scott, Oxford U 1969.

OWL - Original name of Trellis.

Ox - Language for specification of attribute grammars. "User Manual for Ox: An Attribute-Grammar Compiling System based on Yacc, Lex and C", K.M. Bischoff, TR92-30, Iowa State U, Dec 1992.  
info:ox-request@cs.iastate.edu

Oz - U Saarbrucken. Object-oriented concurrent constraint language. Based on constraint communication, a new form of asynchronous communication using logic variables. Partial information about the values of variables is imposed concurrently and incrementally. Supports higher order programming and object-orientation including multiple inheritance. "Object-Oriented Concurrent Constraint Programming in Oz", G. Smolka et al.  
ftp://duck.dfki.uni-sb.de/pub/papers

P+ - "Experience with Remote Procedure Calls in a Real-Time Control System", B. Carpenter et al, Soft Prac & Exp 14(9):901-907 (Sep 1984).

P4 - Rusty Lusk <lusk@anta.mcs.anl.gov>. A macro/subroutine package for parallel programming, using monitors on shared memory machines, message passing on distributed memory machines. Implemented as a subroutine library for C and Fortran. An enhancement of the "Argonne macros", PARMACS.  
ftp://info.mcs.anl.gov/pub/p4t1.2.tar.Z  
info: p4@mcs.anl.gov

PABC - Intermediate language recognized by the Parallel ABC machine, used in the implementation of Concurrent Clean. "The PABC Simulator", E.G.J.M.H. NM-^Zecker, TR 89-19, U Nijmegen 1989.

PACT I - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).  
Version: PACT IA for IBM 704.

PACTOLUS - Digital simulation. Sammet 1969, p.627.

Paddle - Language for transformations leading from specification to program. Used in POPART, a grammar-driven programming environment generator. "POPART: Producer of Paddles and Related Tools, System Builders' Manual", D.S. Wile TR RR-82-21, ISI, Marina del Rey, CA 1982.

PAF - Programmation Automatique des Formules. Dmitri Starynkevitch, 1957. Early language with resemblance to FORTRAN II or BASIC. Papers in French journal RAIRO, 1961-1963? Implemented for the S.E.A. CAB500 computer (France).

PAGE - Typesetting language. "Computer Composition Using PAGE-1", J.L. Pierson, Wiley 1972.

PaiLisp - Parallel Lisp built on Scheme. 1986. "A Parallel Lisp Language PaiLisp and its Kernel Specification", T. Ito et al, in Parallel Lisp: Languages and Systems, T. Ito et al eds, LNCS 441, Springer 1989.

PAISley - Bell Labs. Operational specification language. "An Operational Approach to Requirements Specification for Embedded Systems", P. Zave, IEEE Trans Soft Eng SE-8(3):250-269 (May 1982).

PAL -

1. Paradox Application Language. Language for Paradox, Borland's

relational database.

2. For the AVANCE distributed persistent OS. "PAL Reference Manual", M. Ahlsen et al, SYSLAB WP-125, Stockholm 1987. "AVANCE: An Object Management System", A. Bjornerstedt et al, SIGPLAN Notices 23(11):206-221 (OOPSLA '88) (Nov 1988).

3. Object-oriented Prolog-like language. "Inheritance Hierarchy Mechanism in Prolog", K. Akama, Proc Logic Prog '86, LNCS 264, Springer 1986, pp.12-21.

4. PDP Assembly Language. Assembly language for PDP-8 and PDP-11.

5. Pedagogic Algorithmic Language. "PAL - A Language for Teaching Programming Linguistics", A. Evans Jr, Proc ACM 23rd Natl Conf, Brandon/Systems Press (1968).

Pam - Toy ALGOL-like language used in "Formal Specification of Programming Languages: A Panoramic Primer", F.G. Pagan, P-H 1981.

Pandora - Parlog extended to allow "don't-know" non-determinism. "Pandora: Non-Deterministic Parallel Logic Programming", R. Bahgat et al, Proc 6th Intl Conf Logic Programming, MIT Press 1989 pp.471-486.

PANON - A family of pattern-directed string processing languages based on generalized Markov algorithms. "String Processing Languages and Generalized Markov Algorithms", A. C. Forino, Proc IFIP Working Conf on Symb Manip Languages, pp.141-206, Amsterdam 1968. PANON-1, based on Simple GMA's and PANON-2 based on Conditional Functional GMA's.

Paragon - Mark Sherman. IEEE Software (Nov 1991). [?]

Paralation - PARAllel reLATION. Sabot, MIT 1987. A framework for parallel programming. A "field" is an array of objects, placed at different sites. A paralation is a group of fields, defining nearness between field elements. Operations can be performed in parallel on every site of a paralation. "The Paralation Model: Architecture Independent Programming", G.W. Sabot <gary@think.com>, MIT Press 1988.

Paralation LISP - Embeds the paralation model in Common LISP. Available from MIT Press, (800)356-0343.

Paralation C - Paralation embedded in C. Under development.

ParAlfl - Hudak, Yale. Parallel functional language, a superset of Alfl. Used by the Alfalfa system on Intel iPSC and Encore Multimax. "Para-Functional Programming", P. Hudak, Computer 19(8):60-70 (Aug 1986). "Alfalfa: Distributed Graph Reduction on a Hypercube Multiprocessor", B. Goldberg & P. Hudak, TR, Yale U, Nov 1986.

Parallaxis - U Stuttgart. Data-parallel (SIMD) language, based on Modula-2. "User Manual for Parallaxis Version 2.0", T. Braunl, U Stuttgart. Simulator for workstations, Mac and PC.  
ftp://ftp.informatik.uni-stuttgart.de/pub/parallaxis  
info: Thomas Braunl <braunl@informatik.uni-stuttgart.de>

Parallel C - Never implemented, but influenced the design of C\*.

Parallel FORTH - For the MPP.

Parallel Pascal - Data-parallel language, similar to Actus and Glypnir.  
"Parallel Pascal: An Extended Pascal for Parallel Computers", A. Reeves, J  
Parallel Dist Computing 1:64-80 (1984).

Parallel SML - "Parallel SML: A Functional Language and its Implementation  
in Dactl", Kevin Hammond, Pitman Press 1990.

Parasol - Parallel Systems Object Language. Object-oriented, supports  
network and parallel computing. Modules, exceptions. "The Parasol  
Programming Language", R. Jervis <hjervis!rbj@uunet.uu.net>, Dr Dobbs J,  
Oct 1993, pp.34-41.  
ftp://ftp.uu.net/languages/parasol/\*

Pari - Symbolic math, especially number theory. Version 1.37 for Unix,  
Macintosh, MS-DOS, OS/2.  
info: <pari@alioth.greco-prog.fr>  
ftp://math.ucla.edu/pub/pari/\*

Paris - PARallel Instruction Set. Low-level language for the Connection  
Machine.

Parlance - Concurrent language. "Parallel Processing Structures:  
Languages, Schedules, and Performance Results", P.F. Reynolds, PhD Thesis,  
UT Austin 1979.

Parlog - Clark & Gregory, Imperial College 1983. An AND-parallel Prolog,  
with guards and committed choice [=don't care] nondeterminism. Shallow  
backtracking only. "Parlog: A Parallel Logic Programming Language", K.L.  
Clark and S. Gregory, Imperial College, London, May 1983. ("Parlog83", in  
which the output mechanism was assignment). "Parallel Logic Programming in  
PARLOG, The Language and Its Implementation", S. Gregory, A-W 1987.  
("Parlog86", in which the output mechanism was unification, as in GHC).  
(See Strand). Implementations: MacParlog and PC-Parlog from Parallel Logic  
Programming Ltd, Box 49 Twickenham TW2 5PH, UK.  
ftp://ftp.inria.fr/lang/Parlog.tar.Z  
info: parlog@doc.ic.ac.uk

Parlog++ - Andrew Davison <ad@cs.mu.oz.au>, then Imperial College now U  
Melbourne. Object orientation plus parallel logic, built on top of  
MacParlog. "Parlog++: A Parlog Object-Oriented Language", A. Davison,  
Parlog Group, Imperial College 1988. Sold by PLP Ltd.  
info: parlog@doc.ic.ac.uk

PARMACS - Argonne Natl Lab. The "Argonne macros". A package of macros  
written in m4 for portable parallel programming, using monitors on shared  
memory machines, and message passing on distributed memory machines. E.  
Lusk et al, "Portable Programs for Parallel Processors", HRW 1987. (See  
p4.)  
ftp://netlib.att.com/parmacs

ParMod - "Parallel Programming with ParMod", S. Eichholz, Proc 1987 Intl  
Conf on Parallel Proc, pp.377-380.

PARSEC - Extensible language with PL/I-like syntax, derived from PROTEUS.

"PARSEC User's Manual", Bolt Beranek & Newman (Dec 1972).

Parsley - Barber, Summit Software. A Pascal extension for construction of parse trees. Iterators. "PARSLEY: A New Compiler-Compiler", in Software Development Tools, Techniques and Alternatives, Arlington VA, Jul 1983, pp.232-241.

PARTS - Digital. Visual language for OS/2 2.0.

PARULEL - "The PARULEL Parallel Rule Language", S. Stolfo et al, Proc 1991 Intl Conf Parallel Proc, CRC Press 1991, pp.36-45.

Pascal - (named for the French mathematician Blaise Pascal (1623-1662)) N. Wirth, ca. 1970. Designed for simplicity, in reaction to the complexity of ALGOL 68, and intended as a teaching language. Innovations: enumeration types, subranges, sets, variant records, case statement. Pascal has been extremely influential in programming language design, and has led to a great number of variations and descendants. "The Programming Language Pascal", N. Wirth, Acta Informatica 1:35-63 (1971). "PASCAL User Manual and Report", K. Jensen & N. Wirth, Springer 1975 made significant revisions to the language.

ANSI/IEEE770X3.97-1993, very similar to ISO Pascal, but does not include conformant arrays.

BS 6192, "Specification for Computer Programming Language Pascal", British Standards Institute 1982.

ISO 7185-1983(E). Level 0 and Level 1. Changes from Jensen & Wirth's Pascal include: name equivalence; names must be bound before they are used; loop index must be local to the procedure; formal procedure parameters must include their arguments; conformant array schemas.

Pascal++ - ISO, 1994. An extension of Extended Pascal, inspired by Pascal Plus. Adds concurrency, exceptions and object orientation, including virtual types and multiple inheritance.

ftp://ftp.tdr.dk/pub/pplus

info: pplus@tfl.dk

Pascal- - Pascal subset used in Brinch Hansen on Pascal Compilers, P. Brinch Hansen, P-H 1985.

Pascal-2 - [?]

Pascal-80 - A successor of Platon. Developed at RC International for systems programming. Later it was renamed Real-Time Pascal. "PASCAL80 Report", J. Staunstrup, RC Intl, Denmark Jan 1980.

Pascal+CSP - "Pascal+CSP, Merging Pascal and CSP in a Parallel Processing Oriented Language", J. Adamo, Proc 3rd Intl Conf Distrib Comp Sys, IEEE 1982, pp.542-547.

Pascal-F - Pascal extended to include fixed-point arithmetic. E. Nelson, "Pascal-F: Programming Language for Real-Time Automotive Control", IEEE ElectroTechnol. Rev. (USA), 2:39, 1968.

Pascal-FC - Derived from Pascal-S, provides several types of concurrency: semaphores, monitors, both occam/CSP-style and Ada-style rendezvous. "The Teaching Language Pascal-FC", G.L. Davies et al, Computer J 33(2):147-154 (Apr 1990).

Pascal/L - A SIMD parallel extension of Pascal. "Implementation of an Array and Vector Processing Language", C. Fernstrom, Intl Conf Parallel Proc, IEEE, pp.113-127 (1982)

Pascal-Linda - Ian Flockhart, U Edinburgh, 1991. Under development.

Pascal-m - "Pascal-m: A Language for Loosely Coupled Distributed Systems", S. Abramsky et al in Distributed Computing Systems, Y. Paker et al eds, Academic Press 1986, pp.163-189.

Pascal-P - Variant of Pascal used by the UCSD p-system environment. Extended string and array operations, random access files, separate compilation, etc. Available from Pecan.

Pascal Plus - Jim Welsh & D. Bustard, Queens U, Belfast. Pascal with extensions for object-oriented multiprogramming, uses an 'envelope' construct for both packages and classes. "Pascal Plus - Another Language for Modular Multiprogramming", J. Welsh et al, Soft Prac & Exp 9:947 (1979). "Sequential Program Structures", J. Welsh et al, P-H 1984, ISBN 0-13806828-3.

Pascal/R - Pascal with relational database constructs added. The first successful integrated database language. "Pascal/R Report", J.W. Schmidt et al, U Hamburg, Fachbereich Informatik, Report 66, Jan 1980.

Pascal-S - Simplified Pascal. June, 1975. A strict subset of Pascal, omits scalar types, subranges, sets, files, pointers, packed structures, 'with' and 'goto'. Source for a complete Pascal-S compiler is in "Pascal-S: A Subset and Its Implementation", N. Wirth in Pascal - The Language and Its Implementation, by D.W. Barron, Wiley 1979.  
<ftp://csseq.cs.tamu.edu/mcguire/pascal-s>

Pascal-SC - ESPRIT DIAMOND Project. An extension of Pascal for numerical analysis, with controlled rounding, overloading, dynamic arrays and modules. "PASCAL-SC, A Computer Language for Scientific Computation", G. Bohlender et al, Academic Press 1987.

Pasqual - "Pasqual: A Proposed Generalization of Pascal", R.D. Tennent, TR75-32, Queen's U, Canada, 1975.

PASSIM - Simulation language based on Pascal. "PASSIM: A Discrete-Event Simulation Package for Pascal", D.H Uyeno et al, Simulation 35(6):183-190 (Dec 1980).

PASRO - PASCAL for ROBots. "PASRO - Pascal for Robots", C. Blume et al, Springer 1985.

PAT -

1. Personalized Array Translator. Small subset of APL. Sammet 1969, p.252.

2.  
[info:oed@watsol.uwaterloo.ca](mailto:info:oed@watsol.uwaterloo.ca)

Path Pascal - Parallel extension of Pascal. Processes have shared access

to data objects. Constraints on their synchronization are specified in a path expression. "An Overview of Path Pascal's Design", R.H. Campbell, SIGPLAN Notices 15(9):13-24 (Sep 1980).

PC - Parallel C. U Houston. Extensions to C providing a shared memory SIMD model on message passing machines.

ftp://karazm.math.uh.edu/pub/Parallel/Tools/pc.1.1.1.tar.Z

info: Ridgway Scott <scott@uh.edu>

pC++ - Data parallel extension to C++. Classes and methods for managing distributed collections. "Distributed pC++: Basic Ideas for an Object Parallel Language", F. Bodin et al, Proc Supercomput 91, ACM SIGARCH, pp.273-282.

info: Dennis Gannon <gannon@cs.indiana.edu>

PCF - Simply typed, functional. "Fully Abstract Translations Between Functional Languages", J. Riecke, 18th POPL, pp.245-254 (1991). "LCF Considered as a Programming Language", Theor CS 5:223 (1977).

PCL -

1. Printer Control Language. Document description language used by Hewlett-Packard Laserjet printers, a superset of HP-GL/2. PCL 5 Printer Language Printer Technical Reference Manual, HP 33459-90903. Versions: PCL 3, PCL 5.

2. Portable CommonLoops. Started out as an implementation of CommonLoops. Is now being converted to CLOS, but currently implements only a subset of the CLOS specification.

3. Peripheral Conversion Language. Honeywell. Command language for file transfer between I/O devices on the CP-V and CP-6 operating systems.

4. "PCL - A Process Oriented Job Control Language", V. Lesser et al, Proc 1st Intl Conf Distrib Comp Sys, IEEE 1979, pp.315-329.

5. Programmable Controller Language. Arthur Duncan and John Hutchinson, GE, 1981. A partial implementation of Ada for numerical control applications. Delivered as part of the GE Mark Century 2000. Perhaps the first known example of the use of Ada's features in an embedded system.

PCLIPS - Parallel CLIPS - U Lowell. Concurrent independent CLIPS expert systems. They use 'rassert' (remote assert) to enter facts into each other's database. "PCLIPS: A Distributed Expert System Environment", R. Miller, CLIPS Users Group Conf, Aug 1990.

info: dragon.ulowell.edu:pub/Pclips

PCN - Program Composition Notation. Specification language for parallelism between C and Fortran modules. "Productive Parallel Programming: The PCN Approach", I. Foster et al, Sci Prog 1(1):51-66 (1992).

ftp://info.mcs.anl.gov/pub/pcn

info: Ian Foster <pcn@mcs.anl.gov>

P-code - The intermediate code produced by the Pascal-P compiler. Assembly language for a hypothetical stack machine, the P-machine, said to have been an imitation of the instruction set for the Burroughs Large System. The term was first used in Algorithms + Data Structures = Programs, N. Wirth,

P-H 1976. Byte articles on writing a Pascal Compiler in Northstar BASIC (ca Aug 1978) also used the term. Later used in Apple Pascal, and as the intermediate language in the UCSD P-system. "A Comparison of PASCAL Intermediate Languages", P.A. Nelson, SIGPLAN Notices 14(8):208-213 (Aug 1979). Variants: P2 P-code, P4 P-code, UCSD P-code, LASL P-code.  
info: USUS, Box 1148, La Jolla, CA 92038

PC-TILES - A visual language. [?]

PDEL - Partial Differential Equation Language. Preprocessor for PL/I. "PDEL - A Language for Partial Diferential Equations", A.F. Cardenas, CACM 13(3):184-191 (Mar 1970).

PDELAN - Partial Differential Equation LANguage. "An Extension of FORTRAN Containing Finite Difference Operators", J. Gary et al, Soft Prac & Exp 2(4) (Oct 1972).

PDIL - Agence d'Informatique, France, 1970's. Language for description of communication protocols, part of the RHIN project. [?]

PDL2 - Process Design Language. Developed for the TI ASC computer. "Texas Instruments Process Design Methodology - Design Specification: Process Design Language", Volume I (Sep 1976). Mentioned in "An Overview of Ada" J.G.P. Barnes, Soft Prac & Exp 10:851-887 (1980).

PDS/MaGen - Problem Descriptor System. Generation of matrices and reports for mathematical programming and operations research. "PDS MaGen User Information Manual", Haverly Systems (Dec 1977).

PEARL -

1. Constable, Cornell U, 80's. Constructive mathematics.

2. Process and Experiment Automation Real-Time Language. A real-time language for programming process control systems, widely used in Europe. Size and complexity exceeds Ada. DIN 66253 Teil 2, "Programmiersprache PEARL", Beuth-Verlag, Nov 1980.

3. One of five pedagogical languages based on Markov algorithms, used in "Nonpareil, a Machine Level Machine Independent Language for the Study of Semantics", B. Higman, ULICS Intl Report No ICSI 170, U London (1968). (cf. Brilliant, Diamond, Nonpareil, Ruby[2]).

4. Brian Randell, ca 1970. Multilevel language, mentioned in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974. [?]

Pebble - Polymorphic. "A Kernel Language for Abstract Data Types and Modules", R.M. Burstall & B. Lampson, in Semantics of Data Types, LNCS 173, Springer 1984.

Pebbleman - Jul 1978, revised Jan 1979. DoD requirements that led to APSE.

PECOS - Constraint-based language, built on the object-oriented module of Le-Lisp. "Pecos Reference Manual", ILOG, 1990. ILOG, 12 av Raspail, BP 7, F94251 Gentilly, France.

PEEL - Used to implement version of EMACS on PRIME computer. [?]

PEF - PowerPC Executable Format. Binary object code format used by Apple.

PENCIL - Pictorial ENCodIng Language. On-line system to display line structures. Sammet 1969, 675.

Pepper - Chris Dollin <kers@hplb.hpl.hp.com>. Variant of POP-11.

PEPsy - Prolog extended with parallel modules within which explicit OR-parallelism can be used. "PEPsy: A Prolog for Parallel Processing", M. Ratcliffe et al, ECRC TR CA-17, 1986.

Perl - Practical Extraction and Report Language. Larry Wall <lwall@netlabs.com> An AWK-like interpreted language for scanning text and printing formatted reports. Regular expression primitives, dynamically-scoped variables and functions, extensible runtime libraries, exception handling, packages. Version 5 adds nested data structures and object-oriented features. "Programming Perl", Larry Wall et al, O'Reilly & Assocs.

ftp://ftp.netlabs.com/pub/outgoing/perl.4.0 for Unix, MS-DOS, Amiga

//ftp.netlabs.com/pub/outgoing/perl5.0/perl5a1.tar.Z for Sparc

//rascal.utexas.edu/programming/Perl\_402\_MPW\_CPT\_bin for Mac

uucp: osu-cis

PFL -

1. Persistent Functional Language. A. Poulovasslis, Kings College and Carol Small, Birkbeck College. A functional database language. "A Functional Programming Approach to Deductive Databases", A. Poulovasslis et al, Proc VLDB '91.

info: Alex Poulovasslis <alex@dcs.kcl.ac.uk>

2. Parallel Functional Language. Soeren Holmstrom, Matthews, Chalmers U, early 1980's. The first concurrent extension of ML using CCS. (More recent ones: Poly/ML, Concurrent ML and Facile.) "PFL: A Functional Language for Parallel Programming", S. Holmstrom in Proc Declarative Language Workshop, London 1983, pp.114-139.

Pfortran - Parallel Fortran. U Houston. Extensions to Fortran providing a shared memory SIMD model on message passing machines. Under development.

"Pfortran: A Parallel Dialect of Fortran", L.R. Scott, Fortran Forum 11(3):20-31 (Sep 1992).

info: Ridgway Scott <scott@uh.edu>

pH - Parallel Haskell. A parallel variant of Haskell incorporating ideas from Id and Sisal. Under development.

list: pH@lcs.mit.edu

PHOCUS - Object-oriented Prolog-like language. "PHOCUS: Production Rules, Horn Clauses, Objects and Contexts in a Unification Based System", D. Chan et al, Actes du Sem Prog et Logique, Tregastel (May 1987), pp.77-108.

PIC - Brian Kernighan. Graphics meta-language for textually describing pictures, for use with troff. Featured in Jon Bentley's "More Programming Pearls." "PIC - A Language for Typesetting Graphics", B.W. Kernighan, Soft Prac & Exp 12(1):1-21 (Jan 1982). "PIC - A Graphics Language for Typesetting, Revised User Manual", Bell Labs TR 116, Dec 1984.

Pick BASIC - see Data/BASIC.

PICL - Language on Ncube or iPSC machines?

Pictorial Janus - K. Kahn, Xerox. Visual extension of Janus. Requires Strand88 and a PostScript interpreter.

pidgen+ - For Apple ][. Published in DDJ?  
ftp://wuarchive.wustl.edu/system/apple2/Lang/pidgen/\*

PIE - CMU. Similar to Actus.

PIL - Procedure Implementation Language, subsystem of DOCUS. Sammet 1969, p.678.

PIL/I - Variant of JOSS. Sammet 1969, p.217.

PILE -

1. Polytechnic's Instructional Language for Educators. Similar in use to an enhanced PILOT, but structurally more like Pascal with Awk-like associative arrays (optionally stored on disk). Distributed to about 50 sites by Initial Teaching Alphabet Foundation for Apple ][ and CP/M. "A Universal Computer Aided Instruction System," Henry G. Dietz & Ronald J Juels, Proc Natl Educ Computing Conf '83, pp.279-282.

2. "PILE \_ A Language for Sound Synthesis", P. Berg, Comput Music J 3,1 (1979).

PILOT - Programmed Inquiry Learning Or Teaching. CAI language, many versions. "Guide to 8080 PILOT", J. Starkweather, Dr Dobb's J (Apr 1977). IEEE 1154-1991.  
ftp://locke.ccil.org/pub/esr/pilot-1.6.tar.gz

PINBOL - Decision table language for controlling pinball machines used at Atari. Included a multi-tasking executive and an interpreter that worked on data structures compiled from condition:action lists.

PIRL - Pattern Information Retrieval Language. Language for digraph manipulation, embeddable in FORTRAN or ALGOL, for IBM 7094. "PIRL - Pattern Information Retrieval Language", S. Berkowitz, Naval Ship Res Dev Ctr, Wash DC.

PIT - Language for IBM 650. (See IT).

PL-11 - R.D. Russell, CERN, Nov 1971. High-level machine-oriented language for the PDP-11. Similar to PL360. Written in Fortran IV and cross-compiled on other machines.

PL360 - Structured assembly language for IBM 360 and 370, with a few high-level constructs, syntactically resembles ALGOL 60. Its grammar is defined entirely by operator precedence. "PL/360, A Programming Language for the 360 Computers", N. Wirth, J ACM 15(1):37-74 (Jan 1968).

PL516 - Similar to PL360. "PL 516, An ALGOL-like Assembly Language for the DDP-516", B.A. Wichmann, Natl Phys Lab UK, Report CCU 9, 1970

PL-6 - PL/I-like system language for the Honeywell OS CP-6.

PL.8 - A systems dialect of PL/I, developed originally for the IBM 801 RISC mini, later used internally for IBM RT and R/6000 development. "An Overview of the PL.8 Compiler", M. Auslander et al, Proc SIGPLAN '82 Symp on Compiler Writing.

Pla - High-level music programming language, written in SAIL. Includes concurrency based on message passing. "Pla: A Composer's Idea of a Language", B. Schottstaedt, Computer Music J 7(1):11-20 (Winter 1983).

PLACE - Programming Language for Automatic Checkout Equipment. "The Compiler for the Programming Language for Automatic Chekcout Equipment (PLACE)", AFAPL TR-68-27, Battelle Inst, Columbus, May 1968.

PLAGO - A translator-interpretter for a PL/I subset. "PLAGO/360 User's Manual, Poly Inst Brooklyn.

PLAIN - Programming LAnguage for INteraction. Pascal-like, with extensions for database, string handling, exceptions and pattern matching. "Revised Report on the Programming Language PLAIN", A. Wasserman, SIGPLAN Notices 6(5):59-80 (May 1981).

PLAN - Assembly language for ICL1900 series machines.

Planet - "An Experiment in Language Design for Distributed Systems", D. Crookes et al, Soft Prac & Exp 14(10):957-971 (Oct 1984).

PLANIT - Programming LAnguage for Interaction and Teaching. CAI language. "PLANIT - A Flexible Language Designed for Computer-Human Interaction", S.L. Feingold, Proc FJCC 31, AFIPS (Fall 1967) Sammet 1969, p.706.

Plankalkul - Konrad Zuse, ca. 1945. The first programming language, implemented for the Z3 computer. Included arrays and records. Much of his work may have been either lost or confiscated in the aftermath of WWII. "The Plankalkul of Konrad Zuse", F.L. Bauer et al, CACM 15(7):678-685 (Jul 1972).

PLANNER - C. Hewitt <hewitt@ai.mit.edu> MIT 1967. A language for writing theorem provers. Never fully implemented, see microPLANNER. "PLANNER: A Language for Proving Theorems in Robots", Carl Hewitt, Proc IJCAI-69, Wash DC, May 1969.

PLANS - Programming Language for Allocation and Network Scheduling. A PL/I preprocessor, used for developing scheduling algorithms. "A User's Guide to the Programming Language for Allocation and Network Scheduling", H.R. Ramsey et al, TR SAI-77-068-DEN, Science Applications Inc (Jun 1977).

PLASMA - PLAnner-like System Modeled on Actors. Carl Hewitt, 1975. The first actor language. Originally called Planner-73, and implemented in MacLisp. Lisp-like syntax, but with several kinds of parentheses and brackets. "A PLASMA Primer", B. Smith et al, AI Lab Working Paper 92, MIT Oct 1975. "Viewing Control Structures as Patterns of Passing Messages", C. Hewitt, AI Lab Memo 410, MIT 1976.

Plasyd - A structured assemply language, similar to PL360 but with ICL

instructions instead of IBM. Used internally by ICL for compiler development for the ICL 1900 series.

Platon - Distributed language based on asynchronous message passing. "Message Passing Communication Versus Procedure Call Communication", J. Staunstrup, *Soft Prac & Exp* 12(3):223-234 (Mar 1982). "Platon Reference Manual", S. Soerensen et al, RECAU, U Aarhus, Denmark.

PLAY - 1977. Language for real-time music synthesis. "An Introduction to the Play Program", J. Chadabe ete al, *Computer Music J* 2,1 (1978).

Playground - A visual language for children, developed for Apple's Vivarium Project. OOPSLA 89 or 90?

PL/C - Slight subset of PL/I, aimed at student use. "User's Guide to PL/C", S. Worona et al, Cornell, June 1974. "PL/C - A High Performance Compiler" H.L. Morgan et al, *Proc SJCC, AFIPS* 38:503-510 (1971). Implemented on IBM 370.

PL/I - Programming Language I. George Radin, 1964. Originally named NPL. An attempt to combine the best features of FORTRAN, COBOL and ALGOL 60. Result is large but elegant. One of the first languages to have a formal semantic definition, using the Vienna Definition Language. EPL, a dialect of PL/I, was used to write almost all of the Multics OS. PL/I has no reserved words. Types are fixed, float, complex, character strings with max length, bit strings, and label variables, no user-defined types. Dynamic arrays. Summation, multi-level structures, structure assignment. Untyped pointers, side effects, aliasing. Control flow goto, do-end groups, do-to-by-while-end loops, external procedures, nested procedures and blocks. Procedures may be declared recursive, or grouped into generic families. Controlled variables. Many implementations support concurrency ('call task' and 'wait(event)' amount to fork/join) and compile-time statements. Exception handling. "A Structural View of PL/I", D. Beech, *Computing Surveys*, 2,1 33-64 (1970). ANS X3.53-1976. PL/I is still widely used, at IBM and elsewhere. The CFS file system used at Argonne to manage terabytes of data is written in PL/I.

list: PL1-L@UIUCVMD.BITNET

ftp://wuarchive.wustl.edu/mirrors/msdos/pli/runplila.arc, PL/I interpreter version: LPI for PC's and Unix, Liant <rcg@lpi.liant.com> (508)626-0006

PL/I SUBSET - Early 70's version of PL/I for minis.

PL/I Subset G - ("General Purpose") The commercial PL/I subset (i.e., what was actually implemented by most vendors). ANS X3.74-1981.

PL/I-FORMAC - Variant of FORMAC. "The PL/I-FORMAC Interpreter", J. Xenakis, *Proc 2nd Symp Symbolic and Algebraic Manip*, ACM (Mar 1971). Sammet 1969, p.486.

Plisp - Pattern LISP. 1990. A pattern-matching rewrite-rule language, optimized for describing syntax translation rules. (See LISP70).

PLITS - Programming Language In The Sky. A computational model for concurrency with communication via asynchronous message-passing. "High Level Programming for Distributed Computing", J.A. Feldman, *CACM* 22(6):353-368 (Jun 1979).

PL/M - Programming Language/Microcomputers. Gary Kildall, MAA (later Digital Research) for Intel, 1972. A very low level language incorporating ideas from PL/I, ALGOL and XPL. Integrated macro processor. CP/M was written to support development of the PL/M compiler (not the other way around!). "PL/M-80 Programming Manual", Doc 98-268B, Intel 1976. "PL/M Programmer's Guide", Doc 452161-003, Intel. "A Guide to PL/M Programming for Microcomputer Applications", D. McCracken, A-W 1978. Versions: PL/M-80, PL/M-86, PL/M-286, PL/M-386.

ftp://iecc.com/pub/file/plm.shar.gz parser for PL/M-386  
//locke.ccil.org/pub/retro/plm2c.tar.gz PL/M-286 to C translator

PL/MP - C.J. Tan, IBM TJWRC, 1978. A microprogramming language resembling a subset of PL/I. "Code Optimization for Microcode Compilers", C.J. Tan, AFIPS Conf Proc 47:649-655 (1978).

PL/P - Programming Language, Prime. Russ Barbour, PRIME Computer, late 70's. Subset of PL/I used internally for implementation of PRIMOS. (See SPL[4]).

PL/PROPHET - PL/I-like language for the PROPHET system, used by pharmacologists. "The Implementation of the PROPHET System", P.A. Castleman et al, NCC 43, AFIPS (1974).

PL/S - Programming Language/Systems. IBM late 60's, for the IBM 360 and 370. A machine-oriented language derived from PL/I, permitting inline assembly code and control over register usage. Much of IBM 360 OS/MFT/MVT/SVS/MVS was written in it. Used internally, never released to the public. Documented by various IBM internal ZZ-? publications. "PL/S, Programming Language/Systems", W.R. Brittenham, Proc GUIDE Intl, GUIDE 34, May 14, 1972, pp.540-556. Versions: PLS1, PLSII.

PL/Seq - Programming Language for Sequences. A DSP language. "A General High Level Language for Signal Processors", J. Skytta & O. Hyvarinen, Digital Signal Processing 84, Proc Intl Conf, Florence, Italy, Sep 1984, pp.217-221.

PLZ - Zilog. A high level language for programming microprocessors. A minimal block structured language, goto-less and only DO-OD loops with exit, repeat [from] or continue [from] placed anywhere in the loop. Record structures. Used by Zilog, and by Olivetti in their S6000 series. "Introduction to Microprocessor Programming Using PLZ", Richard Conway et al, Winthrop Pub 1979. Available as PLZ/SYS. "Report on the Programming Language PLZ/SYS", Tod Snook et al, Springer 1978.

PLZ/ASM - Similar to PLZ, but with assembler instructions instead of statements.

PLUM - U. Maryland. Compiler for a substantial subset of PL/I for the Univac 1100. "PL/I Programming with PLUM", M.V. Zelkowitz, Paladin House, 1978.

Plural EuLisp - EuLisp with parallel extensions. "Collections and Garbage Collection", S.C. Merall et al, in Memory Management - IWMM92, Springer 1992, pp.473-489.

PLUS - Late 60's. Machine-oriented systems language used internally by Univac.

PLUSS - Proposition of a Language Useable for Structured Specifications. Algebraic specification language, built on top of ASL. "A First Introduction to PLUSS", M.C. Gaudel, TR, U Paris Sud, Orsay 1984.

PLZ - [?]

PM - "PM, A System for Polynomial Manipulations", G.E. Collins, CACM 9(8):578-589 (Aug 1966).

PML - Parallel ML. "Synchronous Operations as First-Class Values", J.H. Reppy <jhr@research.att.com>, Proc SIGPLAN 88 Conf Prog Lang Design and Impl, June 1988, pp.250-259.

PNU-Prolog - A parallel extension of NU-Prolog, implemented as a preprocessor.  
ftp:munnari.oz.au:/pub/bebop.tar.Z

POCAL - PETRA Operator's CommAnd Language.

POFAC - A subset of Fortran. Mentioned in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, p.273. "POFAC Description", R. Haentjens, Report 19, Centre d'Information, Ecole Royale Militaire, Brussels, 1973.

POGO - Early system on G-15. Listed in CACM 2(5):16 (May 1959).

Polka - Object orientation plus parallel logic, built on top of Parlog. "Polka: A Parlog Object-Oriented Language", Andrew Davison <ad@mullauna.cs.mu.oz.au>, TR, Parlog Group, Imperial College, London 1988.  
info: parlog@doc.ic.ac.uk

Poly -

1. D.C.J. Matthews, Cambridge, early 80's. Polymorphic, block-structured. "An Overview of the Poly Programming Language", D.C.J. Matthews <djcm@cl.cam.ac.uk>, in Data Types and Persistence, M.P. Atkinson et al eds, Springer 1988.

2. St Andrews U, Scotland. Software Prac & Exp, Oct 1986.[?]

3. Polymorphic language used in Polymorphic Programming Languages, David M. Harland, Ellis Horwood 1984.

POLYGOTH - Distributed language integrating classes with a parallel block structure, including multiprocedures and fragments. "Operational Semantics of a Distributed Object-Oriented Language and its Z Formal Specification", M. Benveniste <mbenveni@irisa.irisa.fr>, TR532, IRISA/INRIA-Rennes.

Ponder - Jon Fairbairn, <jf@cl.cam.ac.uk>. Polymorphic, non-strict functional language. Has a type system similar to Girard's System F ("Proofs and Types", J-Y. Girard, Cambridge U Press 1989), also known as Lambda-2 or the polymorphic lambda calculus. Ponder adds extra recursive 'mu' types to those of F, allowing more general recursion. "Ponder and its Type System", J. Fairbairn, TR 31, Cambridge U Computer Lab, Nov 1982. "Subtyping in Ponder", V. Paiva, TR 203, Aug 1990.

POOL2 - Parallel Object-Oriented Language. Philips Research Labs, 1987. Strongly typed, synchronous message passing, designed to run on DOOM (DOOM = Decentralized Object-Oriented Machine). "POOL and DOOM: The Object-Oriented Approach", J.K. Annot, PAM den Haan, in Parallel Computers, Object-Oriented, Functional and Logic, P. Treleaven ed. "Issues in the Design of a Parallel Object-Oriented Language", P. America, Formal Aspects of Computing 1(4):366-411 (1989).

POOL-I - Latest in the line of POOL languages. "A Parallel Object-Oriented Language with Inheritance and Subtyping", P. America et al, SIGPLAN Notices 25(10):161-168 (OOPSLA/ECOOP '90) (Oct 1990).

POOL-T - Object-oriented, concurrent, synchronous. Predecessor of POOL2. "Definition of the Programming Language POOL-T", Esprit Project 415, Doc. 0091, Philips Research Labs, Eindhoven, Netherlands, June 1985.

POP-1 - Package for Online Programming. Edinburgh, 1966. First of the POP family of languages. Used reverse Polish notation. Implemented as a threaded interpreter. EPU-R-17, U Edinburgh (Jul 1966). "POP-1: An Online Language", R. Popplestone, Mach Intell 2, E. Dale et al eds, Oliver and Boyd, Edinburgh 1968.

POP-2 - Robin POPplestone, Edinburgh, 1967. An innovative language incorporating many of Landin's ideas, including streams, closures, and functions as first-class citizens. Algol-like syntax. The first implementation was named Multi-POP, based on a REVPOL function written in POP-1, producing the reverse-polish form as output. "POP-2 Papers", R.M. Burstall et al, Oliver & Boyd 1968. "Programming in POP-2", R.M. Burstall et al, Edinburgh U Press 1971. "POP-2 User's Manual", R. Popplestone, Mach Intell 2, E. Dale et al eds, Oliver and Boyd, Edinburgh 1968.

POP-10 - Julian Davies, 1973. Descendant of POP-2, for the PDP-10. "POP-10 User's Manual", D.J.M. Davies, CS R25, U West Ontario, 1976.

Pop-11 - Robin POPplestone, 1975. Originally for the PDP-11. In some ways, Pop-11 is like FORTH (stack-oriented, extensible, efficient). It's also like LISP (functional, dynamically typed, interactive, garbage-collected). And the syntax is like Pascal (block-structured). "Programming in POP-11", J. Laventhol <jcl@deshaw.com>, Blackwell 1987. Implementations: AlphaPop for Mac (Computable Functions Inc, 413-253-7637). PopTalk [?] POPLOG (U Sussex) available for VAX/VMS and most workstations. info: Robin Popplestone <pop@cs.umass.edu> (413)253-7637

POP-9X - Proposed BSI standard for Pop-11.

POP++ - An object-oriented extension of POPLOG. Available from Integral Solutions.

POPCORN - AI system built on POP-2. "The POPCORN Reference Manual", S. Hardy, Essex U, Colchester, 1973.

Poplar - Morris, 1978. A blend of LISP with SNOBOL4 pattern matching and APL-like postfix syntax. Implicit iteration over lists, sorting primitive. "Experience with an Applicative String-Processing Language", J.H. Morris et al, 7th POPL, ACM 1980, pp.32-46.

POPLER - A PLANNER-type language for the POP-2 environment. "Popler 1.6

Reference Manual", D. Davies et al, U Edinburgh, TPU Report No 1 (May 1973).

POPLOG - U Sussex. Language for the two-stack virtual machine (PVM) which underlies the POPLOG interactive environment. POPLOG supports POP-11, LISP, Prolog and ML via shared data structures and incremental compilation. "POPLOG's Two-Level Virtual Machine Support for Interactive Languages", R. Smith et al, in Research Directions in Cognitive Science, v.5 (1992).

PopTalk - A commercial object-oriented derivative of POP, used in the Expert System MUSE. Cambridge Consultants.

Port - Waterloo Microsystems (now Hayes Canada) ca. 1979. Imperative language descended from Zed. "Port Language" document in the Waterloo Port Development System.

Portable Standard Lisp - "The Portable Standard LISP Users Manual", TR-10, CS Dept, U Utah, Jan 1982. "A Portable Lisp System", M.L. Griss et al, Proc 1982 ACM Symp on Lisp and Functional Prog, Aug 1982.

PORTAL - Process-Oriented Real-Time Algorithmic Language. "PORTAL - A Pascal-based Real-Time Programming Language", R. Schild in Algorithmic Languages, J.W. deBakker et al eds, N-H 1981.

Port Language - "Communicating Parallel Processes", J. Kerridge et al, Soft Prac & Exp 16(1):63-86 (Jan 1986).

POSE - 1967. An early query language. "POSE: A Language for Posing Problems to Computers", S. Schlesinger et al, CACM 10:279-285 (May 1967).

POSTQUEL - POSTGRES QUERY Language. Language used by the database system POSTGRES. "The Design of POSTGRES", M. Stonebraker et al, Proc ACM SIGMOD Conf, June 1986.

<ftp://postgres.berkeley.edu/pub/postgresv4r0.tar.Z> Version 4.0

PostScript - J. Warnock et al, Adobe Systems, ca. 1982. Interpretive FORTH-like language used as a page description language by Apple LaserWriter, and now many laser printers and on-screen graphics systems. "PostScript Language Reference Manual" ("The Red Book"), Adobe Systems, A-W 1985.

POSYBL - PrOgramming SYstem for distriButed appLications. Ioannis Schoinas. A Linda implementation for Unix networks.  
<ftp://ariadne.csi.forth.gr/pub/POSYBL.TAR.Z>  
info: [sxoinas@csd.ucl.ac.uk](mailto:sxoinas@csd.ucl.ac.uk)

PowerFuL - Combines functional and logic programming, using "angelic Powerdomains".

PPL - Polymorphic Programming Language. Harvard U. Interactive and extensible, based on APL. "Some Features of PPL - A Polymorphic Programming Language", T.A. Standish, SIGPLAN Notices 4(8) (Aug 1969).

PPLambda - Essentially the first-order predicate calculus superposed upon the simply-typed polymorphic lambda-calculus. The object language for LCF. "Logic and Computation: Interactive Proof with Cambridge LCF", L. Paulson, Cambridge U Press, 1987.

P-Prolog - Parallel logic language. "P-Prolog: A Parallel Logic Language Based on Exclusive Relation", R. Yang et al, Third Intl Conf on Logic Prog, 1986, pp.255-269.

PRA - PRAGmatics. Language used by COPS for specification of code generators. "Metalanguages of the Compiler Production System COPS", J. Borowiec, in GI Fachgesprach "Compiler-Compiler", ed W. Henhapl, Tech Hochs Darmstadt 1978, pp.122-159.

pre-cc - PREttier Compiler Compiler.  
ftp:ftp.comlab.ox.ac.uk:Programs

PREP - PRogrammed Electronics Patterns. Language for designing integrated circuits. "Computer Assisted Mask Production", R.L. Rosenfeld, Proc IEEE 57(9) Sep 1969.

PRESTO - Bershad et al, U Washington 1987. A parallel language for shared-memory multiprocessors, implemented as a C++ library. Provides classes for threads, spinlocks, monitors and condition variables. "PRESTO: A Kernel for Parallel Programming Environments", B.N. Bershad et al, U Wash CS TR, Jan 1987.  
info: presto@cs.washington.edu  
ftp://cs.washington.edu/pub/prestol.0.tar.Z

PRINT - PRe-edited INTerpreter. Early math for IBM 705. Sammet 1969, p.134.

PRINT I - Early system on IBM 705. Listed in CACM 2(5):16 (May 1959).

PRISM - Distributed logic language. "PRISM: A Parallel Inference System for Problem Solving", S. Kasif et al, Proc 1983 Logic Prog Workshop, pp.123-152.

PRL - Proof Refinement Logic. "PRL: Proof Refinement Logic Programmer's Manual", CS Dept, Cornell, 1983. Versions: micro-PRL, lambda-PRL, nu-PRL. (See NUPRL).

Probe - Object-oriented logic language based on ObjVlisp. "Proposition d'une Extension Objet Minimale pour Prolog", Actes du Sem Prog en Logique, Tregastel (May 1987), pp.483-506.

PROC - Job control language used in the Pick OS. "Exploring the Pick Operating System", J.E. Sisk et al, Hayden 1986.

PROCOL - J. van den Bos, Erasmus U, Rotterdam. A concurrent object-oriented language with protocols, delegation, persistence and constraints. "PROCOL - A Concurrent Object Language with Protocols, Delegation and Constraints", J. van den Bos et al, Acta Informatica 28:511-538 (1991).

PROFILE - Simple language for matching and scoring data. "User's Manual for the PROFILE System", Cambridge Computer Assoc (May 1974).

PROGENY - 1961. Report generator for UNIVAX SS90.

Prograph - Programming in Graphics. Tomasz Pietrzykowski, Technical U, Halifax. A visual dataflow language. Operation icons are connected by

datalinks through which information flows. Supports object orientation. First implemented in Pascal, Acadia U, 1982, later in Prolog at Tech U. Current versions are in C and Prograph. Available for Mac, soon for Windows and Unix. TGS Systems (902) 429-5642.  
info: <szpak@prograph.com>

PROGRES - PROgrammed Graph REwriting Systems. A. Scheurr, Aachen 1991. A very high level language based on graph grammars. Supports multiple inheritance and types of types, declarative specification of graphical attributes, visual specification of graph rewrite rules, builtin backtracking of graph modifications, Used for implementing abstract data types with graph-like internal structure, as a visual language for the graph-oriented database GRAS, and as a rule-oriented language for prototyping nondeterministically specified data/rule base transformations. A. Scheurr, "Introduction to PROGRES, an Attribute Graph Grammar Based Specification Language", in Proc WG'89 Workshop on Graphtheoretic Concepts in Computer Science", LNCS 411, Springer 1991.  
info: Andy Scheurr <andy@i3.informatik.rwth-aachen.de>  
ftp://ftp.informatik.rwth-aachen.de/pub/unix/PROGRES/\* for Sun4

PROJECT - Subsystem of ICES. Sammet 1969, p.616.

Prolog - PROgrammation en LOGique. Alain Colmerauer and Phillippe Roussel, U Aix-Marseille 1971. First implemented 1972 in ALGOL-W. Designed originally for natural-language processing. LUSH (or SLD) resolution theorem proving based on the unification algorithm. No user-defined functions, and no control structure other than the built-in depth-first search with backtracking. "Programming in Prolog", W.F. Clocksin & C.S. Mellish, Springer 1985. Early collaboration between Marseille and R. Kowalski at U Edinburgh continued until about 1975.

C-Prolog - F. Pereira <pereira@research.att.com> et al, Jul 1982. An implementation of Prolog in C. No garbage collection. Not PD.

SB-Prolog - Stony Brook Prolog. PD Prolog implementation for Unix.  
ftp://sbcs.sunysb.edu/pub/sbprolog/v3.0

//ftp.cso.uiuc.edu/amiga/fish/fl/ff140 and ff141 Amiga version 2.3.2

LM-Prolog - Lisp Machine Prolog. K. Kahn et al, 1983. "LM-Prolog User Manual", M. Carlsson et al, Uppsala Dec 1983.

ftp://sics.se/archive/lm-prolog.tar.Z - Prolog interpreter in Zetalisp

Other implementations:

ftp://cpsc.ucalgary.ca/pub/prolog1.1 - Prolog interpreter in Scheme

//aisun1.ai.uga.edu/ai.prolog/eslpro.zip - ESL Prolog for MS-DOS

//aisun1.ai.uga.edu/ai.prolog/? - Open Prolog for Mac

//cs.utah.edu/pub/frolic.tar.Z - Prolog interpreter in Common LISP.

//ai.uga.edu/ai.prolog.standard - Draft ISO standard

//ftp.inria.fr/INRIA/Projects/ChLoE/LOGIC\_PROGRAMMING/wamcc -  
Prolog to C translator via the WAM

Prolog-2 - An implementation of Edinburgh Prolog. "An Advanced Logic Programming Language", Anthony Dodd.

info: Nick Henfrey, ESL (Expert Systems Ltd, Magdalen Centre, Oxford Science Park, Oxford, OX4 4GA, tel 0865 784474.

Prolog-II - Prolog with two new predicates: 'dif' for coroutines and 'freeze' for delayed evaluation. "Prolog II Reference Manual and Theoretical Model", A. Colmerauer, Internal Report, GroupeIA, U Aix-Marseille (Oct 1982). Available from ExperIntelligence, Santa Barbara CA.

Prolog-III - A. Colmerauer, U Aix-Marseille, ca 1984. Marseille Prolog, with unification replaced by constraint resolution. [deferred goals too?] (Not to be confused with Prolog 3, a commercial product.) "Opening the Prolog-III Universe", BYTE 12(9):177-182 (Aug 1987). "An Introduction to Prolog III", A. Colmerauer, CACM 33(7):69-90 (1990). Version 1.2 for MS-DOS.

Prolog+ - [?]

Prolog++ - Phil Vasey, Logic Programming Associates. Prolog with object-oriented features added. For MS-DOS and X-windows. "Prolog++: The Power of Object-Oriented and Logic Programming", Christopher D.S. Moss, A-W 1994, ISBN 0-20-156507-2. Distributed by AI Intl Ltd in England and Quintus (800)542-1283.

Prolog-D-Linda - Embeds the Linda parallel paradigm into SISctus Prolog.  
ftp://ftp.cs.uwa.au  
info: geoff@cs.uwa.edu.au

Prolog-Linda -

1. Prolog extended with Linda-style parallelism. Proc 4th Australian Conf on Artif Intell.  
ftp://bison.cs.uwa.oz.au

2. Neil MacDonald, U Edinburgh 1989. Another Prolog extended with Linda, implemented on a Computing Surface.

PROMAL - PROgrammer's Microapplication Language. Systems Management Assocs. Interpreted C-like language for MS-DOS, C=64 and Apple ][. Computer Language, Mar 1986, pp.128-134.

PROMELA - Language for building finite state machines. [?]  
ftp://netlib.att.com

Pronet - "The Design of a Programming Language Based on Connectivity Networks", R. LeBlanc et al, Proc 3rd Intl Conf Distrib Comp Sys, IEEE 1982, pp.532-541.

PROOF/L - Language with implicit parallelism. Functional, object-oriented. J Parallel Dist Comp 12:202-212 (1991). Forthcoming TR from RADC. [?]

Proposal Writing - Extension of FORTRAN for proposal writing. Sammet 1969, p.170.

PROSE -

1. PROblem Solution Engineering. Numerical problems including differentiation and integration. "Computing in Calculus", J. Thames, Research/Development 26(5) (May 1975).

2. A constraints-and-sequencing system similar to Kaleidoscope. "Reflexive Constraints for Dynamic Knowledge Bases", P. Berlandier et al in Proc First Intl CS Conf '88: AI: Theory and Appls, Dec 1988.

ProSet - U Essen, 1990. Formerly SETL/E. A derivative of SETL with Ada-like syntax. "SETL/E, A Prototyping System Based on Sets", E.E.

Doberkat et al, in Tagungsband TOOL90, W. Zorn ed, pp.109-118, U Karlsruhe, Nov 1990. "ProSet - A Language for Prototyping with Sets", E.-E. Doberkat et al, in Proc Third Intl Workshop on Rapid System Prototyping, N. Kanopoulos ed, IEEE Comp Soc Press, June 1992, pp.235-248.

PROSPER - "PROSPER: A Language for Specification by Prototyping", J. Leszczykowski, Comp Langs 14(3):165-180 (1989).

ProTalk - Quintus. An object-oriented Prolog.

PROTEUS -

1. Extensible language, core of PARSEC. "The Design of a Minimal Expandable Computer Language", J.R. Bell, PhD Thesis, CS, Stanford U (Dec 1968).

2. A language for prototyping parallel languages. Interpreter based on ISETL.  
ftp://particle.cs.unc.edu/proteus

Protosynthex - Query system for English text. Sammet 1969, p.669.

PS 440 - K. Lagally, ca 1974. The system implementation language for the Telefunken TR 440 computer.

PS-ALGOL - Persistent Algol. ca 1981, released 1985. A derivative of S-Algol. Database capability derived from the longevity of data. "The PS-Algol Reference Manual", TR PPR-12-85, CS Dept, U Glasgow 1985. IBM PC version available from CS Dept, U Strathclyde, Glasgow.

pSather - Parallel extension of Sather for clustered shared memory model. Threads synchronized by monitor objects ("gates"). Locality assertions and placement operators. "pSather Monitors: Design, Tutorial, Rationale and Implementation", J.A. Feldman et al, TR-91-031 and TR-93-028, ICSI, Berkeley, CA. Implementation on CM-5.

PSETL - Parallel SETL - An extension of SETL for operating specification and simulation, including process management, I/O and interprocess communication primitives. "Description of Operating Systems Using Very High Level Diction", Gabriel Leshem, NYU 1984.

PSML - Processor System Modeling Language. Simulating computer systems design. A preprocessor to SIMSCRIPT. "Processor System Modeling - A Language and Simulation System", F. Pfisterer, Proc Symp on Simulation of Computer Systems (Aug 1976).

P-TAC - Parallel Three Address Code. "P-TAC: A Parallel Intermediate Language", Z. Ariola et al, Fourth Intl Conf Func Prog Langs and Comp Arch, ACM Sept 1989. (See Kid).

PUB - PUBLISHing. 1972. An early text-formatting language for TOPS-10, with syntax based on SAIL. Influenced TeX and Scribe. "PUB: The Document Compiler", Larry Tesler, Stanford AI Proj Op Note, Sept 1972.

PUFFT - "The Purdue University Fast FORTRAN Translator", Saul Rosen et al, CACM 8(11):661-666 (Nov 1965).

PUMPKIN - "PUMPKIN - (Another) Microprogramming Language", G.R. Lloyd, SIGMICRO Newsletter 5:15-44 (Apr 1974).

PVM - Parallel Virtual Machine. Intermediate language used by the Gambit compiler for Scheme. [Also by Multilisp?]

Python -

1. Guido van Rossum <guido@cwil.nl> 1991. A high-level interpreted language combining ideas from ABC, C, Modula-3, Icon, etc. Intended for prototyping or as an extension language for C applications. Modules, classes, user-defined exceptions. "Linking a Stub Generator (AIL) to a Prototyping Language (Python)", Guido van Rossum et al, Proc 1991 EurOpen Spring Conf. Available for Unix, Amoeba and Mac. Version 1.0.0.  
ftp://ftp.cwi.nl/pub/python/python1.0.0.tar.Z  
list: python-list@cwil.nl

2. Compiler for CMU Common LISP.

Q - Very high level language based on generalized (lazy) sequences. Lexical scope, some support for logical and constraint programming. Macros. Implemented in C++.  
ftp://ftp.cygus.com/pub/Q.\*

QA4 - Question-answering language. A procedural calculus for intuitive reasoning. A LISP-based pattern-matching language for theorem proving. "QA4, A Language for Writing Problem-Solving Programs", J.F. Rulifson et al, Proc IFIP Congress 1968.

QBE - Query By Example. Moshe Zloof, IBM 1975. A user-friendly query language. "QBE: A Language for Office and Business Automation", M.M. Zloof, Computer pp.13-22 (May 1981).

Qlambda - "Queue-based Multi-processing Lisp", R. Gabriel & J. McCarthy, Proc 1984 Symp Lisp and Functional Prog, pp.25-44.

QLISP -

1. SRI 1973. General problem solving, influenced by PLANNER. QA4 features merged with INTERLISP. "QLISP - A Language for the Interactive Development of Complex Systems", E. Sacerdoti et al, NCC 45:349-356, AFIPS (1976).

2. A parallel LISP. "Qlisp", R. Gabriel et al in Parallel Computation and Computers for AI, J. Kowalik ed, 1988, pp.63-89.

QLOG - An integration of logic programming into LISP. "QLOG - The Programming Environment for Prolog in LISP", H.J. Komorowski in Logic Programming, K.L. Clark et al eds, Academic Press 1982.

Q'NIAL - Queen's U, Canada. A portable incremental compiler for NIAL, written in C. "The Q'NIAL Reference Manual", M.A. Jenkins, Queen's U Report, Dec 1983. Versions for Unix and MS-DOS, from NIAL Systems Ltd, Ottawa Canada, (613)234-4188.

QPE - Two-dimensional pictorial query language. "Pictorial Information Systems", S.K. Chang et al eds, Springer 1980.

Q-systems. A. Colmerauer, 1969. A rewrite system with one-way unification, used for English-French translation. It led to Prolog. "The Birth of Prolog" A. Colmerauer et al, SIGPLAN Notices 28(3):37-52 (March 1993).

Quake - Stephen Harrison, DEC SRC, 1993. A string-oriented language designed to support the construction of Modula-3 programs from modules, interfaces and libraries.

QUEASY - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

QUEL - Query language used by the database management system INGRES.

Quest -

1. A language designed for its simple denotational semantics. "The Denotational Semantics of Programming Languages", R. Tennent, CACM 19(8):437-453 (Aug 1976).

2. QUantifiers and SubTypes. Language with a sophisticated type system. Just as types classify values, "kinds" classify types and type operators. Explicit universal and existential quantification over types, type operators, and subtypes. Subtyping is defined inductively on all type constructions, including higher-order functions and abstract types. User-definable higher-order type operators. "Typeful Programming", Luca Cardelli <luca@src.dec.com>, RR 45, DEC SRC 1989. Implemented in Modula-3. <ftp://gatekeeper.dec.com/pub/DEC/Quest/quest12A.tar.Z>

QUICK - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

Quicksilver - dBASE-like compiler for MS-DOS from WordTech, Orinda, CA.

QUIKSCRIPT - Simulation language derived from SIMSCRIPT, based on 20-GATE. "Quikscript - A Simpscript-like Language for the G-20", F.M. Tonge et al, CACM 8(6):350-354 (June 1965).

QUIKTRAN - FORTRAN-like, interactive with debugging facilities. Sammet 1969, p.226.

QUIN - Pyle 1965. Interactive language. Sammet 1969, p.691.

Quintec-Objects - Based on Quintec (not Quintus) Prolog. British.

Quty - Functional plus logic. "Quty: A Functional Language Based on Unification", M. Sato et al, in Conf Fifth Gen Computer Systems, ICOT 1984, pp.157-165.

QX - (meaning "OK", from E.E. Smith SF books). Richard Gillmann, SDC, Santa Monica. Language for digital signal processing of digitized speech. Was part of SDC's speech recognition project.

Raddle - "On the Design of Large Distributed Systems", I.R. Forman, Proc 1st IEEE Intl Conf Comp Langs, pp.25-27 (Oct 1986).

RAIL - Automatix. High-level language for industrial robots.

RAISE - See RSL.

RAL - Expert system.

RAMIS II - Rapid Access Management Information System. Database system. On-Line Software Intl.

Rapidwrite - Method for translating set of abbreviations into the much more verbose COBOL code. Sammet 1969, p.338.

RAPT - "An Interpreter for a Language for Describing Assemblies", R.J. Popplestone et al, Artif Intell 14:79-107 (1980).

RASP - "RASP - A Language with Operations on Fuzzy Sets", D.D. Djakovic, Comp Langs 13(3):143-148 (1988).

RATEL - Raytheon Automatic Test Equipment Language. For analog and digital computer controlled test centers. "Automatic Testing via a Distributed Intelligence Processing System", S.J. Ring, IEEE AUTOTESTCON 77 (Nov 1977).

RATFIV - Successor to RATFOR.

RATFOR - RATIONAL FORTRAN. Kernighan. FORTRAN preprocessor to allow programming with C-like control flow. "Ratfor - A Preprocessor for a Rational Fortran", B.W. Kernighan, Soft Prac & Exp 5:395-406 (Oct 1975). Featured in Software Tools, B.W. Kernighan & P.J. Plauger, A-W 1976. <ftp://wuarhive.wustl.edu/mirrors/unix-c/languages/ratfor.tar-z>

RAWOOP-SNAP - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May 1959).

R:BASE - MS-DOS 4GL from Microrim. Based on Minicomputer DBMS RIM. Was Wayne Erickson the author?

RBASIC - Database language for Revelation, by Cosmos Inc. Combines features of BASIC, Pascal and Fortran.

RBCSP - Roper & Barter's CSP. "A Communicating Sequential Process Language and Implementation", T. Roper & J. Barter, Soft Prac & Exp 11(11):1215-1234 (Nov 1981).

rc -

1. Tom Duff. AT&T Plan 9 shell. Lookalike by Byron Rakitzis <[byron@archone.tamu.edu](mailto:byron@archone.tamu.edu)> <ftp://archone.tamu.edu>

2. Microsoft resource language.

RCC - An extensible language. [?]

RCL - Reduced Control Language. A simplified job control language for OS360, translated to IBM JCL. "Reduced Control Language for Non-Professional Users", K. Appel in Command Languages, C. Unger ed, N-H 1973.

RDL - Requirements and Development Language. "RDL: A Language for Software Development", H.C. Heacox, SIGPLAN Notices 14(9):71-79 (Sep 1979).

Real-Time Euclid - Real-time language, restriction to time-bounded constructs. "Real-Time Euclid: A Language for Reliable Real-Time Systems", E. Kligerman et al, IEEE Trans Software Eng SE-12(9):941-949 (Sept 1986).

Real-Time Mentat - An extension of C++. "Real-Time Mentat: A Data-Driven Object-Oriented System", A.S. Grimshaw et al, Proc IEEE Globecom, Nov 1989 pp.232-241.

Real-Time Pascal - Later name for Pascal-80 by RC Intl, Denmark.

REC - Regular Expression Converter. See CONVERT.

Recital - dBASE-like language/DBMS from Recital Corp. Versions include VAX/VMS.

RECOL - RETrieval COmmand Language. CACM 6(3):117-122 (Mar 1963).

Red - (Also "REDL"). Intermetrics. A language proposed to meet the Ironman requirements which led to Ada. "On the RED Language Submitted to the DoD", E.W. Dijkstra, SIGPLAN Notices 13(10):27 (Oct 1978). "RED Language Reference Manual", J. Nestor and M. van Deusen, Intermetrics 1979.

REDCODE - Proposed as a language for "battle programs" in corewars. (See Computer Recreations column in Scientific American.)

RediLisp - R.M. Keller, U Utah. Dialect of Lisp used on the Rediflow machine, a derivative of FEL.

REDUCE - Anthony Hearn, 1963. Symbolic math, ALGOL-like syntax, written in LISP. "REDUCE, Software for Algebraic Computation", G. Rayna, Springer 1987. Version: Reduce 2, based on Portable Standard LISP.

list: REDUCE-L@DEARN.BITNET

info: reduce@rand.org

server: reduce-netlib@rand.org

Refal - Recursive Functions Algorithmic Language. V. Turchin, Moscow ca 1972 (now at CUNY?). "Supercompiler System Based on the Language Refal", V. Turchin, SIGPLAN Notices 14(2):46-54 (Feb 1979).

REF-ARF - "REF-ARF: A System for Solving Problems Stated as Procedures", R.E. Fikes, Artif Intell J 1(1) (Spring 1970).

Refine - Cordell Green et al, Stanford U. High-level wide-spectrum specification language. Set-theoretic data types (sets, mappings, sequences), first-order logic (for all, there exists). Syntax-directed pattern matching, type inference, and a declarative transformation operator (pre- and post-conditions). Implemented as algorithms which build and transform annotated abstract syntax trees. Compiled into Common Lisp. "Research on Knowledge-Based Software Environments at Kestrel Institute", D.R. Smith et al, IEEE Trans Soft Eng, SE-11(11) (1985). Available from Reasoning Systems, 3260 Hillview Ave, Palo Alto, CA 94304. (See CHI). info: help@reasoning.com

Refined C (RC) - An extension of C to directly specify data access rights so that flow analysis, and hence automatic parallelization, is more effective. Research implementations only. "Refining A Conventional

Language For Race-Free Specification Of Parallel Algorithms," H.G. Dietz et al, Proc 1984 Intl Conf Parallel Proc, pp.380-382.

Refined Fortran (RF) - Similar to Refined C. Research implementations only. "Refined FORTRAN: Another Sequential Language for Parallel Programming," H.G. Dietz et al, Proc 1986 Intl Conf Parallel Proc, pp.184-191.

REG-SYMBOLIC - Early system on IBM 704. Listed in CACM 2(5):16 (May 1959).

REGTRAL - [?] Mentioned in Attribute Grammars, LNCS 323, p.108.

Relational Language. Clark & Gregory. First parallel logic language to use the concept of committed choice. Forerunner of PARLOG. "A Relational Language for Parallel Programming", K.L. Clark et al, Proc ACM Conf on Functional Prog Langs and Comp Arch, pp.171-178, ACM 1981.

RELATIVE - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

RELCODE - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

REL English - Rapidly Extensible Language, English. A formal language based on English. "Practical Natural Language Processing: The REL System as Prototype", Adv in Computers 13, Academic Press 1975.

RenderMan Shading Language. "The RenderMan Companion", S. Upstill, A-W 1989, chaps 13-15.

RENDEZVOUS - Query language, close to natural English. "Seven Steps to Rendezvous with the Casual User", E. Codd in Data Base Management, J.W. Klimbie et al eds, N-H 1974, pp.179-199.

REPL - Restricted EPL. A subset of EPL (the efficient part) used to write the core of Multics.

Required-COBOL - 1961. Minimal subset of COBOL. Later dropped entirely. Sammet 1969, p.339.

Retrieve - Tymshare Corp, 1960's. Query language, inspired JPLDIS which led to Vulcan[1] and then to dBASE II.

Revised ALGOL 60 - Alternate name for ALGOL 60 Revised. Sammet 1969, p.773.

REXX - Restructured EXTended eXecutor. M. Cowlshaw, IBM ca. 1979. (Original name: REX. They also call it "System Product Interpreter"). Scripting language for IBM VM and MVS systems, replacing EXEC2. "The REXX Language: A Practical Approach to Programming", M.F. Cowlshaw, 1985. Versions: PC-Rexx for MS-DOS, and AREXX for Amiga.  
list: REXX-L@UIUCVMD.BITNET.  
ftp://rexx.uwaterloo.ca/pub/freerexx/\* REXX interpreters for Unix

Rez - MacIntosh resource language.

RIGAL - Language for compiler writing. Data structures are atoms, lists/trees. Control based on pattern-matching. "Programming Language

RIGAL as a Compiler Writing Tool", M.I. Augustson, Inst of Math and CS of Latvia U, 1987.

Rigel - Database language? Based on Pascal. Listed by M.P. Atkinson & J.W. Schmidt in a tutorial presented in Zurich, 1989.

RIPscrip - Remote Imaging Protocol scripts. Telegrafix Inc. A protocol/language for describing graphics-painting operations over low speed serial lines. Resembles a small-scale version of NAPLPS. Popular with BBS implementors. Implemented by the freeware communications program RIPterm for MS-DOS. "RIPscrip Graphics Protocol SPecification, Version 1.54", Telegrafix.

RLI - Representation Language Language. A frame language. "A Representation Language Language", R. Greiner and D.B. Lenat, Proc AAAI-80, 1980.

RMAG - Recursive Macro Actuated Generator. Robert A. Magnuson, NIH ca 1970. Stand-alone macroprocessor for IBM 360/370 under VS or OS. Many built-in features and a library of several hundred macros. Several large systems were written in RMAG to generate source code for languages such as IBM JCL, IBM assembly language, COBOL. There was also a system (SLANG: Structured LANGuage compiler) which would generate 370 assembly language from a pseudo-structured-programming language, based on Michael Kessler's structure programming macros developed at IBM. "Project RMAG--RMAG22 User's Guide", R.A. Magnuson, NIH-DCRT-DMB-SSS-UG103, NIH, DHEW, Bethesda, MD 20205 (1977).

ROADS - Subsystem of ICES. Sammet 1969, p.616.

ROBEX - ROBot EXapt. Aachen Tech College. Based on EXAPT. Version: ROBEX-M for micros.

Robotalk - A Forth-based assembly/control language with low level extensions such as special purpose registers and interrupts, and traces of C, Pascal and the HP48 calculator. Used in the game "RoboWar" by David Haris, implemented on the Macintosh. Decompiler by Josh Goldfoot. <ftp://sumex.aim.stanford.edu/info-mac/games>

Roff - Text formatting language/interpreter associated with Unix. (See groff, nroff, troff, RUNOFF).

ROME - Experimental object-oriented language. "The Point of View Notion for Multiple Inheritance", B. Carre et al, SIGPLAN Notices 25(10):312-321 (OOPSLA/ECOOP '90) (Oct 1990).

Rosette - MCC. Concurrent object-oriented language.

RPG - Report Program Generator. IBM 1965. For easy production of sophisticated large system reports. Standard language for programming IBM's S/36 and AS/400 systems. Versions: RPG II, RPG III, RPG/400 for the IBM AS/400. MS-DOS versions by California Software and Lattice. (See CL, OCL).

RPL-1 - Data reduction language. Proc SJCC 30:571-575, AFIPS (Spring 1967).

RPL - Reverse Polish LISP. Language used by HP-28 and HP-48 calculators.

RPT - Unify. Report Writer Language.

RSL - RAISE Specification Language. (RAISE=Rigorous Approach to Industrial Software Engineering). A wide-spectrum specification and design language. Systems may be modular, concurrent, nondeterministic. Specifications may be applicative or imperative, explicit or implicit, abstract or concrete. ESPRIT Project 315, CRI A/S, Denmark. "The RAISE Specification Language", RAISE Language Group, P-H 1992, ISBN 0-13-752833-7.

RTC++ - Real-time extension of C++. "Object-Oriented Real-Time Language Design: Constructs for Timing Constraints", Y. Ishikawa et al, SIGPLAN Notices 25(10):289-298 (OOPSLA/ECOOP '90) (Oct 1990).

RT-CDL - Real-Time Common Design Language. Real-time language for the design of reliable reactive systems. "RT-CDL: A Real-Time Description Language and Its Semantics", L.Y. Lin et al, 11th World Computer Congress IFIP '89 pp.19-26 (Sep 1989).

RTL - Register Transfer Language. Chris Fraser <cwf@research.att.com> & J. Davidson <jwd@virginia.edu>, U Arizona early 80's. Intermediate code for a machine with an infinite number of registers, used for machine-independent optimization. RTL is used by the GNU C compiler and by Davidson's VPCC (Very Portable C compiler). "Quick Compilers Using Peephole Optimization", Davidson et al, Soft Prac & Exp 19(1):79-97 (Jan 1989).

RTL/1 - Real Time Language. Barnes, ICI 1971. A real-time language, the predecessor of RTL/2. "Real Time Languages for Process Control, J.G.P. Barnes, Computer J 15(1):15-17 (Feb 1972).

RTL/2 - John Barnes et al, Imperical Chemical Industries, 1972. Small real-time language based on ALGOL 68, with separate compilation. A program is composed of separately compilable 'bricks' (named modules) which may be datablock, procedure, or stack. A stack is a storage area for use as a workspace by a task. The language is block-structured and weakly typed. Simple types are byte, int, frac and real, no Boolean. Compound types may be formed from arrays, records and refs (pointers). There are no user-defined types. Control consists of if-then-elseif-else-end, for-to-by-do-rep, block-endblock, switch, goto, and label variables. Currently used in the UK and Europe for Air Traffic Control and industrial control. "RTL/2: Design and Philosophy", J.G.P. Barnes, Hayden & Son, 1976. British Standards Inst BS5904 (1980), now being revised.

Ruby -

1. Jones & Sheeran, 1986. Hardware description language. "Ruby - A Language of Relations and Higher-Order Functions", M. Sheeran, Proc 3rd Banff Workshop on Hardware Verification, Springer 1990.  
ftp://ftp.cs.chalmers.se/pub/misc/ruby/\*  
info: graham@cs.chalmers.se

2. One of five pedagogical languages based on Markov algorithms, used in "Nonpareil, a Machine Level Machine Independent Language for the Study of Semantics", B. Higman, ULICS Intl Report No ICSI 170, U London (1968). (cf. Brilliant, Diamond, Nonpareil, Pearl[3]).

RUFL - Rhodes University Functional Language. Rhodes U, Grahamstown, South Africa. Miranda-like.

RUNCIBLE - Early system for math on IBM 650. Listed in CACM 2(5):16 (May 1959).

RUNOFF - An early text-formatting language supported under TOPS-10 on the PDP-10. Ancestral to the troff/nroff/groff family of Unix-based formatters, it resembled a large subset of nroff.

RUSH -

1. Remote Use of Shared Hardware. ca 1966. Interactive dialect of PL/I, related to CPS[1]. "Introduction to RUSH", Allen-Babcock Computing 1969. Sammet 1969, p.309.

2. Adam Sah et al. Extension language, a descendant of Tcl. "An Introduction to the Rush Language", A. Sah et al, Proc Tcl'94 Workshop, June 1994.  
ftp://ginsberg.cs.berkeley.edu/pub/papers/asah/rush-tcl94.\*

Russell - (named for the British mathematician Bertrand Russell (1872-1970)) A. Demers & J. Donahue. A compact, polymorphically typed functional language, with bignums and continuations. Types are themselves first-class values and may be passed as arguments. "An Informal Description of Russell", H. Boehm et al, Cornell CS TR 80-430, 1980. "Understanding Russell: A First Attempt", J.G. Hook in LNCS 173, Springer ftp://parcftp.xerox.com/pub/russell/russell.tar.Z

RUTH - Harrison <D.A.Harrison@newcastle.ac.uk>. Real-time language based on LispKit. Uses timestamps and real-time clocks. "RUTH: A Functional Language for Real-Time Programming", D. Harrison in PARLE: Parallel Architectures and Languages Europe, LNCS 259, Springer 1987, pp.297-314.

S - AT&T. Statistical analysis. "S: An Interactive Environment for Data Analysis and Graphics", Richard A. Becker, Wadsworth 1984. "The New S Language", R.A. Becker et al, Wadsworth & Brooks, 1988. ISBN 0-53-409193-8. Version S-Plus available from StatSci <mtkg@statsci.com>.

S\* - Dasgupta, Simon Fraser U, 1978. A microprogramming language schema, which instantiates to a complete language for any given micromachine. Has Pascal-like syntax, with pre- and post-conditions. "Towards a Microprogramming Language Schema", S. Dasgupta, Proc 11th Ann Workshop Microprogramming (MICRO-11), 1978, pp.144-153.

S\*A - Dasgupta, 1981. A high-level architecture description language, designed to be used with S\*.

S\*M - A nonprocedural hardware description language. "S\*M, An Axiomatic, Non-procedural Hardware Description Language for CLocked Architectures", P.A. Wilsey, MS Thesis, U Southwestern Louisiana, 1985.

S3 - ALGOL-like system language for the ICL 2900 computer.

SAAL - Used on the Univac 1005 in the 1960's by the US Army Material Command.

SAC - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

SAC-1 - G.E. Collins. Early symbolic math system, written in FORTRAN. Proc 2nd Symp Symb Alg Manip pp.144-152 (1971).

SAC2 - Symbolic math system, compiles to FORTRAN or Common LISP. <jma@poly.polytechnique.fr>

SAD SAM - Query language by Lindsay. Sammet 1969, p.669.

SAFARI - Online text editing system by MITRE. Sammet 1969, p.685.

Safe Ada - A subset of Ada for writing safety-critical software. "Safe Ada - Language Study", System AG (Aug 1987).

SAIL -

1. Early system on Larc computer. Listed in CACM 2(5):16 (May 1959).

2. Stanford Artificial Intelligence Language. Dan Swinehart & Bob Sproull, Stanford AI Project, 1970. A large ALGOL-60-like language for the DEC-10 and DEC-20. Its main feature is a symbolic data system based upon an associative store (originally called LEAP). Items may be stored as unordered sets or as associations (triples). Processes, events and interrupts, contexts, backtracking and record garbage collection. Block-structured macros. "Recent Developments in SAIL - An ALGOL-based Language for Artificial Intelligence", J. Feldman et al, Proc FJCC 41(2), AFIPS (Fall 1972). (See MAINSAIL).

SAINT - Symbolic Automatic INTEgrator. J. Slagle, MIT 1961. Written in LISP. Sammet 1969, p.410.

SAL -

1. Single Assignment Language.

2. Simple Actor Language. A minimal actor language, used for pedagogical purposes in Actors, A Model of Concurrent Computation in Distributed Systems, G. Agha, MIT Press 1986.

3. SPARK Annotation Language. ICL, Ltd. Used in the verification of SPARK programs against Z specifications.

SALEM - "SALEM - A Programming System for the Simulation of Systems Described by Partial Differential Equations", S.M. Morris et al, Proc SJCC 33(1), 1968.

S-Algol - Orthogonal data structures on Algol-60. "S-Algol Language Reference Manual", R. Morrison, TR CS/79/1 U St Andrews, 1979. "An Introduction to Programming with S-Algol", A.J. Cole & R. Morrison, Cambridge U Press 1982.

SALT -

1. Symbolic Assembly Language Trainer. Assembly-like language

implemented in BASIC by Kevin Stock, now at Encore in France.

2. Sam And Lincoln Threaded language. A threaded extensible variant of BASIC. "SALT", S.D. Fenster et al, BYTE (Jun 1985) p.147.

SAM76 - Claude Kagan. Macro language, a descendant of TRAC. Version for CP/M. Dr Dobbs J ca 1977.

SAMeDL - SQL Ada Module Description Language. Used to interface Ada applications to SQL-based DBMS's. Compilers for various databases available from Intermetrics. "Rationale for SQL Ada Module Description Language SAMeDL", SEI-92-TR-016.

info: Marc Graham <marc@sei.cmu.edu>

ftp://ajpo.sei.cmu.edu/public/atip/samedl/\*

//ajpo.sei.cmu.edu/pub/documents/92\_reports/tr16.92.ps

//ajpo.sei.cmu.edu/pub/marc/Reference.ps.Z

Sandman - DoD requirements that led to APSE.

SAP - Symbolic Assembler Program. IBM 704 assembly language, late 50's.

SARG - Used on the Univac 1004 in the 1960's by the US Army Material Command.

SAS - Statistical Analysis System. Statistical and matrix language, PL/I-like syntax. "A User's Guide to SAS", A.J. Barr, SAS Inst 1976.

SASL - Saint Andrews Static Language. Turner, 1976. A derivative of ISWIM with infinite data structures. Fully lazy and weakly typed. Designed for teaching functional programming, with very simple syntax. A version of the expert system EMYCIN has been written in SASL. "A New Implementation Technique for Applicative Languages", D.A. Turner, Soft Prac & Exp 8:31-49 (1979).

ftp://a.cs.uiuc.edu/uiuc/kamin.distr/distr/sasl.p

SASL+LV - Unifies logic and functional programming. A more complete version of FGL+LV, in SASL syntax. "Combinator Evaluations of Functional Programs with Logical Variables", G. Bage et al, TR UUCS-87-027, U Utah, Oct 1987.

SASL-YACC - See yacc.

Sather - ("Say-ther", named for the Sather Tower at UCB, as opposed to the Eiffel Tower) Steve M. Omohundro, ICSI, Berkeley 1991. Interactive object-oriented language with simple syntax, similar to Eiffel, but non-proprietary and faster. Sather 0.2 was nearly a subset of Eiffel 2.0, but Sather 1.0 adds many distinctive features. Parameterized classes, multiple inheritance, statically-checked strong typing, garbage collection. Generates C as an intermediate language. Versions for most workstations. (See dpSather, pSather, Sather-K).

ftp://ftp.icsi.berkeley.edu/pub/sather

info: sather-admin@icsi.berkeley.edu

list: sather@icsi.berkeley.edu

Sather-K - Karlsruhe Sather. A sublanguage of Sather used for introductory courses in object-oriented design and typesafe programming.

info: trapp@karlsruhe.gmd.de

SCALLOP - Medium-level language for CDC computers, used to bootstrap the first Pascal compiler.

SCAN -

1. "A Parallel Implementation of the SCAN Language", N.G. Bourbakis, *Comp Langs* 14(4):239-254 (1989).

2. DEC. A real-time language. [same as 1?]

SCEPTRE - Designing and analyzing circuits. "SCEPTRE: A Computer Program for Circuit and Systems Analysis", J.C. Bowers et al, P-H 1971.

Scheme - (originally "Schemer", by analogy with Planner and Conniver.) G.L. Steele & G.J. Sussman, 1975. A LISP dialect, small and uniform, with clean semantics. Scheme is applicative-order and lexically scoped, and treats both functions and continuations as first-class objects.

RRS - "The Revised Report on Scheme", G.L. Steele et al, AI Memo 452, MIT, Jan 1978.

R2RS - "The Revised Revised Report on the Algorithmic Language Scheme", W. Clinger, AI Memo 848, MIT Aug 1985.

R3RS - "The Revised<sup>3</sup> Report on the Algorithmic Language Scheme", J. Rees et al, *SIGPLAN Notices* 21(12):37-79 (Dec 1986).

R3.99RS - R4RS minus the macros

R4RS - "The Revised<sup>4</sup> Report on the Algorithmic Language Scheme", W. Clinger et al, MIT (Nov 1991) (ftp from altdorf.ai.mit.edu).

IEEE P1178-1990, "IEEE Standard for the Scheme Programming Language", ISBN 1-55937-125-0.

Implementations: Scheme86 (Indiana U), MacScheme (Semantic Microsystems), PC Scheme (TI). (See T).

"Orbit: An Optimizing Compiler for Scheme", D.A. Kranz et al, *SIGPLAN Notices* 21(7):281-292 (Jul 1986).

ftp://altdorf.ai.mit.edu/archive/scm/\* SCM interpreter in C, and Hobbit2 compiler (compiles to C), for Amiga, Atari-ST, Mac, MS-DOS, NOS/VE, VAX/VMS, Unix. Conforms to Revised<sup>4</sup> Report and IEEE P1178.

//martigny.ai.mit.edu/archive/scheme-7.3/\* (MIT-Scheme, Liar compiler)

//gatekeeper.dec.com/pub/comp.sources.misc/volume8/elk (for Suns)

//acorn.cs.brandeis.edu/dist/gambit2.0-tar.Z (compiler for 68K's)

//ux1.cso.uiuc.edu/amiga/fish/f7/ff764 (Gambit compiler for Amiga)

//world.std.com/src/lisp/siod-v2.8-shar (Scheme In One Defun, George Carrette <gjc@mittech.com>)

list: scheme@mc.lcs.mit.edu

repository: nexus.yorku.ca:pub/scheme

Scheme-Linda - Ulf Dahlen, U Edinburgh, 1990. On the Computing Surface and the Symmetry. "Scheme-Linda", U. Dahlen et al, EPCC-TN-90-01 Edinburgh 1990.

School - Smalltalk-like but strongly typed, with separate inheritance hierarchies for types and classes. "Types in School", N. de la Rocque Rodriguez et al, *SIGPLAN Notices* 28(8):81-89 (Aug 1993)

Schoonschip - (Dutch for "beautiful ship") M. Veltman, CERN, 1964. Symbolic math, especially High Energy Physics. Algebra only, no derivatives. Originally implemented in CDC-6600 and 7600 assembly

language, currently in 680x0 assembly language. Latest versions (Oct 1991) include Amiga, Atari ST, Sun 3/60, NeXT.  
info: David Williams <dnw@williams.physics.lsa.umich.edu>  
ftp://archive.umich.edu/physics/schip

SCL -

1. System Control Language. Command language for the VME/B operating system on the ICL2900. Block structured, strings, superstrings (lists of strings), int, bool, array types. Can trigger a block whenever a condition on a variable value occurs. Macros supported. Commands are treated like procedure calls. Default arguments. "VME/B SCL Syntax", Intl Computers Ltd 1980.

2. Symbolic Communication Language. Designed primarily for the manipulation of symbolic formulas. Featured pattern matching (which was partly the inspiration for SNOBOL), string operations in buffers, and automatic storage management. "A Language for Symbolic Communication", C.Y. Lee et al, Tech Mem 62-3344-4, Bell Labs, Sept 1962.

Scode - Internal representation used by the Liar compiler for MIT Scheme.

SCOOP - Structured Concurrent Object-Oriented Prolog. "SCOOP, Structured Concurrent Object-Oriented Prolog", J. Vaucher et al, in ECOOP '88, S. Gjessing et al eds, LNCS 322, Springer 1988, pp.191-211.

SCOOPS - Scheme Object-Oriented Programming System. TI, 1986. Multiple inheritance, class variables.

ftp://altdorf.ai.mit.edu/archive/scheme-library/unsupported/CScheme

SCRAP - CSIR, Pretoria, South Africa, late 1970's. Ran on Interdata and Perkin-Elmer computers. In use until the late 1980's.

Scratchpad I - Richard Jenks, Barry Trager, Stephen M. Watt & Robert S. Sutor, IBM Research, ca 1971. General-purpose language originally for interactive symbolic math. It features abstract parametrized datatypes, multiple inheritance and polymorphism. Implementations for VM/CMS and AIX. "Scratchpad User's Manual", RA 70, IBM (June 1975). Version: Scratchpad II. "Scratchpad II Programming Language Manual", R.D. Jenks et al, IBM, 1985. (See AXIOM.)

Scratchpad II Newsletter: Computer Algebra Group, TJWRC, Box 218, Yorktown Hts, NY 10598.

Screamer - An extension of Common Lisp providing nondeterministic backtracking and constraint programming.

ftp://ftp.ai.mil.edu/pub/screamer.tar.Z

Screenwrite - Simple query language. Honeywell late 70's, Level 6 minis.

Scribe - Brian Reid. A text-formatting language.

SCRIPT -

1. Early system on IBM 702. Listed in CACM 2(5):16 (May 1959).

2. Real-time language. "A Communication Abstraction Mechanism and its Verification", N. Francez et al, Sci Comp Prog 6(1):35-88 (1986).

ScriptX - Kaleida Labs. Object-oriented, dynamic, time-based, multithreaded multiplatform language for interactive multimedia. Available soon for Windows and Macintosh.  
info: kaleida.dev@kaleida.com  
<http://web.kaleida.com/official/>

SCROLL - String and Character Recording Oriented Logogrammatic Language.  
"SCROLL - A Pattern Recording Language", M. Sargent, Proc SJCC 36 (1970).

scsh - An extension language?

SDF - Syntax Definition Formalism. CWI. Language for lexical and syntactic specification. In Algebraic Specification, J.A. Bergstra et al eds, ACM Press 1989, Chap 6. "The Syntax Definition Formalism SDF - Reference Manual", J. Heering et al, Centre for Math & CS, Amsterdam, to appear.

SDL -

1. System Software Development Language. System software for the B1700. "System Software Development Language Reference Manual", 1081346, Burroughs Corp (Dec 1974).

2. Specification and Description Language. CCITT. Specification language with both graphical and character-based syntaxes for defining interacting extended finite state machines. Used to specify discrete interactive systems such as industrial process control, traffic control, and telecommunication systems. Proc Plenary Assembly, Melbourne 14-25 Nov 1988, Fasc X.1, CCITT. "Telecommunications Systems Engineering Using SDL", R. Saracco et al, N-H 1989. Available from Verilog, MD. (See XDL).

3. Shared Dataspace Language. "A Shared Dataspace Language Supporting Large-Scale Concurrency", G. Roman et al, Proc 8th Intl Conf Distrib Comp Sys, IEEE 1988, pp.265-272.

4. Structure Definition Language. Used internally by DEC to define and generate the symbols used for VAX/VMS internal data structures in various languages.

5. System Description Language. language used by the Eiffel/S implementation of Eiffel to assemble clusters into a system. (see Lace).

SDL 92 - SDL[2] with object-orientation.

SDMS - Query language.

SEAL - Semantics-directed Environment Adaptation Language.  
<ftp://ftp.cwi.nl/pub/gipe/0092b.ps.Z>

Sed - Stream editor. The input language used by the Unix stream editor.

SEESAW - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

SEL -

1. Self-Extensible Language. "SEL - A Self-Extensible Programming

Language", G. Molnar, Computer J 14(3):238-242 (Aug 1971).

2. Subset-Equational Language. Bharat Jayaraman. Declarative language combining sets and equational programming. Implemented in Quintus Prolog and C. "Towards a Broader Basis for Logic Programming", B. Jayaraman, TR CS Dept, SUNY Buffalo, 1990. "Set Abstraction in Functional and Logic Programming", F.S.K. Silbermann <fs@cs.tulane.edu> et al, ACM Proc [?] 1989.

ftp://ftp.cs.buffalo.edu/users/bharat/SEL2

Self - Small, dynamically-typed object-oriented language, based on prototypes and delegation. Objects may inherit state, and dynamically change their patterns of inheritance. Threads. "Self: The Power of Simplicity", David Ungar <ungar@sun.eng.com> et al, SIGPLAN Notices 22(12):227-242 (OOPSLA '87) (Dec 1987). First implemented: Craig Chambers, 1991 ("Self-91"). Version: 3.0 ("Self-93") in C++ and Self  
ftp://self.stanford.edu/pub/Self-3.0/\* for Sparc

list: self-interest@self.stanford.edu  
request: self-request@self.stanford.edu

SEM - Semantic specification language for COPS. "Metalanguages of the Compiler Production System COPS", J. Borowiec, in GI Fachgesprach "Compiler-Compiler", ed W. Henhapl, Tech Hochs Darmstadt 1978, pp.122-159.

Seneca - See Oberon-V

SEPIA - Standard ECRC Prolog Integrating Applications. Prolog with many extensions including attributed variables ("metaterms") and declarative coroutinging. "SEPIA", Micha Meier <micha@ecrc.de> et al, TR-LP-36 ECRC, March 1988. Version 3.1 available for Suns and VAX. (See ECRC-Prolog).  
info: sepia-request@ecrc.de

Seque - "Seque: A Programming Language for Manipulating Sequences", R.E. Griswold et al, Comp Langs 13(1):13-22 (1988).

Sequel -

1. Precursor to SQL. "System R: Relational Approach to Database Management", IBM Res Lab, San Jose, reprinted in Readings in Database Systems.

2. U Leeds. Theorem prover specification language. Pattern matching notation similar to Prolog. Compiled into Lisp. Proc ICJAI 13.  
ftp://agora.leeds.ac.uk/scs/logic/sequel\*

SESL - State and Event Specification Language. [?]

SETL - SET Language. Courant Inst, early 70's. A very high level set-oriented language. Data types include sets (unordered collections), tuples (ordered collections) and maps (collections of ordered pairs). Expressions may include quantifiers ('for each' and 'exists'). The first Ada translator was written in SETL. "Programming With Sets - An Introduction to SETL", Jacob T. Schwartz et al, Springer 1986.

SETL2 - SETL with more conventional Ada-like syntax, lexical scoping, full block structure, first-class functions and a package and library system.

Kirk Snyder <snyder@spunky.cs.nyu.edu>. "The SETL2 Programming Language", W. Kirk Snyder, Courant Inst TR 490, Jan 1990.  
ftp://cs.nyu.edu, for MS-DOS, OS/2, Mac, Unix workstations

SETL/E - See ProSet.

SETS - Set Equation Transformation System. Symbolic manipulation of Boolean equations. "Efficient Ordering of Set Expressions for Symbolic Expansion", R.G. Worrell et al, J ACM 20(3):482-488 (Jul 1973).

SEUS - R. Weyrauch et al. Language allowing functions to return multiple values. Implemented but never published. Mentioned in "Evolution of Lisp", G.L. Steele et al, SIGPLAN Notices 28(3):231-270 (March 1993).

SEXI - String EXpression Interpreter. Early name of SNOBOL.

SFD-ALGOL - System Function Description-ALGOL. Extension of ALGOL for synchronous systems. Sammet 1969, p.625.

SFL - System Function Language. Assembly language for the ICL2900. "SFL Language Definition Manual", TR 6413, Intl Computers Ltd.

SFLV - Unifies logic and functional programming. SASL+LV with unification moved from actual/formal parameter matching to equational clauses. "Static Analysis of Functional Programs with Logical Variables", G. Lindstrom in Programming Languages Implementation and Logic Programming, P. Deransart et al eds, LNCS 348, Springer 1988.

SGML - Standard Generalized Markup Language. "SGML - The User's Guide to ISO 8879", J.M. Smith et al, Ellis Harwood, 1988. ISO/IEC 8879-1986.

ftp://star.cs.vu.nl/Sgml an SGML parser  
//mailer.cc.fsu.edu/pub/sgml another SGML parser  
//ifi.uio.no/pub/SGML/SGMLS yet another SGML parser  
//ftp.th-darmstadt.de/pub/test/sgml/misc/gf\* a general formatter for many ISO DTD's including SGML, can produce ASCII or Latex output

sh - (or "Shellish"). S.R. Bourne. Command shell interpreter and script language for Unix. "Unix Time-Sharing System: The Unix Shell", S.R. Bourne, Bell Sys Tech J 57(6):1971-1990 (Jul 1978).

SHACO - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

SHADOW - Barnett & Futrelle, 1962. Syntax-directed compiler. Predecessor to SNOBOL? Sammet 1969, p.448, 605.

Sharp APL - "A Dictionary of the APL Language", K. Iverson, Pub 0402, Sharp Assocs, Toronto, 1985.

ftp://watserv1.waterloo.edu/languages/apl/sharp.apl

SHEEP - Symbolic math, especially tensor analysis and General Relativity. Inge Frick, Stockholm, late 70's to early 80's. Implemented in DEC-10 assembly language, then in several LISPs. Current version for Sun-3, based on Portable Standard LISP. "Sheep, a Computer Algebra System for General Relativity", J.E.F. Skea et al in Proc First Brazilian School on Comp Alg, W. Roque et al eds, Oxford U Press 1993, v2.

ftp://galois.maths.gmw.ac.uk/homeftp/pub/sheep

SHELL - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

Short Code or SHORTCODE - John Mauchly, 1949. Pseudocode interpreter for math problems, on Eckert and Mauchly's BINAC, later on UNIVAC I and II. Possibly the first attempt at a higher level language. Sammet 1969, p.129.

Show-And-Tell - Visual dataflow language designed for use by elementary school children. "A Visual Language for Keyboardless Programming", T. Kimura et al, TR WUCS-86-6, CS Dept Washington U, Mar 1986. "Show and Tell: A Visual Language", T.D. Kimura et al in Visual Programming Environments: Paradigms and Systems, E.P. Glinert ed, IEEE Comp Sci Press, 1990, pp.397-404.

SICStus Prolog - SICS (Swedish Inst Comp Sci), Box 1263, S-164 28 Kista, Sweden.

info: sicstus-request@sics.se  
http://www.sics.se/ps/sicstus.html  
list: sicstus-users@sics.se

SIFT - SHARE Internal FORTRAN Translator. Translation utility designed for converting FORTRAN II to FORTRAN IV. The word "sift" was often used as a verb to describe converting code from one language to another. Sammet 1969, p.153.

Sig - Signal Processing, Analysis, and Display program. This is an environment with an associated programming language. Jan Carter, Argonne Natl Lab, (312)972-7250.

SIGLA - SIGma LAnguage. Olivetti. Language for industrial robots. "SIGLA: The Olivetti Sigma Robot Programming Language", M. Salmon, Proc 8th Intl Symp on Industrial Robots, 1978, pp.358-363.

SIGNAL - Le Guernic et al, INRIA. Synchronous dataflow language. An application is a relation described as a set of equations. The compiler includes a formal calculus on sets of instants. "SIGNAL - A Data Flow-Oriented Language for Signal Processing," P. le Guernic, IEEE Trans Acoustics Speech & Signal Proc, ASSP-34(2):362-374 Apr 1986.

SIL -

1. "SIL - A Simulation Language", N. Houbak, LNCS 426, Springer 1990.

2. SNOBOL Implementation Language. Intermediate language forming a virtual machine for the implementation of portable interpreters. "The Design of Transportable Interpreters", F. Druseikis, SNOBOL4 Project Document S4D49, U Arizona (Feb 1975). Version: SIL/2.

Silage - Synchronous DSP specification language. "Silage Reference Manual, Draft 1.0", D.R. Genin & P.N. Hilfinger, Silvar-Lisco, Leuven 1989.

SIMAN - SIMulation ANalysis. C. Dennis Pegden, 1983. Language for simulations, especially manufacturing systems. "Introduction to Simulation using SIMAN", C.D. Pegden et al, McGraw-Hill 1990.

SIMCMP - A simple bootstrap language and compiler, used to compile FLUB. Implementing Software for Non-Numeric Applications, W. M. Waite, P-H 1973.

Implemented in Fortran.

SIML/I - Simulation language, descendant of ASPOL. "The Simulation Language SIML/I", M.H. MacDougall, Proc NCC 1979, pp.39-44.

Simone - A. Hoare et al. Simulation language based on Pascal. "Quasiparallel Programming", W.H. Kaubisch et al, Soft Prac & Exp 6:341-356 (1976).

SIMPAC - Early simulation language with fixed time steps. "Simpac User's Manual", R.P. Bennett et al, TM-602/000/000, Sys Devel Corp, Apr 1962.

SIMPAS - Event scheduling language, implemented as Pascal preprocessor. "SIMPAS - A Simulation Language Based on Pascal", R.M. Bryant in Proc 1980 Winter Sim Conf, T.I Oren et al eds, pp.559-572.

SIMPL -

1. Simulation language, descendant of OPS-4, compiled into PL/I on Multics. "The SIMPL Primer", M.W. Jones et al, Oct 1971.

2. Single Identity Micro Programming Language. C. Ramamoorthy et al, UC Berkeley, 1974. An Algol 60-like microprogramming language, the first to allow sequential specification of horizontal microprograms. Compiler in SNOBOL4. C.V. Ramamoorthy et al, "A High Level Language for Horizontal Microprogramming", IEEE Trans Comput C-23(8):791-801 (Aug 1974).

SIMPLE -

1. Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

2. Simulation of Industrial Management Problems with Lots of Equations. R.K. Bennett, 1958. Predecessor to DYNAMO, for IBM 704.

SIMPL/I - Simulation language implemented as a PL/I preprocessor. "SIMPL/I (Simulation Language Based on PL/I). Program Reference Manual", IBM SH19-5060-0 (June 1972).

SIMPL-T - Base language for a family of languages and compilers. "SIMPL-T, A Structured Programming Language", V.R. Basili, Paladin House 1976.

SIMSCRIPT - Harry Markowitz et al, Rand Corp 1963. Implemented as a Fortran preprocessor on IBM 7090. Large discrete simulations, influenced Simula. "SIMSCRIPT: A Simulation Programming Language", P.J. Kiviat et al, CACI 1973. Versions: SIMSCRIPT I.5 (CACI 1965 - produced assembly language), SIMSCRIPT II, SIMSCRIPT II.5. CACI, (619)457-9681.

SIMULA I - SIMULation LAnguage. Kristen Nygaard & Ole-Johan Dahl, designed 1962, implemented 1964. Extension to ALGOL 60 for the Univac 1107, for discrete simulation. Coroutines.

SIMULA 67 - A general-purpose successor to SIMULA I, in which the simulation support is defined in object-oriented terms. Introduced the record class, leading the way to data abstraction and object-oriented programming. Garbage collection.

SIMULA - Current version of SIMULA 67. Used as the introductory programming language at Lund Inst Tech, Sweden and U Bergen, Norway. "Object-Oriented Programming with SIMULA", Bjorn Kirkerud, A-W 1989. "Data Processing - Programming Languages - SIMULA", Swedish Standard SS 63 61 14 (1987), ISBN 91-7162-234-9, available through ANSI. Association for SIMULA Users, Royal Inst of Tech, S-100 44 Stockholm, Sweden. Simula a.s., Postboks 4403 - Torshov, N-0402 Oslo 4, Norway, versions for almost every computer.  
ftp://ftp.inria.fr/lang/simula/\*  
info: Henry Islo <hio@helios.sunet.se>  
cim@ifi.uio.no

Simulating Digital Systems - FORTRAN-like language for describing computer logic design. Sammet 1969, p.622.

SINA - "An Implementation of the Object-Oriented Concurrent Programming Language SINA", A. Tripathi et al, Soft Prac & Exp 19(3):235-256 (1989).

SIPLAN - SITE PLANning computer language. Interactive language for space planning. "Formal Languages for Site Planning", C.I. Yessios in Spatial Synthesis for Computer-Aided Design, C. Eastman ed, Applied Science Publ 1976.

Siprol - Signal Processing Language. A DSP language. "SIPROL: A High Level Language for Digital Signal Processing", H. Gethoffer, Proc ICASSP-80, 1980, pp.1056-1059.

SIR - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

Siri - Horn <Bruce.Horn@n3.sp.cs.cmu.edu>, CMU 1991. Object-oriented constraint language using a single abstraction mechanism. A conceptual blend of BETA and Bertrand. Similar to Kaleidoscope. "Constraint Patterns as a Basis for Object-Oriented Constraint Programming", B. Horn, OOPSLA '92 (Sept 1992).

SISAL - Streams and Iteration in a Single Assignment Language. James McGraw et al, U Manchester, Lawrence Livermore, DEC and CSU 1983. Single assignment language with strict semantics, automatic parallelization, efficient execution. Outputs a dataflow graph in IF1 (Intermediary Form 1). Derived from VAL, adds recursion and finite streams. Pascal-like syntax. Designed to be a common high-level language for numerical programs on a variety of multiprocessors. "A Report on the SISAL Language Project", J.T. Feo et al, J Parallel and Distrib Computing 10(4):349-366 (Dec 1990). Implementations exist for Cray X-MP, Y-MP, Cray-2, Sequent, Encore Alliant, dataflow architectures, transputers and systolic arrays.  
contact: David Cann <cann@lll-crg.llnl.gov>, Rod Oldehoeft <rro@cs.colostate.edu>.  
ftp://sisal.llnl.gov/pub/sisal  
info: sisal-info@sisal.llnl.gov

SISAL 90 - A SISAL extension with higher order functions, polymorphism.

Sketchpad - I. Sutherland, 1963. Computer-aided design. Constraints using value inference. Introduced the "ring" list structure. "Sketchpad: A Man-Machine Graphical Communication System", I.E. Sutherland, MIT Lincoln Lab, TR 296 (Jan 1963). Sammet 1969, p.678.

Skim - Alain Deutsch <deutsch@poly.polytechnique.fr> et al, France. Scheme implementation with packages and other enhancements.

SKOL - FORTRAN pre-processor for COS (Cray Operating System).

SL5 - String and list processing language with expression-oriented syntax. Coroutines. "An Overview of SL5", Ralph E. Griswold, SIGPLAN Notices 12(4):40-50 (Apr 1977).

SLAM -

1. Simulation Language for Alternative Modeling. Simulation language, descendant of GASP. Implemented as Fortran preprocessor.

2. Continuous simulation language. "SLAM - A New Continuous Simulation Language", N.A. Wallington et al, in SCS Simulation Council Proc Series: Toward Real-Time Simulation (Languages, Models and Systems), R.E. Crosbie et al eds, 6(1):85-89 (Dec 1976).

SLANG -

1. R.A. Sibley. CACM 4(1):75-84 (Jan 1961).

2. Set LANGUAGE. Jastrzebowski, ca 1990. C extension with set-theoretic data types and garbage collection. "The SLANG Programming Language Reference Manual, Version 3.3", W. Jastrzebowski <wojtek@pascal.math.yale.edu>, 1990.

3. Structured LANGUAGE. Michael Kessler, IBM. A language based on structured programming macros for IBM 370 assembly language. "Project RMAG: SLANG (Structured Language) Compiler", R.A. Magnuson, NIH-DCRT-DMB-SSS-UG105, NIH, DHEW, Bethesda, MD 20205 (1980).

4. "SLANG: A Problem Solving Language for Continuous-Model Simulation and Optimization", J.M. Thames, Proc 24th ACM Natl Conf 1969.

S-Lang - Stack-based postfix language, used in the JED editor.

info: John E. Davis <davis@amy.tch.harvard.edu>  
ftp://amy/tch.harvard.edu/pub/slang/\*

SLIM - A VLSI language for translating DFA's into circuits. J.L. Hennessy, "SLIM: A Simulation and Implementation Language for VLSI Microcode", Lambda, Apr 1981, pp.20-28.

SLIP - Symmetric List Processor. J. Weizenbaum, early-60's. Language for list processing using doubly-linked lists. Originally embedded in FORTRAN, later also embedded in MAD and ALGOL. "Symmetric List Processor", J. Weizenbaum CACM 6:524-544(1963). Sammet 1969, p.387.

SLIPS - "An Interpreter for SLIPS - An Applicative Language Based on Lambda-Calculus", V. Gehot et al, Comp Langs 11(1):1-14 (1986).

SLLIC - Intermediate language developed at HP. An infinite-register version of the Precision Architecture instruction set?

Sloop - "Parallel Programming in a Virtual Object Space", S. Lucco, SIGPLAN

Notices 22(12):26-34 (OOPSLA '87) (Dec 1987).

SMALGOL - Small ALGOL. Subset of ALGOL 60. "SMALGOL-61", G.A. Bachelor et al CACM 4(11):499-502 (Nov 1961). Sammet 1969.

SMALL -

1. Functional, lazy, untyped. "SMALL - A Small Interactive Functional System", L. Augustsson, TR 28, U Goteborg and Chalmers U, 1986.

2. A toy language used to illustrate denotational semantics. "The Denotational Description of Programming Languages", M.J.C. Gordon, Springer 1979.

Small-C - A subset of C. Compiler source in C producing 8080 code in Dr Dobb's J, May 1980 and Dec 1982. "The Small-C Handbook," James Hendrix, Reston 1984.

[ftp://apple.com/ArchiveVoll/unix\\_lang](ftp://apple.com/ArchiveVoll/unix_lang)

Smalltalk - Software Concepts Group, Xerox PARC, led by Alan Kay, early 70's. Took the concepts of class and message from Simula-67 and made them all-pervasive, the quintessential object-oriented language. Innovations included the bitmap display, windowing system and use of mouse. Early versions: Smalltalk-72, Smalltalk-74, Smalltalk-76 (inheritance taken from Simula, and concurrency), and Smalltalk-78. "The Smalltalk-76 Programming System Design and Implementation", D.H. Ingalls, 5th POPL, ACM 1978, pp.9-16.

Smalltalk-80 - "Smalltalk-80: The Language and Its Implementation" ("The Blue Book"), Adele Goldberg et al, A-W 1983. BYTE 6(8) (Aug 1981).

archive: <st.cs.uiuc.edu/pub/ISA>

<ftp://prep.ai.mit.edu/pub/gnu> GNU Smalltalk v1.1

archive: <st.cs.uiuc.edu/pub/MANCHESTER>

mail server: [goodies-lib@r5.cs.man.ac.uk](mailto:goodies-lib@r5.cs.man.ac.uk)

SmalltalkAgents - QKS. Smalltalk with closures. [?]

Smalltalk DB - Formerly OPAL. Language of the object-oriented database GemStone. "Making Smalltalk a Database System", G. Copeland et al, Proc SIGMOD'84, ACM 1984, pp.316-325.

Smalltalk/V - First widely available version of Smalltalk, for PC, Mac. Digital, 1986.

SmallVDM - "SmallVDM: An Environment for Formal Specification and Prototyping in Smalltalk", in Object Oriented Specification Case Studies, K. Lano et al eds, P-H 1993.

SmallWorld - Object-oriented language. "SW 2 - An Object-based Programming Environment", M.R. Laff et al, IBM TJWRC, 1985.

SMART - For MS-DOS?

SMIL - Machine language for a Swedish computer.

<ftp://wuarchive.wustl.edu/mirrors/info-mac/lang/smil-emulator.hqx>

SML -

1. Standard ML. R. Milner <rm@lfcs.edinburgh.ac.uk> ca. 1984. Aimed to unify the dialects of ML, has evolved into a robust general-purpose language. Functional, with imperative features. Environment based, strict. Adds to ML the call-by-pattern of Hope, recursive data types, reference types, typed exceptions, and modules. (The "core" language excludes the modules.) "A Proposal for Standard ML", R. Milner, ACM Symp on LISP and Functional Prog 1984, pp.184-197.

Implementations:

SML/NJ - Standard ML of New Jersey, Version 0.75 ftp://cs.yale.edu/pub/ml and //research.att.com/dist/ml

Edinburgh SML - Core language only. Byte-code interpreter in C. Ported to Amiga, Atari, Archimedes and IBM PC.

info: <lfcs@ed.ac.uk>

ftp://ftp.dcs.ed.ac.uk/pub/edml/EDML4 EdML V 0.44

Moscow SML - Sergei Romanenko <sergei-romanenko@refal.msk.su> Core language only. Based on CAML Light and the ML Kit. For Unix, MS-DOS, and MS Windows.

ftp://dina.kvl.dk/pub/Peter.Sestoft/mosml/\*

info: Peter Sestoft <sestoft@id.dtu.dk>

POPLOG ML - U Sussex. For the Poplog system.

Poly/ML - Implemented in Poly[1], for MC68020 and SPARC. Abstract Hardware Ltd <ahl@ahl.co.uk>.

ANU ML - Aust Natl U. For MC68020, VAX and Pyramid.

Micro ML - U Umea, Sweden. Interpreter of an ML subset, for MS-DOS.

list: sml-request@cs.cmu.edu

ftp://sbcs.sunysb.edu, a lazy version

sml2c - portable, written in SML. Language extensions include first-class continuations, asynchronous signal handling.

info: <david.tarditi@cs.cmu.edu>

ftp://dravido.soar.cs.cmu.edu/usr/nemo/sml2c:sml2c.tar.Z

2. Small Machine Language. Barnes, ICI 1969. Real-time language, an ALGOL variant, and the predecessor of RTL. "SML User's Guide", J.G.P. Barnes, ICI, TR JGPB/69/35 (1969).

SML# - An extension of SML/NJ with polymorphic field selection and nondestructive field update. "A Compilation Method for ML-style Polymorphic Record Calculi", A. Ohori, POPL 1992.

ftp://ftp.cis.upenn.edu/pub/sml#/\*

SML/NJ - Standard ML of New Jersey. An implementation of SML by Andrew Appel at Princeton <Appel@princeton.edu> and Dave MacQueen at AT&T <dbm@research.att.com>. "Standard ML of New Jersey", A. Appel et al, "Proc Third Intl Symp on Prog Lang Impl and Logic Programming", LNCS Springer 1991. Versions for Unix, Mac

ftp://research.att.com/dist/ml/\* Version 0.93

list: sml@cs.cmu.edu

SMoLCS - Specification metalanguage used for a formal definition of Ada. "An Introduction to the SMoLCS Methodology", E. Astesiano, U Genova 1986.

SMP - Steven Wolfram's earlier symbol manipulation program, before he turned to Mathematica. "SMP Handbook", C. Cole, S. Wolfram et al, Caltech 1981.

SNAP -

1. Early interpreted text-processing language for beginners, close to basic English. "Computer Programming in English", M.P. Barnett, Harcourt Brace 1969. [For IBM 360?]

2. "Some Proposals for SNAP, A Language with Formal Macro Facilities", R.B. Napper, Computer J 10(3):231-243 (1967). [same as 1?]

SNOBOL - StriNg Oriented symBolic Language. David Farber, Ralph Griswold & I. Polonsky, Bell Labs 1962-3. String processing language for text and formula manipulation. "SNOBOL, A String Manipulating Language", R. Griswold et al, J ACM 11(1):21 (Jan 1964).

SNOBOL2 - Brief existence, featured built-in functions, but not programmer-defined ones. "SNOBOL2", D.J. Farber, R.E. Griswold and I.P. Polonsky, TR Bell Labs, Apr 1964.

SNOBOL3 - 1965. SNOBOL with user-defined functions. SNOBOL 6.3 compiler for PDP-6 and PDP-10, written in SNOBOL. "The SNOBOL3 Programming Language", D.J. Farber et al, Bell Sys Tech J 45(6):895-944 (Jul 1966).

SNOBOL4 - Griswold et al, 1967. Quite distinct from its predecessors. Declarative with dynamic scope. Patterns are first-class data objects that can be constructed by concatenation and alternation. Success and failure used for flow control. Delayed (unevaluated) expressions can be used to implement recursion. Table data type. Strings generated at run-time can be treated as programs and executed. "The SNOBOL4 Programming Language", Ralph E. Griswold et al, P-H 1971. (See FASBOL, SPITBOL, SIL).  
ftp://apple.com/ArchiveVoll1/unix\_lang

    //cs.uarizona.edu//pub/snobol4/budne/\* Unix port of the original macro implementation

SITBOL - "SITBOL Version 3.0", J.F. Gimpel, TRS4D30b, Bell Labs 1973.

SNOOPS - Craske, 1988. An extension of SCOOPS with meta-objects that can redirect messages to other objects. "SNOOPS: An Object-Oriented language Enhancement Supporting Dynamic Program Reconfiguration", N. Craske, SIGPLAN Notices 26(10): 53-62 (Oct 1991).

SO 2 - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

SOAP - Symbolic Optimal Assembly Program. IBM 650 assembly language. "Optimal" refers to rearranging instructions on slowly rotating drum memory. Listed in CACM 2(5):16 (May 1959). Versions: SOAP I, SOAP II, CASE SOAP III.

SOAR - State, Operator And Result. A. Newell, early 80's. A general problem-solving production system architecture, intended as a model of human intelligence. Originally implemented in LISP and OPS5, currently in Common Lisp. "The SOAR Papers", P.S. Rosenbloom et al eds, MIT Press 1993.  
Version: Soar6  
info: soar@cs.cmu.edu  
doc: soar-doc@cs.cmu.edu

SOCRATIC - [Not a language?] Bolt, Beranek & Newman. Early interactive learning system. Sammet 1969, p.702.

SODA - Symbolic Optimum DEUCE Assembly Program. Symbolic assembler for a one-level storage virtual machine for the English ELeCtric DEUCE. "SODA Manual of Operation", R. C> Brigham & C. G. Bell, School of Elec Eng, U New S Wales, Sydney, NSW (1958).

SODAS - D.L. Parnas & J.A. Darringer. Proc FJCC 31:449-474, AFIPS (Fall 1967).

SOHIO - Early system on IBM 705. Listed in CACM 2(5):16 (May 1959).

SOL -

1. Simulation Oriented Language. Knuth & McNeley. ALGOL extension for discrete simulation. "SOL - A Symbolic Language for General Purpose System Simulation", D.E. Knuth et al, IEEE Trans Elec Comp, EC-13(4):401-408 (Aug 1964). Sammet 1969, p.656.

2. Second-Order Lambda calculus. A typed lambda calculus. "Abstract Types have Existential Type", J. Mitchell et al, 12th POPL, ACM 1985, pp.37-51.

3. Semantic Operating Language. Language for manipulating semantic networks for building cognitive models, particularly for natural language understanding. "Explorations in Cognition", D.A. Norman et al, W.H. Freeman 1974.

SOLO - Name inspired by SOL[3] + LOGO. A variant of LOGO with primitives for dealing with semantic networks and pattern matching rather than lists. "A User-Friendly Software Environemnt for the Novice Programmer", M. Eisenstadt <marc@open.ac.uk>, CACM 27(12):1056-1064 (1983).

Solve - Parallel object-oriented language. "Message Pattern Specifications: A New Technique for Handling Errors in Parallel Object-Oriented Systems", J.A. Purchase et al, SIGPLAN Notices 25(10):116-125 (OOPSLA/ECOOP '90) (Oct 1990).

SP - Simplicity and Power. Prolog-like. "Simplicity and Power - Simplifying Ideas in Computing", J.G. Wolff, Computer J 33(6):518-534 (Dec 1990).

SPADE - Specification Processing And Dependency Extraction. Specification language. G.S. Boddy, ICL Mainframes Div, FLAG/UD/3DR.003

SPAR - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

SPARK - Southampton U and Program Validation, Ltd. An annotated subset of Ada for formal verification. Eliminates derived types, anonymous types, access types, variant records, recursion, generics, tasking and exceptions. "SPARK - An Annotated Ada Subset for Safety-Critical Programming", Tri-Ada '90.  
info: pvl@cix.compulink.co.uk

SPARKS - FORTRAN superset, used in Fundamentals of Data Structures, E. Horowitz & S. Sahni, Computer Science Press 1976.

Speakeasy - Simple array-oriented language with numerical integration and

differentiation, graphical output, aimed at statistical analysis.  
"Speakeasy", S. Cohen, SIGPLAN Notices 9(4), (Apr 1974). "Speakeasy-3 Reference Manual", S. Cohen et al. 1976.

Spec - Specification language. Expresses black-box interface specifications for large distributed systems with real-time constraints. It incorporates conceptual models, inheritance and the event model. A descendant of MSG.84. "An Introduction to the Specification Language Spec", V. Berzins et al, IEEE Software 7(2):74-84 (Mar 1990).

SPECIAL - SRI specification language. [HDM?] "SPECIAL - A Specification and Assertion Language", L. Robinson et al, TR CSL-46, SRI, Jan 1987.

SPECOL - "SPECOL - A Computer Enquiry Language for the Non-Programmer", B.T. Smith, Computer J 11:121 (1968).

SPEED - Early system on LGP-30. Listed in CACM 2(5):16 (May 1959).

Speedcoding - John Backus, 1953. A pseudocode interpreter for math on IBM 701, IBM 650. Sammet 1969, p.130.

Speedcoding 3 - Listed in CACM 2(5):16 (May 1959).

SPEEDEX - Early system on IBM 701. Listed in CACM 2(5):16 (May 1959).

SP/k - Subset PL/I, k=1..8. A series of PL/I subsets, simplified for student use. "SP/k: A System for Teaching Computer Programming", R.C. Holt et al, CACM 20(5):301-309 (May 1977).

SPG - System Program Generator. A compiler-writing language. "A System Program Generator", D. Morris et al, Computer J 13(3) (1970).

SPIT - Language for IBM 650. (See IT).

SPITBOL - SPeedy ImplemenTation of snOBOL. "Macro SPITBOL - A SNOBOL4 Compiler", R.B.K. Dewar et al, Soft Prac & Exp 7:95-113, 1971. Current versions: SPITBOL-68000, Sparc SPITBOL from Catspaw Inc, (719)539-3884.

SPL -

1. Synchronous Programming Language. A DSP language. "Introduction to the SPL Compiler", Computalker Consultants, 1986.

2. Space Programming Language. Realtime language used by the US Air Force for aerospace software. Aka SPL/J6. Similar to JOVIAL. "Space Programming Language Development", SAMSO TP 70-325, System Development Corp (Sep 1970). (See CLASP).

3. System Programming Language. HP, 1977. An ALGOL-like language for the HP3000 computer allowing inline assembly code. MPE, the OS for the HP3000 was written in SPL. Pub.No.30000-90024, HP.

4. Systems Programming Language. PRIME Computer, 80's. A variant of PL/I used on PRIME computers. PL/I subset G, less I/O plus a few extensions. SPL User's Reference Guide, Prime. (See PL/P.)

5. Systems Programming Language. D.B. Wortman, U Toronto. Philips Data

Sys, Netherlands, 1971. PL/I subset/extension for the P1000. Symbolic constants, pointer arithmetic, inline assembly code. Used to implement compilers, operating systems, and database. "Experiences With SPL", J. Klunder in Machine Oriented Higher Level Languages, W. van der Poel, N-H 1974, pp.385-393. [can 4 and 5 possibly be the same?]

6. Student Programming Language. A translator-interpretor for a dialect of PL/I.  
SPLash! -

1. Software Research Northwest, 1987. Compiler for SPL[3].

2. Systems Programming Language for Software Hackers. Mentioned in TeX for the Impatient, Paul W. Abrahams, A-W 1990.

SPL/I - Signal Processing Language I. Intermetrics. General language designed for acoustic signal processing. Graphics and multiprocessing features. "SPL/I Language Reference Manual", M.S. Kosinski, Intermetrics Report 172-1 (July 1976).

SPLINTER - PL/I interpreter with debugging features. Sammet 1969, p.600.

Split-C - Parallel extension of C for distributed memory multiprocessors. Aims to provide efficient low-level access to the underlying machine.  
ftp://ftp.cs.berkeley.edu/ucb/CASTLE/Split-C (for the CM5)  
info: split-c@boing.cs.berkeley.edu

SPLX - Specification Language for Parallel cross-product of processes and sequential modules. "Parallel Module Specification on SPLX", C.F. Nourani, SIGPLAN Notices 27(1):114-115 (Jan 1992).

SPM - Sequential Parlog Machine. Language of a virtual machine for Parlog implementation.

Spool - Object-oriented logic. "An Experience with a Prolog Based Language", K. Fukunaga et al, SIGPLAN Notices 21(11):224-231 (Nov 1986) (OOPSLA '86).

SPRING - String PROCESSING language. "From SPRING to SUMMER: Design, Definition and Implementation of Programming Languages for String Manipulation and Pattern Matching", Paul Klint, Math Centre, Amsterdam 1982.

SPRINT - List processing language involving stack operations. "SPRINT - A Direct Approach to List Processing Languages", C.A. Kapps, Proc SJCC 30 (1967). Sammet 1969, p 462.

SPS - Symbolic Programming System. Assembly language for IBM 1620.

SPSS - Statistical Programs for the Social Sciences. "SPSS X User's Guide", SPSS Inc. 1986.

SPUR - Early system on IBM 650. Listed in CACM 2(5):16 (May 1959).

Squiggol - See BMF.

SQL - Structured Query Language. IBM, 1970's, for use in System R. The de

facto standard relational database interface language, often embedded in other programming languages. "A Guide to the SQL Standard," C.J. Date, A-W 1987.

SQL Module Language. Used to interface other languages (Ada, C, COBOL, etc) to SQL-based DBMS's. ANSI standard. Version: Ada/SAME by Informix (SAME=Standard ANSI Module language with Extensions.)

Square - Query language, precursor to SQL. "Specifying Queries as Relational Expressions: The SQUARE Data Sublanguage", R.E. Boyce et al, CACM 18(11):621-628 (Nov 1975).

Squeak - "Squeak: A Language for Communicating with Mice", L. Cardelli et al, Comp Graphics 19(3):199-204 (July 1985) (See Newsqueak).

SR - Synchronizing Resources. A language for concurrent programming. A 'resource' is the encapsulation of a process and its variables, and can be separately compiled. Supports local and remote procedure call, rendezvous, message passing, multicast, dynamic process creation, semaphores, and shared memory. "An Overview of the SR Language and Implementation", G. Andrews, ACM TOPLAS 10:51-86 (Jan 1988). "The SR Programming Language: Concurrency in Practice", G.R. Andrews et al, Benjamin/Cummings 1993, ISBN 0-8053-0088-0. Version: 2.3  
ftp://cs.arizona.edu/sr/sr.tar.Z  
info: info-sr-request@cs.arizona.edu

SRC Modula-3 - From DEC/SRC, Palo Alto, CA. "Modula-3 Report (revised)" Luca Cardelli et al.  
ftp://gatekeeper.dec.com

SRDL - Small algebraic specification language, allows distfix operators. "A Constructive Method for Abstract Algebraic Software Specification", H. Klaeren, Theor Computer Sci 30, pp.134-204, 1984.

Srl -

1. Bharat Jayaraman. "Towards a Broader Basis for Logic Programming", B. Jayaraman, TR CS Dept, SUNY Buffalo, 1990.

2. Schema Representation language. "SRL/2 Users Manual", J.M. Wright et al, Robotics Inst, CMU, 1984.

3. Structured Robot Language. C. Blume & W. Jacob, U Karlsruhe.

SSL -

1. Synthesizer Specification Language. Specification language based on term algebra and attribute grammar. Used by the Synthesizer Generator, a generator for language-based editors such as the Cornell Program Synthesizer. "Generating Language Based Environments", T. Reps, MIT Press 1984.

2. Syntax Semantic Language. R.C. Holt & J. Cordy. A specification language for recursive descent parsers. "An Introduction to S/SL: Syntax/Semantic Language", R.C. Holt et al, ACM TOPLAS 4(2):149-178 (Apr 1982).

ftp://ftp.cs.toronto.edu/pub/ssl.tar.Z

STAB-11 - "The Translation and Interpretation of STAB-11", A.J.T. Colin et al, Soft Prac & Exp 5(2):123-138 (Apr 1975).

STAC - Storage Allocation and Coding Program. Symbolic macro-assembler for the English Electric DEUCE. "DEUCE STAC Programming Manual", DEUCE News No. 38, Report K/AA y 1 DEUCE Library Service, Data Processing and Control Systems Division, English Electric Company, Kidsgrove, June.

STAGE2 - A macro language. "The Mobile Programming System: STAGE2", W. M. Waite, CACM 13:415 (1970). Implemented in FLUB.

Standard Lisp - A. Hearn. Subset of Lisp 1.5 developed primarily for implementing REDUCE. Replaced by Portable Standard LISP. "Standard LISP Report", J. Marti et al, SIGPLAN Notices 14(10):48-58 (Oct 1979).

Standard ML - See SML.

STAR 0 - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May 1959).

StarLISP - See \*LISP.

StarMOD - See \*MOD.

Starset - Portable storage/retrieval language for distributed databases. "Starset programming Language", M.M. Gilula et al, Nauka, Moscow 1991, ISBN 5-02-006831-4.

Statemate - Language for building finite state machines. [?]

Steelman - DoD, June 1978. Fifth and last of the series of DoD requirements that led to Ada. "Steelman Requirements for High Order Programming Languages", AD-A059 444, US Dept of Defense, June 1978. (See Strawman, Woodenman, Tinman, Ironman).

STENSOR - L. Hornfeldt, Stockholm, mid-80's. Symbolic math, especially General Relativity. Implemented on top of SHEEP and MACSYMA.

Sticks&Stones - Hardware description language. Functional, polymorphic, loosely based on ML. "An Algebraic Approach to Hardware Description and Verification", L. Cardelli, Thesis, Edinburgh U, 1982. "Sticks&Stones II: A Functional Language VLSI Layout Generation Tool", Andrew Butterfield <butrfeld@cs.tcd.ie>, Thesis, Trinity College, 1990.

STIL - Statistical Interpretive Language. "STIL User's Manual", C.F. Donaghey et al, Indust Eng Dept, U Houston (Aug 1969).

STING - A parallel dialect of Scheme intended to serve as a high-level operating system for symbolic programming languages. First-class threads, and processors, and customizable scheduling policies. "A Customizable Substrate for Concurrent Languages", S. Jagannathan et al, ACM SIGPLAN Notices, 1992.  
info: suresh@research.nj.dec.com

STOIC - String Oriented Interactive Compiler. Smithsonian Astrophysical Observatory. Similar to FORTH for strings, includes many VAX-specific

items.

Stoneman - HOLWG, DoD, Feb 1980. DoD requirements that led to APSE. "Requirements for Ada Programming Support Environments: STONEMAN", US Dept of Defense, Feb 1980.

STP4 - Statistical language.

Strand -

1. AND-parallel logic programming language. Essentially flat Parlog83 with sequential-and and sequential-or eliminated. "Strand: New Concepts on Parallel Programming", Ian Foster et al, P-H 1990. Strand88, available commercially from Strand Software, Beaverton, OR. info: strand88@sstl.uucp

2. Query language, implemented on top of INGRES (an RDBMS). "Modelling Summary Data", R. Johnson, Proc ACM SIGMOD Conf 1981.

Strawman - HOLWG, DoD, Apr 1975. The first of the series of DoD requirements that led to Ada. (See Woodenman, Tinman, Ironman, Steelman).

STREAM - "STREAM: A Scheme Language for Formally Describing Digital Circuits", C.D. Kloos in PARLE: Parallel Architectures and Languages Europe, LNCS 259, Springer 1987.

STRESS - STRuctual Engineering Systems Solver. Structural analysis problems in Civil Engineering. Superseded by STRUDL. "STRESS: A User's Manual", S.J. Fenves et al, MIT Press 1964. Sammet 1969, p.612.

STROBES - Shared Time Repair of Big Electronic Systems. Computer testing. Sammet 1969, p.699.

STRUDL - STRUctured Design Language. Dynamic and finite-element analysis, steel and concrete structures. Subsystem of ICES. "ICES STRUDL-II Engineering User's Manual", R68-91, CE Dept MIT (Nov 1968) Sammet 1969, p.613.

STRUM - Algol-like microprogramming language for the Burroughs D Machine. "STRUM: A Structured Microprogram Development System for Correct Firmware", D.A. Patterson, IEEE Trans Comput C25(10):974-985 (1976).

STRUM2 - A variant of STRUM used in the V-compiler.

STSC APL - Implementation of APL by Scientific Time-Sharing Corp.

STUDENT - D.G. Bobrow 1964. Early query system. Sammet 1969, p.664.

Student PL/I - A translator-intepreter for a PL/I subset derived from SPL[6]. "Student PL/I Compiler, R.A. Vowels, RMIT, Melbourne, 1971.

STUDIO - "STUDIO - A Modular, Compiled, Actor-Oriented Language, Based Upon a Multitask Runtime System", A. Hadjadji et al, Joint Modular Languages Conference, U Ulm, Sept 1994.

Successor - Language for distributed computing derived from SR. "Successor: Refinements to SR", R.A. Olsson et al, TR 84-3, U Arizona 1984.

Sue - System language, used to write an OS for the IBM 360. Cross between Pascal and XPL. Allows type checked separate compilation of internal procedures using a program library. "The System Language for Project Sue", B.L. Clark et al, SIGPLAN Notices 6(9):79-88 (Oct 1971).

SUGAR - Westfield College, U London. Simple lazy functional language used in Principles of Functional Programming, Hugh Glaser et al, P-H 1984.

SUIF - Stanford University Intermediate Format. Register-oriented intermediate language.

SUILVEN - A microprogramming language. "Towards Machine-Independent Microprogramming", J.F. Sommerville, Euromicro J 5(4):219-224 (1979).

SUMMER - Klint & Sint, CWI late 70's. String manipulation and pattern matching. Recently used as the input and implementation language for the Dataflow Compiler Project at CWI. "An Overview of the SUMMER Programming Language", Paul Klint, 7th POPL, ACM 1980, pp.47-55.

SUMMER SESSION - Early system on MIT's Whirlwind. Listed in CACM 2(5):16 (May 1959).

SUPER - Successor to LOGLISP, based on LNF. "New Generation Knowledge Processing: Final Report on the SUPER System", J Alan Robinson et al, CASE Center TR 8707, Syracuse U, 1987.

SUPERMAC - General-purpose macro language, embeddable in existing languages as a run-time library. "SUPERMAC - A Macro Facility That can be Added to Existing Compilers", P.J. Brown, Soft Prac & Exp 10(6):431-434.

Super Pascal - Pascal variant used in Data Structures and Algorithms, A. Aho, Hopcroft & Ullman, A-W 1983. Adds nonnumeric labels, return statement, expressions as names of types.

SuperTalk - Silicon Beach Software. A superset of HyperTalk used in SuperCard.

Sure - Bharat Jayaraman. "Towards a Broader Basis for Logic Programming", B. Jayaraman, TR CS Dept, SUNY Buffalo, 1990.

SURGE - Sorter, Updater, Report Generator, Etc. IBM 704, 1959. Sammet 1969, p.8.

SweetLambda - Sugared lambda-calculus?

SYDEL - Jan Garwick, ca 1974. System language, fully typed, with inline assembly code.

SYGMA - Novosibirsk. For the BESM-6, M-220 and Minsk-22. "SYGMA, A Symbolic Generator and Macroassembler", A.P. Ershov et al, in Symbol Manipulation Languages and Techniques, D.G. Bobrow ed, N-H 1968, pp.226-246.

Sylvan - [Distributed language?]

SYMBAL - SYMBolic ALgebra. Max Engeli, late 60's. Symbolic math language

with ALGOL-like syntax. Implemented for CDC6600. "User's Manual for the Formula Manipulation Language SYMBAL", M. Engeli, TRM-8.00, Comp Ctr UT Austin, June 1968.

SymbMath - Small symbolic math package for MS-DOS. Has the ability to learn new facts. Version 2.1.1  
ftp://simtel20//calculator/sml4a.zip

SYMBOLANG - Lapidus & Goldstein, 1965. Symbol manipulating FORTRAN subroutine package for IBM 7094, later CDC6600. "Some Experiments in Algebraic Manipulation", CACM 8:501-508 (1965).

SYMBOLIC ASSEMBLY - Early system on IBM 705. Listed in CACM 2(5):16 (May 1959).

Symbolic Mathematical Laboratory - On-line system under CTSS for symbolic math. Used display screen and light pen. Sammet 1969, p.514.

Symmetric LISP - A parallel LISP in which environments are first-class objects. Implemented in Common LISP. "Parallelism, Persistence and Meta-Cleanliness in the Symmetric Lisp Processor", D. Gelernter et al, SIGPLAN Notices 22(7):274-282 (July 1987). "A Programming Language Supporting First-Class Parallel Environments", S. Jagannathan, MIT-LCS/TR 434, 1989. info: Suresh Jagannathan <suresh@research.nj.nec.com>

SYMPL - SYStems Programming Language. CDC. A derivative of Jovial. Non-reentrant block structured language with extensive bit manipulation facilities and linkable with FORTRAN. Major parts of CDC systems during the 70's were written in SYMPL.

SYN - Syntactic specification language for COPS. "Metalanguages of the Compiler Production System COPS", J. Borowiec, in GI Fachgesprach "Compiler-Compiler", ed W. Henhapl, Tech Hochs Darmstadt 1978, pp.122-159.

Synchronous C++ - Ecole Polytechnic Federale de Lausanne.

SYSLISP - System language used in the implementation of Portable Standard Lisp. Mentioned in "The Evolution of Lisp", G.L. Steele et al, SIGPLAN Notices 28(3):231-270 (Mar 1993).

T -

1. True. Rees, 1982. LISP dialect with static scope, a near-superset of Scheme. "The T Manual", Johnathan A. Rees <jar@zurich.ai.mit.edu> et al, Yale U, 1984. Maintained by David Kranz <kranz@masala.lcs.mit.edu>. ftp://ftp.ai.mit.edu/pub/systems/t3.1 Unix source

2. Functional. "T: A Simple Reduction Language Based on Combinatory Term Rewriting", Ida et al, Proc of Prog Future Generation Computers, 1988.

TABLET - Query language. "Human Factor Comparison of a Procedural and a Non-procedural Query Language", C. Welty et al, ACM Trans Database Sys 6(4):626-649 (Dec 1981).

TABSOL - T.F. Kavanaugh. Early system oriented language. Proc FJCC 18:117-136, AFIPS (Fall 1960). (See GECOM).

TAC - Translator Assembler-Compiler. For Philco 2000.

TACL - Tandem Advanced Command Language. Tandem, about 1987. The shell language used in Tandem computers.

TACPOL - PL/I-like language used by US Army for command and control.

TAL - Tandem Application Language. A cross between C and Pascal. Primary system programming language on Tandem computers. (Tandem has no assembler and until recently did not have C or Pascal.) Derived from SPL[3]? (see microTAL).

TALE - Typed Applicative Language Experiment. M. van Leeuwen. Lazy, purely applicative, polymorphic. Based on typed second order lambda calculus. "Functional Programming and the Language TALE", H.P. Barendregt et al, in Current Trends in Concurrency, LNCS 224, Springer 1986, pp.122-207.

TALL - TAC List Language. "TALL - A List Processor for the Philco 2000", J. Feldman, CACM 5(9):484-485 (Sep 1962).

TAO -

1. Lisp dialect with concurrency, object-orientation and logic. "Concurrent Programming in TAO - Practice and Experience", I. Takeuchi in Parallel Lisp: Languages and Systems, T. Ito et al eds, LNCS 441, Springer 1989, pp.271-299.

2. (Formerly APESE) The language of the APE100 SIMD machine.  
<http://slacvm.slac.stanford.edu:5080/FIND/FREEHEP/NAME/APESE/FULL>

TARTAN - A simpler proposed language to meet the Ironman requirements. "TARTAN - Language Design for the Ironman Requirements: Reference Manual", Mary Shaw et al, SIGPLAN Notices 13(9):36-58 (Sep 1978).

TASM - Turbo Assembler. MS-DOS assembler from Borland.

TASS - Template ASSEMBLY language. Intermediate language produced by the Manchester SISAL compiler.

TAWK - Tiny AWK.

Taxis - "A Language Facility for Designing Database-Intensive Applications", J. Mylopoulos et al, ACM Trans Database Sys 5(2):185-207 (June 1980).

TBIL - Tiny Basic Interpreter Language. Inner interpreter of Tom Pittman's set of Tiny Basics in Dr Dobb's J.

Tbl -

1. M.E. Lesk. Language for formatting tables, a preprocessor to nroff.
2. Table Building Language. Robt. Freiburghouse, MIT. Simple language which combines user-defined actions into an abstract machine. Used to build table-driven predictive parsers and code generators in the MULTICS Fortran compiler and several PL/I compilers, including VAX-11 PL/I.

"Engineering A Compiler: VAX-11 Code Generation and Optimization", P. Anklam et al, Digital Press 1977.

Tcl -

1. ("tickle") Tool Command Language. John Ousterhout, UCB. <ouster@sprite.berkeley.edu> A string language for issuing commands to interactive programs. Each application can extend tcl with its own set of commands. "Tcl: An Embeddable Command Language", J. Ousterhout, Proc 1990 Winter USENIX Conf.  
ftp://ucbvax.berkeley.edu

2. Terminal Control Language. The command language used in the Pick OS. "Exploring the Pick Operating System", J.E. Sisk et al, Hayden 1986.

3. Tymshare Conversational Language. "Interactive Languages: Design Criteria and a Proposal", R.K. Moore et al, Proc FJCC 33(1) (1968).

Tcode - Intermediate language used by the Spineless Tagless G-machine (an abstract machine based on graph reduction) designed as a target for compilation of non-strict functional languages. "The Spineless tagless G-machine", S. Peyton Jones et al, Fourth Intl Conf Func Prog Langs and Comp Arch pp.184-201, ACM Sept 1989.

TCOL - CMU. Tree-based intermediate representation produced by the PQCC compiler generator. "An Overview of the Production Quality Compiler-Compiler Projects", B.W. Leverett et al, IEEE Computer 13(8): 38-49 (Aug 1980). (See LG).

TCOL.Ada - CMU, 1980. An intermediate representation for Ada, was merged with AIDA to form Diana. "TCOL.Ada: Revised Report on an Intermediate Representation for the DOD Standard Programming Language", J.M. Newcomer et al, CMU-CS-79-128 (June 1979).

tcsh - Command language for Unix, a dialect of csh.

Tcsim - Time (Complex) Simulator. Complex arithmetic version of Tsim. ZOLA Technologies, ZOLA@Applelink.Apple.com.

TDF - Intermediate language, a close relative of ANDF. A TDF program is an ASCII stream describing an abstract syntax tree. "TDF Specification", Defence Research Agency/Electronics Division, Malvern UK, +44 684 895314. info: Nic Peeling <peeling@hermes.mod.uk>

TDFL - Dataflow language. "TDFL: A Task-Level Dataflow Language", P. Suhler et al, J Parallel Dist Comp 9(2):103-115 (Jun 1990).

TECO - Text Editor and COrrector. (Originally "Tape Editor and COrrector"). Macro language for text editing, screen handling and keyboard management. Has a reputation for being cryptic and hard to learn. (TECO programs are said to resemble line noise.) The first EMACS editor was written in TECO.  
ftp://usc.edu/pub/teco, for VAX/VMS, Unix, MS-DOS, Mac, Amiga

TELCOMP - Variant of JOSS. Sammet 1969, p.217.

Telescript - General Magic. [?]

Telon - by Panasophic [?]

TELOS -

1. The EuLisp Object System. The object-oriented core of EuLisp. Incorporates ideas from CLOS, ObjVLisp and OakLisp. Total merging of types with classes, and message-passing with normal function application. (See EuLisp).

2. Pascal-based AI language. "Design Rationale for TELOS, a Pascal-based AI Language", Travis et al, SIGPLAN Notices 12(8) (Aug 1977).

TELSIM - Busch, ca 1966. Digital simulation. Sammet 1969, p.627.

TempLog - A clausal subset of first-order temporal logic, with discrete time. "Temporal Logic in Programming", M. Abadi et al, Intl Symp Logic Prog pp.4-16 (1987).

TEMPO - Simple syntax and semantics. Designed for teaching semantic and pragmatic aspects of programming languages. "TEMPO: A Unified Treatment of Binding Time and Parameter Passing Concepts in Programming Languages", N.D. Jones et al, LNCS 66, Springer 1978.

Tempura - Language based on temporal logic. "Executing Temporal Logic Programs", B. Moszkowski, Camb U Press 1986.

Ten15 - A universal intermediate language, redecessor to TDF. Polymorphic? "Ten15: An Overview", P. Core et al, Royal Signals Radar Establishment TR 3977 (Sept 1986).

TERMAC - Interactive matrix language. "Users Guide to TERMAC", J.S. Miller et al, MIT Dec 1968.

Terse - Language for decryption of hardware logic. "Hardware Logic Simulation by Compilation", C. Hansen, 25th ACM/IEEE Design Automation Conf, 1988.

TeX - Donald Knuth, 1978. Language for formatting and typesetting text, especially mathematical formulas. Macro extensible. Version 3.0, March 1990.

TeX-78 - An old (incompatible) version of TeX.

TeX-82 - The version of TeX described in The TeXbook, Donald Knuth, A-W 1984.

<ftp://labrea.stanford.edu/tex/TeX3.14.tar.Z>

Well-known extensions to TeX:

LaTeX - Lamport TEX. Incorporates document styles. "LaTeX, A Document Preparation System", Leslie Lamport <lamport@gatekeeper.dec.com>, A-W 1986.

MuTeX - for typesetting music.

<ftp://nic.stolaf.edu/pub/mutex/MuTeX.tar.Z>

PiCTeX - for pictures.

<ftp://june.cs.washington.edu/tex/PiCTeX.tar.Z>

SliTeX - for slide preparation. Distributed with LaTeX.

TFDL - "TFDL : A Task-level Dataflow Language", P.A. Suhler et al, J Parallel and Distrib Comput 9:103-115 (1990).

TGS-II - Translator Generator System. Contained TRANDIR. Sammet 1969, p.640.

THEO - Frame language. "Theo: A Framework for Self-Improving Systems", Mitchell et al, in Architectures for Intelligence, K. VanLehn ed, Erlbaum, 1989.

Theseus - Based on Euclid, never implemented. "Theseus - A Programming Language for Relational Databases", J.E. Shopiro, ACM Trans Database Sys 4(4):493-517 (Mar 1979).

ThingLab - Simulation system written in Smalltalk-80. Solves constraints using value inference. "The Programming Language Aspects of ThingLab, A Constraint-Oriented Simulation Laboratory", A. Borning, ACM TOPLAS 3(4):353-387 (Oct 1981). Version: ThingLab II.

Tinman - HOLWG, DoD, Jan 1976. Third of the series of DoD requirements that led to Ada. (See Strawman, Woodenman, Ironman, Steelman).

tinman+ - Macro language for Apple ][? Published in DDJ?

TINT - Interpreted version of JOVIAL. Sammet 1969, p.528.

Tiny - Concurrency through message-passing to named message queues.

TIP - Texas Instruments Pascal.

TIPL

1. Teach IPL. Interpretive IPL teaching system. Sammet 1969, p.393.
2. Dialect of IGL.[?]

TK!Solver - Software Arts 1983. Numerical constraint-oriented language. "The TK!Solver Book", M. Konopasek et al, McGraw-Hill 1984.

TL0 - Thread Language Zero. The instruction set of the TAM (Threaded Abstract Machine), used to implement Id. "Fine-grain Parallelism with Minimal Hardware Support", David Culler et al, SIGPLAN Notices 26(4):164-175 (ASPLOS-IV Proc) (Apr 1991).

TL1 - Transaction Language 1. Bellcore. A subset of CCITT's MML with simpler syntax. Similar to USL[2]. For use in communications between telephone operating systems and remote network test equipment. OTGR, TR-TSY-000439, section 12, Bellcore. (1-800-521-CORE)

TL/I - An intermediate language for Turing machines. "Examples of Formal Semantics", D. Knuth in Symp on Semantics of Algorithmic Languages, E. Engeler ed, LNM 188, Springer 1971, pp.212-235.

TMDL - Target-Machine Description Language. Machine-description language used in the Graham-Glanville code generator. "Table-Driven Code Generation", S.L. Graham, IEEE Computer 13(8):25-34 (Aug 1980).

TMG - TransMoGrifier. Early language for writing recursive descent compilers. Macro'd from the IBM 1604 to the 709 to the 7094 to the GE635, where it was used by McIlroy and Morris to write the EPL compiler for

Multics. "TMG - A Syntax-Directed Compiler", R.M. McClure, Proc ACM 20th Natl Conf (1965). Sammet 1969, p.636.

TOK - Referred to in Ursula K. LeGuin's "Always Coming Home." Seems to be similar to the original BASIC.

Toronto Euclid - The standard dialect of Euclid, as compared to Ottawa Euclid.

TPL -

1. Table Producing Language. "The Bureau of Labor Statistics Table Producing Language (TPL)", R.C. Mendelssohn, Proc ACM Annual Conf (1974).

2. Fleming Nielson. Concurrent, functional. [?]

3. Terminal Programming Language. Texas Inst, late 70's. Used on the TI-990/1 Small Business Computer and the TI-771 Intelligent Terminal.

TPS - Tree Pruning System. "An Adaptive Tree Pruning System: A Language for Programming Heuristic Tree Searches", E.W. Kozdrowicki, Proc ACM 23rd Natl Conf 1968.

TPU - Text Processing Utility. DEC. Language for creation of text-processing interfaces, used to implement DEC's Extensible VAX Editor (EVE).

TRAC - Text Reckoning And Compiling. Calvin N. Mooers and Peter Deutsch <Deutsch@SMLI.Eng.Sun.Com>. An interactive macrogenerator language for string manipulation. Derived ideas from Macro SAP. "TRAC: A Procedure-Describing Language for the Reactive Typewriter", Calvin N. Mooers, CACM 9(3):215-219 (Mar 1966). Rockford Research Inst, 1972. Sammet 1969, pp.448-454. Macro Processors, A.J. Cole, Cambridge U Press. Versions for PDP-1, PDP-8, PDP-10, PDP-11 (See MINT, SAM76)  
info: Preston Briggs <preston@rice.edu>  
ftp://locke.ccil.org/pub/retro Implementation in Perl

Trafola-H - A specification language for program transformations. Functional, statically typed, polymorphic, with extended constructs for pattern-matching. "Trafola-H Reference Manual", R. Heckmann et al, U Saarlandes, Saarbrucken 1991.

Traits - Early object-oriented language. Supported multiple inheritance and overriding of operations. "Traits: An Approach to Multiple-Inheritance Subclassing", G. CUrny er al, ACM SIGOA Conf on Office Automation Systems, 1982.

TRANDIR - TRANSlation DIRector. A language for syntax-directed compiling. Sammet 1969, p.640.

TRANQUIL - 1966. ALGOL-like language with sets and other extensions, for the Illiac IV. "TRANQUIL: A Language for an Array Processing Computer", N.E. Abel et al, Proc SJCC 34 (1969).

TRANS - TRAFFic Network Simulation Language. "A Model for Traffic Simulation and a Simulation Language for the General Transportation Problem", Proc FJCC 37 (1970).

TRANS-USE - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May

1959).

TRANSCODE - Early system on Ferut computer. Listed in CACM 2(5):16 (May 1959).

TRANSIT - Subsystem of ICES. Sammet 1969, p.616.

TRANSLANG - A microassembly language for the Burroughs D Machine.

TREET - E.C. Haines, 1964. An experimental variant of LISP1.5, implemented on the STRETCH computer. Basic structure was a trinary tree. "The TREET Time-Sharing System", H.A. Bayard et al, Proc 2nd Symp Symb and Alg Manip, ACM (Mar 1971). Sammet 1969, pp.457-461.

TREETRAN - FORTRAN IV subroutine package for tree manipulation.

Trellis - (formerly named OWL). DEC. Object-oriented, with static type-checking and encapsulation. "An Introduction to Trellis/OWL", C. Schaffert et al, OOPSLA 1986. "Trellis Object-Based Environment Language Reference Manual", C. Schaffert et al, DEC-TR-372, DEC OSG, Hudson, MA (Nov 1985).  
info: Jerry Smith <smith@pipe.enet.dec.com>  
ftp://tk.telematik.informatik.uni-karlsruhe.de:/pub/tnt\* TNT system  
info: tnt@tk.telematik.informatik.uni-karlsruhe.de

TRIGMAN - Symbolic math, especially Celestial Mechanics.

Trilogy - Paul Voda <voda@mff.uniba.cs>, UBC, 1988. Logic programming language with numerical constraint-solving over the natural numbers. Syntactically a blend of Prolog, LISP and Pascal. Contains three types of clauses: predicates (backtracking but no assignable variables), procedures (if-then-else but no backtracking; assignable variables), and subroutines (like procedures, but with input and system calls; callable only from top level or from other subroutines). "The Constraint Language Trilogy: Semantics and Computations", P. Voda <voda@wimsey.cs.ubc.ca>, Complete Logic Systems, 741 Blueridge Ave, North Vancouver BC, V7R 2J5. Papers by Voda, and BYTE review.[?]

TRIX - Language for a family of line-oriented text editors used on CDC 7600 and CRAY machines under LTSS at Lawrence Livermore. Derived from TRAC.

Troff - Text formatting language/interpreter, a variant of Unix roff. (See groff, nroff, RUNOFF).

TROLL - Array language for continuous simulation, econometric modeling, statistical analysis. "TROLL Reference Manual", D0062, Info Proc Services, MIT (1973-76).

True BASIC - John Kemeny & Thomas E. Kurtz. A compiled BASIC requiring no line numbers.

TS - Typed Smalltalk. Ralph Johnson, U Illinois <johnson@speedy.cs.uiuc.edu>. "TS: An Optimizing Compiler for Smalltalk", R.E. Johnson et al, SIGPLAN Notices 23(11) (Nov 1988).

Tsim - Time Simulator. Stack-based simulation language. ZOLA Technologies, ZOLA@Applelink.Apple.com.

TSL-1 - Task Sequencing Language. Language for specifying sequences of tasking events in Ada programs. "Task Sequencing Language for Specifying Distributed Ada Systems", D.C. Luckham et al in PARLE: Parallel Architectures and Langs Europe, LNCS 259, Springer 1987, pp.444-463.

Tui - Functional. "Tui Language Manual", B. Boutel, TR CSD-8-021, Victoria U of Wellington, 1988.

Tuki - An intermediate code for functional languages. "Another Implementation Technique for Applicative Languages", H. Glaser et al, ESOP86, LNCS 213, Springer 1986.

TUPLE - Toyohashi University Parallel Lisp Environment. A parallel Lisp based on KCL. "Memory Management and Garbage Collection of an Extended Common Lisp System for Massively Parallel SIMD Architecture", Taiichi Yuasa, in Memory Management, IWMM92, Springer 1992, 490-507.

Tuple Space Smalltalk - "Using Tuple Space Communication in Distributed Object-Oriented Languages", S. Matsuoka et al, SIGPLAN Notices 23(11):276-284 (Nov 1988).

Turbo Pascal - Borland Intl's Pascal. Perhaps the first integrated development environment for MS-DOS.

Versions 1.0-3.0: standard Pascal with a few extensions

Versions 4.0 (1987) and 5.0: separate compilation

Version 5.5: object-oriented facilities

Version 6.0: Turbo Vision OOP library

Turbo Prolog - 1986. A Prolog-like language with strong typing and user-defined domains. Programs are arranged in sections: DOMAINS, CLAUSES, PREDICATES, DATABASE and GOAL. Currently known as PDC Prolog, and distributed by Prolog Development Center, Atlanta (404)873-1366 <pdcmc@mail.com>?

Turing - R.C. Holt <holt@csri.toronto.edu> & J.R. Cordy <cordy@qucis.queensu.ca>, U Toronto, 1982. Descendant of Concurrent Euclid, an airtight super-Pascal. Used mainly for teaching programming at both high school and university level. "Turing Language Report", R.C. Holt & J.R. Cordy, Report CSRI-153, CSRI, U Toronto, Dec 1983. "The Turing Programming Language", R.C. Holt & J.R. Cordy, CACM 31(12) (Dec 1988). Available from Holt Software Assocs, Toronto. Versions for Sun, MS-DOS, Mac, etc.

info: <distributed@turing.toronto.edu>

Turing Plus - Systems programming language, a concurrent descendant of Turing. "The Turing Plus Report", R.C. Holt & J.R. Cordy, CSRI, U Toronto, Feb 1987. Available from Holt Software Assocs, Toronto <distributed@hsa.on.ca>.

Turingol - D. Knuth. High-level language for programming Turing machines? Subject of the first construction of a nontrivial attribute grammar. "Semantics of Context-Free Languages", D. Knuth, Math Sys Thy 2:127-145 (1975).

TUTOR - Scripting language on PLATO systems from CDC. "The TUTOR Language", Bruce Sherwood, Control Data, 1977.

Twentel - Functional. "The TWENTEL System (Version 1).", H. Kroeze, CS Dept TR, U Twente, 1986.

TWIG - Tree-Walking Instruction Generator. A code-generator language. "Twig Language Manual", S.W.K. Tijang, CS TR 120, Bell Labs, 1986. A variant, ML-Twig in SML by Jussi Rintanen <jur@cs.hut.fi> comes with SML/NJ.

TXL - Tree Transformation Language. J.R. Cordy et al, Queens U, Canada, 1988. A hybrid functional/rule-based language with unification, implied iteration and deep pattern match. Intended for source-to-source translation and program transformation tasks. "TXL: A Rapid Prototyping System for Programming Language Dialects", J.R. Cordy et al, Comp Langs 16(1) (Jan 1991). "Specification and Automatic Prototype Implementation of Polymorphic Objects in Turing Using the TXL Dialect Processor", J.R. Cordy & E.M. Promislow, Proc IEEE Intl Conf on Comp Lang ICCL'90 (Mar 1990). [ftp://ftp.qucis.queensu.ca/pub/txl/\\*](ftp://ftp.qucis.queensu.ca/pub/txl/*) Version 7.4, source in ANSI C

TYPOL - A specialized logic programming language. "TYPOL: A Formalism to Implement Natural Semantics", T. Despeyroux, RR 94, INRIA, 1988.

UAN - User Action Notation. VPI. A notation for representation of graphical user interfaces, e.g. mice and icons, H. Hartson et al, ACM Trans on Info Sys, July 1990.

UBASIC - Y. Kida <kida@rkmath.rikkyo.ac.jp>. Extension of BASIC oriented for symbolic math and number theory. Includes bignums, fractions, complex numbers and polynomials, integer factorization. Version for MS-DOS, written in assembly language. Reviewed by W.D. Neumann, Notices of AMS (Mar 1991).  
<ftp://wsmr-simtel20.army.mil>  
[//shape.mps.ohio-state.edu/pub/msdos/ubasic](ftp://shape.mps.ohio-state.edu/pub/msdos/ubasic)

uC++ - Micro-C++. U Waterloo. A concurrent extension of C++ with coroutines and mutual exclusion. Soft Prac & Exp (Feb 1992). Version 3.7 for Unix using GCC.  
<ftp://plg.uwaterloo.ca/pub/uSystem/u++-3.7.tar.Z>

UCSD Pascal - see Pascal-P.

U-Code - Universal Pascal Code. Intermediate language, a generalization of P-code for easier optimization. Developed originally for the Los Alamos Cray-1 and the Lawrence Livermore S-1. A refined version currently used by MIPS compilers is descended from one at Stanford U. "Machine Independent Pascal Code Optimization", D.R. Perkins et al, SIGPLAN Notices 14(8): 201-201 (1979). "A Transporter's Guide to the Stanford U-Code Compiler System", P. Nye et al, TR CSL Stanford U, June 1983. (See HPCode).

UDL/I - Unified Design Language for Integrated circuits. 1991. A hardware description language for VLSI design. "UDL/I Language Reference Manual", UDL/I Committee, Japan Elec Indus Devel Assoc, Sep 1993. Version 2.0.3.  
[ftp://ftp.kuis.kyoto-u.ac.jp:/cad/udli/\\*](ftp://ftp.kuis.kyoto-u.ac.jp:/cad/udli/*)

UHELP - Linear programming. "UHELP User's Manual", D. Singh, Indus Eng Dept, U Houston (Oct 1969).

UGLIAC - Early system on Datatron 200 series. Listed in CACM 2(5):16 (May

1959).

UHELP - Mathematical language, listed [?] 1976.

UIL - User Interface Language? Distributed with Motif.

ULP - Small structured language for use in microprocessors. "User's Guide to the ULP Language for the PDP-11", CS TR 536, U Maryland (May 1977).

uML - Micro ML. An interpreter for a subset of SML that runs on MS-DOS.  
ftp://ftp.cs.umu.se/pub/umlexe01.zoo

UNBASIC - Eric S. Raymond, 1981-1982. An extension to IBM BASIC, adding Pascal-like procedures and control constructs. Implemented as a preprocessor to IBM ROM Basic. Distributed by MicroCorp Inc. Became obsolete when the Microsoft BASIC compiler for the IBM PC was released -- but for a few months it was the only IBM-PC language to support procedures.

UNCOL - UNiversal Computer Oriented Language. A universal intermediate language, discussed but never implemented. "A First Version of UNCOL", T.B. Steel, Proc JCC 19:371-378 (Winter 1961). Sammet 1969, p.708.

UNICODE - Pre-FORTRAN on the IBM 1103, similar to MATH-MATIC. Sammet 1969, p.137.

UNIFORM - An intermediate language developed for reverse engineering both COBOL and FORTRAN. "The REDO Compendium", H. van Zuylen ed, Wiley 1993.

UNIQUE - A portable job control language, used. "The UNIQUE Command Language - Portable Job Control", I.A. Newman, Proc DATAFAIR 73, 1973, pp.353-357.

UNISAP - Early system on UNIVAC I or II. Listed in CACM 2(5):16 (May 1959).

UNITY - A high-level parallel language. "Parallel Program Design", K.M. Chandry and Misra, A-W 1988.  
ftp://sanfrancisco.ira.uka.de/pub/maspar/maspar\_unity.tar.Z a translator into MPL.

Uranus - Hideyuki Nakashima <nakashim@el.go.jp>, 1993. A logic-based knowledge representation language. An extension of Prolog written in Common Lisp, with Lisp-like syntax. Extends Prolog with a multiple world mechanism, plus term descriptions to provide functional programming.  
ftp://etlport.etl.go.jp/pub/uranus/ftp

USE - Early system on IBM 1103 or 1103A. Listed in CACM 2(5):16 (May 1959).

USL -

1. Query language, close to natural English.

2. User System Language. Bellcore, "Operations Technology Generic Requirements: User System Interface", TR-825.

USSA - B. Burshteyn, Pyramid, 1992. Object-oriented state language.

doc: primost.cs.wisc.edu:/pub/ussa.ps.Z

utility-coder - Data manipulation and report generation. "User's Manual for utility-coder", Cambridge Computer Assoc (Jul 1977).

UTOPIST - E. Tyugu, Acad Sci Estonia, Tallinn, early 1980's. Specification language for attribute grammars. "Synthesis of a Semantic Processor from an Attribute Grammar", Prog and Comp Soft, 9(1):29-39 (Jan 1983).

V - Wide-spectrum language used in the knowledge-based environment CHI. "Research on Knowledge-Based Software Environments at Kestrel Inst", D.R. Smith et al, IEEE Trans Soft Eng SE-11(11):1278-1295 (1985).

VAL -

1. Value-oriented Algorithmic Language. J.B. Dennis, MIT 1979. Single assignment language, designed for MIT dataflow machine. Based on CLU, has iteration and error handling, lacking in recursion and I/O. "A Value-Oriented Algorithmic Language", W.B. Ackermann et al, MIT LCS TR-218, June 1979. "The VAL Language: Description and Analysis", J.R. McGraw, TOPLAS 4(1):44-82 (Jan 1982).

2. Variable Assembly Language. Unimation. Language for industrial robots. Version: VAL II - "VAL II: A New Robot Control System for Automatic Manufacturing", B.E. Shimano et al, Proc IEEE Intl Conf on Robotics 1984, pp.278-292.

3. VHDL Annotation Language. Stanford.  
ftp://wilbur.stanford.edu/pub/valbin-sun3-4.0-0.1.3.tar.Z  
source in Ada available from Larry M. Augustin <lma@sierra.stanford.edu>

Valid - Dataflow language. "A List-Processing-Oriented Data Flow Machine Architecture", Makoto Amamiya et al, AFIPS NCC, June 1982, pp.143-151.

VCODE -

1. Intermediate language used in the compilation of NESL.  
"Implementation of a Portable Nested Data-Parallel Language", Guy Blelloch et al, in Fourth ACM SIGPLAN Symposium on Princ and Practice of Parallel Programming, ACM, 1993.

2. Intermediate language used in the compilation of C+@.

VDM++ - Object-oriented extension of VDM-SL. "Object-Oriented Specification in VDM++", in Object Oriented Specification Case Studies, K. Lano et al eds, P-H 1993.

VDM-SL - Vienna Development Method Specification Language. (Also known as Meta-IV.) Model-oriented specification language, upon which the Vienna Development Method is based. "The Vienna Development Method: The Meta-Language", D. Bjorner et al eds, LNCS 61, Springer 1978. "The VDM-SL Reference Guide", J. Dawes, Pitman 1991. "Systematic Software Development Using VDM", C.B. Jones, P-H 1989. ISO draft Apr 1993. Version: BSI/VDM.  
ftp://gateway.dec.com/pub/vdmsl\_standard  
info: derek@mcs.le.ac.uk

Vector C - CMU? Variant of C similar to ACTUS.

VECTRAN - FORTRAN with array extensions. "The VECTRAN Language: An Experimental Language for Vector/Matrix Array Processing, Report G320-3334, IBM (Aug 1975).

Verdi - (named for the Italian composer Giuseppe Verdi (1813-1901))  
Provable systems language. Descendant of Ottawa Euclid.

VEL - See LISP70.

Verilog - Phil Moorby, Gateway Design Information, 1983. A hardware description language. "The Verilog Hardware Description Language", Donald E. Thomas & Philip Moorby, Kluwer 1991. Available from Cadence Design Systems, (408)943-1234,  
info: kmd@cadence.com

VGQF - Query language. [?]

VHDL - VHSIC Hardware Description Language. (VHSIC = Very High Speed Integrated Circuit) A large language with Ada-like syntax. The DoD standard for hardware description. "VHSIC Hardware Description Language", M.R. Shahdad et al, IEEE Computer 18(2):94-103 (Feb 1985). IEEE Std 1076 (1987). Version: VHDL92.

Vienna Definition Language - IBM Vienna Labs. A language for formal definition via operational semantics. Used to specify the semantics of PL/I. "The Vienna Definition Language", P. Wegner, ACM Comp Surveys 4(1):5-63 (Mar 1972).

Vienna Fortran - Hans Zima <zima@sophie.par.univie.ac.at>, U Vienna. Data-parallel extension of Fortran 77 for distributed memory multiprocessors. "Programming In Vienna Fortran", B. Chapman et al, Scientific Programming 1(1):31-50 (Aug 1992).

Views - A Smalltalk extension for computer algebra. "An Object Oriented Approach to Algebra System Design", K. Abdali et al, in Symp Symb Alg Manip, ACM 1986, pp.24-30.

VIF - VHDL Interface Format. Intermediate language used by the Vantage VHDL compiler. "A VHDL Compiler Based on Attribute Grammar Methodology", R. Farrow et al, SIGPLAN NOTICES 24(7):120-130 (Jul 1989).

Viron - "Five Paradigm Shifts in Programming Language Design and Their Realization in Viron, a Dataflow Programming Environment", V. Pratt, 10th POPL, ACM 1983, pp.1-9.

VITAL - Semantics language using FSL. Mondschein, 1967. Sammet 1969, p.641.

VIVID - Numerical constraint-oriented language. "VIVID: The Kernel of a Knowledge Representation Environment Based on the Constraints Paradigm of Computation", J. Maleki, Proc 20th Annual Hawaii Intl Conf on System Sciences (Jan 1987) pp.591-597.

viz - Visual language for specification and programming. "viz: A Visual Language Based on Functions", C.M. Holt, 1990 IEEE Workshop on Visual Langs, Oct 1990, pp.221-226.

Vlisp - Patrick Greussay <pg@litp.ibp.fr> ca 1973. A Lisp dialect with a fast interpreter to a portable virtual machine. Introduced the "chronology", a dynamic environment for implementing interrupts. Led to Le\_Lisp. (See ObjVlisp). "Contribution a la Definition Interpretive et a l'Implementation des Lambda-Langages", P. Greussay, These d'Etat, U Paris VI, Nov 1977.

VML - VODAK Model Language. Language for extensible object-oriented database. "Object-Oriented Modeling for Hypermedia Systems Using the Object-Oriented VODAK Model Language (VML)" Wolfgang Klas et al, in Object-Oriented Database Management Systems, NATO ASI Series, Springer 1993. info: aberer@ darmstadt.gmd.de

VMPL - A microprogramming language with PL/I-like syntax, for an abstract micromachine. The program is first translated into intermediate language, then compiled into the target microcode. "Design Objectives for High Level Microprogramming Languages", K. Malik et al, Proc 11th Ann Workshop Microprogramming (MICRO-11), 1978, pp.154-160.

VPL - Dataflow language for interactive image processing. "VPL: An Active, Declarative Visual Programming System, D. Lau-Kee et al, 1991 IEEE Workshop on Vis Langs, Oct 1991, pp.40-46.

VSP - Very Simple Prolog+. [?]

VULCAN -

1. Wayne Ratliff ca. 1980. CP/M port of JPLDIS which evolved into dBASE II.

2. The dBASE-like interpreter and compiler sold by RSPI with their Emerald Bay product. [same as 1?]

3. Early string manipulation language. "VULCAN - A String Handling Language with Dynamic Storage Control", E.P. Storm et al, Proc FJCC 37, AFIPS (Fall 1970).

4. Kahn et al, Xerox PARC. Concurrent object orientation and logic programming. An object-oriented preprocessor for FCP. "Vulcan: Logical Concurrent Objects", K. Kahn et al in Research Directions in Object-Oriented Programming, A.B. Shriver et al eds, MIT Press 1987.

WAFL - WARwick Functional Language. Warwick U, England. LISP-like.

WAM - Intermediate language for compiled Prolog, used by the Warren Abstract Machine. "An Abstract Prolog Instruction Set", D.H.D. Warren, TR 309, SRI 1983.

WATBOL - WATERloo COBOL, for MVS.

WATFIV - WATERloo Fortran IV. U Waterloo, Canada. Student-friendly variant of FORTRAN IV.

WATFOR - WATERloo FORtran. U Waterloo, Canada. Student-friendly variant of FORTRAN. "WATFOR - The University of Waterloo FORTRAN IV Compiler", P.W. Shantz et al, CACM 10(1):41-44 (Jan 1967).

WAVE - Robotics language. "WAVE: A Model-Based Language for Manipulator Control", R.P. Paul, *Ind Robot* 4(1):10-17 (1979).

WEB - Knuth's self-documenting brand of programming, with algorithms and documentation intermixed in one file. They can be separated using Weave and Tangle. Versions exist for Pascal, C etc. Spiderweb can be used to create versions for other languages (<ftp://princeton.edu>). "Literate Programming", D.E. Knuth, *Computer J* 27(2):97-111 (May 1984).

WFL - Work Flow Language. Burroughs, ca 1973. A job control language for the B6700/B7700 under MCP. WFL was a compiled block-structured language similar to ALGOL-60, with subroutines and nested begin-end's. "Work Flow Management User's Guide", Burroughs Manual 5000714 (1973). "Burroughs B6700/B7700 Work Flow Language", R.M. Cowan in *Command Languages*, C. Unger ed, N-H 1975.

Wisp - "An Experiment with a Self-Compiling Compiler for a Simple List-Processing Language", M.V. Wilkes, *Ann Rev Automatic Programming* 4:1-48 (1964).

Wizard - Lehigh U, ca 1975. [?]

Woodenman - HOLWG, DoD, 1975. Second of the series of DoD requirements that led to Ada. "Woodenman Set of Criteria and Needed Characteristics for a Common DoD High Order Programming Language", David A. Fisher, *Inst for Def Anal Working Paper*, Aug 1975. (See Strawman, Tinman, Ironman, Steelman).

WOOL - Window Object Oriented Language. Colas Nahaboo <[colas@sophia.inria.fr](mailto:colas@sophia.inria.fr)> Small Common Lisp-like extension language. Claims to be the fastest interpreted language in C with run-time types. Version 1 used by the GWM window manager. Version 2 has an object system. <ftp://export.lcs.mit.edu/contrib/gwm/>\*

WPL+ - Word-oriented language internal to PRODOS Applewriter 2.1. Available on GENie.

WPOP - WonderPop. Robert Rae <[rhr@aiai.ed.ac.uk](mailto:rhr@aiai.ed.ac.uk)>, Edinburgh 1976. An implementation of POP for the PDP-10 that used cages for different data types. Introduced processes, properties, and some typed identifiers.

WRITEACOURSE - CAI language, for IBM 360. "WRITEACOURSE: An Educational Programming Language", E. Hunt et al, *Proc FJCC* 33(2) (1968).

WSFN - Which Stands For Nothing. Atari 1983. Beginner's language with emphasis on graphics, for Atari home computers. Version: Advanced WSFN.

WSL - Waterloo Systems Language. A C-like systems programming language. "Waterloo Systems Language: Tutorial and Language Reference", F.D. Boswell, WATFAC Publications Ltd, Waterloo, Canada. ISBN 0-919884-00-8.

X-1 - Early system on UNIVAC I or II. Listed in *CACM* 2(5):16 (May 1959).

Xbase - Generic term for the dBASE family of languages. Coined in response to threatened litigation over use of the trademark "dBASE."

XBASIC - eXtended BASIC. 1972. An extension of BASIC, including matrix operations and Algol-like procedures. For the Univac 1108.

XC - Declarative extension of C++. "XC - A Language for Embedded Rule Based Systems", E. Nuutila et al, SIGPLAN Notices 22(9):23-32 (Sep 1987).

XDL - An object-oriented extension to CCITT's SDL[2]. "XDL: An Object-Oriented Extension to SDL", S.J. Ochuodho et al in [?]

Xfun - S. Dalmás <dalmás@sophia.inria.fr>, INRIA, 1991. A cross between SML and Russell, intended for computer algebra. "A Polymorphic Functional language Applied to Symbolic Computation", S. Dalmás, Proc Intl Symp Symb Alg Comp, Berkeley 1992.

Xi - VLSI design language. "The Circuit Design Language Xi", S.I. Feldman, unpublished memo, Bell Labs, 1982. Mentioned in Computational Aspects of VLSI, J.D. Ullman, CS Press 1984.

XICS - Xerox. Page description language.

XL - A tuple language used as the intermediate form in the code generator generator XGEN. "A Knowledge Based Code Generator Generator", Proc Symp AI and Prog Lang, Aug 1977, pp.126-129.

XLISP - eXperimental LISP. David Betz <dbetz@apple.com>. LISP variant with object-oriented extensions, portable source in C. BYTE article. Version 2.0. Version 2.1 by Tom Almy <toma@sail.labs.tek.com> is closer to Common Lisp.

<ftp://cs.orst.edu/pub/xlisp/>\*

[//glia.biostr.washington.edu/pub/xlisp/xlisp21e.tar.Z](http://glia.biostr.washington.edu/pub/xlisp/xlisp21e.tar.Z)

XLISP-PLUS - An extension of XLISP used in the WINTERP OSF/Motif Widget interpreter.

<http://www.eit.com/software/winterp/winterp/html>

[ftp://ftp.x.org/contrib/devel\\_tools/winterp\\*](ftp://ftp.x.org/contrib/devel_tools/winterp*)

XNF - Hardware description language?

XPC - eXplicitly Parallel C. Dialect of Parallel C which is mode independent, i.e. efficiently compilable to both SIMD and MIMD architectures. "Toward Semantic Self-Consistency in Explicitly Parallel Languages," M.J. Phillip & H.G. Dietz, Proc 4th Intl Conf on Supercomputing, Santa Clara, CA, May 1989, v.1, pp.398-407. Research implementations only.

XPL - Stanford, 1967-69. Small dialect of PL/I used for compiler writing. One-dimensional arrays. I/O achieved with character pseudo-variable INPUT and OUTPUT, e.g. OUTPUT = 'This is a line'; Inline machine code.

"Programmers are given all the rope they ask for. Novices tend to hang themselves fairly frequently." Implemented on IBM 360, Univac 1100, ICL System 4, CDC6000 and Cyber series, XDS Sigma-5 and Sigma-7, and DEC PDP-10. "A Compiler Generator," W.M. McKeeman et al, P-H 1970. Also JCC, AFIPS 1968.

XPOP - Extensible macro assembly language with user-redefinable grammar, for use with FAP. "XPOP: A Meta-language Without Metaphysics", M.I. Halpern, Proc FJCC 25:57-68, AFIPS (Fall 1964).

XScheme - David Betz. Scheme with object-oriented extensions. Source in C. Versions for PC, Macintosh, Atari, Amiga.  
ftp://labrea.stanford.edu/pub/xscheme.tar.gz  
comp.sources.amiga/volume90, version 0.28 for Amiga

XTRAN - FORTRAN-like, interactive. [?]

Y - General purpose language syntactically like RATFOR, semantically like C. Lacks structures and pointers. Used as a source language for the Davidson/Fraser peephole optimizer. "The Y Programming Language", D.R. Hanson, SIGPLAN Notices 16(2):59-68 (Feb 1981).  
ftp://ftp.cs.princeton.edu/pub/y+po.tar.Z

Yaa - Yet Another Assembler - Macro assembler for GCOS 8 and Mark III on Bull DPS-8 machines. Available from Bull as part of U Waterloo Tools package (maintained by <pjf@thinkage.on.ca>).

yacc - Yet Another Compiler Compiler. Language used by the Yacc LALR parser generator. "YACC - Yet Another Compiler Compiler", S.C. Johnson, CS TR 32, Bell Labs (Jul 1975).

Implementations:

ayacc - UC Irvine. written in Ada, produces Ada output  
ftp://liege.ics.uci.edu/pub/irus/aflex-ayacc\_1.2a.tar.Z

Bison - from GNU

ftp://prep.ai.mit.edu/pub/gnu/bison-1.21.tar.Z

Bison++ - produces C++ output.

ftp://psuvax1.cs.psu.edu/pub/src/gnu/bison++-1.04.tar.Z

perl-byacc - produces perl output

ftp://ftp.sterling.com/local/perl-byacc1.8.2.tar.Z

SASL-yacc - "Yacc in SASL - An Exercise in Functional Programming", Simon Peyton-Jones, Software Prac & Exp 15:807-820 (1985). Mentions also a BCPL implementation.

yacc++ - 1990. An object-oriented rewrite of yacc, supports regular expressions, produces an LR(1) parser. "YACC Meets C++", S.C. Johnson, USENIX Spring '88 Conf. Chris Clark, Compiler Resources Inc, Barbara Zino <bz%compres.UUCP@primerd.cv.com> (508) 435-5016.

MLYACC - Implementation and output in SML/NJ

ftp://research.att.com/dist/ml/75.tools.tar.Z

YALLL - Yet Another Low Level Language. Patterson et al, UC Berkeley, 1979. A microprogramming language resembling conventional assembly language. "Towards an Efficient Machine-Independent Language or Microprogramming", D.A. Patterson et al, Proc 12th Ann Workshop Microprogramming (MICRO-12), 1979, pp.22-35

YAPS - Yet Another Production System? College Park Software. Commercial production rule language, simpler than OPS5. Allows knowledge bases to be attached to instances of CLOS objects.

info: Liz Allen <liz@grian.cps.altadena.cs.us>

YASOS - Yet Another Scheme Object System.

info: Ken Dickey <kend@newton.apple.com>

Yay - Yet Another Yacc - An extension of Yacc with LALR(2) parsing. Available from Bull as part of U Waterloo Tools package (maintained by <pjf@thinkage.on.ca>).

Yellow - SRI. A language proposed to meet the Ironman requirements which led to Ada. "On the YELLOW Language Submitted to the DoD", E.W. Dijkstra, SIGPLAN Notices 13(10):22-26 (Oct 1978).

Yerk - (named for Yerkes Observatory) A public domain reincarnation of Neon.

ftp://oddjob.uchicago.edu/pub/Yerk  
info: Bob Lowenstein <rfl@oddjob.uchicago.edu>

YLISP - Hewlett-Packard. A variant of Xlisp for the HP-95LX palmtop.  
ftp://hpcsos.col.hp.com/mirrors/.scsi5/hp95lx/languages/ylisp\*

Z -

1. ("zed") Programming Research Group, Oxford U, ca 1980. A specification language based on axiomatic set theory and first order predicate logic. Uses many non-ASCII symbols. Used in the IBM CICS project? "Understanding Z", J.M. Spivey, Cambridge U Press 1988. An embedding of Z in HOL available from ICL <ProofPower-server@win.icl.co.uk>

2. A stack-based, complex arithmetic simulation language. ZOLA Technologies, ZOLA@Applelink.Apple.com.

Z++ - Object-oriented extension of Z. "Z++, an Object-Oriented Extension to Z", Lano, Z User Workshop, Oxford 1990, Springer Workshops in Computing, 1991, pp.151-172.

ZAP - Language for expressing transformational developments. "A System for Assisting Program Transformation", M.S. Feather, ACM TOPLAS 4(1):1-20 (Jan 1982).

Zed - 1978. Software Portability Group, U Waterloo. Eh, with types added. Similar to C. Implementation language for the Thoth realtime operating system. Added a few simple types for greater efficiency on byte-addressed machines. String constants in case statements. Enforces the naming convention: MANIFESTS, Externals and locals. "Porting the Zed Compiler", G.B. Bonkowski et al, SIGPLAN Notices 14(8):92-97 (Aug 1979).

ZENO - U Rochester 1978. Euclid with asynchronous message-passing. "Preliminary ZENO Language Description", J.E. Ball et al, SIGPLAN Notices 14(9):17-34 (Sep 1979).

ZERO - Object oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

ZEST - Object oriented extension of Z. "Object Orientation in Z", S. Stepney et al eds, Springer 1992.

ZetaLisp - Maclisp dialect used on the LISP Machine. The many extensions to Maclisp include vectors, closures, flavors, stack groups, locatives and invisible pointers. "LISP Machine Manual", D. Weinreb and D. Moon, MIT AI Lab, 1981.

ZIL - Zork Implementation Language. Language used by Infocom's Interactive Fiction adventure games. Interpreted by the zmachine, for Unix and Amiga.  
ftp://plains.nodak.edu/Minix/st.contrib.Infocom.tar.Z

Zipcode - [?] Parallel language at Lawrence Livermore?

zsh - Sh with list processing and database enhancements.  
ftp://cs.ucsd.edu/pub/zsh/zsh2.1.0.tar.Z

ZOPL - Geac. [?] A low-level Pascal?

ZUG - Geac. [?] A low-level Awk?

Zuse - (named for Konrad Zuse, the designer of the first modern programming language Plankalkul.) Christian Collberg <collberg@dna.lth.se>, PhD thesis 1991. A descendant of Ada, Modula-2, Mesa and Oberon-1 supporting several levels of information hiding. The Zuse type system includes fully hidden types (similar to Modula-2 opaque types but without any implementation restriction), semi-open pointer types (same as Modula-2 opaque types), extensible record types (similar to Oberon-1 public projection types but without the compiler hint), enumeration types, extensible enumeration types, and extensible subrange types. A type can also be protected by specifying the operations that particular modules may perform (similar to C++ friend classes and Ada private types). Zuse also includes hidden and extensible constants and hidden inline procedures. In order to support the higher levels of information hiding the implementation employs partial intermediate code linking. Version for Sun-3.

Zz - An extensible language.

<http://slacvm.slac.stanford.edu:5080/FIND/FREEHEP/NAME/ZZ/FULL>

## APPENDIX A

### Selected Works on Computer Language History and Design

=====

AFIPS (American Federation of Information Processing Societies), SJCC (Spring Joint Computer Conference) and FJCC (Fall Joint Computer Conference), replaced after 1972 by the NCC (National Computer Conference).  
TK7885 .A1 J6

Andrews, G.R. and F.B. Schneider, "Concepts and Notations for Concurrent Programming", ACM Comp Surveys 15(1):1-43 (Mar 1983).

Bal, H.E., "Programming Languages for Distributed Systems", ACM Comp Surveys 21(3):261-322 (Sep 1989).

Blume, Christian, "Programming Languages for Industrial Robots", Springer 1986.

Bobrow, D.G. & Raphael, "New Programming Languages for Artificial Intelligence Research", ACM Comp Surveys 6:155 (Nov 1974).

Computer Languages QA 76.7 .C647 ISSN 0096-0551

Conference Record of the 19XX ACM Symposium on Lisp and Functional Programming, QA 76.73 .L23 A26

Horowitz, Ellis, "Programming Languages, A Grand Tour", 3rd ed, Springer  
1987. QA76.7 .P78

Hudak, Paul, "Conception, Evolution and Application of Functional  
Programming Languages", ACM Comp Surveys 21(3):359-411 (Sep 1989).

IEEE Transactions on Software Engineering, QA76 .I33x

LFP - Proceedings of the 19XX ACM Conference on Lisp and Functional  
Programming. QA 76.73 .L23 A24

Metropolis, N. et al, "A History of Computing in the Twentieth Century.  
Proceedings of the International Research Conference on the History of  
Computing", A-P 1986. QA 75.5 .H58 1986 ISBN 0-12-491650-3.

Milutinovic, V., "Microprogramming and Firmware Engineering", IEEE Computer  
Soc Press 1989. QA 76.6 .M524 1989 ISBN 0-8186-0839-0.

POPL - Conference Record of the [Nth] Annual ACM Symposium on Principles of  
Programming Languages. QA 76.7 .A25a

Sammet, Jean E., <sammet@vtvml.cc.vt.edu> "Programming Languages: History  
and Fundamentals", P-H 1969. QA76.5 .S213. The definitive work on early  
computer language development.

Sammet, Jean E., "Programming Languages: History and Future", CACM  
15(7):601-610 (Jul 1972).

Sammet, Jean E., "Roster of Programming Languages"  
Computers & Automation 16(6):80-82 (June 1967)  
Computers & Automation 17(6):120-123 (June 1968)  
Computers & Automation 18(7):153-158 (June 1969)  
Computers & Automation 19(6B):6-11 (Nov 30, 1970)  
Computers & Automation 20(6B):6-13 (June 30, 1971)  
Computers & Automation 21(6B) (Aug 30, 1972)  
Computing Reviews 15(4): 147-160 (April 1974)  
CACM 19(12):655-669 (Dec 1976)  
SIGPLAN Notices 13(11):56 (Nov 1978).

Shapiro, E., "The Family of Concurrent Logic Programming Languages", ACM  
Comp Surveys, 21(3):413-510 (Sep 1989).

ACM SIGPLAN Notices, QA76.7 .A13

Software: Practice and Experience, QA 76.5 .S653

ACM TOPLAS (Transactions on Programming Languages and Systems) QA76.7 .A16

W. van der Poel, Machine Oriented Higher Level Languages, N-H 1974.

Wexelblat, R.L. ed, "Proceedings: ACM SIGPLAN History of Programming  
Languages Conference" (Los Angeles, July 1978). 758 pp. Academic Press  
1981. An entertaining account of the history of roughly 15 major  
programming languages. Jean Sammet co-organized the conference. ACM No  
548780. ACM Order Dept: (800)342-6626. QA76.7 .H56

## APPENDIX B

### A Chronology of Influential Computer Languages

=====  
Konrad Zuse in Nazi Germany may have developed the first real computer programming language, "Plankalkul" ca. 1945. This is mentioned in the 1978 ACM History of Programming Languages FORTRAN session.

According to Sammet, over 200 programming languages were developed between 1952 and 1972, but she considered only about 13 of them to be significant.

1957	FORTRAN
1958	ALGOL
1960	LISP
1960	COBOL
1962	APL
1962	SIMULA
1964	BASIC
1964	PL/I
1966	ISWIM
1970	Prolog
1972	C
1975	Pascal
1975	Scheme
1977	OPS5
1978	CSP
1978	FP
1980	dBASE II
1983	Smalltalk-80
1983	Ada
1983	Parlog
1984	Standard ML
1986	C++
1986	CLP(R)
1986	Eiffel
1988	CLOS
1988	Mathematica
1988	Oberon
1990	Haskell