The LISP 2 project

Paul McJones

DRAFT of April 28, 2016

1 Introduction

LISP 2 was an ambitious project to define and implement a potential successor to the original LISP 1.5, which John McCarthy and his colleagues developed between 1958 and 1962 at the Massachussetts Institute of Technology. LISP 2 went beyond LISP 1.5 in a number of ways, including ALGOL-like syntax, efficient support for arrays, and a retargettable, compiler-based implementation.

Early design work on the LISP 2 project took place at MIT and Stanford during 1963 and 1964 and included one or more conferences to bring together the interested people. Starting around late 1964, arrangements were made with System Development Corporation (SDC) and Information International Incorporated (III) to complete the detailed design and carry out the implementation. A fairly complete system was running on SDC's AN/FSQ-32 computer by the fall of 1966.

The Q-32 was used for LISP 2 development because it was available and had a timesharing system that facilitated development by a multiperson project. A substantial part of LISP 2 was completed on the Q-32, but the system exceeded the capacity of the available memory (approximately 48,000 48-bit words). The structure of LISP 2 reflected the intention to port it to other systems — at various times, ports to the IBM System/360, the GE 635, and the DEC PDP-6 were discussed. Design work was done for some of these ports, but none were carried out and in 1967 work on the system was terminated. The potential users of LISP 2, who were in artificial intelligence projects at Stanford, MIT, and other universities and corporate research laboratories, mostly migrated to LISP 1.5 implementations on the DEC PDP-6 and its follow-on the PDP-10.

Through the foresight of project members Jeff Barnett and Clark Weissman, as well as historian Herbert Stoyan, a number of project documents and source code listings were preserved and made available to the Computer History Museum; most of those have been

digitized and are available on the internet[1]. One goal of this paper is to serve as an introduction to people wishing to study these artifacts.

2 Requirements and early design work (MIT and Stanford, 1963-1964)

At the first History of Programming Languages conference, McCarthy [1978] gave this rationale for the LISP 2 project:

> "As a programming language LISP had many limitations. Some of the most evident in the early 1960s were ultra-slow numerical computation, inability to represent objects by blocks of registers and garbage collect the blocks, and lack of a good system for input-output of symbolic expressions in conventional notations. All these problems and others were to be fixed in LISP 2. In the meantime, we had to settle for LISP 1.5[2] developed at M.I.T. which corrected only the most glaring deficiencies."

The earliest known LISP 2 document is a one-page agenda[3] for a "LISP 2 Specifications Conference" held by the Artificial Intelligence Group at Stanford. Section 2 of this agenda was "Proposals for LISP 2.0", and included:

a. Linear Free Storage
b. Numbers and other full words
c. Auxiliary Storage
d. Input language - infix notation
e. Arrays
f. Freer output format
g. Sequence of implementation
h. Comments
i. Documentation and maintenance
j. Hash Coding
k. Subroutine linkage
l. Storage conventions
m. Effect of various I-O apparatus
n. Interaction with programs in other languages
o. Expressions having property lists
p. Data structures
q. Fitting into monitor

[1] See McJones [2010].
[2] See McCarthy et al. [1962].
[3] See Anonymous [1963?]. The document is not signed or dated, but McCarthy [1963] mentions "At the LISP 2 meeting in July".

Section 3 was "Objections to LISP 1.5" and included:

1. Cluttered
2. Slow numberical [sic] calculations
3. Data structure
4. Slow interpreter

At the First International LISP Conference, held in Mexico City over New Years 1964, John McCarthy was scheduled to give a talk, "The LISP 2 Compiler", but no written record of the talk is known.[4] In any case, memos on various LISP 2 design topics began appearing in the report series from the Stanford and MIT Artificial Intelligence projects: McCarthy [1963] and Levin [1964b] on storage conventions, Levin [1964a] on syntax, and Bobrow [1964b] on string manipulation.

3 Development (SDC and III, 1964-1967)

3.1 Building a team

As the scope of the system evolved, it was obvious that more resources were required than were available at the Stanford and MIT AI projects. Luckily, the System Development Corporation (SDC) had people and a timesharing system running on the AN/FSQ-32 (Q-32) computer.[5] SDC, located in Santa Monica, California, was a non-profit organization that had been spun out of the RAND Corporation to write the software for the SAGE air defense system. The Q-32 was originally planned as a second-generation SAGE computer, but when those plans were canceled the Q-32 prototype at SDC became available for ARPA-funded projects in timesharing and networking.

While the SDC people had extensive expertise in systems programming, they lacked specific LISP experience, so a contract was negotiated with Information International, Incorporated (in Cambridge, Massachusetts and New York City) to obtain the services of some additional people with LISP experience.[6] The combined team was described in this Foreword from

[4] See McJones [2012a].

[5] See Schwartz et al. [1964] and Schwartz and Weissman [1967].

[6] Robert W. Taylor, former ARPA IPTO director, recalled: "I believe ARPA supported a single contract with SDC to provide an 'umbrella' for work carried out on the Q-32, rather than specific contracts for specific activities. ... Non–SDC employees, like triple I, were probably funded by SDC as subcontractors." (Personal communication to Paul McJones, July 13, 2012.) See Fredkin [1964] for the contract proposal, which proposed implementing the system not only on the Q-32 but also the GE 635, presumably because Project MAC was planning to use that machine for the Multics timesharing system. After reading the proposal in 2012, Jeff Barnett noted: "Virtually all the work was done in Los Angeles, not Boston as implied and MIT wasn't really involved except we'd visit Minsky a few times a year and tell him what we were up to." (Personal communication to Paul McJones, June 26, 2012.)

an SDC LISP 2 technical memo:[7]

> "LISP II is a joint development of SDC and III. The idea for LISP II as a language combining the properties of an algebraic language like ALGOL and the list-processing language LISP was conceived by M. Levin of MIT. Development of the concepts of LISP II was carried forth in a series of conferences held at MIT and Stanford University. Contributions in concepts and detail were made by Prof. John McCarthy of Stanford University, Prof. Marvin Minsky of MIT, and the LISP II project team consisting of M. Levin, L. Hawkinson, R. Saunders and P. Abrahams of III, and S. Kameny, C. Weissman, E. Book, Donna Firth, J. Barnett and V. Schorre of SDC."

III employee Lowell Hawkinson assumed the role of manager for the LISP 2 project. As he recalled in 2010:[8]

> "In April of 1965, I joined Information International, Inc., a small software and hardware company, in good part because I could in principle serve the military interests of the U.S. in a way different from being a soldier in Vietnam. Soon thereafter, Information International became a subcontractor to the System Development Corporation for the implementation of LISP 2. ... Though I was the most junior person involved [22 years old], I effectively led the LISP 2 implementation effort because of my LISP and compiler building experience."

About six months after he arrived, Jeff Barnett came onboard, and became the main compiler implementer. Michael Levin was at SDC in Santa Monica for the summer of 1965 and continued to play an important role specifying parts of the language.[9] Paul Abrahams worked out of his apartment on East 23rd Street in New York City, and also spent a lot of time in Santa Monica.[10]

In 2012, Levin recalled:[11]:

> "I left the project some time around September or October, 1965. My recollection was that just before I left Santa Monica, there was a significant finalizing conference in Santa Monica, at which the many ideas added by the SDC people such as Book, Kameny and Weissman were added. I believe that I moderated this meeting, then flew to my parent's home in NJ and documented the syntax of the resulting specification from memory. This would have been with pencil

[7] See Kameny and Hawkinson [1965]. Levin and Abrahams had worked on LISP 1.5 at MIT, Saunders had ported LISP 1.5 to the Q-32, and Hawkinson had implemented his own version of LISP 1.5 at Yale University and the University of Mexico (see McJones [2012b]).

[8] Personal communication to Paul McJones, April 7, 2010.

[9] Personal communication from Lowell Hawkinson to Paul McJones, April 8, 2010.

[10] See Norberg [2006].

[11] Personal communication to Paul McJones, July 19, 2012; date confirmed as 1965 on July 25, 2012.

> and paper since portable or home computers did not exist, and a typewriter would have been too messy. I probably mailed this off to Lowell."

3.2 Early projects

The first LISP project at SDC was to port LISP 1.5 to the Q-32. In 2005, Robert A. Saunders recalled:[12]

> "The Q-32 implementation was the first LISP that did not run on an interpreter: all code was compiled before execution. Someone subsequently wrote an interpreter for it, but I have no idea why other than for the fun and instruction of doing it. [As stated in Berkeley and Bobrow[13]], the LISP was developed on Stanford's 709/90 system, compiled there, and taken to Santa Monica as machine instructions on magnetic tape. Some fundamental code (e.g. CONS, and later the garbage collector) was done in machine language on the Q-32 itself."

As another early LISP project, Erwin Book produced a LISP 1.5 version of the Meta Compiler, a syntax-directed translator that would later be used for the LISP 2 source language-to-intermediate language translator.[14]

3.3 Development proceeds

With a team in place, a LISP 1.5 implementation, and the Meta Compiler, development on LISP 2 picked up its pace. Early design memos were fairly informal, for example sometimes lacking a title.[15] But by the end of 1965 the memos followed a standard style and were subjected to a release control process, presumably derived from SDC's earlier experience developing the SAGE software.[16]

In 2010, Jeff Barnett related some memories of this period.[17] Various people visited SDC from other organizations to help out with the project. One of them was Warren Teitelman, who worked on a pattern-directed facility for LISP 2 that was similar to COMIT and METEOR.[18] Another was Fred Blair, from IBM Yorktown. Blair spent a summer [1965?] at

[12] Robert A. Saunders, personal communication to Paul McJones, May 6, 2005.

[13] See Saunders [1964].

[14] See Book [1965]. This Meta Compiler was based on a system by Val Schorre and Lee Schmidt; see Schorre [1964].

[15] See Levin [1965a] through Levin [1965b] available via McJones [2010].

[16] See TM-2710 technical memorandum series, beginning with Kameny and Hawkinson [1965], available via McJones [2010].

[17] Personal communication from Jeff Barnett to Paul McJones, June 22, 2010.

[18] See Yngve [1962] and Bobrow [1964a].

SDC, and worked on LAP, the assembler. He also did some work on symbolic algebra which wasn't part of LISP 2 but was a topic of general interest in those days. The SDC people also visited other organizations. Barnett recalled that he and Hawkinson visited Fran Allen in San Jose (where she was working on IBM's ACS project) to discuss advanced compiler optimizations. The team also made periodic trips to visit the project's "godfathers": John McCarthy at Stanford and Marvin Minsky at MIT. After the MIT visits, which included visiting the "hackers" at the AI Lab, they would borrow a Landrover and drive to the Courant Institute in Greenwich Village (New York City) and visit John Cocke and former project member Paul Abrahams.[19] Despite their hard work, the project members had time for a little fun. Project member Stan Kemeny and visitor Fred Blair were rock climbers, and Fred would climb up the brick facades of buildings. A humorous one-page flyer for the project was typed up by someone at III. It began, "Are storage allocation headaches getting you down? Are your symbols coming out of your machine with bits missing and frayed at the decrement?" and was signed, "Bottled in bond by Information International, Inc."[20]

While LISP 2 development proceeded, other people were developing unrelated applications using the Q-32 LISP 1.5 implementation. Anthony C. Hearn, working remotely from Stanford University, ported his REDUCE computer algebra system from Stanford's IBM 7090 to the Q-32.[21] Adolfo Guzmán and Harold V. McIntosh at CENAC (Centro Nacional de Calculo, Instituto Politecnico Nacional, Mexico) used the Q-32 to develop their CONVERT pattern transformation language.[22] Finally, project member Clark Weissman was writing the *LISP 1.5 Primer*,[23] which became the standard LISP textbook for many years.

3.4 Structure of the compiler

Although people had been writing LISP compilers since the original LISP project, it was still a new field. Lowell Hawkinson had these recollections:[24]

> "As original architect of the LISP 2 compiler (when I left the project in 1967, that role passed to Jeff Barnett), I'd describe its design as having had several roots. First, of course, there was the original LISP 1.5 compiler developed at M.I.T. by Michael Levin [and Tim Hart] and believed by him to have been the first compiler ever to have compiled itself. For me, the influence of this compiler was quite general and conceptual, since I was not acquainted with its

[19] Abrahams' oral history Norberg [2006] says he moved from III to Courant in 1967.

[20] See Anonymous [1967?].

[21] See Hearn [1968].

[22] See Guzmán and McIntosh [1966].

[23] See Weissmann [1967]. Weissman also designed the LISP 2 I/O subsystems, was responsible for documentation, and played a role in SDC management and ARPA interface.

[24] Personal communication to Paul McJones, July 18, 2012.

details. Secondly, there were the LISP compilers I had built, beginning with YULISP on the IBM 709 at Yale in the winter of 1962-63 and continuing at the University of Florida and the Instituto Politecnico in Mexico City (there with Robert Yates) from late 1963 to early 1965. Finally, there was the LISP 1.5 compiler of Robert Saunders for the AN/FSQ-32 at SDC."

Jeff Barnett supplied some details:[25]

> "The organization of the LISP 2 compiler was interesting. When an expression was compiled a set of global variables were bound to indicate what an easy-to-use result would look like, e.g., an unboxed integer in a register. The compiler described the result actually produced in another set of global variables, e.g., a real number at memory location POP. The compiler than needed to convert the given result to something that could be consumed.
>
> The Q-32 instruction set was somewhere between arcane and obscure and Bob Saunders was the only one who knew all the secret incantations that could do conversions. So in the middle of the night we would call him at home (394-6259 aka EXHOCKEY) to get help. Bob, unlike the rest of us, was married and had a life away from LISP. In desperation he wrote a rather massive function called EXHOCKEY (naturally) that one called whenever it was necessary to convert a produced state to a desired state. He then threatened us with death if we called too late at night."

3.5 Memory problems

A July 12, 1966 progress report[26] from Lowell Hawkinson included these highlights:

> "An operating version of LISP 2 was demonstrated to ARPA in mid-May.
>
> A paper on LISP 2 has been accepted for presentation to the Fall Joint Computer Conference."

The report ended with this bad news:

> "TECHNICAL PROBLEMS
>
> A continuing problem has been the limited core storage available on the Q-32 and projected for SDC's IBM 360/65. The lack of space on the Q-32 has caused a slowdown in the rate of updating the system to include the syntax translator and language modifications. The 50K maximum task size specified

[25] Personal communication to Paul McJones, July 18, 2012.

[26] See Hawkinson [1966].

> for the 360/65 time sharing system required much extra work in specifying the internal storage conventions for that machine."

Jeff Barnett expanded on the situation:[27]

> "The Q-32 version was so cramped for space (approximately 48,000 48 bit words) that an application-level swapper was written: If a function was needed but not in memory 1) if memory space was available, the function was paged in by LISP from an ordinary data file, 2) if space was not available, the memory manager tried to create space by shuffling the in-memory area holding binary code to make space, if this did not work some in-memory code was excised or the GC invoked then 3) the dynamic loader read the binary code into memory."

While SDC was planning to acquire an IBM System/360 Model 50 for the Adept time-sharing project, it would not have virtual memory, and the maximum user address space would have been smaller than the Q-32. A complete set of design documentation for the System/360 version of LISP 2 was published on April 26, 1967[28], but the port was never carried out.

The 1966 FJCC paper noted the system was complete except for the pattern-directed features:[29]

> "A fourth major change, the introduction of pattern-driven data manipulation facilities, along the lines of COMIT [5] and METEOR [6], is still in the process of implementation. Because of the open-ended nature of LISP 2, these facilities can be added without disrupting the existing system structure. We mention this facility here, despite the fact that it does not yet exist, because it is an integral part of the over-all design of the language. Since the specifications are not final as of this writing, however, we shall not discuss them further."

4 Aftermath

By 1967 it would have been very difficult to displace LISP 1.5. In his review of the AFIPS paper, Weizenbaum [1967] does a good job explaining the strengths of LISP and the difficulties in displacing it. LISP 2 did not displace LISP 1.5; until Common Lisp arrived several decades later, many incompatible dialects of LISP flourished.[30] McCarthy [1978] looked back:

[27] Personal communication to Paul McJones, July 17, 2012.
[28] See the document series TM-3417 in McJones [2010].
[29] See Abrahams et al. [1966].
[30] See Steele and Gabriel [1993], who inexplicably omit LISP 2.

> "Unfortunately, the Q32 at SDC was never equipped with more than 48K words of this memory. When it became clear that the Q32 had too little memory, it was decided to develop the language for the IBM 360/67 and the Digital Equipment PDP-6 — SDC was acquiring the former, while III and M.I.T. and Stanford preferred the latter. The project proved more expensive than expected, the collaboration proved more difficult than expected, and so LISP 2 was dropped. From a 1970s point of view, this was regrettable, because much more money has since been spent to develop LISPs with fewer features. However, it was not then known that the dominant machine for AI research would be the PDP-10, a successor of the PDP-6. A part of the AI community, e.g. BBN and SRI made what proved to be an architectural digression in doing AI work on the SDS 940 computer."

Lowell Hawkinson recalled:[31]

> "Even if the project had had more mature leadership, it probably could not have overcome the unfortunate choice of an ALGOL-like syntax and type system, overemphasis on compiled code efficiency, and the obscure choice of first machine to build for (the IBM AN/FSQ-32, a machine that was time-shared using teletype terminals). LISP's fundamental strengths were lost in LISP 2's design."

LISP 2 took a pioneering step beyond the experience gained from the first generation of programming languages. Its combination of ALGOL-like syntax and static type checking with LISP-style garbage collection and runtime types foreshadowed later languages such as the research language ECL[32], Common Lisp[33], and Java.[34]

5 Artifacts

McJones [2010] contains digitized copies of the LISP 2 documents donated to the Computer History Museum by Herbert Stoyan[35] and Clark Weissman[36] and from Jeff Barnett's personal collection. In particular, Barnett's collection includes source code listings, for which he provided annotations.

[31] Personal communication to Paul McJones, April 7, 2010.
[32] See Wegbreit [1971].
[33] See Pitman [1994].
[34] See Gosling et al. [1996].
[35] See McJones [2011].
[36] Lot X6720.2013, cataloging in progress.

6 Acknowledgments

I thank Jeff Barnett, Clark Weissman, and Herbert Stoyan for making documents available. I thank Jeff Barnett for his extended commentary on LISP 2 before and during this project, and also to Fred Blair, Lowell Hawkinson, Anthony Hearn, Michael Levins, Robert Saunders, and Robert W. Taylor for answering my questions.

Bibliography

Paul W. Abrahams, Jeffrey A. Barnett, Erwin Book, Donna Firth, Stanley L. Kameny, Clark Weissman, Lowell Hawkinson, Michael I. Levin, and Robert A. Saunders. The LISP 2 programming language and system. In *Proceedings of the Fall Joint Computer Conference*, volume 29, pages 661–676. AFIPS, November 1966. URL http://doi.ieeecomputersociety.org/10.1109/AFIPS.1966.106.

Anonymous. LISP 2 specifications conference. Agenda, July? 1963? URL http://www.softwarepreservation.org/projects/LISP/stanford/Stanford-LISP_2_Specifications_Conference.pdf.

Anonymous. Are storage allocation headaches getting you down? Humorous one-page flyer for LISP 2, Information International, Inc., 1967? URL http://www.softwarepreservation.org/projects/LISP/lisp2/III-Bottled_in_bond.pdf.

D. G. Bobrow. METEOR, a LISP interpreter for string transformation. In *The Programming Language LISP: Its Operation and Applications*. Information International, Inc. and MIT Press, 1964a. URL http://www.softwarepreservation.org/projects/LISP/book/III_LispBook_Apr66.pdf#page=161.

Daniel G. Bobrow. String manipulation in the new language. Technical Report Memo 71 and Memorandum MAC-M-176, Artificial Intelligence Project and Project MAC, Massachusetts Institute of Technology, May 1964b. URL ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-071.pdf.

Erwin Book. The LISP version of the Meta Compiler. Technical report, System Development Corporation, November 1965. URL http://www.bitsavers.org/pdf/sdc/q32/TM-2710_330_00_The_LISP_Version_Of_The_Meta_Compiler_Nov65.pdf.

Edward Fredkin. A LISP II programming system. Proposal No. 64-12 submitted to System Development Corporation, December 1964. URL http://www.softwarepreservation.org/projects/LISP/lisp2/Fredkin-Proposal-LISP_II_Programming_System.pdf.

James Gosling, William N. Joy, and Guy L. Steele Jr. *The Java Language Specification*. Addison-Wesley, 1996. ISBN 0-201-63451-1.

Adolfo Guzmán and Harold V. McIntosh. CONVERT. *Communications of the ACM*, 9 (8):604–615, August 1966. URL http://dx.doi.org/10.1145/365758.365787.

Lowell Hawkinson. Technical progress report — LISP 2 — period ending July 6, 1966. Information International, Inc., July 1966. URL http://www.softwarepreservation.org/projects/LISP/lisp2/Hawkinson-Technical_Progress_Report-Jul1966.pdf.

Anthony C. Hearn. REDUCE: A user-oriented interactive system for algebraic simplification. In M. Klerer and J. Reinfelds, editors, *Interactive Systems for Experimental Applied Mathematics*, pages 79–90. Academic Press, 1968.

S. L. Kameny and L. Hawkinson. LISP II intermediate language. Technical Report TM-2710/220/00, System Development Corporation, November 1965. URL http://www.softwarepreservation.org/projects/LISP/lisp2/TM-2710_220_00-LISP_II_Intermediate_Language.pdf.

M. Levin. LISP II project. memo 7. Information International, Inc., 1965a. URL http://www.softwarepreservation.org/projects/LISP/lisp2/Levin-Memorandum_7-1965.pdf.

M. Levin. Programming examples: Memo 13. Information International, Inc., 1965b. URL http://www.softwarepreservation.org/projects/LISP/lisp2/Levin-Programming_Examples-Memorandum_13.pdf.

Michael Levin. Syntax of the new language. Technical Report Memo 68 and Memorandum MAC-M-158, Artificial Intelligence Project and Project MAC, Massachusetts Institute of Technology, May 1964a. URL ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-068.pdf.

Michael Levin. New language storage conventions. Technical Report Memo 69 and Memorandum MAC-M-159, Artificial Intelligence Project and Project MAC, Massachusetts Institute of Technology, May 1964b. URL ftp://publications.ai.mit.edu/ai-publications/pdf/AIM-069.pdf.

J. McCarthy. Storage conventions in LISP 2. Technical Report AIM-8, Stanford University Artificial Intelligence Project, September 1963. URL http://www.softwarepreservation.org/projects/LISP/stanford/Stanford-AIM-8.pdf.

John McCarthy. History of LISP. *SIGPLAN Not.*, 13(8):217–223, August 1978. ISSN 0362-1340. doi: 10.1145/960118.808387. URL http://doi.acm.org/10.1145/960118.808387.

John McCarthy, Paul W. Abrahams, Daniel J. Edwards, Timothy P. Hart, and Michael I. Levin. *Lisp 1.5 Programmer's Manual*. MIT Press, March 1962. URL http://www.softwarepreservation.org/projects/LISP/lisp15_family#Lisp_15_Programmers_Manual_.

Paul McJones. Lisp 2 section — History of LISP web site. Digitized copies of manuals and source code listings, August 2010. URL http://www.softwarepreservation.org/projects/LISP/lisp2_family/#LISP_2_.

Paul McJones. Guide to the Herbert Stoyan collection on LISP programming, 2011. URL http://www.computerhistory.org/collections/accession/102703236.

Paul McJones. The first international LISP conference (1963). Dusty Decks blog, April 2012a. URL http://www.mcjones.org/dustydecks/archives/2012/04/23/456/.

Paul McJones. Harold V. McIntosh and his students: Lisp escapes MIT. Dusty Decks blog, July 2012b. URL http://www.mcjones.org/dustydecks/archives/2012/07/06/239/.

Arthur Norberg. Paul W. Abrahams interview: October 15, 16, and 17, 2007; Deerfield, MA. In *ACM Oral History interviews*, New York, NY, USA, 2006. ACM. doi: 10.1145/1141880.1380529. URL http://doi.acm.org/10.1145/1141880.1380529. Interviewee-Abrahams, Paul W.

Kent Pitman. ANSI/INCITS 226-1994 : Programming Language Common Lisp. Technical report, InterNational Committee for Information Technology Standards (formerly NCITS), January 1994. URL http://www.lispworks.com/documentation/common-lisp.html.

Robert A. Saunders. The LISP system for the Q-32 computer. In Edmund C. Berkeley and Daniel G. Bobrow, editors, *The Programming Language LISP: Its Operation and Applications*, pages 220–238. The MIT Press, 1964. URL http://www.softwarepreservation.org/projects/LISP/book/III_LispBook_Apr66.pdf#page=230.

D. V. Schorre. META II a syntax-oriented compiler writing language. In *Proceedings of the 1964 19th ACM national conference*. ACM, 1964. URL http://doi.acm.org/10.1145/800257.808896.

Jules I. Schwartz and Clark Weissman. The SDC time-sharing system revisited. In *Proceedings of the 1967 22nd national conference*, ACM '67, pages 263–271, New York, NY, USA, 1967. ACM. doi: 10.1145/800196.805996. URL http://doi.acm.org/10.1145/800196.805996.

Jules I. Schwartz, Edward G. Coffman, and Clark Weissman. A general-purpose time-sharing system. In *Proceedings of the April 21-23, 1964, spring joint computer conference*, AFIPS '64 (Spring), pages 397–411, New York, NY, USA, 1964. ACM. doi: 10.1145/1464122.1464163. URL http://doi.acm.org/10.1145/1464122.1464163.

Guy L. Steele, Jr. and Richard P. Gabriel. The evolution of Lisp. In *The second ACM SIGPLAN conference on History of programming languages*, HOPL-II, pages 231–270, New York, NY, USA, 1993. ACM. ISBN 0-89791-570-4. doi: 10.1145/154766.155373. URL http://doi.acm.org/10.1145/154766.155373.

Ben Wegbreit. The ECL programming system. In *Proceedings of the November 16-18, 1971, fall joint computer conference*, AFIPS '71 (Fall), pages 253–262, New York, NY, USA, 1971. ACM. doi: 10.1145/1479064.1479113. URL http://doi.acm.org/10.1145/1479064.1479113.

Clark Weissmann. *LISP 1.5 Primer*. Dickenson Publishing Company, Inc., 1967. URL http://www.softwarepreservation.org/projects/LISP/book/Weismann_LISP1.5_Primer_1967.pdf.

Joel Weizenbaum. Review of 'The LISP 2 programming language and system' by P. W. Abrahams et al. *IEEE Transactions on Electronic Computers*, pages 236–238, April 1967. URL http://www.softwarepreservation.org/projects/LISP/lisp2/Weizenbaum-Review-LISP2.pdf.

V. Yngve. *COMIT Reference Manual*. MIT Press, 1962.