References

Peter Lee – Realistic Compiler Generation.
Denotational Semantics in several passes.
https://www.amazon.ca/Realistic-Compiler-Generation-Peter-Lee/dp/0262121417

Holt, Cordy, Wortman
S/SL
Syntax Semantic Language
Primitives for Parsing
Primitives (mechanisms) for *handles* to semantic checking
https://research.cs.queensu.ca/home/cordy/pub/downloads/ssl/
(paper in pt+ssl/doc/ssl.intro and various other places)

Cordy
OCG
Orthogonal Code Generator
Portable code generator in two passes, declarative
https://books.google.ca/books?id=X0OaMQEACAAJ&dq=bibliogroup:%22University+of+Toronto+Computer+Systems+Research+Institute+Technical+Report+CSRI%22&hl=en&sa=X&ved=2ahUKEwig1Legm8bqAhWvlHIEHYzzBYEQ6AEwBHoECAEQAQs

Fraser Davidson
https://www.researchgate.net/publication/220404697_The_Design_and_Application_of_a_Retargetable_Peephole_Optimizer
Portable code generator in two passes
(Used in gcc)

Smalltalk Blue Book
implementation of Smalltalk in a subset of Smalltalk
http://www.mirandabanda.org/bluebook/

Ron Cain
SmallC
very straight-forward implementation of a C compiler that could (IMO) be ported to any modern language
https://ia801008.us.archive.org/19/items/dr_dobbs_journal_vol_05_201803/dr_dobbs_journal_vol_05.pdf
(see number 45, May 1980 “A Small C Compiler for the 8080’s)

Anatomy of Lisp
John R. Allen
a Lisp Compiler written in a functional style
https://www.amazon.ca/Anatomy-Lisp-John-Allen/dp/007001115X

Flow-Based Programming
J. P. Morrison

https://jpaulm.github.io/fbp/
book: https://jpaulm.github.io/fbp/book.html
google group: https://groups.google.com/d/forum/flow-based-programming
slack:
https://join.slack.com/t/fbphq/shared_invite/enQtOTM4ODkzMTYyODE3LTJiMmNlZjhiMWY1MD
Y1ODA4Y2YzNDBlNDZlMTBkMDNlMjcwNzg2MGZhZjA2NjJjYTliYTM0OTIyYmM0Yzk0MDQ

Nils Holm
PROLOG Control in Six Slides
PROLOG is fundamentally very simple – it is just an exhaustive pattern matching tool
Nils Holm created the easiest (IMO) to understand description of PROLOG...
https://www.t3x.org/bits/prolog6.html

Eiffel, The Language
Bertrand Meyer
https://www.amazon.ca/Eiffel-The-Language/dp/0132479257

Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Volume One
(see Tiny Basic)
https://ia801302.us.archive.org/24/items/dr_dobbs_journal_vol_01/dr_dobbs_journal_vol_01.pdf

Dr. Dobb's Journal of Computer Calisthenics & Orthodontia, Volume Four
(my article "Potpourri of Lisp Functions" in number 38)
https://ia801007.us.archive.org/30/items/dr_dobbs_journal_vol_04_201803/
dr_dobbs_journal_vol_04.pdf

Jonathan Edwards
Getting to Simple
https://alarmingdevelopment.org/?p=766

Paul Bassett
Framing Software Reuse
https://www.amazon.ca/Framing-Software-Reuse-Lessons-World/dp/013327859X
(see also XVCL) https://www.sciencedirect.com/science/article/pii/S0167642304000978

Frits van der Wateren
Lisp for MC6800
https://github.com/guitarvydas/frits-van-der-wateren-lisp

Cordy, TXL
Source Transformation by Example
(functional, backtracking, parsing)
http://www.txl.ca/

R.C. Holt
Data Descriptors
https://dl.acm.org/doi/abs/10.1145/24039.24051

Greenspun’s Tenth Rule
https://en.wikipedia.org/wiki/Greenspun's_tenth_rule

Ilya Prigogine
Order Out of Chaos
https://www.amazon.com/Order-Out-Chaos-Dialogue-Thinkers-ebook/dp/B01N7H2J3B/ref=sr_1_1?dchild=1&keywords=ilya+prigogine+order&qid=1595281139&sr=8-1

C.A.R. Hoare CSP (Communicating Sequential Processes)
https://www.cs.cmu.edu/~crary/819-f09/Hoare78.pdf

Ron Cain
Small-C
https://en.wikipedia.org/wiki/Small-C

Ohm-js
object-oriented language for JS (based on PEG and META-II)
https://github.com/harc/ohm

PEG.js
PEG parser for JS (I feel that Ohm is “better”)
https://pegjs.org/

ESRAP
PEG parser for CL (Common Lisp)
https://scymtym.github.io/esrap/

Bryan Ford
PEG parsing – the definitive source
https://bford.info/packrat/

JQ tutorial
https://codefaster.substack.com/p/mastering-jq-part-1-59c

Rob Pike “Concurrency is not Parallelism”
https://vimeo.com/49718712