classmate
Date
Page

frontends
↳ commodity components

processors
↳ performance sensitive

optimization
code generation
in focus

BALANCE

~~static~~ single-assignment form
list scheduling
graph-coloring register allocation
data-flow analysis
scalar optimization

PHILOSOPHY

virtual registers
call-by-name param passing
↳ algol-60
tail recursion
↳ scheme

MOTIVATION FOR CONSTRUCTION

manage hardware latencies
predict execution flow
demand for memory ↑
independence / dependence

greedy heuristic search
deterministic finite automata
fixed-point algorithms
simple-theorm provers
algebraic simplifiers
pattern matchers
solvers for diophantine equations
pressburger arithmetic

APPROACH

engineering design
choice of intermediate representation
; profound impact → rest of compiler
(short shrift)

classmate
Date
Page

FRONT END

scanning
parsing
context-sensitive analysis

recognizers
finite automata
regular expressions
automating → scanner:

parsing:
context-free grammers
top-down recursive descent parsers
bottom-up table-driven LR(1) parsers

type systems:
context-sensitive analy
attribute grammers

INFRASTRUCTURE

intermediate representations:
trees
graphs
linear codes
symbol tables

runtime abstractions:
procedures
namespaces
linkage conventions
memory management

OPTIMIZATION

~~effects-based~~ taxonomy
scalar optimization

CODE GENERATION

instruction selection (tree) pattern matching

peephole-style matchers

instruction scheduling

list scheduling

register allocation

approximate NP-complete problem: 3-satisfiability.

CROSSCUTTING IDEAS

fixed-point algorithms

scanners

parsers

finite automata
↳ scanning
↳ LR(1) table construction
↳ pattern matchers
↳ for instruction selection