

Grontends	MOTIVATION FOR CONSTRUCTION
commodity components	
	manage handware latencies
processors	predict execution flow
performance sensitive	demand for memory?
	independence / dependence
optimization	
Code generation	greedy heuristic search
in focus	deterministic finite automata
	fixed-point algorithms
	Simple - theorm provers
	algebraic simplifiers
BALANCE	pattern matchers
	solvers for diophantine equations
Static single-assignment form	press burger anithmetic
list scheduling	
graph-coloring register allocation	on
data-flow analysis	APPROACH
scalar optimization	
	engineering design
	Choice of intermediate representation
PHILOSOPHY	: profound impact > rest of compiler
	(short shnift)
virtual registers	
call-by-name param passing	
(algol-60	
tail recursion	
(> scheme	

Classmate FRONT END vecognizers scanning finite automata parsing regular expressions context-sensitive analysis automating - scanner: parsing: type systems: context-free grammers Context - sensitive analy top-down recursive descent parsers attribute grammers

bottom-up table-driven LR(1) parsers

INFRASTRUCTURE

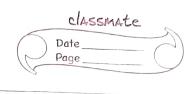
OPTIMIZATION

intermediate representations:

procedures trees namespaces linkage conventions linear codes memory management symbol tables

runtime abstractions:

s-based taxonomy scalar optimization



approximate NP-complete problem:

3-satisfiability.

	CODE	GrE	NEF	ZAT	ION
П					
ш					
П					
П					

instruction selection
ctreed pattern matching
peephole-style matchers

instruction scheduling list scheduling

register allocation

CROSS CUTTING IDEAS

fixed-point algorithms

sanners

parsers

Scanning

LR(1) table construction pattern matchers

finite automata

La for instruction selection