

CS202 Compiler Construction

March 11, 2003

CS 202 18

1

Midterm exam:

Good job!

Generally, results displayed a good mastery of concepts.

CS 202 18

18

2

Most common problems:

- expressions are not allowed at the top-level in programs (problem 6.a)
- sentences may have more than one derivation in an unambiguous grammar (problem 5)

Any questions?

CS 202 18

18

3

Next phase of compiler construction: IR transformation

IR is bridge between:

Parse trees

language specific

high level

Machine code

architecture specific

low level

CS 2023 18

IR

4

IR is more general than either

Can describe many languages

Can describe many architectures

IR is intermediate level

Lower level constructs (closer to machine level)

Many specifics abstracted

CS 2023 18

IR

5

Like parse trees,

IR made of *exprs* and *stmts*

Control structures translated into labeled *jumps*

Most other *stmts* *moves*

3-operand form typical

target = left op right

Also have function calls

CS 2023 18

IR *stmts*

6

IR statements:

```
MOVE expr,expr  
JUMP expr,{<const,label> list},label  
CJUMP expr,true label,false label  
LABEL  
CALL expr,{expr}  
SEQ stmt,stmt  
ESEQ stmt,expr
```

CS 202 18

IR statements

7

IR exprs have no side-effects

Each operand becomes either:
memory (can have single +/-)
register

Only integer constants in IR
Chars, integers, pointers

CS 202 18

IR exprs

8

IR transformation "flattens" expressions

Turns complex expressions into list of simple IR expressions

For example,
 $a = b + ((c=3)^d)$
becomes
move c,3
move t1,c*,d
move a,b+,t1

CS 202 18

example

9
