

CMSC 430 Final
Tuesday, December 17, 10:30am

Prof. Zelkowitz, Section 0101

December 17, 1996

Instructions

1. Do not open this exam until told to do so.
2. Answer all questions in the exam book.
3. If you want to have your grade posted under `~mvz/final.grades`, write the following on the inside cover of your exam book, **and then sign the statement**; if you do not sign, it will not be posted.

Please post my grade next to the code number _____ . (Remember this number!)

Name _____

4. All projects are due no later than noon on December 18, 1996.
5. Note: Computer accounts will be deleted December 25. If you want to save anything, move it to another machine today.
6. Do NOT turn page until told to do so.

- 1 [15]. Consider the regular expression $(0(01)^*01)^*$.
- (a) Give a nondeterministic finite state automaton that recognizes the same set.
 - (b) Give a regular grammar for this set.
 - (c) Give the minimal state deterministic finite state automaton that recognizes the same set.

2 [15]. Show that the following grammar is LALR(1). Give the parsing tables for the grammar.

```
S → A b | B C
A → a A | ε
B → a B | ε
C → C d | d
```

3 [15]. (a) Describe the set of strings accepted by the following grammar G_1 :

```
S → ( ) | ( S ) | S ( ) | S ( S )
```

(b) Describe the set of strings accepted by the following grammar G_2 :

```
P → T
T → ( | ) | T ( | T )
```

(c) Define a set of attributes for grammar G_2 such that $P.valid$ is true if and only if the string is also accepted by G_1 .

4 [15]. Assume the NIP language syntax were changed, so that like C, procedures could not be nested. A NIP program in this modified language would then look like:

```
P procedure
  ...
Q procedure
  ...
R procedure
  ...
start P
```

For each of the following structures, tell whether their implementation would change if you implemented this modified language as compared to the language actually implemented during the semester; and if the implementation changed, how you would change it.

(a) Activation records would still need to be implemented on a stack on procedure entry. (Doesn't change? Changes? If so, how?)

- (b) Static links are still needed in activation records. (Doesn't change? Changes? If so, how?)
- (c) Dynamic links are still needed in activation records. (Doesn't change? Changes? If so, how?)
- (d) Arrays need dope vectors. (Doesn't change? Changes? If so, how?)
- (e) Symbol table still needs a block structured design with scope rules. (Doesn't change? Changes? If so, how?)

5 [15]. Consider the code generation of a NIP program on the hac430 computer.

Assume variable A is at offset 16 in the local activation record.

Assume variable B is at offset 17 in the activation record of the procedure that is one level external to the currently executing procedure.

Assume variable C is at offset 18 in the local activation record.

Assume register 6 points to the start of the local activation record.

(a) Give the machine language instructions for executing the following statement:

```
A := B * C
```

(b) Consider the following code fragment:

```
A := 3;
B := 4;
if A-B then C := D; E := F + G end;
H := I + J * K;
write(1+L, M)
```

Give a sequence of quads for this program fragment.

(c) Draw boxes around each basic block in your answer to (b).

6 [15]. Consider the following nip-like program:

```

p procedure
integer i;
integer array a[5];

q procedure(integer: j; integer: b);
begin
j:=j+1;
i:=i+1;
b:=b+1;
call r(j,b)
end

r procedure(integer: j; integer: b)
begin
j:=j+1;
i:=i+1;
b:=b+1;
write(i,j,b)
end

begin
for i:= 0 to 5 do a[i]:= i;
i := 0;
call q(i, a[i]);
write(i, a[0], a[1], a[2], a[3], a[4], a[5])
end

start p

```

in registers, (e) Pipelined architecture, (f) High clock rate.

- (a) If all parameters are call by reference, what is printed?
- (b) If all parameters are call by value, what is printed?
- (c) If all parameters are call by name, what is printed?
- (d) If all parameters are call by value result, what is printed?

7 [10]. Answer each of the following.

(a) Assume the lower bound for all arrays start at 1, the storage for array A begins at location 2000, and each integer takes one storage location. Give the virtual origin for the array A declared as:

```
integer A[10,4];
```

- (b) What is a display? Why is it needed in some languages?
- (c) Which of the following (one or more) account for the increased execution speed of a RISC processor?
 - (a) cache memory, (b) Multiple registers, (c) Simple instruction format, (d) Built-in stack operations