1. Answer all questions in the exam book.

2. All projects are due no later than noon on May 22, 2003.

3. Note: Computer accounts will be deleted May 25. If you want to save anything, move it to another machine today.

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.

\[
\begin{align*}
S & \rightarrow \text{A b | B C} \\
A & \rightarrow \text{a A | } \epsilon \\
B & \rightarrow \text{a B | } \epsilon \\
C & \rightarrow \text{C d | d}
\end{align*}
\]

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

\[
\begin{align*}
S & \rightarrow ( ) | ( S ) | S ( ) | S ( S )
\end{align*}
\]
(b) Describe the set of strings accepted by the following grammar \(G_2\):

\[
\begin{align*}
P & \rightarrow T \\
T & \rightarrow ( | ) | T ( | T )
\end{align*}
\]
(c) Define a set of attributes for grammar \(G_2\) such that \(P.\text{valid}\) is true if and only if the string is also accepted by \(G_1\).

4 [15]. Assume the NIP language is like C, similar data structures and arrays and procedures cannot be nested. A NIP program in this modified language would then look like:

```c
main procedure
...
P procedure
...
Q procedure
...
R procedure
...
```

For each of the following structures, are they needed for NIP implementation? In one or two sentences explain how they are implemented.
(a) Activation records would still need to be implemented on a stack on procedure entry.
(b) Static links are still needed in activation records.
(c) Dynamic links are still needed in activation records.
(d) Arrays need dope vectors.
(e) Symbol table still needs a block structured design with scope rules.

5 [15]. Consider the code generation of a NIP-2 program, which is like NIP only procedures may have internally nested procedures, on a standard CISC-like processor like the Intel pentium.

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:
\[ A := B \times C \]

(b) Consider the following code fragment:

\[
\begin{align*}
A &:= 3; \\
B &:= 4; \\
\text{if } A-B \text{ then } C &:= D; \\
E &:= F + G \text{ end;}
\end{align*}
\]

\[
\begin{align*}
H &:= I + J \times K; \\
\text{write}(1+L, M)
\end{align*}
\]

Give a sequence of quads (three address code) for this program fragment.

(c) Draw boxes around each basic block in your answer to (b).
6 [15]. Consider the following NIP-like program:

```c
/* i and array a are global */
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

main procedure
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
```

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