

Midterm Review

CptS 355

Spring Semester, 2004

EECS, Washington State University

Structure

- 10:10-11AM AM Wednesday, March 10
- 10-20 multiple choice, true/false, short answer, essay questions
- Be precise, be verbose
- Bring pen or pencil
- Open book/notes/reference sheets

Readings

- Lecture notes
- Sebesta
 - Chapter 1 –Intro
 - Chapter 3 – Syntax and Semantics
 - Chapter 5 – Names
 - Chapter 14 – Scheme section
 - PostScript and Scheme notes

Introduction

- History
- Language evaluation criteria – (particularly as related to languages used so far as well as C and C++)
 - Readability
 - Writability
 - Reliability
- Implementation methods
 - Compilation – (lexical analysis, syntax analysis, semantic analysis, int. code generation, optimization, code generation)
 - Interpretation
 - Hybrid Implementation
 - Virtual machines

Sample Questions

- What programming language was developed for scientific computing in the 50s and 60s and is still in use today?
- What is a C programming language statement or feature that increases reliability? Why does it increase reliability?

Syntax – Chapter 3

- Tokens
- BNF – grammar
 - Implementing associativity (L or R) using recursion
 - Implementing operator precedence by grammar construction
- Derivation/Parse trees
 - Take a stream of tokens, and draw a parse tree
- Recognizer – syntax analyzer

Sample Questions

- In the following C program fragment, list the tokens.

```
if (x == y) {printf("hi");}
```

- List every sentence that is legal by the rules in the grammar given below.

$\langle P \rangle \rightarrow \langle Q \rangle \mid \langle Q \rangle + \langle Q \rangle$

$\langle Q \rangle \rightarrow - \langle F \rangle \mid \langle F \rangle$

$\langle F \rangle \rightarrow x$

- In the grammar given above, show the parse tree for the following sequence of tokens.

$x + - x$

Semantics – Chapter 3

- Operational – translate high-level construct to lower-level instructions for a virtual machine
- Axiomatic
 - Assignment axiom
 - Statement sequences
 - If
 - Loop axiom – find invariant

Sample Questions

- Give an operational semantics for a C while loop, given below, using the following virtual machine instructions: `goto`, `if`.

```
while (expr1) do { expr2; }
```

- Compute the weakest precondition for the following.

```
a = b + 3; a = a + 2; {a = 20}
```

- Give a precondition and loop invariant for the following program

```
x = 1;  
n = 3;  
while (x < n) do x = x + 1;  
{x = 3}
```

Names – Chapter 4

- Names
 - Variables
 - Functions
 - Properties
- Types
 - Simple
 - Compound – arrays, lists, sets
- Binding
 - When - Static vs. dynamic
 - What – Type vs. address vs. value

Bindings – Chapter 4

- Type bindings
 - Static vs. dynamic
 - Declarations: implicit vs. explicit
- Storage bindings
 - Storage model - static, stack, heap
 - Static
 - Stack dynamic
 - Explicit heap dynamic
 - Implicit heap dynamic
 - Allocation, deallocation, lifetime

Sample Questions

- Is C strongly typed, why or why not?
- Is Scheme strongly typed?
- Some languages (including C) allow a variable to be declared as a **static** variable, which defines its lifetime to be the program's entire execution time even if the variable is declared inside an inner block. Static variables behave like global variables in respect to their lifetimes. Discuss the advantages of this feature. Provide an example where such a variable would be useful.

More Names – Chapter 4

- Type checking and compatibility
 - Name equivalence
 - Structure equivalence
 - Strong typing
- Scoping
 - Static vs. dynamic
 - Namespaces
 - Referencing environment

Sample Questions

- In the following C program, to the right of each storage binding, list the kind of storage binding (e.g., static, stack-dynamic, explicit heap-dynamic, implicit heap-dynamic).

```
int x;

void main() {
    char *str;

    str = malloc(100);
    x = 4;
    first(x);
    free(str);
}

void first(int z) {
    char ch[100];
    int *f;
    char *ch2[100];

    second(z);
}

void second(int z) {
    z = 324;
}
```

Sample Questions

- In the following C program, draw the namespaces. What is the scope of the variable `ch2`? In the assignment marked `/* HERE */`, what is the referencing environment?

```
int x;

void main() {
    char *str;

    str = malloc(100);
    x = 4;
    first(x);
    free(str);
}

void first(int z) {
    char ch[100];
    int *f;
    char *ch2[100];

    second(z);
}

void second(int z) {
    z = 324; /* HERE */
}
```

Sample Questions

- What value would be printed by the following C program? *If* C had dynamic scoping (it doesn't) what value would be printed?

```
void main() {
    int x = 4;
    first();
    printf("%d", x);
}
int x = 3;
void first() {
    x = 5;
}
```

Sample Questions

Scheme and PostScript

- Create or modify a program
- Calisthenics
 - (list '(1 2) 3) =>
 - 3 4 add dup =>
- Name scoping
- Recursive functions for list processing (Scheme)