

Programming Languages Midterm Exam

Please write all answers in the blue book. Keep your answers brief!

1. (a) Can the following set be specified using a regular expression?
 - The set of all strings containing only the letters a, b, and c such that the as come before the bs, and the bs come before the cs.If so, provide the regular expression. If not, explain why not.
- (b) Can the following set be specified using a context free grammar (CFG)?
 - The set of all strings, containing only the letters a, b, and c, that are palindromes (are identical when read forward and backward, e.g. abcba).If so, provide the CFG. If not, explain why not.

2. (a) For each of pass-by-value, pass-by-reference, pass-by-value-result and pass-by-name, what would the following program print if that parameter passing mechanism was used?

```
program one;
  a: array[1..5] of integer;
  i: integer;

  procedure f(x: integer; y: integer)
  begin
    y := y + 1;
    x := x * 2;
    a[i] := a[i] + 15;
  end

begin (* main program *)
  for j := 1 to 5 loop
    a[j] := j;
  end loop;
  i := 1;
  f(a[i], i);
  for j := 1 to 5 loop
    print(a[j]);
  end loop;
end (* main program *)
```

- (b) What do I mean when I say “Java uses pass-by-value with pointer semantics for objects”?
3. (a) What does concurrency mean in the context of programming languages? Be sure your answer covers the situation where a program is executed on a single processor.
- (b) Write a simple Ada program that exhibits concurrency, so that the user might see different output in different runs of the program.
4. (a) Write a `map-list` function in Scheme, where `map-list` takes a list of functions, `funcs`, and a list of values, `L`, and returns a list of the results of applying each function in `funcs` to the corresponding value in `L`. For example,

```
> (map-list (list (lambda (x) (+ x 1)) (lambda (x) (* x 3))) '(2 5))
(3 15)
```

- (b) Write a function `ho-map-list` (where “ho” stands for “higher-order”) that takes a list of functions, `funcs`, and returns a function that takes a list of values, `L`, and applies each function in `funcs` to the corresponding element of `L`. For example,

```
> (define f (ho-map-list (list (lambda (x) (+ x 1)) (lambda (x) (* x 3)))))
> (f '(2 5))
(3 15)
```

In writing `ho-map-list`, you cannot call any external user-defined function (such as `map-list` above). That is, `ho-map-list` has to be self-contained.

5. (a) Assuming the following program is written in a statically scoped language, what would it print?

```
program two;
  x: integer;
  y: integer;

  procedure A()
  begin
    print(x,y);
  end;

  procedure B(procedure C())
    x:integer;
  begin
    x := 7;
    C();
  end;

begin (* main program *)
  x := 1;
  y := 2;
  B(A);
end;
```

- (b) What would the above program print if it were written in a dynamically scoped language?

6. **Note: The λ -calculus will not be on the Spring 2015 midterm (but will be on the final exam).**

- (a) What does “normal form” mean in the context of the λ -calculus?
- (b) Give an example of a term that is in normal form.
- (c) Give an example of a term that has no normal form and justify your answer.
- (d) Give an example, if possible, of a term that would be reduced to normal form under normal order evaluation but would not be reduced to normal form under applicative order evaluation. If an example is not possible, state why not, and if it is possible, explain it.
- (e) Give an example, if possible, of a term that would be reduced to normal form under applicative order evaluation but would not be reduced to normal form under normal order evaluation. If an example is not possible, state why not, and if it is possible, explain it.