

Sample Mid-Term Exam 1

CS 5510, Fall 2009

September 18

Instructions: You have fifty minutes to complete this open-book, open-note, closed-computer exam. Please write all answers in the provided space, plus the back of the exam if necessary.

1) Given the following grammar:

```
⟨weed⟩ = 'leaf
        | (list 'branch ⟨weed⟩ ⟨weed⟩)
        | (list 'stem ⟨weed⟩)
```

Which of the following expressions are examples of ⟨weed⟩?

- a) (list 'leaf)
- b) (list 'stem)
- c) (list 'branch (list 'branch 'leaf 'leaf) 'leaf)
- d) (list 'stem 'leaf)

2) Explain why the following is a ⟨weed⟩:

```
(list 'branch (list 'stem 'leaf) (list 'branch 'leaf 'leaf))
```

3) Provide a `define-type` declaration for `Weed` that is a suitable representation for `<weed>`s.

4) Implement the function `weed-forks`, takes a `<weed>` and returns the number of `branches` that it contains. Your implementation must follow the shape of the data definition.

5) Given the following expression in the book language with `with` and `fun`:

```
{with {g {fun {z} {f z}}}
  {with {f {fun {z} {g z}}}
    {with {y {with {f {fun {z} {f {+ z x}}}}
      {f y}}}
      {+ y q}}}}
```

- a) Draw arrows on the above expression from each bound variable to its binding occurrence.
- b) List the free variables: _____ and bound variables: _____

6) Given the following expression:

```
{with {g {fun {x} {fun {y} {+ y x}}}}
  {with {x 13}
    {with {f {g 6}}
      {f x}}}}
```

Describe a trace of the evaluation in terms of arguments to an `interp` function for every call. (There will be 16 calls.) The `interp` function takes two arguments — an expression and a deferred substitution — so show both for each call. For number, variable, and `fun` expressions, show the result value, which is immediate. Use the back of the exam for additional space, and use the following abbreviations to save time:

$$\begin{aligned} E_0 &= \text{the whole expression} \\ E_1 &= \{\text{fun } \{x\} \{\text{fun } \{y\} \{+ y x\}\}\} \\ E_2 &= \{\text{with } \{x 13\} \{\text{with } \{f \{g 6\}\} \{f x\}\}\} \\ E_3 &= \{\text{with } \{f \{g 6\}\} \{f x\}\} \end{aligned}$$

Answers

1) (c) and (d)

2) Since 'leaf is a <weed> by line 1 of the definition, then by line 3, (list 'stem 'leaf) is a <weed>, and by line 2, (list 'branch 'leaf 'leaf) is a <weed>. Finally, then, by line 2 again, (list 'branch (list 'stem 'leaf) (list 'branch 'leaf 'leaf)) is a <weed>.

3) (define-type Weed
 [leaf]
 [stem (rest Weed?)]
 [branch (left Weed?)
 (right Weed?)])

4) ; weed-forks : Weed -> num
 (define (weed-forks w)
 (type-case Weed w
 [leaf () 0]
 [stem (rest) (weed-forks rest)]
 [branch (l r) (+ 1
 (weed-forks l)
 (weed-forks r))]))
 (test (weed-forks (leaf)) 0)
 (test (weed-forks (stem (leaf))) 0)
 (test (weed-forks (stem (branch (leaf) (leaf)))) 1)
 (test (weed-forks (branch (branch (leaf) (leaf)) (leaf))) 2)

5)

```

      v-----,
{with {g {fun {z} {f z}}}
  ^-----,
      v---+-,
{with {f {fun {z} {g z}}}
  ^-----,
      v---+-----,
{with {y {with {f {fun {z} {f {+ z x}}}}
  ^-----,
  |      {f y}}}
  |
  {+ y q}}}}

```

Free: f, x, y, q Bound: z, g, f, y

6)

expr = E_0
 subs = (mtSub)

expr = E_1
 subs = (mtSub)
 result = (closureV 'x {fun {y} {+ y x}} (mtSub)) = C_1

expr = E_2
 subs = (aSub 'g C_1 (mtSub)) = S_1

```

expr  = 13
subs  = S1
result = (numV 13)

expr  = E3
subs  = (aSub 'x (numV 13) S1) = S2

expr  = {g 6}
subs  = S2

expr  = g
subs  = S2
result = C1

expr  = 6
subs  = S2
result = (numV 6)

expr  = {fun {y} {+ y x}}
subs  = (aSub 'x (numV 6) (mtSub)) = S3
result = (closureV 'y {+ y x} S3) = C2

expr  = {f x}
subs  = (aSub 'f C2 S2) = S4

expr  = f
subs  = S4
result = C2

expr  = x
subs  = S4
result = (numV 13)

expr  = {+ y x}
env   = (aSub 'y (numV 13) S3) = S5

expr  = y
env   = S5
result = (numV 13)

expr  = x
env   = S5
result = (numV 6)

```