Sample Mid-Term Exam 1

CS 5510, Fall 2011

September 22

Instructions: You have eighty 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 `branch`es 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 evalaution 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. Assume a variant of `interp` that interprets `with` directly. 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:

E_0	=	the whole expression
E_1	=	`{fun {x} {fun {y} {+ y x}}}`
E_2	=	`{with {x 13} {with {f {g 6}} {f x}}}`
E_3	=	`{with {f {g 6}} {f x}}`

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 = $\boxed{E_0}$
subs = `(mtSub)`

expr = $\boxed{E_1}$
subs = `(mtSub)`
result = `(closureV 'x` $\boxed{\texttt{\{fun \{y\} \{+ y x\}\}}}$ `(mtSub)) =` C_1

expr = $\boxed{E_2}$
subs = `(aSub 'g` C_1 `(mtSub)) =` S_1

expr = $\boxed{\texttt{13}}$
subs = S_1
result = (numV 13)

expr = $\boxed{E_3}$
subs = (aSub ’x (numV 13) S_1) = S_2

expr = $\boxed{\texttt{\{g 6\}}}$
subs = S_2

expr = $\boxed{\texttt{g}}$
subs = S_2
result = C_1

expr = $\boxed{\texttt{6}}$
subs = S_2
result = (numV 6)

expr = $\boxed{\texttt{\{fun \{y\} \{+ y x\}\}}}$
subs = (aSub ’x (numV 6) (mtSub)) = S_3
result = (closureV ’y $\boxed{\texttt{\{+ y x\}}}$ S_3) = C_2

expr = $\boxed{\texttt{\{f x\}}}$
subs = (aSub ’f C_2 S_2) = S_4

expr = $\boxed{\texttt{f}}$
subs = S_4
result = C_2

expr = $\boxed{\texttt{x}}$
subs = S_4
result = (numV 13)

expr = $\boxed{\texttt{\{+ y x\}}}$
env = (aSub ’y (numV 13) S_3) = S_5

expr = $\boxed{\texttt{y}}$
env = S_5
result = (numV 13)

expr = $\boxed{\texttt{x}}$
env = S_5
result = (numV 6)