

CMPE 320 PRINCIPLES OF PROGRAMMING LANGUAGES MIDTERM ANSWERS

1. $\langle \text{assign} \rangle \rightarrow \langle \text{var} \rangle = \langle \text{expr} \rangle$
 $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term1} \rangle \mid \langle \text{expr} \rangle - \langle \text{term1} \rangle \mid \langle \text{term1} \rangle$
 $\langle \text{term1} \rangle \rightarrow \langle \text{term1} \rangle * \langle \text{term2} \rangle \mid \langle \text{term1} \rangle / \langle \text{term2} \rangle \mid \langle \text{term2} \rangle$
 $\langle \text{term2} \rangle \rightarrow \langle \text{term3} \rangle ^ \langle \text{term2} \rangle \mid \langle \text{term3} \rangle$
 $\langle \text{term3} \rangle \rightarrow (\langle \text{expr} \rangle) \mid \langle \text{var} \rangle$
 $\langle \text{var} \rangle \rightarrow a \mid b \mid \dots \mid z$

2. The underlined parts are the syntaxes and the remaining parts denote the semantic rules of the attribute grammar.

```

 $\langle \text{expr}_1 \rangle \rightarrow \langle \text{expr}_2 \rangle + \langle \text{term} \rangle$ 
 $\langle \text{expr}_1 \rangle.m \leftarrow \langle \text{expr}_2 \rangle.m$ 
 $\langle \text{expr}_1 \rangle.n \leftarrow \langle \text{expr}_2 \rangle.n$ 
for (i = 1 to  $\langle \text{expr}_2 \rangle.m$ )
  for (j = 1 to  $\langle \text{expr}_2 \rangle.n$ )
     $\langle \text{expr}_1 \rangle.value(i, j) \leftarrow \langle \text{expr}_2 \rangle.value(i, j) + \langle \text{term} \rangle.value(i, j)$ 
  endfor
endfor

```

```

 $\langle \text{expr} \rangle \rightarrow \langle \text{term} \rangle$ 
 $\langle \text{expr} \rangle.m \leftarrow \langle \text{term} \rangle.m$ 
 $\langle \text{expr} \rangle.n \leftarrow \langle \text{term} \rangle.n$ 
for (i = 1 to  $\langle \text{term} \rangle.m$ )
  for (j = 1 to  $\langle \text{term} \rangle.n$ )
     $\langle \text{expr} \rangle.value(i, j) \leftarrow \langle \text{term} \rangle.value(i, j)$ 
  endfor
endfor

```

```

 $\langle \text{term}_1 \rangle \rightarrow \langle \text{term}_2 \rangle * \langle \text{factor} \rangle$ 
 $\langle \text{term}_1 \rangle.m \leftarrow \langle \text{term}_2 \rangle.m$ 
 $\langle \text{term}_1 \rangle.n \leftarrow \langle \text{factor} \rangle.n$ 
for (i = 1 to  $\langle \text{term}_2 \rangle.m$ )
  for (j = 1 to  $\langle \text{factor} \rangle.n$ )
     $\langle \text{term}_1 \rangle.value(i, j) \leftarrow 0$ 
    for (k = 1 to  $\langle \text{term}_2 \rangle.n$ )
       $\langle \text{term}_1 \rangle.value(i, j) \leftarrow \langle \text{term}_1 \rangle.value(i, j) + \langle \text{term}_2 \rangle.value(i, k) * \langle \text{factor} \rangle.value(k, j)$ 
    endfor
  endfor
endfor

```

```

 $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle$ 
 $\langle \text{term} \rangle.m \leftarrow \langle \text{factor} \rangle.m$ 
 $\langle \text{term} \rangle.n \leftarrow \langle \text{factor} \rangle.n$ 
for (i = 1 to  $\langle \text{factor} \rangle.m$ )
  for (j = 1 to  $\langle \text{factor} \rangle.n$ )
     $\langle \text{term} \rangle.value(i, j) \leftarrow \langle \text{factor} \rangle.value(i, j)$ 
  endfor
endfor

```

```

 $\langle \text{factor} \rangle \rightarrow ( \langle \text{expr} \rangle )$ 
 $\langle \text{factor} \rangle.m \leftarrow \langle \text{expr} \rangle.m$ 
 $\langle \text{factor} \rangle.n \leftarrow \langle \text{expr} \rangle.n$ 
for (i = 1 to  $\langle \text{expr} \rangle.m$ )
  for (j = 1 to  $\langle \text{expr} \rangle.n$ )
     $\langle \text{factor} \rangle.value(i, j) \leftarrow \langle \text{expr} \rangle.value(i, j)$ 
  endfor
endfor

```

```

<factor> -> <id>
<factor>.m <- <id>.m
<factor>.n <- <id>.n
for (i = 1 to <id>.m)
  for (j = 1 to <id>.n)
    <factor>.value(i,j) <- <id>.value(i,j)
  endfor
endfor

```

3. $M_r(\text{repeat } \langle \text{st-list} \rangle \text{ until } \langle \text{bool} \rangle, s) \equiv$
- ```

if $M_{sl}(\langle \text{st-list} \rangle, s) = \text{error}$
 then error
 else if $M_b(\langle \text{bool} \rangle, M_{sl}(\langle \text{st-list} \rangle, s)) = \text{error}$
 then error
 else if $(M_b(\langle \text{bool} \rangle, M_{sl}(\langle \text{st-list} \rangle, s)) = \text{true})$
 then $M_{sl}(\langle \text{st-list} \rangle, s)$
 else $M_r(\text{repeat } \langle \text{st-list} \rangle \text{ until } \langle \text{bool} \rangle, M_{sl}(\langle \text{st-list} \rangle, s))$

```

$M_b(\langle \text{var} \rangle_1 = \langle \text{var} \rangle_2, s) \equiv$

```

if $\text{VarMap}(\langle \text{var} \rangle_1, s) = \text{undef}$
 then error
 else if $\text{VarMap}(\langle \text{var} \rangle_2, s) = \text{undef}$
 then error
 else if $\text{VarMap}(\langle \text{var} \rangle_1, s) = \text{VarMap}(\langle \text{var} \rangle_2, s)$
 then true
 else false

```

$M_{sl}(\langle \text{ass-st} \rangle \langle \text{st-list} \rangle, s) \equiv$

```

if $M_a(\langle \text{ass-st} \rangle, s) = \text{error}$
 then error
 else $M_{sl}(\langle \text{st-list} \rangle, M_a(\langle \text{ass-st} \rangle, s))$

```

$M_{sl}(\langle \text{ass-st} \rangle, s) \equiv$

```

 $M_a(\langle \text{ass-st} \rangle, s)$

```

$M_a(\langle \text{var} \rangle_1 = \langle \text{var} \rangle_2, s) \equiv$

```

if $\text{VarMap}(\langle \text{var} \rangle_2, s) = \text{undef}$
 then error
 else $\{ \langle i_1, v_1 \rangle, \dots, \langle i_n, v_n \rangle \}$ where
 $v_j = \text{VarMap}(i_j, s)$, if $i_j \neq \langle \text{var} \rangle_1$
 $\text{VarMap}(\langle \text{var} \rangle_2, s)$, if $i_j = \langle \text{var} \rangle_1$

```

4. (DEFINE (Points lis team)
- ```

(IF (NULL? lis)
  0
  (+ (COND ((EQ? (CAAR lis) team)
            (COND ((> (CADDR (CAR lis)) (CADDR (CAR lis))) 3)
                  ((= (CADDR (CAR lis)) (CADDR (CAR lis))) 1)
                  (ELSE 0)))
      ((EQ? (CADAR lis) team)
       (COND ((> (CADDR (CAR lis)) (CADDR (CAR lis))) 0)
              ((= (CADDR (CAR lis)) (CADDR (CAR lis))) 1)
              (ELSE 3)))
      (ELSE 0))
    (Points (CDR lis) team))))

```