

Compiler Construction

Lecture 19: Code Generation IV

(Translation Example & Correctness)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

`noll@cs.rwth-aachen.de`

`http://www-i2.informatik.rwth-aachen.de/i2/cc08/`

Summer semester 2008

- 1 Repetition: Translation of EPL into AM Programs
- 2 A Translation Example
- 3 Correctness of the Translation

The Abstract Machine AM

Definition (Abstract machine for EPL)

The **abstract machine for EPL (AM)** is defined by the **state space**

$$S := PC \times DS \times PS$$

with

- the **program counter** $PC := \mathbb{N}$,
- the **data stack** $DS := \mathbb{Z}^*$ (top of stack to the right), and
- the **procedure stack** (or: **runtime stack**) $PS := \mathbb{Z}^*$ (top of stack to the left).

Thus a state $s = (l, d, p) \in S$ is given by

- a program label $l \in PC$,
- a data stack $d = d.r : \dots : d.1 \in DS$, and
- a procedure stack $p = p.1 : \dots : p.t \in PS$.

Structure of Procedure Stack

The semantics of procedure and transfer instructions requires a particular structure of the procedure stack $p \in PS$: it must be composed of **frames** (or: **activation records**) of the form

$$sl : dl : ra : v_1 : \dots : v_k$$

where

static link sl : points to frame of surrounding declaration environment
 $\implies$ used to access non-local variables

dynamic link dl : points to previous frame (i.e., of calling procedure)
 $\implies$ used to remove topmost frame after termination of procedure call

return address ra : program label after termination of procedure call
 $\implies$ used to continue program execution after termination of procedure call

local variables v_i : values of locally declared variables

The Initial Symbol Table

By Definition 17.9, an EPL program $P = \text{in/out } I_1, \dots, I_n; K. \in Pgm$ has the **semantics** $\mathfrak{M}[[P]] : \mathbb{Z}^n \dashrightarrow \mathbb{Z}^n$.

Given $(z_1, \dots, z_n) \in \mathbb{Z}^n$, we choose the **initial state**

$$s := (1, \varepsilon, \underbrace{0 : 0 : 0 : z_1 : \dots : z_n}_{\text{I/O frame}}) \in S = PC \times DS \times PS$$

Thus the corresponding **initial symbol table** has n entries:

$$\text{st}_{\text{I/O}}(I_j) := (\text{var}, 0, j) \quad \text{for every } j \in [n]$$

Translation of EPL into AM Programs

Translation mapping $\text{trans} : Pgm \dashrightarrow AM$ defined using

- symbol table:

$$\begin{aligned} Tab := \{st \mid st : Ide \dashrightarrow (\{const\} \times \mathbb{Z}) \\ \cup (\{var\} \times Lev \times Off) \\ \cup (\{proc\} \times PC \times Lev \times Size)\} \end{aligned}$$

- $\text{update} : Dcl \times Tab \times Lev \dashrightarrow Tab$
for extending the symbol table by **declarations**
- $kt : Block \times Tab \times PC \times Lev \dashrightarrow AM$ for translating **blocks**
- $dt : Dcl \times Tab \times Lev \dashrightarrow AM$
for translating the **bodies of procedure declarations**
- $ct : Cmd \times Tab \times PC \times Lev \dashrightarrow AM$
for translating **commands**
- $bt : BExp \times Tab \times PC \times Lev \dashrightarrow AM$
for translating **Boolean expressions**
- $at : AExp \times Tab \times PC \times Lev \dashrightarrow AM$
for translating **arithmetic expressions**

- 1 Repetition: Translation of EPL into AM Programs
- 2 A Translation Example
- 3 Correctness of the Translation

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

Intermediate code:

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.  
  
trans(in/out  $I_1, \dots, I_n; K.$ ) :=  
  1 : CALL( $a, 0, \text{size}(K)$ );  
  2 : JMP(0);  
  kt( $K, \text{st}_{I/0}, a, 1$ )  
  
 $\text{st}_{I/0} = [x \mapsto (\text{var}, 0, 1)]$ 
```

Intermediate code:

```
trans(in/out x;  $K.$ )
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

$kt(D\ C, st, a, l) :=$

```
dt(D, update(D, st, l), l)  
ct(C, update(D, st, l), a, l)  
a' : RET;
```

$st_{I/O} = [x \mapsto (\text{var}, 0, 1)]$

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
   kt(K, stI/O, a0, 1)
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
  F()  
y := 1;  
F();  
x := y.
```

$\text{update}(\text{var } I_1, \dots, I_n; \text{st}, l) :=$

$\text{st}[I_1 \mapsto (\text{var}, l, 1), \dots, I_n \mapsto (\text{var}, l, n)]$

$\text{update}(\text{proc } I_1; K_1; \dots; I_n; K_n; \text{st}, l) :=$

$\text{st}[I_1 \mapsto (\text{proc}, a_1, l, \text{size}(K_1)), \dots, I_n \mapsto (\text{proc}, a_n, l, \text{size}(K_n))]$

$\text{st}' = [x \mapsto (\text{var}, 0, 1),$

$y \mapsto (\text{var}, 1, 1),$

$F \mapsto (\text{proc}, a_1, 1, 0)]$

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
   dt(D, update(D, stI/0, 1), 1)  
   ct(C, update(D, stI/0, 1), a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
  F()  
y := 1;  
F();  
x := y.
```

$\text{dt}(\text{proc } I_1; K_1; \dots; I_n; K_n; \text{st}, l) :=$
 $\text{kt}(K_1, \text{st}, a_1, l + 1)$

$\vdots$
 $\text{kt}(K_n, \text{st}, a_n, l + 1)$
where $\text{st}(I_j) = (\text{proc}, a_j, \dots, \dots)$ for every $j \in [n]$

$\text{st}' = [\text{x} \mapsto (\text{var}, 0, 1),$
 $\text{y} \mapsto (\text{var}, 1, 1),$
 $\text{F} \mapsto (\text{proc}, a_1, 1, 0)]$

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
    dt(D, st', 1)  
    ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

$kt(D\ C, st, a, l) :=$

```
dt(D, update(D, st, l), l)  
ct(C, update(D, st, l), a, l)  
a' : RET;
```

$st' = [x \mapsto (\text{var}, 0, 1),$
 $y \mapsto (\text{var}, 1, 1),$
 $F \mapsto (\text{proc}, a_1, 1, 0)]$

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
   kt(KF, st', a1, 2)  
   ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
x := y.
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
   ct(CF, st', a1, 2)  
a3 : RET;  
   ct(C, st', a0, 1)  
a2 : RET;
```

```
ct(if B then C1 else C2, st, a, l) :=  
  bt(B, st, a, l)  
  a' : JFALSE(a'');  
  ct(C1, st, a' + 1, l)  
  a'' - 1 : JMP(a''');  
  ct(C2, st, a'', l)  
  a''' :
```

```
st' = [x ↦ (var, 0, 1),  
       y ↦ (var, 1, 1),  
       F ↦ (proc, a1, 1, 0)]
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

$\text{bt}(A_1 > A_2, \text{st}, a, l) := \text{at}(A_1, \text{st}, a, l)$
 $\quad \text{at}(A_2, \text{st}, a', l)$
 $\quad a'' : \text{GT};$

$\text{st}' = [\text{x} \mapsto (\text{var}, 0, 1),$
 $\quad \text{y} \mapsto (\text{var}, 1, 1),$
 $\quad \text{F} \mapsto (\text{proc}, a_1, 1, 0)]$

Intermediate code:

```
1 : CALL( $a_0$ , 0, 1);  
2 : JMP(0);  
   bt(x > 1, st',  $a_1$ , 2)  
 $a_4$  : JFALSE( $a_3$ );  
      ct( $C_1$ , st',  $a_4 + 1$ , 2)  
 $a_3$  : RET;  
      ct( $C$ , st',  $a_0$ , 1)  
 $a_2$  : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

$at(I, st, a, l) :=$

$$\begin{cases} a : LIT(z); & \text{if } st(I) = (\text{const}, z) \\ a : LOAD(l - lev, off); & \text{if } st(I) = (\text{var}, lev, off) \end{cases}$$
$$st' = [x \mapsto (\text{var}, 0, 1), \\ y \mapsto (\text{var}, 1, 1), \\ F \mapsto (\text{proc}, a_1, 1, 0)]$$

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
   at(x, st', a1, 2)  
   at(1, st', a', 2)  
   GT;  
a4 : JFALSE(a3);  
      ct(C1, st', a4 + 1, 2)  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.  
  
 $at(z, st, a, l) := a : LIT(z);$   
  
 $st' = [x \mapsto (var, 0, 1),$   
         $y \mapsto (var, 1, 1),$   
         $F \mapsto (proc, a_1, 1, 0)]$ 
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
       $at(1, st', a', 2)$   
      GT;  
a4 : JFALSE(a3);  
      ct(C1, st', a4 + 1, 2)  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.
```

```
ct(I := A, st, a, l) :=  
  at(A, st, a, l)  
  a' : STORE(l - lev, off);  
  if st(I) = (var, lev, off)  
st' = [x ↦ (var, 0, 1),  
       y ↦ (var, 1, 1),  
       F ↦ (proc, a1, 1, 0)]
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
      LIT(1);  
      GT;  
a4 : JFALSE(a3);  
      ct(C1, st', a4 + 1, 2)  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.  
  
 $\text{at}(A_1 + A_2, \text{st}, a, l) := \text{at}(A_1, \text{st}, a, l)$   
                                   $\text{at}(A_2, \text{st}, a', l)$   
                                   $a'' : \text{ADD};$   
  
 $\text{st}' = [x \mapsto (\text{var}, 0, 1),$   
          $y \mapsto (\text{var}, 1, 1),$   
          $F \mapsto (\text{proc}, a_1, 1, 0)]$ 
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
      LIT(1);  
      GT;  
a4 : JFALSE(a3);  
       $\text{at}(y * x, \text{st}', a_4 + 1, 2)$   
      STORE(1, 1);  
       $\text{at}(x - 1, \text{st}', a', 2)$   
      STORE(2, 1);  
      ct(F(), st', a'', 2)  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
  F()  
  y := 1;  
  F();  
  x := y.  
  
ct(I(), st, a, l) :=  
  a : CALL(ca, l - lev, loc);  
if st(I) = (proc, ca, lev, loc)  
st' = [x ↦ (var, 0, 1),  
       y ↦ (var, 1, 1),  
       F ↦ (proc, a1, 1, 0)]
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
      LIT(1);  
      GT;  
a4 : JFALSE(a3);  
      LOAD(1, 1);  
      LOAD(2, 1);  
      MULT;  
      STORE(1, 1);  
      LOAD(2, 1);  
      LIT(1);  
      SUB;  
      STORE(2, 1);  
      ct(F(), st', a'', 2)  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.  
  
 $st' = [x \mapsto (\text{var}, 0, 1),$   
          $y \mapsto (\text{var}, 1, 1),$   
          $F \mapsto (\text{proc}, a_1, 1, 0)]$ 
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
      LIT(1);  
      GT;  
a4 : JFALSE(a3);  
      LOAD(1, 1);  
      LOAD(2, 1);  
      MULT;  
      STORE(1, 1);  
      LOAD(2, 1);  
      LIT(1);  
      SUB;  
      STORE(2, 1);  
      CALL(a1, 1, 0);  
a3 : RET;  
      ct(C, st', a0, 1)  
a2 : RET;
```

Example: Factorial Function I

Example 19.1 (Factorial function)

Source code:

```
in/out x;  
var y;  
proc F;  
  if x > 1 then  
    y := y * x;  
    x := x - 1;  
    F()  
  y := 1;  
  F();  
  x := y.  
  
 $st' = [x \mapsto (\text{var}, 0, 1),$   
          $y \mapsto (\text{var}, 1, 1),$   
          $F \mapsto (\text{proc}, a_1, 1, 0)]$ 
```

Intermediate code:

```
1 : CALL(a0, 0, 1);  
2 : JMP(0);  
a1 : LOAD(2, 1);  
      LIT(1);  
      GT;  
a4 : JFALSE(a3);  
      LOAD(1, 1);  
      LOAD(2, 1);  
      MULT;  
      STORE(1, 1);  
      LOAD(2, 1);  
      LIT(1);  
      SUB;  
      STORE(2, 1);  
      CALL(a1, 1, 0);  
a3 : RET;  
a0 : LIT(1);  
      STORE(0, 1);  
      CALL(a1, 0, 0);  
      LOAD(0, 1);  
      STORE(1, 1);  
a2 : RET;
```

Example: Factorial Function II

Example 19.2 (Factorial function; continued)

Code with symbolic addresses:

```
1 : CALL(a0,0,1);
2 : JMP(0);
a1 : LOAD(2,1);
    LIT(1);
    GT;
a4 : JFALSE(a3);
    LOAD(1,1);
    LOAD(2,1);
    MULT;
    STORE(1,1);
    LOAD(2,1);
    LIT(1);
    SUB;
    STORE(2,1);
    CALL(a1,1,0);
a3 : RET;
a0 : LIT(1);
    STORE(0,1);
    CALL(a1,0,0);
    LOAD(0,1);
    STORE(1,1);
a2 : RET;
```

Linearized

($a_0 = 17, a_1 = 3, a_2 = 22, a_3 = 16, a_4 = 6$):

```
1 : CALL(17,0,1);
2 : JMP(0);
3 : LOAD(2,1);
4 : LIT(1);
5 : GT;
6 : JFALSE(16);
7 : LOAD(1,1);
8 : LOAD(2,1);
9 : MULT;
10 : STORE(1,1);
11 : LOAD(2,1);
12 : LIT(1);
13 : SUB;
14 : STORE(2,1);
15 : CALL(3,1,0);
16 : RET;
17 : LIT(1);
18 : STORE(0,1);
19 : CALL(3,0,0);
20 : LOAD(0,1);
21 : STORE(1,1);
22 : RET;
```

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

<i>PC</i>	<i>DS</i>	<i>PS</i>
1	ε	<u>0 : 0 : 0 : 2</u>

```
1: CALL(17,0,1);
2: JMP(0);
3: LOAD(2,1);
4: LIT(1);
5: GT;
6: JFALSE(16);
7: LOAD(1,1);
8: LOAD(2,1);
9: MULT;
10: STORE(1,1);
11: LOAD(2,1);
12: LIT(1);
13: SUB;
14: STORE(2,1);
15: CALL(3,1,0);
16: RET;
17: LIT(1);
18: STORE(0,1);
19: CALL(3,0,0);
20: LOAD(0,1);
21: STORE(1,1);
22: RET;
```

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);			
4 : LIT(1);			
5 : GT;			
6 : JFALSE(16);			
7 : LOAD(1,1);			
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	0 : 0 : 0 : 2
2 : JMP(0);	17	ε	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
3 : LOAD(2,1);	18	1	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
4 : LIT(1);			
5 : GT;			
6 : JFALSE(16);			
7 : LOAD(1,1);			
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;			
6 : JFALSE(16);			
7 : LOAD(1,1);			
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);			
7 : LOAD(1,1);			
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);			
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT ;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);			
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16) ;	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;			
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);			
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);	8	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
11 : LOAD(2,1);			
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:

	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT ;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);	8	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
12 : LIT(1);			
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);	8	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
12 : LIT(1);	10	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
13 : SUB;			
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ϵ	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ϵ	4 : 3 : 2 : 0 : <u>0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0</u> : 0 : 0 : 0 : 2
4 : LIT(1);	19	ϵ	<u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
5 : GT;	3	ϵ	3 : 2 : 20 : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
6 : JFALSE(16);	4	2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
9 : MULT;	7	ϵ	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
10 : STORE(1,1);	8	1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
12 : LIT(1);	10	2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
13 : SUB;	11	ϵ	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 2</u> : 0 : 0 : 0 : 2
14 : STORE(2,1);			
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	4 : 3 : 2 : 0 : <u>0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0</u> : 0 : 0 : 0 : 2
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
5 : GT;	3	ε	3 : 2 : 20 : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
6 : JFALSE(16);	4	2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
9 : MULT;	7	ε	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
10 : STORE(1,1);	8	1	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
12 : LIT(1);	10	2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 1</u> : 0 : 0 : 0 : 2
13 : SUB;	11	ε	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 2</u> : 0 : 0 : 0 : 2
14 : STORE(2,1);	12	2	<u>3 : 2 : 20</u> : <u>4 : 3 : 2 : 2</u> : 0 : 0 : 0 : 2
15 : CALL(3,1,0);			
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	0 : 0 : 0 : 2
2 : JMP(0);	17	ε	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
3 : LOAD(2,1);	18	1	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
4 : LIT(1);	19	ε	4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
5 : GT;	3	ε	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
6 : JFALSE(16);	4	2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
7 : LOAD(1,1);	5	2 : 1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
8 : LOAD(2,1);	6	1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
9 : MULT;	7	ε	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
10 : STORE(1,1);	8	1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
11 : LOAD(2,1);	9	1 : 2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
12 : LIT(1);	10	2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
13 : SUB;	11	ε	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
14 : STORE(2,1);	12	2	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
15 : CALL(3,1,0);	13	2 : 1	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
16 : RET;			
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);	8	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
12 : LIT(1);	10	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
13 : SUB;	11	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
14 : STORE(2,1);	12	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
15 : CALL(3,1,0);	13	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
16 : RET;	14	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
17 : LIT(1);			
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	<u>0 : 0 : 0 : 2</u>
2 : JMP(0);	17	ε	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
3 : LOAD(2,1);	18	1	<u>4 : 3 : 2 : 0 : 0 : 0 : 0 : 2</u>
4 : LIT(1);	19	ε	<u>4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
5 : GT;	3	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
6 : JFALSE(16);	4	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
7 : LOAD(1,1);	5	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
8 : LOAD(2,1);	6	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
9 : MULT;	7	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
10 : STORE(1,1);	8	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
11 : LOAD(2,1);	9	1 : 2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
12 : LIT(1);	10	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2</u>
13 : SUB;	11	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
14 : STORE(2,1);	12	2	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
15 : CALL(3,1,0);	13	2 : 1	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
16 : RET;	14	1	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2</u>
17 : LIT(1);	15	ε	<u>3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1</u>
18 : STORE(0,1);			
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	1 : 2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);			
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	1 : 2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);			
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT ;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	1 : 2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	1 : 1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);			
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;			

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	0 : 0 : 0 : 2
2 : JMP(0);	17	ε	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
3 : LOAD(2,1);	18	1	4 : 3 : 2 : 0 : 0 : 0 : 0 : 2
4 : LIT(1);	19	ε	4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
5 : GT;	3	ε	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
6 : JFALSE(16);	4	2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
7 : LOAD(1,1);	5	2 : 1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
8 : LOAD(2,1);	6	1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
9 : MULT;	7	ε	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
10 : STORE(1,1);	8	1	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
11 : LOAD(2,1);	9	1 : 2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
12 : LIT(1);	10	2	3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2
13 : SUB;	11	ε	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
14 : STORE(2,1);	12	2	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
15 : CALL(3,1,0);	13	2 : 1	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
16 : RET;	14	1	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2
17 : LIT(1);	15	ε	3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1
18 : STORE(0,1);	3	ε	6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1
19 : CALL(3,0,0);	4	1	6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1
20 : LOAD(0,1);	5	1 : 1	6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1
21 : STORE(1,1);	6	0	6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1
22 : RET;	16	ε	6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	16	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	20	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	1 : 2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	2 : 1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	1 : 1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	16	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	20	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	21	2	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	16	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	20	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	21	2	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	22	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0) ;	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	16	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	20	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	21	2	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	22	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
	2	ε	$0 : 0 : 0 : 2$

Example: Factorial Function II

Example 19.3 (Factorial function; continued)

Computation for $x = 2$:	<i>PC</i>	<i>DS</i>	<i>PS</i>
1 : CALL(17,0,1);	1	ε	$0 : 0 : 0 : 2$
2 : JMP(0);	17	ε	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
3 : LOAD(2,1);	18	1	$4 : 3 : 2 : 0 : 0 : 0 : 0 : 2$
4 : LIT(1);	19	ε	$4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
5 : GT;	3	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
6 : JFALSE(16);	4	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
7 : LOAD(1,1);	5	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
8 : LOAD(2,1);	6	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
9 : MULT;	7	ε	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
10 : STORE(1,1);	8	1	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
11 : LOAD(2,1);	9	$1 : 2$	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
12 : LIT(1);	10	2	$3 : 2 : 20 : 4 : 3 : 2 : 1 : 0 : 0 : 0 : 2$
13 : SUB;	11	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
14 : STORE(2,1);	12	2	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
15 : CALL(3,1,0);	13	$2 : 1$	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
16 : RET;	14	1	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
17 : LIT(1);	15	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
18 : STORE(0,1);	3	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
19 : CALL(3,0,0);	4	1	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
20 : LOAD(0,1);	5	$1 : 1$	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
21 : STORE(1,1);	6	0	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
22 : RET;	16	ε	$6 : 2 : 16 : 3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	16	ε	$3 : 2 : 20 : 4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	20	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	21	2	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 1$
	22	ε	$4 : 3 : 2 : 2 : 0 : 0 : 0 : 2$
	2	ε	$0 : 0 : 0 : 2$
	0	ε	$0 : 0 : 0 : 2$

- 1 Repetition: Translation of EPL into AM Programs
- 2 A Translation Example
- 3 Correctness of the Translation

Theorem 19.4 (Correctness of translation)

For every $P \in Pgm$, $n \in \mathbb{N}$, and $(z_1, \dots, z_n), (z'_1, \dots, z'_n) \in \mathbb{Z}^n$:

$$\begin{aligned} & \mathfrak{M}[[P]](z_1, \dots, z_n) = (z'_1, \dots, z'_n) \\ \iff & [[\text{trans}(P)]](1, \varepsilon, 0 : 0 : 0 : z_1 : \dots : z_n) = (0, \varepsilon, 0 : 0 : 0 : z'_1 : \dots : z'_n) \end{aligned}$$

Theorem 19.4 (Correctness of translation)

For every $P \in Pgm$, $n \in \mathbb{N}$, and $(z_1, \dots, z_n), (z'_1, \dots, z'_n) \in \mathbb{Z}^n$:

$$\begin{aligned} & \mathfrak{M}[[P]](z_1, \dots, z_n) = (z'_1, \dots, z'_n) \\ \iff & [[\text{trans}(P)]](1, \varepsilon, 0 : 0 : 0 : z_1 : \dots : z_n) = (0, \varepsilon, 0 : 0 : 0 : z'_1 : \dots : z'_n) \end{aligned}$$

Proof.

see M. Mohnen: *A Compiler Corectness Proof for the Static Link Technique by means of Evolving Algebras*, Fundamenta Informaticae 29(3), 1997, pp. 257–303 □

Correct Translation of Arithmetic Expressions

The simplest subcase of the correctness proof is the following.

Lemma 19.5 (Correctness of at)

Let $A \in AExp$, $\rho \in Env$, $\sigma \in Stt$, $z \in \mathbb{Z}$, $st \in Tab$, $a \in PC$, $l \in Lev$, $d \in DS$, and $p \in PS$ where, for every $I \in Ide$ in A ,

- $\rho(I) = z \in \mathbb{Z} \implies st(I) = (\mathbf{const}, z)$ and
- $\rho(I) = \alpha \in Loc, \sigma(\alpha) = z \implies st(I) = (\mathbf{var}, lev, off)$ for some $lev \in Lev$ and $off \in Off$ such that $p.(base(p, l - lev) + off + 2) = z$.

Then, with $P := at(A, st, a, l)$,

$$\mathcal{A}[[A]] \rho \sigma = z \implies [[P]](a, d, p) = (a + |P|, d : z, p).$$

Correct Translation of Arithmetic Expressions

The simplest subcase of the correctness proof is the following.

Lemma 19.5 (Correctness of at)

Let $A \in AExp$, $\rho \in Env$, $\sigma \in Stt$, $z \in \mathbb{Z}$, $st \in Tab$, $a \in PC$, $l \in Lev$, $d \in DS$, and $p \in PS$ where, for every $I \in Ide$ in A ,

- $\rho(I) = z \in \mathbb{Z} \implies st(I) = (\mathbf{const}, z)$ and
- $\rho(I) = \alpha \in Loc, \sigma(\alpha) = z \implies st(I) = (\mathbf{var}, lev, off)$ for some $lev \in Lev$ and $off \in Off$ such that $p.(base(p, l - lev) + off + 2) = z$.

Then, with $P := at(A, st, a, l)$,

$$\mathfrak{A}[[A]] \rho \sigma = z \implies [[P]](a, d, p) = (a + |P|, d : z, p).$$

Proof.

on the board

