Figure 1: FlowChart program example: dictionary search

```
1    read (name, namelist, valuelist):
2    search: if name = car namelist goto found else cont;
3    cont  : valuelist ← cdr valuelist;
4            namelist ← cdr namelist ;
5            goto search;
6    found : return (car valuelist);
```

Figure 2: A mix-generated target program without transition compression

```
1    (search, (z, (x y z))): goto (cont, (z, (x y z)));
2    (cont  , (z, (x y z))): valuelist ← cdr valuelist;
3                            goto (search, (z, (y z)));
4    (search, (z, (y z)))  : goto (cont, (z, (y z)));
5    (cont  , (z, (y z)))  : valuelist ← cdr valuelist;
6                            goto (search, (z, (z)));
7    (search, (z, (z)))    : goto (found, (z, (z)));
8    (found , (z, (z)))    : return (car valuelist);
```

Figure 3: A mix-generated target program after transition compression

```
1    (search, (z, (x y z))): valuelist ← cdr valuelist;
2                            valuelist ← cdr valuelist;
3                            return (car valuelist);
```

Figure 4: Turing Machine interpreter written in FlowChart

```
1   read       (Q, Right);
2   init:      Qtail ← Q; Left ← '();
3   loop:      if Qtail = '() goto stop else cont;
4   cont:      Instruction ← first_instruction Qtail;
5              Qtail       ← rest Qtail;
6              Operator    ← cadr Instruction;
7
8              if Operator = 'right goto do-right else cont1;
9   cont1:     if Operator = 'left  goto do-left  else cont2;
10  cont2:     if Operator = 'write goto do-write else cont3;
11  cont3:     if Operator = 'goto  goto do-goto  else cont4;
12  cont4:     if Operator = 'if    goto do-if    else error;
13
14  do-right:  Left       ← cons (first_symbol Right, Left);
15             Right      ← cdr  Right;
16             goto loop;
17  do-left:   Right      ← cons (first_symbol Left, Left);
18             Left       ← cdr  Left;
19             goto loop;
20  do-write:  Symbol     ← caddr Instruction;
21             Right      ← cons  (Symbol, car Right);
22             goto loop;
23  do-goto:   NextLabel ← caddr     Instruction;
24             Qtail     ← new_tail (NextLabel, Q);
25             goto loop;
26  do-if:     Symbol     ← caddr    Instruction;
27             NextLabel ← caddddr Instruction;
28             if Symbol = first_symbol Right goto jump else loop;
29
30  jump:      Qtail ← new_tail (NextLabel, Q);
31             goto loop;
32
33  error:     return ('syntaxerror: Instruction);
34
35  stop:      return Right;
```

Figure 5: Annotated Turing Machine interpreter written in FlowChart

```
1   read       (Q, Right);
2   init:      Qtail ← Q; Left ← '();
3   loop:      if Qtail = '() goto stop else cont;
4   cont:      Instruction ← first_instruction Qtail;
5              Qtail       ← rest Qtail;
6              Operator    ← cadr Instruction;
7
8              if Operator = 'right goto do-right else cont1;
9   cont1:     if Operator = 'left  goto do-left  else cont2;
10  cont2:     if Operator = 'write goto do-write else cont3;
11  cont3:     if Operator = 'goto  goto do-goto  else cont4;
12  cont4:     if Operator = 'if    goto do-if    else error;
13
14  do-right:  Left        ← cons (first_symbol Right, Left);
15             Right       ← cdr  Right;
16             goto loop;
17  do-left:   Right       ← cons (first_symbol Left, Left);
18             Left        ← cdr  Left;
19             goto loop;
20  do-write:  Symbol      ← caddr Instruction;
21             Right       ← cons  (Symbol, cdr Right);
22             goto loop;
23  do-goto:   NextLabel   ← caddr     Instruction;
24             Qtail       ← new_tail (NextLabel, Q);
25             goto loop;
26  do-if:     Symbol      ← caddr    Instruction;
27             NextLabel   ← caddddr  Instruction;
28             if Symbol = first_symbol Right goto jump else loop;
29
30  jump:      Qtail       ← new_tail (NextLabel, Q);
31             goto loop;
32
33  error:     return ('syntaxerror: Instruction);
34
35  stop:      return Right;
```

Figure 6: Turing Machine interpreter written in FlowChart

```
1   read (Q, Right);
2   init:     Qtail ← Q; [Left ← '()];
3   loop:     if Qtail = '() goto stop else cont;
4   cont:     Instruction ← first_instruction Qtail;
5             Qtail       ← rest Qtail;
6             Operator    ← cadr Instruction;
7
8             if Operator = 'right goto do-right else cont1;
9   cont1:    if Operator = 'left  goto do-left  else cont2;
10  cont2:    if Operator = 'write goto do-write else cont3;
11  cont3:    if Operator = 'goto  goto do-goto  else cont4;
12  cont4:    if Operator = 'if    goto do-if    else error;
13
14  do-right: [Left ← cons (first_symbol Right, Left)];
15            [Right ← cdr Right];
16            goto loop;
17  do-left:  [Right ← cons (first_symbol Left, Right)];
18            [Left ← cdr Left];
19            goto loop;
20  do-write: Symbol    ← caddr Instruction;
21            [Right ← cons (Symbol, cdr (Right)];
22            goto loop;
23  do-goto:  NextLabel ← caddr     Instruction;
24            Qtail     ← new_tail (NextLabel, Q);
25            goto loop;
26  do-if:    Symbol    ← caddr    Instruction;
27            NextLabel ← caddddr  Instruction;
28            [if (Symbol = first_symbol Right goto jump else loop];
29
30  jump:     Qtail     ← new_tail (NextLabel, Q);
31            goto loop;
32
33  error:    return ('syntaxerror: Instruction);
34
35  stop:     [return Right];
```

Figure 7: TM program example

```
0: if 0 goto 3
1: right
2: goto 0
3: write 1
```

Target	Int	Static variables *vs*			
label	label	Instruction	Qtail	Symbol	NextLabel
lab0	init	()	()	()	()
lab1	cont	right	(2:goto 0, 3:write 1)	0	3
lab2	jump	if 0 goto 3	(1:right, 2:goto 0, 3:write 1)	0	3

Figure 8: A mix-generated target program

```
read: (Right);
lab0: Left ← '();
      if '0 = first_symbol Right goto lab2 else lab1;
lab1: Left ← cons (first_symbol Right, Left);
      Rigth ← cdr Right;
      if '0 = first_symbol Right goto lab2 else lab1;
lab2: Right ← cons ('1, cdr Right);
      return (Right);
```