Übersetzung in Stackcode

trans (F(x) = if x>0 then x-1 else F(F(x+2)))

```
= 1:    exptrans( if x>0 then x-1 else F(F(x+2)), 1)
        RET
```

```
= 1:    exptrans ( x>0 , 1.1)

        JMC 1.3;
        exptrans ( x-1 , 1.2)

        JMP 1.4;
  1.3:  exptrans (F(F(x+2)) , 1.3)

  1.4:  RET
```

```
= 1:        exptrans ( x , 1.1.1)
            exptrans ( 0 , 1.1.2)
            EXEC >;
            JMC 1.3;
            exptrans ( x , 1.2.1)
            exptrans ( 1 , 1.2.2)
            EXEC -;
            JMP 1.4;
  1.3:      exptrans ( F(x+2) , 1.3.1)

            CREATE(1, 1.3.0);
            JMP 1;
1.3.0,
1.4:        RET
```

trans (F(x) = if x>0 then x-1 else F(F(x+2)))

```
= 1:      exptrans ( x , 1.1.1)
          exptrans ( 0 , 1.1.2)
          EXEC >;
          JMC 1.3;
          exptrans ( x , 1.2.1)
          exptrans ( 1 , 1.2.2)
          EXEC -;
          JMP 1.4;
  1.3:    exptrans ( F(x+2) , 1.3.1)









          CREATE(1, 1.3.0);
          JMP 1;
1.3.0,
1.4:      RET
```

```
= 1:      LOAD 1;
          EXEC 0;
          EXEC >;
          JMC 1.3;
          LOAD 1;
          EXEC 1;
          EXEC -;
          JMP 1.4;
  1.3:    exptrans ( x+2 , 1.3.1.1)



          CREATE(1,1.3.1.0);
          JMP 1;
1.3.1.0:  CREATE(1, 1.3.0);
          JMP 1;
1.3.0,
1.4:      RET
```

trans (F(x) = if x>0 then x-1 else F(F(x+2)))

```
= 1:      LOAD 1;
          EXEC 0;
          EXEC >;
          JMC 1.3;
          LOAD 1;
          EXEC 1;
          EXEC -;
          JMP 1.4;
  1.3:    LOAD 1;
          EXEC 2;
          EXEC +;
          CREATE(1,1.3.1.0);
          JMP 1;
1.3.1.0:  CREATE(1, 1.3.0);
          JMP 1;
1.3.0,
1.4:      RET
```

LADEN

Standardadressform

```
= 1:      LOAD 1;
  2:      EXEC 0;
  3:      EXEC >;
  4:      JMC 9;
  5:      LOAD 1;
  6:      EXEC 1;
  7:      EXEC -;
  8:      JMP 16;
  9:      LOAD 1;
  10:     EXEC 2;
  11:     EXEC +;
  12:     CREATE(1,14);
  13:     JMP 1;
  14:     CREATE(1, 16);
  15:     JMP 1;
  16:     RET
```

Zustandsfolge der Stackmaschine

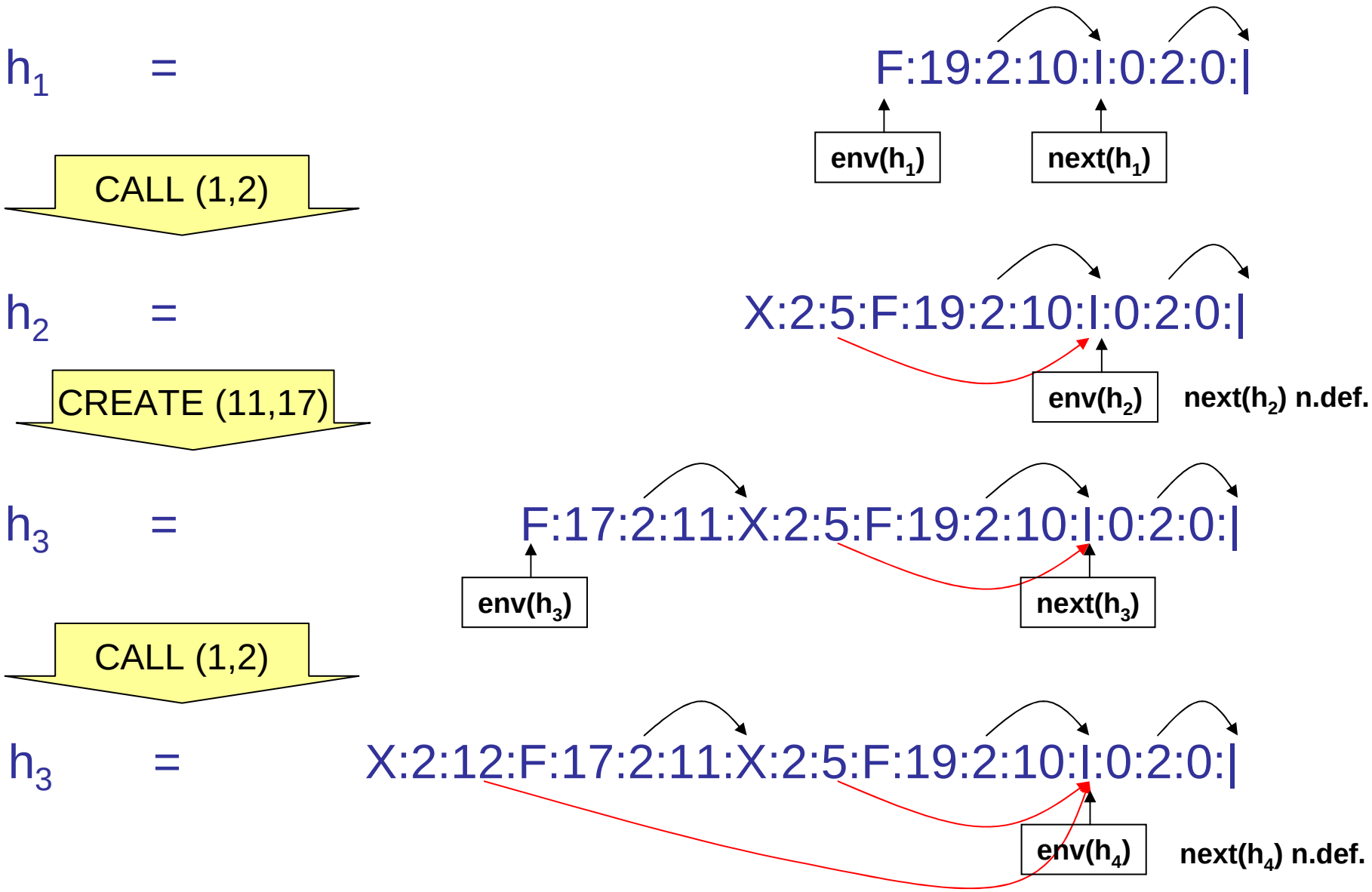
Programmspeicher

```
1:   LOAD 1;
2:   EXEC 0;
3:   EXEC >;
4:   JMC 9;
5:   LOAD 1;
6:   EXEC 1;
7:   EXEC -;
8:   JMP 16;
9:   LOAD 1;
10:  EXEC 2;
11:  EXEC +;
12:  CREATE(1,14);
13:  JMP 1;
14:  CREATE(1, 16);
15:  JMP 1;
16:  RET
```

BZ	DK	FK .
1	|	0:2:0|
2	|0	0:2:0|
3	|0:0	0:2:0|
4	|ff	0:2:0|
9	|	0:2:0|
10	|0	0:2:0|
11	|0:2	0:2:0|
12	|2	0:2:0|
13	|	14:2:2:0:2:0|
1	|	14:2:2:0:2:0|
2	|2	14:2:2:0:2:0|
3	|2:0	14:2:2:0:2:0|
4	|tt	14:2:2:0:2:0|
5	|	14:2:2:0:2:0|
6	|2	14:2:2:0:2:0|
7	|2:1	14:2:2:0:2:0|
8	|1	14:2:2:0:2:0|
16	|1	14:2:2:0:2:0|
14	|1	0:2:0|
15	|	16:2:1:0:2:0|

BZ	DK	FK .
1	|	16:2:1:0:2:0|
2	|1	16:2:1:0:2:0|
3	|1:0	16:2:1:0:2:0|
4	|tt	16:2:1:0:2:0|
5	|	16:2:1:0:2:0|
6	|1	16:2:1:0:2:0|
7	|1:1	16:2:1:0:2:0|
8	|0	16:2:1:0:2:0|
16	|0	16:2:1:0:2:0|
16	|0	0:2:0|
0	|0	|

Funktionskellerzustände (nicht-strikte Semantik)



Übersetzung in Stackcode (nicht-strikte Semantik)

trans (F(x) = if x>0 then x-1 else F(F(x+2)))

```
= 1:      exptrans( if x>0 then x-1 else F(F(x+2)), 1)
          RET
```

```
= 1:          exptrans ( x , 1.1.1)
              exptrans ( 0 , 1.1.2)
              EXEC >;
              JMC 1.3;
              exptrans ( x , 1.2.1)
              exptrans ( 1 , 1.2.2)
              EXEC -;
              JMP 1.4;
  1.3:        JMP 1.3.0;
  1.3.1:      exptrans ( F(x+2) , 1.3.1)
              RET;
  1.3.0:      CREATE(1, 1.3.0);
              JMP 1;
  1.3.2, 1.4:RET
```

```
= 1:          CALL ( 1 , 1.1.1.0);
  1.1.1.0:    EXEC 0;
              EXEC >;
              JMC 1.3;
              CALL ( 1 , 1.2.1.0)
  1.2.1.0:    EXEC 1;
              EXEC -;
              JMP 1.4;
  1.3:        JMP 1.3.0;
  1.3.1:      JMP 1.3.1.0;
  1.3.1.1:    CALL (1, 1.3.1.1.1.0)
  1.3.1.1.1.0: EXEC 2;
              EXEC +;
              RET;
  1.3.1.0:    CREATE(1.3.1.1,1.3.1.2);
              JMP 1;
  1.3.1.2:    RET;
  1.3.0:      CREATE(1.3.1, 1.3.0);
              JMP 1;
  1.3.2, 1.4: RET
```

trans (F(x) = if x>0 then x-1 else F(F(x+2)))

Standardadressform

```
= 1:            CALL ( 1 , 1.1.1.0);
  1.1.1.0:      EXEC 0;
                EXEC >;
                JMC 1.3;
                CALL ( 1 , 1.2.1.0)
  1.2.1.0:      EXEC 1;
                EXEC -;
                JMP 1.4;
  1.3:          JMP 1.3.0;
  1.3.1:        JMP 1.3.1.0;
  1.3.1.1:      CALL (1, 1.3.1.1.1.0)
  1.3.1.1.1.0: EXEC 2;
                EXEC +;
                RET;
  1.3.1.0:      CREATE(1.3.1.1,1.3.1.2);
                JMP 1;
  1.3.1.2:      RET;
  1.3.0:        CREATE(1.3.1, 1.3.0);
                JMP 1;
  1.3.2, 1.4:   RET
```

LADEN

```
= 1:    CALL ( 1 , 2);
  2:    EXEC 0;
  3:    EXEC >;
  4:    JMC 9;
  5:    CALL ( 1 , 6)
  6:    EXEC 1;
  7:    EXEC -;
  8:    JMP 20;
  9:    JMP 18;
  10:   JMP 15;
  11:   CALL (1, 12)
  12:   EXEC 2;
  13:   EXEC +;
  14:   RET;
  15:   CREATE(11,17);
  16:   JMP 1;
  17:   RET;
  18:   CREATE(10, 20);
  19:   JMP 1;
  20:   RET
```

Zustandsfolge der modifizierten Stackmaschine

Programmspeicher

```
= 1:    CALL ( 1 , 2);
  2:    EXEC 0;
  3:    EXEC >;
  4:    JMC 9;
  5:    CALL ( 1 , 6)
  6:    EXEC 1;
  7:    EXEC -;
  8:    JMP 20;
  9:    JMP 18;
  10:   JMP 15;
  11:   CALL (1, 12)
  12:   EXEC 2;
  13:   EXEC +;
  14:   RET;
  15:   CREATE(11,17);
  16:   JMP 1;
  17:   RET;
  18:   CREATE(10, 20);
  19:   JMP 1;
  20:   RET
```

BZ	DK	FK
1	|	I:0:2:0|
2	|0	I:0:2:0|
3	|0:0	I:0:2:0|
4	|ff	I:0:2:0|
9	|	I:0:2:0|
18	|	I:0:2:0|
19	|	F:20:2:10:I:0:2:0|
1	|	F:20:2:10:I:0:2:0|
10	|	X:2:5:F:20:2:10:I:0:2:0|
15	|	X:2:5:F:20:2:10:I:0:2:0|
16	|	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
1	|	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
11	|	X:2:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
12	|0	X:2:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
13	|0:2	X:2:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
14	|2	X:2:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
2	|2	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
3	|2:0	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
4	|tt	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
5	|	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
11	|	X:6:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|

Zustandsfolge der modifizierten Stackmaschine

Programmspeicher

```
= 1:    CALL ( 1 , 2);
  2:    EXEC 0;
  3:    EXEC >;
  4:    JMC 9;
  5:    CALL ( 1 , 6)
  6:    EXEC 1;
  7:    EXEC -;
  8:    JMP 20;
  9:    JMP 18;
  10:   JMP 15;
  11:   CALL (1, 12)
  12:   EXEC 2;
  13:   EXEC +;
  14:   RET;
  15:   CREATE(11,17);
  16:   JMP 1;
  17:   RET;
  18:   CREATE(10, 20);
  19:   JMP 1;
  20:   RET
```

BZ	DK	FK
11	|	X:6:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
12	|0	X:6:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
13	|0:2	X:6:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
14	|2	X:6:12:F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
6	|2	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
7	|2:1	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
8	|1	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
20	|1	F:17:2:11:X:2:5:F:20:2:10:I:0:2:0|
17	|1	X:2:5:F:20:2:10:I:0:2:0|
2	|1	F:20:2:10:I:0:2:0|
3	|1:0	F:20:2:10:I:0:2:0|
4	|tt	F:20:2:10:I:0:2:0|
5	|	F:20:2:10:I:0:2:0|
10	|	X:6:5:F:20:2:10:I:0:2:0|
15	|	X:6:5:F:20:2:10:I:0:2:0|
16	|	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
1	|	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
11	|	X:2:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
12	|0	X:2:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
13	|0:2	X:2:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
14	|2	X:2:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|

Zustandsfolge der modifizierten Stackmaschine

Programmspeicher

```
= 1:    CALL ( 1 , 2);
  2:    EXEC 0;
  3:    EXEC >;
  4:    JMC 9;
  5:    CALL ( 1 , 6)
  6:    EXEC 1;
  7:    EXEC -;
  8:    JMP 20;
  9:    JMP 18;
  10:   JMP 15;
  11:   CALL (1, 12)
  12:   EXEC 2;
  13:   EXEC +;
  14:   RET;
  15:   CREATE(11,17);
  16:   JMP 1;
  17:   RET;
  18:   CREATE(10, 20);
  19:   JMP 1;
  20:   RET
```

BZ	DK	FK
14	|2	X:2:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
2	|2	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
3	|2:0	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
4	|tt	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
5	|	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
11	|	X:6:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
12	|0	X:6:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
13	|0:2	X:6:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
14	|2	X:6:12:F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
6	|2	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
7	|2:1	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
8	|1	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
20	|1	F:17:2:11:X:6:5:F:20:2:10:I:0:2:0|
17	|1	X:6:5:F:20:2:10:I:0:2:0|
6	|1	F:20:2:10:I:0:2:0|
7	|1:1	F:20:2:10:I:0:2:0|
8	|0	F:20:2:10:I:0:2:0|
20	|0	F:20:2:10:I:0:2:0|
20	|0	I:0:2:0|
0	|0	|