Introduktion til Programmering og Problemløsning (PoP)

Jon Sporring
Department of Computer Science
2020/09/07

UNIVERSITY OF COPENHAGEN

Håndkøring (tracing)

simple.fsx

```
1 let a = 3
2 printfn "%d" a
```

Step	Line	Env.	Bindings and evaluations
0	-	E_0	()
1	1	E_0	a=3
2	2	E_0	output="3"

Håndkøring (tracing)

function.fsx

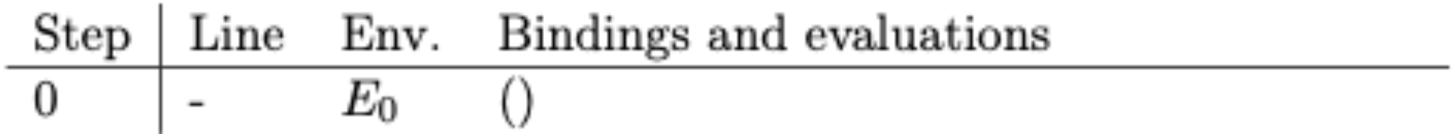
```
1 let double n =
2   2*n
3
4 let result = double 5
5 printfn "%d" result
```

Step	Line	Env.	Bindings and evaluations
0	-	E_0	()
1	1	E_0	double=((n), 2*n, ())
2	4	E_0	result=?
3	1	E_1	((n=5), 2*n, ())
4	2	E_1	return=10
5	4	E_0	result=10
6	5	E_0	output="10"

Håndkøring (tracing)

loop.fsx

```
1 let printDoubles n =
2   for i = 1 to n do
3     printf "%d " (2*i)
4   printfn ""
5
6 printDoubles 3
```

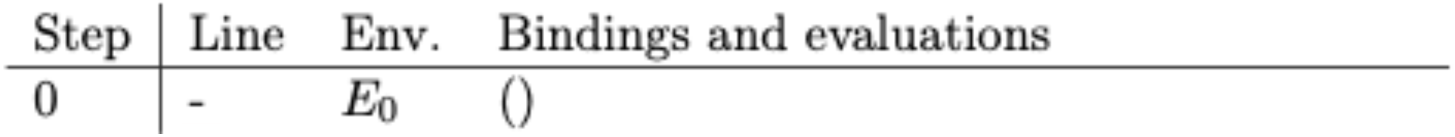


Step	Line	Env.	Bindings and evaluations
0	-	E_0	()

Håndkøring (tracing)

loopUnfolded.fsx

```
01 let printDoubles n =
02   // Unfolding assuming n = 3
03   //  for i = 1 to n do
04   //    printf "%d " (2*i)
05   (
06     let i = 1
07     printf "%d " (2*i)
08   )
09   (
10     let i = 2
11     printf "%d " (2*i)
12   )
13   (
14     let i = 3
15     printf "%d " (2*i)
16   )
17   printfn ""
18
19 printDoubles 3
```



Step	Line	Env.	Bindings and evaluations
0	-	E_0	()

Håndkøring (tracing)

loop.fsx

```
1 let printDoubles n =
2   for i = 1 to n do
3     printf "%d " (2*i)
4   printfn ""
5
6 printDoubles 3
```

Step	Line	Env.	Bindings and evaluations
0	-	E_0	()
1	1	E_0	printDoubles = ((n), body, ())
2	6	E_0	return = ?
3	1	E_1	((n=3), body, ())
4	2	E_2	((i=1), for-body, (n=3))
5	3	E_2	output = "2␣"
6	2	E_3	((i=2), for-body, (n=3))
7	3	E_3	output = "4␣"
8	2	E_4	((i=3), for-body, (n=3))
9	3	E_4	output = "6␣"
10	4	E_1	output = "↵"
11	4	E_1	return = ()
12	6	E_0	return = ()

Håndkøring (tracing)

loopMutable.fsx

```
1 let stringDoubles n =
2   let mutable str = ""
3   for i = 1 to n do
4     str <- str + " " + string (2*i)
5   str.[1..]
6
7 printfn "%s" (stringDoubles 3)
```

Step	Line	Env.	Bindings and evaluations
0	-	E_0	()
1	1	E_0	stringDoubles=((n), body, ())
2	7	E_0	output=?
3	7	E_1	((n=3), body, ())
4	2	E_1	str = α
5	3	E_2	((i=1), for-body, (n=3, str=α))
6	4	E_2	str ← "" + "␣" + "2"
7	3	E_3	((i=2), for-body, (n=3, str=α))
8	4	E_3	str ← "␣2" + "␣" + "4"
9	3	E_4	((i=3), for-body, (n=3, str=α))
10	4	E_4	str ← "␣2␣4" + "␣" + "6"
11	5	E_1	return="2␣4␣6"
12	7	E_0	output="2␣4␣6"

Step	Value
0	-
4	α = ""
6	α = "␣2"
8	α = "␣2␣4"
10	α = "␣2␣4␣6"

Resumé

I denne video hørte du om:

- Håndkøring (tracing)
  - Simpelt program
  - Funktion
  - Løkke
  - Variable