

# UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i                    INF3110/4110 — Programmeringsspråk

Eksamensdag:            2. desember 2003

Tid for eksamen:        14.30–17.30

Oppgavesettet er på 7 sider.

Vedlegg:                 Ingen

Tillatte hjelpemidler: Alle trykte og skrevne

Kontroller at oppgavesettet er komplett før  
du begynner å besvare spørsmålene.

## Løsningsforslag

### Oppgave 1 ML (14 poeng)

#### 1a Typer (8 poeng)

- `f1 : 'a -> 'a list`
- `f2 : int -> int list`
- `f3 : 'a * 'b * 'c -> 'a * 'c`
- `f4 : int * 'a * bool -> int`
- `f5 : int -> int`
- `f6 : ('a -> 'b) * 'a -> 'b`
- `f7 : ('a -> 'b) -> 'a -> 'b`
- `f8 : "a -> "a -> "a`

#### 1b Programmering (6 poeng)

(3 p) (a) En løsning *uten* map:

```
fun dbl [] = []  
  | dbl (x::xs) = (x * 2) :: dbl xs;
```

(Fortsettes på side 2.)

```
fun double [] = []
  | double (x::xs) = dbl x :: double xs;
```

En løsning med map:

```
fun double ll = map (fn l => map (fn x => x * 2) l) ll;
```

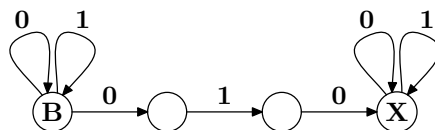
(3 p) (b)

```
fun init [] l = true
  | init s [] = false
  | init (x::xs) (y::ys) = x = y andalso init xs ys;
```

### Oppgave 2 Syntaks (10 poeng)

(1,5 p) (a 1)  $[0|1]^* 010 [0|1]^*$

(1,5 p) (a 2)

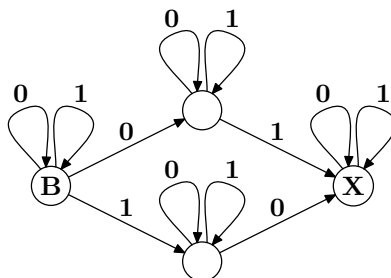


(1,5 p) (b 1)

```
[ [0|1]* 0 [0|1]* 1 [0|1]*
  | [0|1]* 1 [0|1]* 0 [0|1]* ]
```

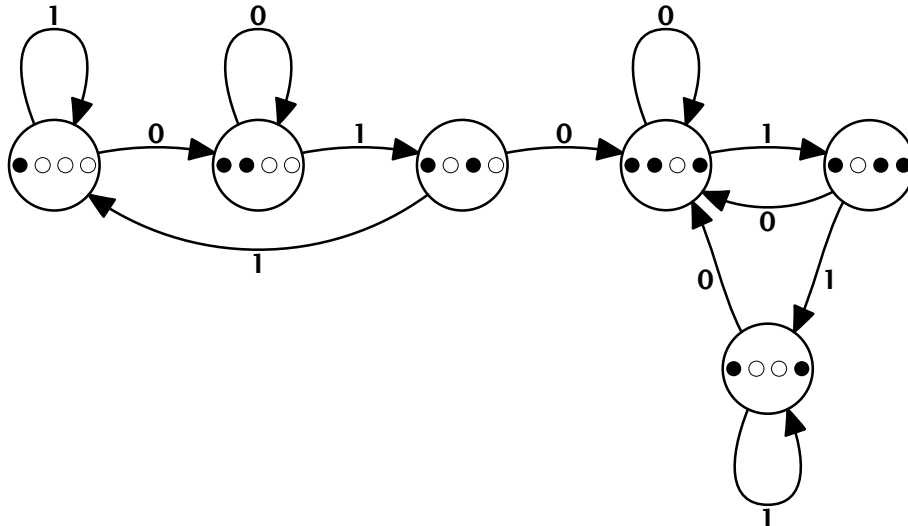
Alternativt:  $[0^+1|1^+0] [0|1]^*$

(1,5 p) (b 2)



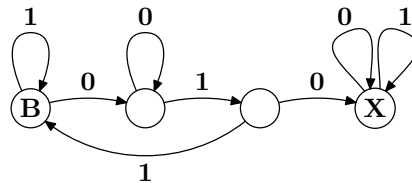
(4 p) (c)

(Fortsettes på side 3.)



De tre tilstandene lengst til høyre er slutttilstander.

Hvis vi slår sammen slutttilstandene og ikke viser alle delmengdene, så får vi:



### Oppgave 3 Grammatikker (9 poeng)

(5 p)

- (1) (a) (b)
- (2) (a)
- (3) (a) (b) (c) (d)
- (4) (d)
- (5) (a)

#### 3a (1 poeng)

Språket beskriver strenger på formen  $\mathbf{a}^n \mathbf{b}^n$ , dvs.  $\underbrace{\mathbf{a} \dots \mathbf{a}}_n \underbrace{\mathbf{b} \dots \mathbf{b}}_n$ , hvor  $n \geq 1$ .

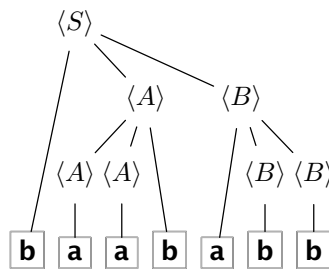
(Fortsettes på side 4.)

**3b** (1 poeng)

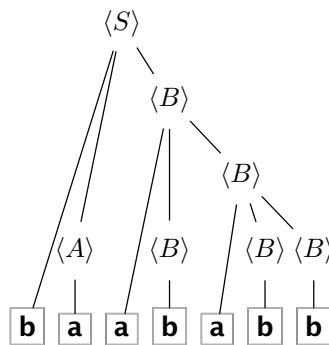
Språket beskriver strenger som er like uansett om de leses forfra eller bakfra.

**3c** (2 poeng)

(1 p) **(a)** Syntakstre for strengen **baababb**



(1 p) **(b)** Viser at grammatikken er flertydig ved å tegne enda et syntakstre for strengen:

**Oppgave 4**  $\lambda$ -kalkyle (7 poeng)

Her vises konversjoner med tegnet  $\rightsquigarrow$  og substitusjoner med tegnet  $=$ .

(1 p) **1.**

$$(\lambda x.x)z \rightsquigarrow x[x/z] \\ = z$$

(2 p) **2.**

$$((\lambda y.x)(\lambda x.x)x)[x/y] \rightsquigarrow ((\lambda z.x)(\lambda x.x)x)[x/y] \\ = (\lambda z.y)(\lambda x.x)y \rightsquigarrow y[z/(\lambda x.x)]y \\ = yy$$

(Fortsettes på side 5.)

I første linje her endres  $y$  til  $z$ ; dette er  $\alpha$ -konversjon.

(2 p) 3.

$$\begin{aligned}
 & (\lambda x.xx)((\lambda b.b)(\lambda c.c)) && \rightsquigarrow && xx[x/((\lambda b.b)(\lambda c.c))] \\
 = & ((\lambda b.b)(\lambda c.c))((\lambda b.b)(\lambda c.c)) && \rightsquigarrow && (b[b/(\lambda c.c)])(\lambda b.b)(\lambda c.c) \\
 = & (\lambda c.c)((\lambda b.b)(\lambda c.c)) && \rightsquigarrow && (\lambda c.c)(b[b/(\lambda c.c)]) \\
 = & (\lambda c.c)(\lambda c.c) && \rightsquigarrow && c[c/(\lambda c.c)] \\
 = & \lambda c.c
 \end{aligned}$$

Alternativt (svarer til *call by value* i stedet for *call by name*):

$$\begin{aligned}
 & (\lambda x.xx)((\lambda b.b)(\lambda c.c)) && \rightsquigarrow && (\lambda x.xx)(b[b/(\lambda c.c)]) \\
 = & (\lambda x.xx)(\lambda c.c) && \rightsquigarrow && xx[x/(\lambda c.c)] \\
 = & (\lambda c.c)(\lambda c.c) && \rightsquigarrow && c[c/(\lambda c.c)] \\
 = & \lambda c.c
 \end{aligned}$$

(2 p) 4.

$$\begin{aligned}
 & (\lambda xy.xyx)(\lambda b.b)(\lambda c.c) && \rightsquigarrow && (\lambda y.xyx)[x/(\lambda b.b)](\lambda c.c) \\
 = & (\lambda y.(\lambda b.b)y(\lambda b.b))(\lambda c.c) && \rightsquigarrow && ((\lambda b.b)y(\lambda b.b))[y/(\lambda c.c)] \\
 = & (\lambda b.b)(\lambda c.c)(\lambda b.b) && \rightsquigarrow && b[b/(\lambda c.c)](\lambda b.b) \\
 = & (\lambda c.c)(\lambda b.b) && \rightsquigarrow && c[c/(\lambda b.b)] \\
 = & \lambda b.b && \rightsquigarrow &&
 \end{aligned}$$

(Fortsettes på side 6.)

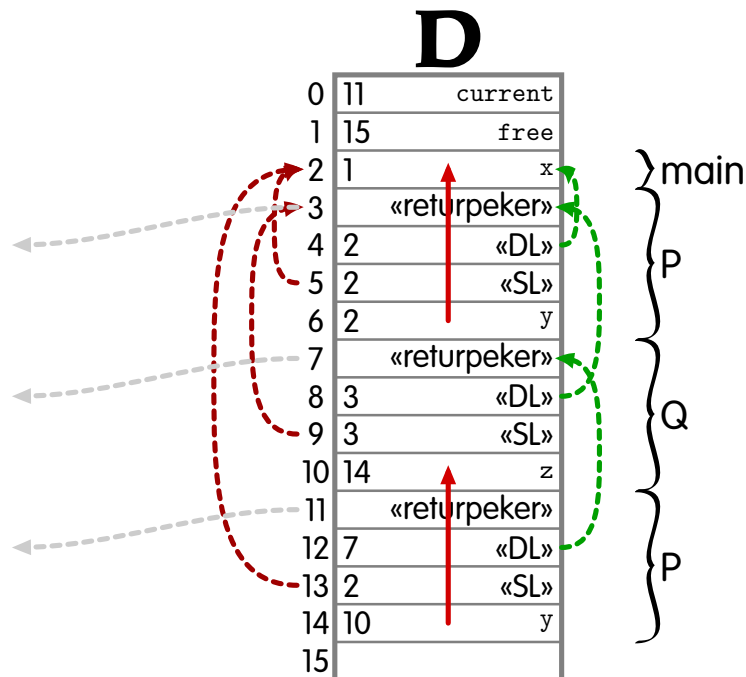
## Oppgave 5 Kjøresystemer/parameteroverføring (12 poeng)

Nedenfor er det vist et lite program i språket C4" fra læreboken.

Anta at prosedyren P overfører sine parametre ved *referanseoverføring* og at prosedyren Q overfører sine parametre ved *verdioverføring*.

### 5a (10 poeng)

Her er statiske linker tegnet på venstre side (i rødt) og dynamiske linker tegnet på høyre side (i grønt). De to pilene i midten viser at y er en peker (siden y blir overført med referanse).



#### Forslag til poengfordeling:

Aktiveringsblokkene (main - P - Q - P): 2 poeng

Dynamiske linker: 2 poeng

Statiske linker: 2 poeng

Current + free: 2 poeng

Pekere for y: 2 poeng

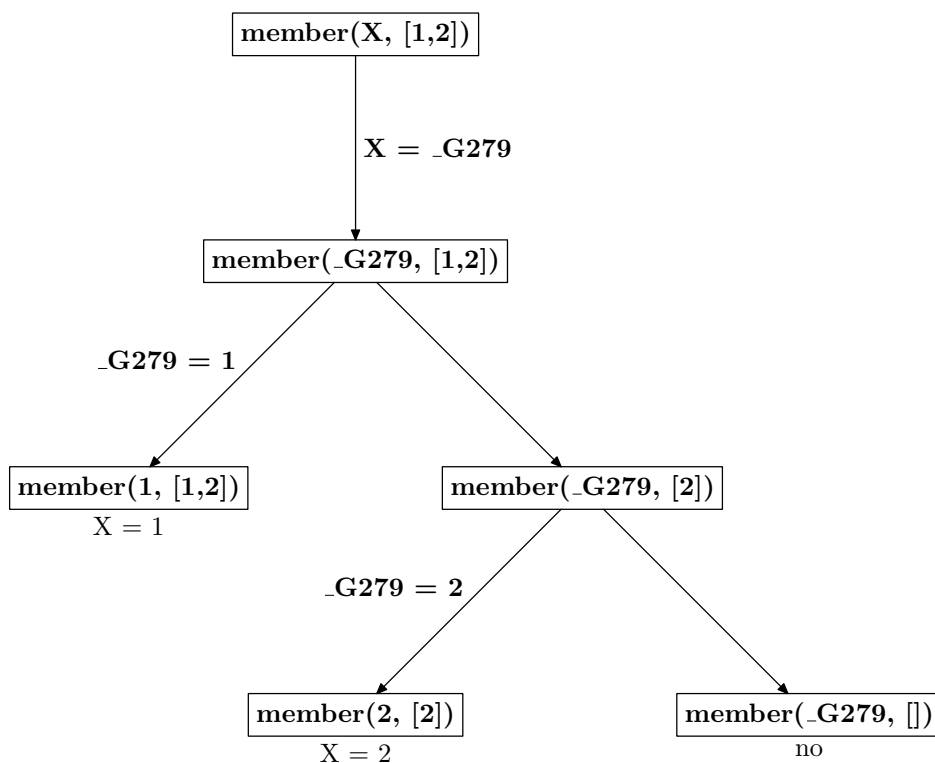
(Fortsettes på side 7.)

**5b** (2 poeng)

Nei. Q er ikke ferdig med eksekveringen av sine kall. Hvis Q endelig hadde nådd slutten, etter at P var ferdig, så ville verdien ha blitt kopiert tilbake til lagerplassen som y (i P) opptar.

**Oppgave 6 PROLOG** (8 poeng)**6a Unifikasjon** (4 poeng)

1.  $G = [1, 2, 3, 4]$
2.  $Z = h(a) \quad U = a$
3. No
4.  $X = \_G149 \quad Y = 2 \quad Z = \_G149$

**6b Member** (4 poeng)

Prolog svarer altså:  $X = 1$ ;  $X = 2$ ; No.