

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.

Først noen generelle råd og bemerkninger:

- Oppgavesettet består av seks uavhengige deler. Det er mange oppgaver og dermed viktig at du disponerer tiden godt.
- Poengene angitt på hver del antyder hvor mye vekt de forskjellige delene vil bli tillagt ved sensuren. Totalt kan du oppnå 60 poeng for denne besvarelsen.
- Om oppgaveteksten på noe punkt er uklar eller upresis, kan du gjøre dine egne presiseringer. Formulér i så fall disse tydelig i oppgavebesvarelsen din.
- Legg vekt på å gi korte og klare forklaringer.

Lykke til!

(Fortsettes på side 2.)

Oppgave 1 ML (14 poeng)

1a Typer (8 poeng)

Hvilken type vil ML gi til følgende funksjoner? Husk at ML forsøker å gi den mest generelle/polymorfe typen for hvert uttrykk. Med andre ord: for hver funksjon, hva er den mest generelle typen til denne funksjonen? (Se eksemplene nedenfor.)

- `fun f1(x) = [x]`
- `fun f2(x) = [42, x, x, x]`
- `fun f3(x,y,z) = (x,z)`
- `fun f4(x,z,b) = if b then x else 10`
- `fun f5(x) = if (x > 0) then x*f5(x-1) else 1`
- `fun f6(g,x) = g(x)`
- `fun f7 g x = g(x)`
- `fun f8(i) = fn j => if (i = j) then i else j`

Eksempler:

<code>fun f(x) = (42, "h")</code>	får typen	<code>'a -> int * string.</code>
<code>fun f(x) = x</code>	får typen	<code>'a -> 'a.</code>

1b Programmering (6 poeng)

Merk: Her er det ikke tillatt å benytte seg av standardbiblioteket. Alt skal være rent funksjonelt; man kan ikke benytte seg av imperative konstruksjoner.

(a) Skriv en funksjon

```
fun double (ll : int list list) : int list list = ...
```

som tar en liste av lister av heltall og som fordobler hvert tall slik:

```
double [[1,2,3],[2,3,4]] gir [[2,4,6],[4,6,8]]
```

(b) Skriv en funksjon

```
fun init (seq: 'a list) (lst: 'a list) : bool = ...
```

(Fortsettes på side 3.)

som er slik at `init seq lst` returnerer `true` hvis og bare hvis `seq` er den første delen av listen `lst`:

```
init [1,2] [1,2,3]    gir    true
init [1,2] [0,1,2,3] gir    false
```

Oppgave 2 Syntaks (10 poeng)

I denne oppgaven skal vi se på forskjellige språk over alfabetet $\{0,1\}$.

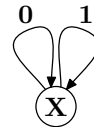
For (a) og (b) under: angi (1) et regulært uttrykk og (2) en ikke-deterministisk automat som godtar *nøyaktig* de angitte strengene.

(a) Strenger som inneholder sekvensen **010**.

(b) Strenger som inneholder både **0** og **1**.

Eksempel: Alle strenger over $\{0,1\}$.

(1) Et regulært uttrykk: $[[0|1]]^*$.



(2) En endelig deterministisk automat:

(c) Gjør om den ikke-deterministiske automaten i punkt (a) til en *deterministisk* automat. La det gå klart frem (f.eks. av tegningen) hvordan denne overgangen er gjort.

Oppgave 3 Grammatikker (9 poeng)

3a (5 poeng)

Angi hvilke alternativer som er riktige. (Det er minst ett korrekt alternativ for hver oppgave, men det kan også være flere riktige alternativer. Angi da nøyaktig de riktige alternativene.)

(1) En LL(1)-grammatikk

- (a) kan ha tomme høyresider
- (b) kan være høyrerekursiv
- (c) kan være venstreerekursiv
- (d) ingen av disse passer

(Fortsettes på side 4.)

- (2) For enhver kontekstfri grammatikk så fins det
- (a) en kontekstsensitiv grammatikk for det samme språket
 - (b) en regulær grammatikk for det samme språket
 - (c) en LL(1)-grammatikk for det samme språket
 - (d) ingen av delene
- (3) Rekursive definisjoner er tillatt i
- (a) regulære grammatikker
 - (b) kontekstfrie grammatikker
 - (c) kontekstsensitive grammatikker
 - (d) LL(1)-grammatikker
- (4) En kontekstfri grammatikk må være
- (a) regulær
 - (b) entydig
 - (c) LL(1)
 - (d) ingen av delene
- (5) Uttrykkskraften til klassisk BNF er
- (a) lik uttrykkskraften til utvidet BNF
 - (b) større enn uttrykkskraften til utvidet BNF
 - (c) mindre enn uttrykkskraften til utvidet BNF

(At **A** har større uttrykkskraft enn **B** betyr her at **A** kan beskrive alle språkene som **B** kan beskrive, men ikke omvendt.)

3b (1 poeng)

Beskriv med så få ord som mulig hvilket språk følgende grammatikk definerer.

$$\langle S \rangle \rightarrow \mathbf{a} \langle S \rangle \mathbf{b} \mid \mathbf{ab}$$

3c (1 poeng)

Beskriv med så få ord som mulig hvilket språk følgende grammatikk definerer.

$$\langle S \rangle \rightarrow \mathbf{a} \langle S \rangle \mathbf{a} \mid \mathbf{b} \langle S \rangle \mathbf{b} \mid \varepsilon$$

(Fortsettes på side 5.)

3d (2 poeng)

La følgende grammatikk være gitt.

$$\langle S \rangle \rightarrow \mathbf{a} \langle B \rangle \mid \mathbf{b} \langle A \rangle \langle B \rangle$$
$$\langle A \rangle \rightarrow \mathbf{a} \mid \langle A \rangle \langle A \rangle \mathbf{b}$$
$$\langle B \rangle \rightarrow \mathbf{b} \mid \mathbf{b} \langle S \rangle \mid \mathbf{a} \langle B \rangle \langle B \rangle$$

(a) Tegn et syntakstre for strengen **baababb**

(b) Vis at grammatikken er flertydig.

Oppgave 4 λ -kalkyle (7 poeng)

Reduser følgende λ -termer så mye som mulig. Vis hvert reduksjonssteg og substitusjonssteg:

1. $(\lambda x.x)z$
2. $((\lambda y.x)(\lambda x.x)x)[x/y]$
3. $(\lambda x.xx)((\lambda b.b)(\lambda c.c))$
4. $(\lambda xy.xyx)(\lambda b.b)(\lambda c.c)$

(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)

Tegn hvordan datalageret/"run-time"-stakken ser ut rett før Q blir kalt for *andre* gang. Pass på å få med eventuelle dynamiske linker, statiske linker, parametre og lokale variable. Tegn disse med alle verdier.

```
1 main()
2 {
3     int x = 0;
4
5     P(int y)      \\ y overføres ved referanse
6     {
7         Q(int z)  \\ z overføres ved verdi
8         {
9             z := z + 12;
10            P(z);
11        }
12
13        y := y + 1;
14        Q(y);
15    }
16
17    P(x);
18 }
```

5b (2 poeng)

Hvis Q hadde overført sine parametre ved *verdi-resultatoverføring*, ville datalageret sett annerledes ut? Hvorfor/hvorfor ikke?

(Fortsettes på side 7.)

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

Denne oppgaven dreier seg om PROLOGs unifikasjon.. Gi PROLOGs svar for hver av spørringene under.

1. $x(G) = x([1, 2, 3, 4])$.
2. $f(a, Z) = f(U, h(U))$.
3. $f(U, h(V)) = f(a, g(a))$.
4. $numbers(1, 2, X) = numbers(1, Y, Z)$.

Eksempel:

?- $f(a) = f(X)$.

PROLOG svarer $X = a$.

6b Member (4 poeng)

Her er predikatet member, som finner ut hvilke elementer som er medlem av en liste.

```
member(E, [E|_]).  
member(E, [_|Rest]) :- member(E, Rest).
```

Eksempel på bruk:

?- member(2, [1, 2, 3]).

Yes

Vis et fullstendig søketre for spørringen ?- member(X, [1, 2]) og gi alle svarene PROLOG gir.