

TDT4165 PROGRAMMING LANGUAGES

Fall 2012

Exercise 03 Deklarativ beregningsmodell

Før du starter:

Relevant lesestoff:

- Kapittel 2.2 - 2.4, 2.6, 2.8.2 i V&H.

Du skal levere:

- teori03.txt: Svar på de teoretiske spørsmålene.
- kode03.oz: Løsning på programmeringsoppgavene.
- sporsmal03.txt (valgfritt): Mulige spørsmål du har.
- tilbakemelding03.txt (valgfritt): Tilbakemeldinger om faget: Hva er hovedproblemene? Hvordan kan det forbedres? Andre kommentarer?

Du må svare på alle deloppgaver

Om du ikke klarer å løse en deloppgave må du forklare hva som er vanskelig med deloppgaven (din løsningsstrategi og hvorfor den ikke fungerer).

Task 1 Løse og bundne identifikatorer

For hver identifikator i de følgende utsagn, skriv om den er løs eller bundet. Dersom en identifikator binder (maps) til flere forskjellige variabler i minnet (single-assignment store), oppgi om den er løs eller bundet for hver forskjellige binding (mapping).

- a) local X in {P X Y} end
- b) local X in local Y in {X Y Z} end end
- c) proc {P X} local Y in {Q Z Y} end end
- d) case X of f(Y) then {P Y} else {Q Y} end

Task 2 Eksterne referanser

Før opp de eksterne referansene i følgende prosedyrer:

- a) proc {P X Y} {Q X Y} end
- b) proc {P X Y} {P X Y} end
- c) proc {P X Y} {Q Z U} end
- d) proc {P X Y} local Z in {Q Z U} end end
- e) proc {P X Y} local Z in {Q Z Y} end end

Task 3 Omskriving til kjernespråk

Du skal nå skrive om noen kjente funksjoner fra det utvidete språket. Du kan bare bruke syntaks fra tabell 2.1 og 2.2 i V&H. Dette er faktisk et subsett av det deklaratve kjernespråket (som også tillater syntaktisk sukker som `Record.Field` og `X=Y+Z`).

- a) fun {Append Xs Ys}
 case Xs
 of nil then
 Ys
 [] X|Xr then
 X|{Append Xr Ys}
 end
end

- b) Du kan her anta at identifikatorene `Equals`, `Plus` og `Minus` representerer variabler som er bundet til prosedyrer på formen

```
proc $ Arg1 Arg2 Out ... end
```

Kudos om du også klarer å binde disse selv – i kjernespråket – i stedet for å anta dem gitt.

```
fun {Max N M}
  if N==0 then
    M
  elseif M==0 then
    N
  else
    1+{Max N-1 M-1}
  end
end
```

Task 4 Deklarasjon av flere variable

Lag en regel for utvidelse av kjernespråket slik at det er mulig å deklare flere variable per `local` utsagn. Det skal være mulig å skrive for eksempel `local X Y Z in X = Y + Z end` i stedet for den mer tungvinte varianten med å måtte ha en `local` per variabel.

Vi betrakter altså et utsagn av formen

```
local <x>1...<x>n in <s> end
```

Regelen må ta hensyn til eventuelle endringer i minnet (store) og miljø (environment), og dytte `<s>` til stakken. Den skal skrives i samme stil som reglene i V&H 2.4.3.

Task 5 Eksekvering med den abstrakte maskinen

Her skal du eksekvere noen små programmer skrevet i det deklorative kjerne-språket (pluss utvidelsen fra oppgave 4) for hånd ved å bruke den abstrakte maskinen. Bruk notasjonen fra kapittel 2.4 i V&H. Oppgi hvilken regel du bruker på hvert steg.

- a) `local B in`
 `if B then skip else skip end`
 `end`

Hint: utførelsestilstanden i steg 0 er:

$$([(\text{local } B \text{ in if } B \text{ then skip else skip end end}, \phi), \phi)$$

- b) `local B in`
 `B = false`

```

        if B then skip else skip end
    end

c) local X Z A B P in
    proc {P X Y}
        Y = X + Z
    end
    Z = 7
    X = 4
    {P X A}
    {P A B}
end

```

Task 6 Unifikasjon

Hvilke av de følgende uttrykk vil føre til en unifikasjonsfeil? For de som kan unifiseres, vis den unifiserte datastrukturen og alle de brukte variablene, og hva de peker til i minnet. For alle andre forklar hvorfor unifikeringen feiler. Du kan anta at før utførelsen av disse uttrykkene er variablene ubundet. Prøv å løse disse oppgavene uten å bruke Mozart.

- a) $X = 2 \quad X = 3$
- b) $X = Y = a = Z$
- c) $D = 1|2|X = 1|2|3|4|Y|\text{nil}$
- d) $D = [1 \ Y \ X \ 3 \ 4] = 1|2|\text{nil}|3|4|Y$
- e) $X = 1|2|Y = 1|2|3|4|Y$