

INF2080 – Logikk og beregninger

Forelesning 10: Datastrukturer



UiO • Institutt for informatikk

Sist oppdatert: 2012-02-15 11:23

10.1 Beskrivelse — datastruktur

Binære trær

- Ønsker beskrivelse som ikke avhenger av regnemediet
- Datamaskin er en syntaksmaskin
- Informasjonsbiter og syntaktiske operasjoner
- Binære trær — bygd opp fra **nil** ved bruk av **cons**
- $\mathcal{B} :: \text{nil} | \text{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, ...
- Signatur
 - **nil** : \mathcal{U}
 - **cons** : $\mathcal{U} \times \mathcal{U} \rightarrow \mathcal{U}$
 - \prec : $\mathcal{U} \times \mathcal{U} \rightarrow \text{bool}$ — konstruert før (deltre)

Binære trær — nye begrep

$$\begin{aligned}x \preceq y & : & x \prec y \wedge x = y \\ \forall x \prec y. Fx & : & \forall x (x \prec y \rightarrow Fx) \\ \exists x \prec y. Fx & : & \exists x (x \prec y \wedge Fx) \\ \forall x \preceq y. Fx & : & Fy \wedge \forall x \prec y. Fx \\ \exists x \preceq y. Fx & : & Fy \vee \exists x \prec y. Fx \\ x = \mathbf{hd}(y) & : & \exists z \prec y. y = \mathbf{cons}(x, z) \\ x = \mathbf{tl}(y) & : & \exists z \prec y. y = \mathbf{cons}(z, x) \\ x \prec \mathbf{hd}(y) & : & \exists u \prec y. \exists v \prec y. (y = \mathbf{cons}(u, v) \wedge x \preceq v) \\ x \prec \mathbf{tl}(y) & : & \exists u \prec y. \exists v \prec y. (y = \mathbf{cons}(u, v) \wedge x \preceq u) \\ x \preceq \mathbf{hd}(y) & : & x = \mathbf{hd}(y) \vee x \prec \mathbf{hd}(y) \\ x \preceq \mathbf{tl}(y) & : & x = \mathbf{tl}(y) \vee x \prec \mathbf{tl}(y) \\ x = \mathbf{hd}^*(y) & : & x \preceq y \wedge \forall z \preceq y (x \prec z \rightarrow x \preceq \mathbf{hd}(z)) \\ x = \mathbf{tl}^*(y) & : & x \preceq y \wedge \forall z \preceq y (x \prec z \rightarrow x \preceq \mathbf{tl}(z))\end{aligned}$$

Binære trær — formler

- Konnektiver — enkelt, dele opp i alle tilfeller
- Kvantorer — vanskelig, ubegrenset søk
- Begrensete kvantorer — som generelle konnektiver, ganske enkelt
- Begreps hierarki


$\Delta_0 = \Pi_0 = \Sigma_0$: Ingen ubegrensete kvantorer

Π_{n+1} : \forall -kvantor utenfor Σ_n

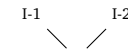
Σ_{n+1} : \exists -kvantor utenfor Π_n

Δ_n : Både Π_n og Σ_n

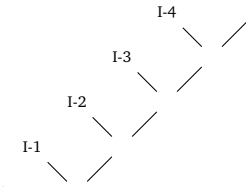
Binære trær — koding

- En informasjonsbit 

- Par av informasjonsbiter



- Endelig sekvens av informasjonsbiter
- Et deltre x er en informasjonsbit av y : $x = \mathbf{hd\ tl}^*y$
- Alt dette er Δ_0 — ingen ubegrensete kvantorer



Binære trær — hva kan kodes

Vesentlig at det er binære trær

Δ_0 : Beskrives med begrensete kvantorer

- Syntaks = Δ_0
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

Σ_1 : Fins x med syntaktiske krav

- Noe kan aksepteres av maskin
- Noe er bevisbart

Π_1 : Noe er ikke bevisbart, kan ikke aksepteres

Π_2 : Spesifikasjon — $\forall \text{inn} \exists \text{ut} \dots$