# INF2080 – Logikk og beregninger

## Forelesning 10: Datastrukturer

UiO : Institutt for informatikk

Sist oppdatert: 2012-02-15 11:23

## 10.1 Beskrivelse — datastruktur

# Binære trær

# Binære trær

- Ønsker beskrivelse som ikke avhenger av regnemediet
- Datamaskin er en syntaksmaskin

# Binære trær

- Ønsker beskrivelse som ikke avhenger av regnemediet
- Datamaskin er en syntaksmaskin
- Informasjonsbiter og syntaktiske operasjoner

# 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**

# 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**
- $\mathcal{B} :: \textbf{nil} | \textbf{cons}(\mathcal{B}, \mathcal{B})$

# 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**
- $\mathcal{B} :: \mathbf{nil} | \mathbf{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, ...

# 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**
- $\mathcal{B} :: \mathbf{nil} | \mathbf{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, ...
- Signatur

# 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**
- $\mathcal{B} :: \mathbf{nil} | \mathbf{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, . . .
- Signatur
    - $\mathbf{nil} : \mathcal{U}$

# 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**
- $\mathcal{B} :: \mathbf{nil} | \mathbf{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, ...
- Signatur
  - $\mathbf{nil} : \mathcal{U}$
  - $\mathbf{cons} : \mathcal{U} \times \mathcal{U} \to \mathcal{U}$

# 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**
- $\mathcal{B} :: \textbf{nil}|\textbf{cons}(\mathcal{B}, \mathcal{B})$
- Basis for Lisp, Scheme, Haskell, Prolog, ...
- Signatur
  - $\textbf{nil} : \mathcal{U}$
  - $\textbf{cons} : \mathcal{U} \times \mathcal{U} \to \mathcal{U}$
  - $\prec : \mathcal{U} \times \mathcal{U} \to \textbf{bool}$ — konstruert før (deltre)

# Binære trær — nye begrep

$$x \preceq y \quad : \quad x \prec y \wedge x = y$$

# Binære trær — nye begrep

$$
\begin{aligned}
x \preceq y &\quad : \quad x \prec y \wedge x = y \\
\forall x \prec y.Fx &\quad : \quad \forall x(x \prec y \to Fx)
\end{aligned}
$$

# Binære trær — nye begrep

$$
\begin{array}{lcl}
x \preceq y & : & x \prec y \wedge x = y \\
\forall x \prec y.Fx & : & \forall x(x \prec y \rightarrow Fx) \\
\exists x \prec y.Fx & : & \exists x(x \prec y \wedge Fx)
\end{array}
$$

# Binære trær — nye begrep

$$
\begin{array}{lcl}
x \preceq y & : & x \prec y \wedge x = y \\
\forall x \prec y.Fx & : & \forall x(x \prec y \rightarrow Fx) \\
\exists x \prec y.Fx & : & \exists x(x \prec y \wedge Fx) \\
\forall x \preceq y.Fx & : & Fy \wedge \forall x \prec y.Fx
\end{array}
$$

# Binære trær — nye begrep

$$
\begin{array}{rcc}
x \preceq y & : & x \prec y \wedge x = y \\
\forall x \prec y.Fx & : & \forall x(x \prec y \rightarrow Fx) \\
\exists x \prec y.Fx & : & \exists x(x \prec y \wedge Fx) \\
\forall x \preceq y.Fx & : & Fy \wedge \forall x \prec y.Fx \\
\exists x \preceq y.Fx & : & Fy \vee \exists x \prec y.Fx
\end{array}
$$

# Binære trær — nye begrep

$$
\begin{array}{rcc}
x \preceq y & : & x \prec y \wedge x = y \\
\forall x \prec y.Fx & : & \forall x(x \prec y \rightarrow Fx) \\
\exists x \prec y.Fx & : & \exists x(x \prec y \wedge Fx) \\
\forall x \preceq y.Fx & : & Fy \wedge \forall x \prec y.Fx \\
\exists x \preceq y.Fx & : & Fy \vee \exists x \prec y.Fx \\
x = \mathbf{hd}(y) & : & \exists z \prec y.y = \mathbf{cons}(x, z)
\end{array}
$$

# Binære trær — nye begrep

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

# Binære trær — nye begrep

$$
\begin{array}{rcl}
x \preceq y & : & x \prec y \land x = y \\
\forall x \prec y.Fx & : & \forall x(x \prec y \to Fx) \\
\exists x \prec y.Fx & : & \exists x(x \prec y \land Fx) \\
\forall x \preceq y.Fx & : & Fy \land \forall x \prec y.Fx \\
\exists x \preceq y.Fx & : & Fy \lor \exists x \prec y.Fx \\
x = \mathbf{hd}(y) & : & \exists z \prec y.y = \mathbf{cons}(x, z) \\
x = \mathbf{tl}(y) & : & \exists z \prec y.y = \mathbf{cons}(z, x) \\
x \prec \mathbf{hd}(y) & : & \exists u \prec y.\exists v \prec y.(y = \mathbf{cons}(u, v) \land x \preceq v)
\end{array}
$$

# Binære trær — nye begrep

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

# Binære trær — nye begrep

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

# Binære trær — nye begrep

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

# Binære trær — nye begrep

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

# Binære trær — nye begrep

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

# Binære trær — formler

# Binære trær — formler

- Konnektiver — enkelt, dele opp i alle tilfeller
- Kvantorer — vanskelig, ubegrenset søk

# Binære trær — formler

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

# Binære trær — formler

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

# 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

# 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

  $\Pi_{n+1}$ : $\forall$-kvantor utenfor $\Sigma_n$

# Binære trær — formler

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

$$\begin{aligned} \Delta_0 = \Pi_0 = \Sigma_0 &: \text{ Ingen ubegrensete kvantorer} \\ \Pi_{n+1} &: \ \forall\text{-kvantor utenfor } \Sigma_n \\ \Sigma_{n+1} &: \ \exists\text{-kvantor utenfor } \Pi_n \end{aligned}$$

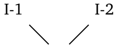
# Binære trær — formler

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

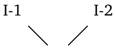
$$\begin{aligned} \Delta_0 = \Pi_0 = \Sigma_0 &: \text{ Ingen ubegrensete kvantorer} \\ \Pi_{n+1} &: \forall\text{-kvantor utenfor } \Sigma_n \\ \Sigma_{n+1} &: \exists\text{-kvantor utenfor } \Pi_n \\ \Delta_n &: \text{ Både } \Pi_n \text{ og } \Sigma_n \end{aligned}$$

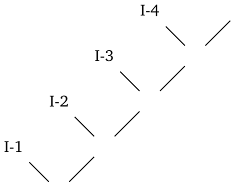
# Binære trær — koding

# Binære trær — koding

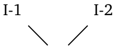
- En informasjonsbit 
- Par av informasjonsbiter 

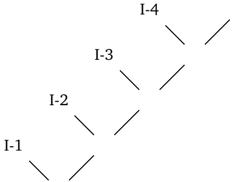
# Binære trær — koding

- En informasjonsbit 
- Par av informasjonsbiter 

- Endelig sekvens av informasjonsbiter 

# Binære trær — koding

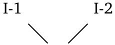
- En informasjonsbit 
- Par av informasjonsbiter 

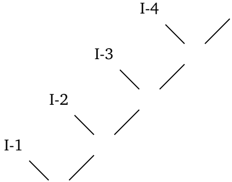
- Endelig sekvens av informasjonsbiter 

- Et deltre $x$ er en informasjonsbit av $y$ : $x = \mathbf{hd}\ \mathbf{tl}^* y$

# Binære trær — koding

- En informasjonsbit 

- Par av informasjonsbiter 


- Endelig sekvens av informasjonsbiter
- Et deltre $x$ er en informasjonsbit av $y$ : $x = \textbf{hd tl}^{*} y$
- Alt dette er $\Delta_0$ — ingen ubegrensete kvantorer

# Binære trær — hva kan kodes

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks = $\Delta_0$

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks $= \Delta_0$
- En kjøring

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks $= \Delta_0$
- En kjøring
- Et bevis

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks = $\Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks = $\Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

$\Sigma_1$ : Fins $x$ med syntaktiske krav

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks = $\Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

$\Sigma_1$ : Fins $x$ med syntaktiske krav

- Noe kan aksepteres av maskin

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks = $\Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

$\Sigma_1$ : Fins $x$ med syntaktiske krav

- Noe kan aksepteres av maskin
- Noe er bevisbart

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks $= \Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

$\Sigma_1$ : Fins $x$ med syntaktiske krav

- Noe kan aksepteres av maskin
- Noe er bevisbart

$\Pi_1$ : Noe er ikke bevisbart, kan ikke aksepteres

# Binære trær — hva kan kodes

Vesentlig at det er binære trær

$\Delta_0$ : Beskrives med begrensete kvantorer

- Syntaks $= \Delta_0$
- En kjøring
- Et bevis
- En sekvens av info-biter med syntaktiske krav

$\Sigma_1$ : Fins $x$ med syntaktiske krav

- Noe kan aksepteres av maskin
- Noe er bevisbart

$\Pi_1$ : Noe er ikke bevisbart, kan ikke aksepteres

$\Pi_2$ : Spesifikasjon — $\forall$**inn**$\exists$**ut**...