



Formalisering af Metateori for Metaprogrammering i et Metalogisk System

af Stine Søndergaard

Fredag den 3. oktober 2008

Metaprogrammering

- Kode der genererer kode
- Trinvis beregning

$$power \equiv \mathbf{fix} \ p : \mathbf{nat} \rightarrow \mathbf{nat} \rightarrow \mathbf{nat} .$$
$$\lambda x : \mathbf{nat} . \lambda n : \mathbf{nat} .$$
$$\mathbf{case} \ n \ \mathbf{of} \ \mathbf{z} \quad \Rightarrow \ \mathbf{s} \ \mathbf{z}$$
$$| \ \mathbf{s} \ n' \Rightarrow \mathit{times} \ x \ (p \ x \ n')$$
$$power_2 \equiv \lambda x : \mathbf{nat} . \mathit{times} \ x \ (\mathit{times} \ x \ (\mathbf{s} \ \mathbf{z}))$$

- I specialet: Modallogisk motiverede metaprogrammeringssprog

Metateori

- Teori hvis emne er en anden teori
- Eksempel på teori:
 - Evaluering: $e \hookrightarrow v$
 - Typning: $\Gamma \vdash e : \tau$
- Eksempel på metateori:
 - Determinisme: $e \hookrightarrow v, e \hookrightarrow v' \Rightarrow v = v'$
 - Typesundhed: Et veltypet program går ikke galt

Metalogisk system

- LF (Edinburgh Logical Framework)
 - Uniform repræsentation af syntaks, semantik og metateori
 - *Metasprog*: Afhængigt typet λ -kalkule udstyret med signatur
 - *Metodologi*: Domme-som-typer og HOAS (LFs variable bruges til at repræsentere objektsprogets variable)
- Elf
 - Logikprogrammeringsfortolkning af LF
 - Konstanter i LF-signatur tildeles operationel betydning
 - Forespørgsler bliver til søgning efter lukkede termer af en bestemt type
- Twelf
 - Ekstra (meta) faciliteter der muliggør maskinverificering af metateori (totaliteten af logiske relationer tjekkes)

Problemstilling

- Kan flertrinsprogrammeringssprog repræsenteres ved brug af HOAS? Eller skaber det problemer, at der lever variable på forskellige beregningstrin?
- Er en eventuel repræsentation til at arbejde med, når man ønsker at få verificeret typiske metateoretiske resultater?
- Konklusion: Ja, det lykkedes! Dog med visse forbehold...

Mini-ML_{ex}[□] (eksplicit) : Syntaks

- Typer:

$$\tau ::= \mathbf{nat} \mid \tau_1 \rightarrow \tau_2 \mid \tau_1 \times \tau_2 \mid \square \tau$$

- Udtryk:

$$e ::= x \mid u \mid \lambda x : \tau . e \mid e_1 e_2 \mid \mathbf{fix} x : \tau . e \mid \\ \langle e_1, e_2 \rangle \mid \mathbf{fst} e \mid \mathbf{snd} e \mid \\ \mathbf{z} \mid \mathbf{s} e \mid \mathbf{case} e \mathbf{of} \mathbf{z} \Rightarrow e_1 \mid \mathbf{s} x \Rightarrow e_2 \mid \\ \mathbf{box} e \mid \mathbf{let} \mathbf{box} u = e_1 \mathbf{in} e_2$$

- Værdier:

$$v ::= \mathbf{z} \mid \mathbf{s} v \mid \lambda x : \tau . e \mid \langle v_1, v_2 \rangle \mid \mathbf{box} e$$

Mini-ML_{ex}[□] : Operationssemantik

- Storskridtsevaluering $e \hookrightarrow^{\text{ex}} v$:

$$\frac{}{\lambda x : \tau . e \hookrightarrow^{\text{ex}} \lambda x : \tau . e} \text{ EVAL}^{\text{ex}}\text{-ABS}$$

$$\frac{e_1 \hookrightarrow^{\text{ex}} \lambda x : \tau . e'_1 \quad e_2 \hookrightarrow^{\text{ex}} v_2 \quad \{v_2/x\} e'_1 \hookrightarrow^{\text{ex}} v}{e_1 e_2 \hookrightarrow^{\text{ex}} v} \text{ EVAL}^{\text{ex}}\text{-APP}$$

$$\frac{}{\mathbf{box} e \hookrightarrow^{\text{ex}} \mathbf{box} e} \text{ EVAL}^{\text{ex}}\text{-BOX}$$

$$\frac{e_1 \hookrightarrow^{\text{ex}} \mathbf{box} e'_1 \quad \{e'_1/u\} e_2 \hookrightarrow^{\text{ex}} v}{\mathbf{let box} u = e_1 \mathbf{in} e_2 \hookrightarrow^{\text{ex}} v} \text{ EVAL}^{\text{ex}}\text{-LET-BOX}$$

Mini-ML[□]_{ex} : Eksempel

- Potensopløftning:

$$\begin{aligned} power^{ex} &\equiv \mathbf{fix} \ p : \mathbf{nat} \rightarrow \square (\mathbf{nat} \rightarrow \mathbf{nat}) . \\ &\quad \lambda n : \mathbf{nat} . \\ &\quad \quad \mathbf{case} \ n \ \mathbf{of} \ \mathbf{z} \quad \Rightarrow \ \mathbf{box} \ (\lambda x : \mathbf{nat} . \mathbf{s} \ \mathbf{z}) \\ &\quad \quad \quad | \ \mathbf{s} \ m \Rightarrow \ \mathbf{let} \ \mathbf{box} \ u = p \ m \\ &\quad \quad \quad \quad \mathbf{in} \ \mathbf{box} \ (\lambda x : \mathbf{nat} . \mathit{times} \ x \ (u \ x)) \end{aligned}$$
$$\begin{aligned} power^{ex} \ (\mathbf{s} \ (\mathbf{s} \ \mathbf{z})) &\hookrightarrow^{ex} \ \mathbf{box} \ (\lambda x : \mathbf{nat} . \mathit{times} \ x \\ &\quad \quad \quad ((\lambda x : \mathbf{nat} . \mathit{times} \ x \ ((\lambda x : \mathbf{nat} . \mathbf{s} \ \mathbf{z}) \ x)) \ x)) \end{aligned}$$

- Overflødige β -redekser

Mini-ML_{ex}[□] : Typesystem

- Omgivelser:

Globale: $\Delta ::= \cdot \mid \Delta, u :: \tau$ Lokale: $\Gamma ::= \cdot \mid \Gamma, x : \tau$

- Typedom $\Delta \mid \Gamma \vdash^{\text{ex}} e : \tau$:

$$\frac{\Gamma(x) = \tau}{\Delta \mid \Gamma \vdash^{\text{ex}} x : \tau} \text{TP}^{\text{ex}}\text{-VAR-X} \qquad \frac{\Delta \mid \Gamma\{x \mapsto \tau_1\} \vdash^{\text{ex}} e : \tau_2}{\Delta \mid \Gamma \vdash^{\text{ex}} \lambda x : \tau_1. e : \tau_1 \rightarrow \tau_2} \text{TP}^{\text{ex}}\text{-ABS}$$

$$\frac{\Delta \mid \Gamma \vdash^{\text{ex}} e_1 : \tau_1 \rightarrow \tau_2 \quad \Delta \mid \Gamma \vdash^{\text{ex}} e_2 : \tau_1}{\Delta \mid \Gamma \vdash^{\text{ex}} e_1 e_2 : \tau_2} \text{TP}^{\text{ex}}\text{-APP}$$

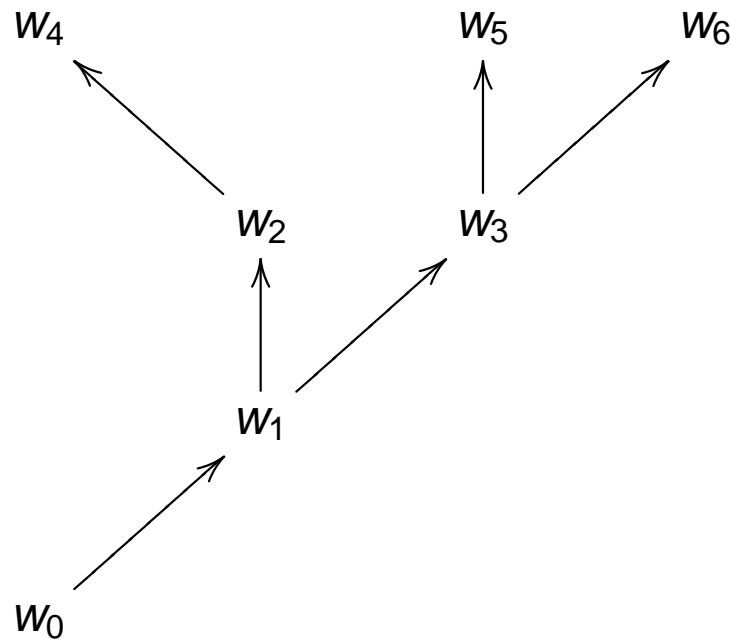
$$\frac{\Delta(u) = \tau}{\Delta \mid \Gamma \vdash^{\text{ex}} u : \tau} \text{TP}^{\text{ex}}\text{-VAR-U} \qquad \frac{\Delta \mid \cdot \vdash^{\text{ex}} e : \tau}{\Delta \mid \Gamma \vdash^{\text{ex}} \mathbf{box} e : \square \tau} \text{TP}^{\text{ex}}\text{-BOX}$$

$$\frac{\Delta \mid \Gamma \vdash^{\text{ex}} e_1 : \square \tau_1 \quad \Delta\{u \mapsto \tau_1\} \mid \Gamma \vdash^{\text{ex}} e_2 : \tau_2}{\Delta \mid \Gamma \vdash^{\text{ex}} \mathbf{let box} u = e_1 \mathbf{in} e_2 : \tau_2} \text{TP}^{\text{ex}}\text{-LET-BOX}$$

- Problem: I LF kan antagelser ikke fjernes, når de først er tilføjet
- Løsning: Visse antagelser gøres ubrugelige

Mini-ML \Box _{ex} : Verdener

- Udvidelse af Curry-Howard-isomorfien til at omfatte nødvendighedsoperatoren \Box
- Et beregningstrin svarer til en verden i Kripke-modellen:



Mini-ML_{ex}[□] : LF-repræsentation

- Intrinsic indkodning af verdener

- LF-signaturen Σ_{ex} :

tp : **type**
nat : **tp**
code : **tp** \rightarrow **tp**
world : **type**
exp : **world** \rightarrow **tp** \rightarrow **type** ...

- LF-omgivelser:

$$\Lambda = w_1 : \mathbf{world}, x_{11} : \mathbf{exp} w_1 T_{11}, \dots, x_{1 m_1} : \mathbf{exp} w_1 T_{1 m_1},$$
$$\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots$$
$$w_n : \mathbf{world}, x_{n1} : \mathbf{exp} w_n T_{n1}, \dots, x_{n m_n} : \mathbf{exp} w_n T_{n m_n},$$
$$u_1 : (\prod w : \mathbf{world}. \mathbf{exp} w T_1), \dots, u_m : (\prod w : \mathbf{world}. \mathbf{exp} w T_m)$$

- **Lemma:** Et kanonisk LF-objekt M med

$$\Lambda \vdash_{\Sigma_{\text{ex}}}^{\text{LF}} M : \mathbf{exp} w_j T$$

kan kun indeholde frie lokale variable blandt $x_{i1}, \dots, x_{i m_i}$

Mini-ML_{ex}[□] : LF-repræsentation

- Repræsentation af typeregler:

box : $\prod W : \mathbf{world} . \prod T : \mathbf{tp} .$

$(\prod W' : \mathbf{world} . \mathbf{exp} W' T) \rightarrow \mathbf{exp} W (\mathbf{code} T)$

let_box : $\prod W : \mathbf{world} . \prod T_1 : \mathbf{tp} . \prod T_2 : \mathbf{tp} .$

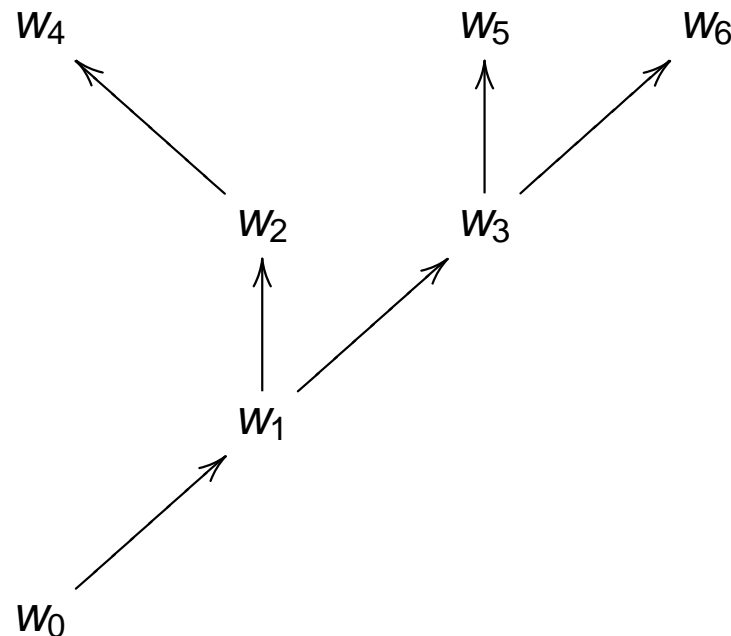
$\mathbf{exp} W (\mathbf{code} T_1) \rightarrow$

$((\prod W' : \mathbf{world} . \mathbf{exp} W' T_1) \rightarrow \mathbf{exp} W T_2) \rightarrow \mathbf{exp} W T_2$

- I **box** e skal e kunne types i en vilkårlig verden og kan derfor ikke gøre brug af lokale antagelser
- I **let box** $u = e_1$ **in** e_2 kan u types i en vilkårlig verden inde i e_2

Mini-ML[□] (implicit) : Motivation

- I Mini-ML_{ex}[□] genereres kodestumper eksplicit til brug for fremtidige trin
- I Mini-ML[□] overføres kodestumper på en mere implicit måde mellem beregningstrin (efterrationalisering)



Mini-ML[□] : Syntaks

- Typer:

$$\tau ::= \mathbf{nat} \mid \tau_1 \rightarrow \tau_2 \mid \tau_1 \times \tau_2 \mid \square \tau$$

- Udtryk:

$$\begin{aligned} m ::= & x \mid \lambda x : \tau . m \mid m_1 m_2 \mid \mathbf{fix} x : \tau . m \mid \\ & \langle m_1, m_2 \rangle \mid \mathbf{fst} m \mid \mathbf{snd} m \mid \\ & \mathbf{z} \mid \mathbf{s} m \mid \mathbf{case} m \mathbf{of} \mathbf{z} \Rightarrow m_1 \mid \mathbf{s} x \Rightarrow m_2 \mid \\ & \mathbf{box} m \mid \mathbf{unbox} p \end{aligned}$$

- Popudtryk:

$$p ::= m \mid \mathbf{pop} p$$

- Omgivelser:

Lokale: $\Gamma ::= \cdot \mid \Gamma, x : \tau$

Stakke: $\Psi ::= \cdot \mid \Psi; \Gamma$

$(\Psi; \Gamma) \vdash^{\text{im}} m : \tau$

Mini-ML[□] : Eksempel

- Potensopløftning:

$$\begin{aligned} power^{ex} &\equiv \text{fix } p : \text{nat} \rightarrow \square (\text{nat} \rightarrow \text{nat}) . \\ &\quad \lambda n : \text{nat} . \\ &\quad \text{case } n \text{ of } z \quad \Rightarrow \text{box } (\lambda x : \text{nat} . s z) \\ &\quad \quad | s m \Rightarrow \text{let box } u = p m \\ &\quad \quad \quad \text{in box } (\lambda x : \text{nat} . \text{times } x (u x)) \end{aligned}$$
$$\begin{aligned} power^{im} &\equiv \text{fix } p : \text{nat} \rightarrow \square (\text{nat} \rightarrow \text{nat}) . \\ &\quad \lambda n : \text{nat} . \\ &\quad \text{case } n \text{ of } z \quad \Rightarrow \text{box } (\lambda x : \text{nat} . s z) \\ &\quad \quad | s m \Rightarrow \text{box } (\lambda x : \text{nat} . \text{times } x \\ &\quad \quad \quad ((\text{unbox } (\text{pop } (p m)))) x)) \end{aligned}$$

- I stil med quasiquote / unquote

Mini-ML[□] : LF-repræsentation

- Samme LF-trick som i det eksplicitte system, men her fokuseres der på de enkelte verdensovergange i Kripketræet

Mini-ML[□] \rightsquigarrow Mini-ML_{ex}[□]

- Operationssemantik nemmere at formulere for Mini-ML_{ex}[□], hvor ræsonneringen ikke går bagud i tid
- Typebevarende oversættelse fra Mini-ML[□] til Mini-ML_{ex}[□]
 - *Idé*: Opsamling af alle forekomster af unbox-udtryk i liste og generering af let-box-bindinger
 - *Problem*: Håndtering af frie u -variable undervejs i processen
 - *Løsning*: Brug af indekserede labels i stedet for u -variable
 - *Resultat*: Kilde + målsprog bruger HOAS, mens oversættelsen bruger førsteordens-repræsentation

Sammenligning af de formaliserede systemer

- Mini-ML_{ex}[□] og Mini-ML[□]
 - Udtryk af typen $\square \tau$ har ingen frie ikke-globale variable
 - ⇒ Kode kan genereres og udføres sikkert i samme arbejdsgang
 - ⇒ Overflødige β -redekser i genereret kode
- Mini-ML[○]
 - Udtryk af typen $\circ \tau$ kan godt indeholde fri variable
 - ⇒ Overflødige β -redekser i genereret kode kan undgås
 - ⇒ Mister muligheden for at udføre genereret kode (dennes frie ikke-globale variable fremgår ikke af typen)
- Mini-ML_{co}
 - Det bedste fra begge verdener! Der holdes styr på ikke-globale variable i genereret kode: $[x_1 : \tau_1, \dots, x_n : \tau_n] \tau$
 - Generalisering af Mini-ML_{ex}[□]

Andre formaliserede metateoretiske resultater

- Determinisme for $\text{Mini-ML}_{\text{ex}}^{\square}$, Mini-ML° , $\text{Mini-ML}_{\text{co}}$
- Indlejring af $\text{Mini-ML}_{\text{ex}}^{\square}$ i $\text{Mini-ML}_{\text{co}}$
- Ækvivalens mellem stor- og lilleskridtssemantik for $\text{Mini-ML}_{\text{co}}$
- Typesundhed: Fremskridt og typebevaring (det sidste fulgte gratis pga. intrinsisk typning) for $\text{Mini-ML}_{\text{co}}$
- Alt sammen gik ganske glat!

Opsummering og afrunding

- Formaliseret 4 sprog (syntaks, typeregler, operationssemantik), typebevarende oversættelse, ækvivalens af semantikker, determinisme, typesundhed
- Adequacy-beviserne er centrale og bruges også i formaliserede beviser
- Alle metasætninger er verificeret i Twelf (omkring 5000 linier)
- Muligt fremtidigt arbejde:
 - Overføre den erhvervede erfaring til formaliseret ræsonnering om andre metaprogrammeringssystemer