Dependent Type Theory with Pattern-Matching and Size-Change Termination

TYPES 2006 Nottingham

David Wahlstedt

Chalmers University of Technology, Göteborg, Sweden

davidw@cs.chalmers.se

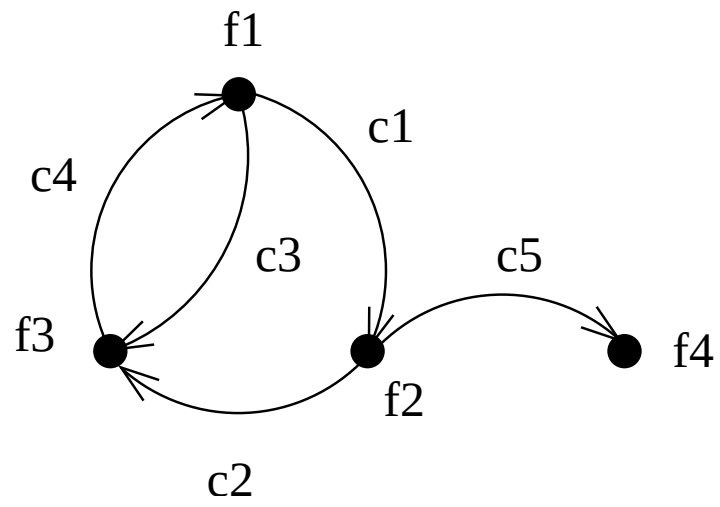
Contribution

A proof of normalisation for Martin-Löf's Logical Framework (Martin-Löf '86, in Nordström, Petersson, Smith '90) extended with first-order parametric data types and recursively defined constants with pattern-matching.

Recursion is proved well-founded if the definitions obey the *Size-Change Principle for Program Termination*. (Lee, Jones, Ben-Amram '01)

Pattern-matching definitions **Call Graph**

$$
\begin{array}{lcl}
f_1\ p_{11}\ \ldots\ p_{1n} & = & t_1 \\
\vdots & \vdots & \vdots \\
f_n\ p_{m1}\ \ldots\ p_{mn} & = & t_m
\end{array}
\quad \Rightarrow
$$



For each **call**, c_k, to f_j in t_i there is an arc from f_i to f_j.

For each recursive call c,

relate formal parameters and actual parameters:

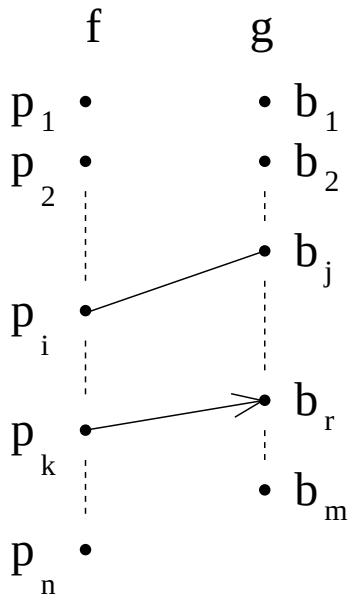
Recursive call c $\Rightarrow$ Size-change graph

$$f\ p_1\ \ldots\ p_n =$$

$$\ldots$$

$$g\ u_1\ \ldots\ u_m$$

$$\ldots$$



whenever

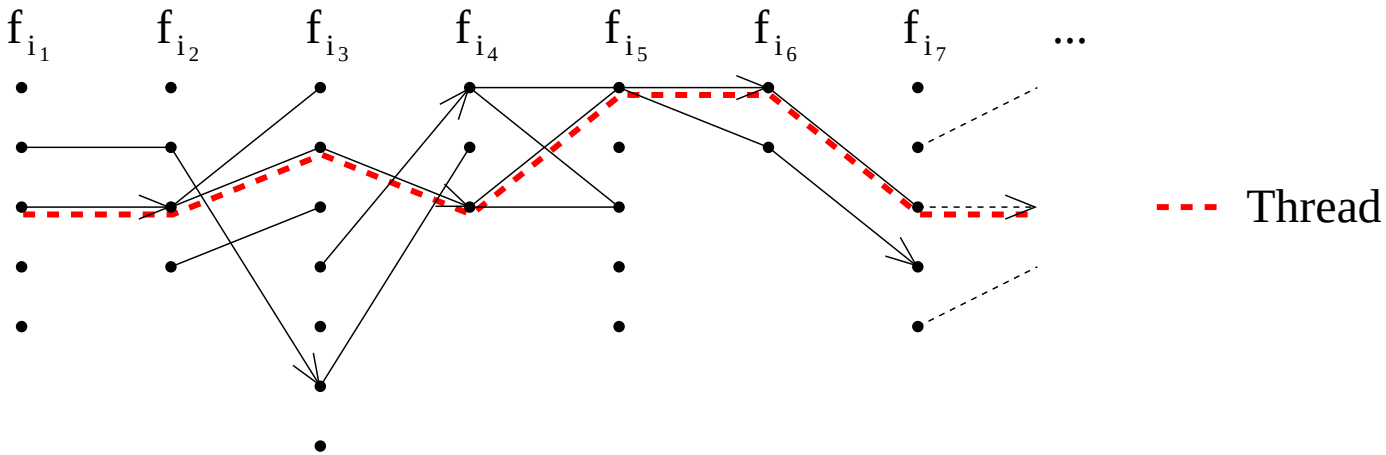
$p_i = b_j$

$p_k > b_r$

The Size-Change Principle

All the infinite call sequences contain an infinitely decreasing thread.

Path



Example: "zipping" lists

```
alternate : (A : Set) -> List A -> List A -> List A
alternate A nil ys = ys
alternate A (cons x xs) ys = cons x (alternate A ys xs)
```

Martin-Löf's logical framework

$$
\begin{array}{llll}
Terms & t, u, v & ::= & x \mid t\ u \mid \lambda x.t \\
Types & T, U, V & ::= & \mathrm{Set} \mid \mathrm{El}\ t \mid \mathrm{Fun}\ T\ (\lambda x.U)
\end{array}
$$

The notation $(x : T) \to U$ is shorthand for $\mathrm{Fun}\ T\ (\lambda x.U)$.

$(x_1 : T_1, \ldots\ ,\ x_n : T_n) \to T$
is shorthand for
$(x_1 : T_1) \to \ldots \to (x_n : T_n) \to T$.

Extended language

Defined constants

$$f : (x_1 : T_1, \ \ldots \ , x_n : T_n) \to T$$

with pattern-matching rules $\quad f \ p_1 \ \ldots \ p_n = t$

Patterns are of the form $\quad p ::= x \mid c \ p \ \ldots \ p$

First-order parametric data types $\quad d : \mathrm{Set}^n \to \mathrm{Set}$
with polymorphic constructors

$$c : \mathrm{El} \ e_1 \to \ldots \to \mathrm{El} \ e_n \to \mathrm{El} \ (d \ x_1 \ \ldots \ x_k)$$

with $\ e ::= x \mid d \ e \ \ldots \ e \ $ and $\ FV(e_i) \subseteq \{x_1, \ldots \ , x_k\}$

Extensions f, d, c are contained in the *signature* Σ.

Reduction rules

$t \rightsquigarrow t'$ reduction in one step

$t \rightsquigarrow^* t'$ reduction in zero or more steps

$$\frac{}{(\lambda x.t)\ u \rightsquigarrow_\beta t[x = u]}$$

$$\frac{}{f\ (p_1\ \gamma)\ \ldots\ (p_n\ \gamma) \rightsquigarrow_\iota t\ \gamma}\ f\ p_1\ \ldots\ p_n = t$$

Equality

$t =_{\beta\iota} u \ \Leftrightarrow\ t \rightsquigarrow^* v \ \wedge\ u \rightsquigarrow^* v.$

Reduction is confluent—equality is transitive.

Example: Set-valued functions

```
data T = Big | Small | T ==> T

subtype : T -> T -> Set
subtype Small Big = Unit
subtype Big Small = Empty
...
subtype (t1 ==> t2) (u1 ==> u2) =
   And (subtype u1 t1) (subtype t2 u2)
```

Context formation

$$\frac{}{() \text{ well-formed}} \qquad \frac{\Gamma \text{ well-formed} \quad \Gamma \vdash T}{(\Gamma, x : T) \text{ well-formed}}$$

Type formation

$$\frac{\Gamma \text{ well-formed}}{\Gamma \vdash \mathrm{Set}} \qquad \frac{\Gamma \vdash t : \mathrm{Set}}{\Gamma \vdash \mathrm{El}\ t} \qquad \frac{\Gamma \vdash U \quad (\Gamma, x : U) \vdash V}{\Gamma \vdash (x : U) \to V}$$

Type inhabitation

$$\frac{\Gamma \text{ well-formed}}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma \vdash T \quad \Gamma \vdash t : U}{\Gamma \vdash t : T}\ U =_{\beta\iota} T$$

$$\frac{\Gamma \vdash t : (x : U) \to V \quad \Gamma \vdash u : U}{\Gamma \vdash t\ u : V[u/x]} \qquad \frac{\Gamma \vdash U \quad (\Gamma, x : U) \vdash v : V}{\Gamma \vdash \lambda x.v : (x : U) \to V}$$

$$\frac{\Gamma \text{ well-formed}}{\Gamma \vdash f : \Sigma(f)} \qquad \frac{\Gamma \text{ well-formed}}{\Gamma \vdash d : \Sigma(d)} \qquad \frac{\Gamma \vdash u_1 : \mathrm{Set} \quad \ldots \quad \Gamma \vdash u_k : \mathrm{Set}}{\Gamma \vdash c : \Sigma(c)[u_1,\ \ldots\ , u_k]}$$

The cartesian product of a family of sets

$$\frac{}{\Gamma \vdash \Pi : (x : \mathrm{Set},\ \mathrm{El}\ x \to \mathrm{Set}) \to \mathrm{Set}}$$

$$\frac{\Gamma \vdash u : \mathrm{Set} \quad \Gamma \vdash t : \mathrm{El}\ u \to \mathrm{Set}}{\Gamma \vdash \mathrm{fun} : ((x : \mathrm{El}\ u) \to \mathrm{El}\ (t\ x)) \to \mathrm{El}\ (\Pi\ t\ u)}$$

We can define sets by recursion

$F : (n : \mathrm{El}\ \mathrm{Nat}) \to \mathrm{Set}$

$F\ \mathbf{zero} = \mathrm{Nat}$

$F\ (\mathbf{succ}\ n) = \Pi\ (F\ n)\ (\lambda x.\mathrm{Nat})$

We can only type-check β-normal terms

$$\begin{array}{rcl} s & ::= & h\ s_1\ \ldots\ s_n \mid \lambda x.s \\ h & ::= & x \mid f \mid c \mid d \mid \Pi \mid \mathrm{fun} \end{array}$$

Checking type inhabitation

$$\frac{\Gamma \vdash s_i \uparrow T_i[s_1,\ \ldots\ ,s_{i-1}]}{\Gamma \vdash x\ s_1\ \ldots\ s_n \uparrow U} \begin{cases} \Gamma(x) = (x_1 : T_1,\ \ldots\ ,\ x_n : T_n) \to T \\ U =_{\beta\iota} T[s_1,\ \ldots\ ,s_n] \end{cases}$$

$$\frac{\Gamma, x : U \vdash s \uparrow V}{\Gamma \vdash \lambda x.s \uparrow (x : U) \to V}\ x \notin \Gamma$$

$$\frac{\Gamma \vdash s_i \uparrow T_i[s_1, \ \ldots \ , s_{i-1}]}{\Gamma \vdash f \ s_1 \ \ldots \ s_n \uparrow U} \begin{cases} \Sigma(f) = (x_1 : T_1, \ \ldots \ , \ x_n : T_n) \to T \\ U =_{\beta\iota} T[s_1, \ \ldots \ , s_n] \end{cases}$$

$$\frac{\Gamma \vdash s_i \uparrow \mathrm{El}\ e_i[u_1, \ \ldots \ , u_k]}{\Gamma \vdash c \ s_1 \ \ldots \ s_n \uparrow \mathrm{El}\ u} \begin{cases} \Sigma(c) = (\mathrm{El}\ e_1, \ \ldots \ , \mathrm{El}\ e_n) \\ \qquad\qquad \to \mathrm{El}\ (d \ x_1 \ \ldots \ x_k) \\ \mathrm{El}\ u \rightsquigarrow^* \mathrm{El}\ (d \ u_1 \ \ldots \ u_k) \end{cases}$$

$$\frac{\Gamma \vdash s_i \uparrow \mathrm{Set}}{\Gamma \vdash d \ s_1 \ \ldots \ s_n \uparrow \mathrm{Set}^m \to \mathrm{Set}} \ \Sigma(d) = \mathrm{Set}^{n+m} \to \mathrm{Set}$$

$$\frac{\Gamma \vdash s_1 \uparrow \mathrm{Set} \quad \Gamma \vdash s_2 \uparrow \mathrm{El}\ s_1 \to \mathrm{Set}}{\Gamma \vdash \Pi\ s_1\ s_2 \uparrow \mathrm{Set}}$$

$$\frac{\Gamma \vdash s \uparrow (x : \mathrm{El}\ u) \to \mathrm{El}\ (v\ x)}{\Gamma \vdash \mathrm{fun}\ s \uparrow U}\ U \rightsquigarrow^* \mathrm{El}\ (\Pi\ u\ v)$$

Theorem

If $\overset{\sim}{\prec}$ well-founded, Σ valid and $\Gamma \vdash t : T$,
then t normalisable.

Reducibility method
(Gödel '41, '58, Tait '67, Girard '71, Martin-Löf '72,
C. Coquand '96)

$\mathrm{RED}_T(t)$ (simplified version)

- T atomic:

  t normalizable.

- T is a function type $U \to V$:

  t maps reducible terms in U to reducible terms in V.

Given a new constant

$$f : (x_1 : T_1,\ \ldots,\ x_n : T_n) \to T,$$

assuming

$$\mathsf{RED}_{T_1}(t_1),\ \mathsf{RED}_{T_2[t_1]}(t_2),\ \ldots\ ,\ \mathsf{RED}_{T_n[t_1,\ \ldots\ ,\ t_{n-1}]}(t_n),$$

we have to prove

$$\mathsf{RED}_{T[t_1,\ \ldots\ ,\ t_n]}(f\ t_1\ \ldots\ t_n)$$

Call relation

- Formal call

  $g\ u_1\ \ldots\ u_m \prec f\ p_1\ \ldots\ p_n$ whenever

  there is a rule $f\ p_1\ \ldots\ p_n = t$

  with $g\ u_1\ \ldots\ u_m$ subterm of t.

- Call instance

  $(g\ u_1\ \ldots\ u_m)\ \gamma\ \delta \mathrel{\tilde{\prec}} (f\ p_1\ \ldots\ p_n)\ \gamma$ whenever

  $g\ u_1\ \ldots\ u_m \prec f\ p_1\ \ldots\ p_n$

  with γ and δ normal.

In the case when $t_i \equiv p_i\ \gamma$ and $(f\ p_1\ \ldots\ p_n)\ \gamma \rightsquigarrow_\iota s\ \gamma$, knowing γ reducible and $\Delta \vdash s : T[p_1\ \ldots\ p_n]$, prove $\mathsf{RED}_{T[p_1\ \ldots\ p_n]\gamma}(s\ \gamma)$.

By completeness we get $\Delta \vdash s \uparrow T[p_1\ \ldots\ p_n]$, and we prove the goal by induction on the type-checking and the well-foundedness of $\tilde{\prec}$.