Termination of Mutually Recursive Functions

Andreas Abel

POP Seminar, CMU Computer Science
May 26, 2000

Slide 1

1. Introduction
2. The foetus Project
3. Mutually Recursive Functions with One Argument
4. Mutually Recursive Functions with Several Arguments

Recursion over Inductive Types

Slide 2

- Functional programming languages and logical frameworks base upon λ-calculus enriched with inductive types.
  Examples: ML, LEGO
- Definition of functions/constants by recursion over inductive type possible.
- Standard means: recursor/elimination. Ensures totality.
  Example:

  $$\begin{aligned}\text{half'} &= R^N\,(\lambda x^B.\,0)\,(\lambda x^N \lambda f^{B\to N}.\,R^B\,(f\,\text{true})\,(1 + (f\,\text{false})))\\ \text{half} &= \lambda n^N.\,\text{half'}\,n\,\text{false}\end{aligned}$$

  Drawback: Misses intuition, readability, usability.

Slide 3

Pattern Matching

- Alternative: "free recursive definitions".
  Example:

  half 0 = 0
  half 1 = 0
  half n+2 = (half n)+1

  But: syntax permits non-total functions $\Longrightarrow$ totality check required!
- LEGO allows to implement proofs by pattern matching, but fails to perform totality check $\Longrightarrow$ invalid proofs possible!

Slide 4

The foetus Project

1996 **Mu**nich **T**ype **T**heory **I**mplementation (T. Altenkirch)

1998 Implementation of termination checker foetus for a sublanguage of MuTTI (A. Abel)

1999 Reimplementation of termination checker into Agda (C. Coquand, Chalmers, Sweden)

1999 Verification I: Wellfoundedness of domains [AA99]

2000 Verification II: Single Recursive Functions [Abe00]
Verification III: Mutually Recursive Functions (in progress)

Wellfoundedness and Accessibility

Let S be a set and $<$ a relation on S. The accessible part $\mathsf{Acc}_{>} \subseteq S$ is defined as the smallest set closed under

$$w \in \mathsf{Acc}_{>} :\Longleftrightarrow \forall v < w.\ v \in \mathsf{Acc}_{>}$$

Accessible part induction (wellfounded induction):

Slide 5

$$\frac{\forall w \in S.\ (\forall v < w.\ P(v)) \Rightarrow P(w)}{\forall w \in \mathsf{Acc}_{>}.\ P(w)}$$

Wellfounded part $\mathsf{WF} \subseteq S$:

$$w \in \mathsf{WF}_{>} :\Longleftrightarrow \nexists f : \mathbb{N} \to S.\ f(0) = w \wedge \forall n \in \mathbb{N}.\ f(n) > f(n+1)$$

Brouwer's bar theorem (axiom of bar induction):

$$\mathsf{WF}_{>} \subseteq \mathsf{Acc}_{>}$$

(Classically provable.)

Single Recursive Function

Slide 6

- Assume a wellfounded domain $(\mathcal{D}, <)$, i.e., $\mathcal{D} = \mathsf{Acc}_{>}$.
- Provided that:
  1. all statements (except the recursive calls) in f terminate
  2. in each recursive call the argument v is smaller than the function input w

  we can define termination of function f at argument $w \in \mathcal{D}$ as:

$$\frac{\forall v < w.\ f@v \Downarrow}{f@w \Downarrow}$$

- Goal: $\forall w \in \mathcal{D}.\ f@w \Downarrow$
- Proof by wellfounded induction.

Mutually Recursive Functions with a Single Argument

Slide 7

- Let $\mathcal{F}$ be a finite set of function symbols.

$$g \preceq f :\iff f \longrightarrow g \qquad \text{“}f \text{ calls } g\text{”}$$

- Straightforward extension of predicate “terminates at”:

$$\begin{aligned} f@w \Downarrow \quad &:\iff \quad \forall g \preceq f,\ v < w.\ g@v \Downarrow \\ \mathcal{F}@w \Downarrow \quad &:\iff \quad \forall f \in \mathcal{F}.\ f@w \Downarrow \end{aligned}$$

- Goal: $\forall w \in \mathcal{D}.\ \mathcal{F}@w \Downarrow$
- Proof by wellfounded induction.
- But: criterion to strict!

Call Graphs

Slide 8

- Sufficient: In each call *cycle*

$$f \longrightarrow g \longrightarrow \ldots \longrightarrow f$$

the argument is decreased once.

- Functions and calls can be organized in a labelled directed graph:

$$\begin{tikzcd} f \arrow[r, "?"] & g \arrow[r, bend left, "\leq"] & h \arrow[l, bend left, "<"] \end{tikzcd}$$

- Indirect (combined) calls:

$$\frac{f \xrightarrow{R} g}{f \xrightarrow{R}^{+} g} \qquad \frac{f \xrightarrow{R}^{+} g \qquad g \xrightarrow{S}^{+} h}{f \xrightarrow{S \star R}^{+} h}$$

$\star$	$<$	$\leq$	?
$<$	$<$	$<$	?
$\leq$	$<$	$\leq$	?
?	?	?	?

Good Call Graphs

Slide 9

- Let $\mathcal{C}$ be a call graph.

$$\mathcal{C} \text{ good} \quad :\Longleftrightarrow \quad \forall f \in \mathcal{F}.\, \forall f \xrightarrow{R_1} f_1 \xrightarrow{R_2} \ldots \xrightarrow{R_n} f_n \xrightarrow{R_{n+1}} f.\, \prod_{i=1}^{n+1} R_i = \text{“}<\text{”}$$

- Good call graphs have two properties:
  Each cycle

$$f \xrightarrow{\vec{R}} f$$

  1. contains only calls that are at least preserving:

$$\forall i.\; R_i \in \{<, \leq\}$$

  2. contains at least one decreasing call:

$$\exists i.\; R_i = \text{“}<\text{”}$$

No Infinite Call Sequences

Slide 10

- Goal: All call sequences $f(w) \rightsquigarrow g(v) \rightsquigarrow \ldots$ terminate.
- Evaluation ordering $\ll$ on $\mathcal{F} \times \mathcal{D}$ must fulfill

$$\begin{aligned} (g, v) \ll (f, w) \quad \Leftarrow \quad & f \xrightarrow{\textbf{?}} g \\ & \vee (f \xrightarrow{\leq} g \wedge v \leq w) \\ & \vee (f \xrightarrow{\leq} g \wedge v < w) \end{aligned}$$

- Theorem: For good call graphs the most general ordering $\ll$ is wellfounded:

$$WF_{\gg} = \mathcal{F} \times \mathcal{D}$$

- Proof: Consider an infinite call sequence. Since $\mathcal{F}$ is finite, one particular function symbol f must appear infinitly often. Goodness of the call graph implies an infinite descend on the argument of f. Contradiction!

Classical Termination Proof

- New (weaker) termination predicate:

$$f@w \Downarrow \;:\Longleftrightarrow \forall (g,v) \ll (f,w).\ g@v \Downarrow$$

Slide 11

- Goal: $\forall f \in \mathcal{F}, w \in \mathcal{D}.\ f@w \Downarrow$.
- Proof by wellfounded induction, making use of the bar theorem.
- **Question 1:** Can we proof termination constructively without bar induction?

Alternative Goodness Characterization

- A call graph $\mathcal{C}$ is good if there is a bijective naming

$$f^{11}, \dots, f^{1m_1}, \dots, f^{n1} \dots f^{nm_n}$$

of the function symbols in $\mathcal{F}$ s.th.

Slide 12

$$\begin{aligned} f^{i_1 j_1} \xrightarrow{?} f^{i_2 j_2} &\quad\Rightarrow\quad i_1 > i_2 \\ f^{i_1 j_1} \xrightarrow{\leq} f^{i_2 j_2} &\quad\Rightarrow\quad i_1 > i_2 \vee (i_1 = i_2 \wedge j_1 > j_2) \\ f^{i_1 j_1} \xrightarrow{<} f^{i_2 j_2} &\quad\Rightarrow\quad i_1 \geq i_2 \end{aligned}$$

- This characterization has been used, e.g., by Frank Pfenning and Carsten Schürmann for termination checking in the Twelf system [PS98].
- **Question 2:** Are the two criteria equivalent?

Ordering on Function Symbols

Slide 13

- Define two relations $\prec, \lhd$ on $\mathcal{F}$ by

$$\begin{aligned} g \prec f \quad &:\Longleftrightarrow \quad f \longrightarrow g \wedge g \not\longrightarrow^{+} f \\ g \lhd f \quad &:\Longleftrightarrow \quad f \xrightarrow{\leq} g \end{aligned}$$

- Theorem: Both relations are wellfounded.
- Proof: In both cases the transitive closure is irreflexive. Since $\mathcal{F}$ is finite, this entails wellfoundedness.

$$\begin{aligned} f \prec^{+} f \quad &\Rightarrow \quad f \longrightarrow^{+} f \wedge f \not\longrightarrow^{+} f \\ f \lhd^{+} f \quad &\Rightarrow \quad f \xrightarrow{\leq}{}^{+} f \qquad \text{(contradicts goodness)} \end{aligned}$$

- The modified lexicographic product $\prec \otimes' \lhd$ is wellfounded, too, and can be completed to a total ordering. **Answer 2: yes!**

$$g \ \prec\otimes'\lhd \ f \quad :\Longleftrightarrow \quad g \prec f \vee (g \preceq f \wedge g \lhd f)$$

Wellfounded Evaluation Ordering

Slide 14

- Define relation $\ll$ on $\mathcal{F} \times \mathcal{D}$:

$$\begin{aligned} (g, v) \ll (f, w) \ :\Longleftrightarrow \quad & g \prec f \\ \vee \quad & (g \preceq f \ \wedge \quad v < w \\ & \qquad \vee \quad (v \leq w \quad \wedge g \lhd f)) \end{aligned}$$

- Theorem: $\ll$ is a wellfounded evaluation ordering.
- Proof: Wellfounded: $\ll$ is a modified lexicographic product of wellfounded relations.
  Evaluation ordering:

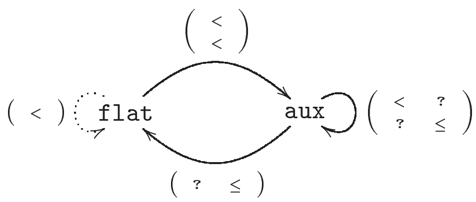
$$\begin{aligned} f \xrightarrow{?} g \quad &\Rightarrow \quad g \prec f && \text{(Goodness property 1)} \\ f \xrightarrow{\leq} g \wedge v < w \quad &\Rightarrow \quad g \preceq f \wedge v < w \\ f \xrightarrow{\leq} g \wedge v \leq w \quad &\Rightarrow \quad g \preceq f \wedge v \leq w \wedge g \lhd f \end{aligned}$$

- Now we can proof $\forall w \in \mathcal{D}, f \in \mathcal{F}.\ f@w \Downarrow$ by wellfounded induction.
  Answer 1: yes!

Towards Functions with Several Arguments

Slide 15

```
fun flat []          = []
  | flat (l::ls)     = aux l ls
and aux  []      ls = flat ls
  | aux  (x::xs) ls = x :: aux xs ls;
```



Call Graphs for Functions with Several Arguments

Slide 16

- Let $\mathcal{F}$ be a finite set of function symbols with arity mapping $\mathsf{ar} : \mathcal{F} \to \mathbb{N}$
- A call graph is a labelled directed multi-graph with edges

$$f \xrightarrow{\sigma,a} g$$

  s.th.

$$\begin{array}{lcll} \sigma & : & \mathsf{ar}(g) \to \mathsf{ar}(f) & \text{permutation of arguments} \\ a & : & \mathsf{ar}(g) \to \{<, \leq, ?\} & \text{size change information} \end{array}$$

- A call graph is good iff

$$\forall f \xrightarrow{\sigma,a}{}^{+} f.\ \exists k.\ \sigma = \mathsf{id} \restriction k \wedge \mathsf{lex}^k_<(a)$$

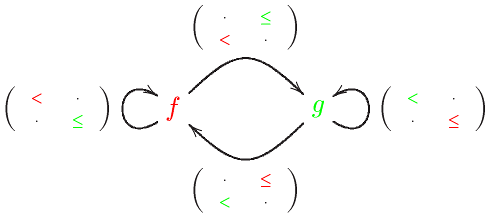
  where we refer to k as number of relevant arguments and

$$\begin{array}{lcl} \mathsf{lex}^k_<(a) & :\Longleftrightarrow & \exists k' < k.\ a(k') = \text{“}<\text{”} \wedge \forall i < k'.a(i) = \text{“}\leq\text{”} \\ \mathsf{lex}^k_=(a) & :\Longleftrightarrow & \forall i < k.\ a(i) = \text{“}\leq\text{”} \\ \mathsf{lex}^k_\leq(a) & :\Longleftrightarrow & \mathsf{lex}^k_<(a) \vee \mathsf{lex}^k_=(a) \end{array}$$

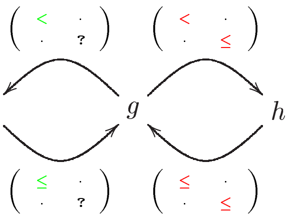
Complications

Slide 17

- Attributes "decreasing" ($<$) and "preserving" ($\leq$) of a call are no longer global. The call $f \longrightarrow g$ is decreasing for f and preserving for g.



- Two call cycles may have a different number of relevant arguments. Here $k(g \to f \to g) = 1$ and $k(g \to h \to g) = 2$.



Argument Trace

Slide 18

- Arguments are being permuted $\Rightarrow$ we need an argument trace

$$\tau_{f \to g} : \mathsf{ar}(g) \to \mathsf{ar}(f) \qquad \text{for all } f, g \in \mathcal{F}$$

- Requirements: For each cycle $h \longrightarrow^* f \xrightarrow{\sigma,a} g \longrightarrow^* h$ with k relevant arguments

$$\tau_{h \to h} = \mathsf{id} \restriction k \tag{1}$$

$$\tau_{f \to h} = \sigma \circ \tau_{g \to h} \restriction k \tag{2}$$

- Example: $\tau_{g\to f} = \mathsf{id}$, not $\tau_{g\to f} = (1\ 2)$.

Slide 19

$$\begin{tikzcd} f \arrow[r, bend left, "{\begin{pmatrix}<&\cdot&\cdot\\ \cdot&\cdot&<\\ \cdot&<&\cdot\end{pmatrix} \quad \begin{pmatrix}\leq&\cdot&\cdot\\ \cdot&\leq&\cdot\\ \cdot&\cdot&\leq\end{pmatrix}}"] & g \arrow[l, bend left, "{\begin{pmatrix}\leq&\cdot&\cdot\\ \cdot&<&\cdot\\ \cdot&\cdot&?\end{pmatrix}}"] \end{tikzcd}$$

Call Classification

- We classify the calls as decreasing resp. (strictly) preserving by ($R \in \{<, =, \leq\}$):

$$\mathsf{class}^h_R(f \xrightarrow{\sigma,a} g) \;:\Longleftrightarrow\; \forall Z = h \longrightarrow^* f \xrightarrow{\sigma,a} g \longrightarrow^* h.\ \mathsf{lex}^{k(Z)}_R(a \circ \tau_{g\to h})$$

- Property 1. In each cycle each call is preserving

Slide 20

$$\forall Z = h \longrightarrow^* f \xrightarrow{\sigma,a} g \longrightarrow^* h.\ \mathsf{class}^h_{\leq}(f \xrightarrow{\sigma,a} g)$$

- Classification of transitions:

$$\begin{aligned} f \xrightarrow{<} g \quad &:\Longleftrightarrow \quad \exists h \approx f.\ \forall f \xrightarrow{\sigma,a} g.\ \mathsf{class}^h_<(f \xrightarrow{\sigma,a} g) \\ f \xrightarrow{\leq} g \quad &:\Longleftrightarrow \quad \forall h \approx f.\ \exists f \xrightarrow{\sigma,a} g.\ \mathsf{class}^h_=(f \xrightarrow{\sigma,a} g) \\ f \xrightarrow{?} g \quad &:\Longleftrightarrow \quad g \not\longrightarrow f \end{aligned}$$

$h \approx f$ is defined as $h \longrightarrow^* f \longrightarrow^* h$.

Evaluation Ordering

- We define $g \prec f$ as before and

$$g \lhd f :\Longleftrightarrow f \xrightarrow{\leq} g$$

- Theorem: Both relations are wellfounded.

Slide 21

- Define $v <^h_{f \to g} w$ as "v is smaller than w wrt. to h in a call from f to g". This relation is wellfounded.
- Theorem: The relation $\ll$ defined by

$$\begin{aligned}(g, v) \ll (f, w) \quad :\Longleftrightarrow \quad & g \prec f \vee g \approx f \wedge (\forall h.\ v \leq^h_{f \to g} w) \\ & \wedge ((\exists h.\ v <^h_{f \to g} w) \vee g \lhd f)\end{aligned}$$

  is a wellfounded evaluation ordering.
- Proof: $\ll$ is a lexicographic product of three wellfounded relations. The second of these is a multiset ordering of wellfounded relations indexed by h.

Further Extensions

Weaken the definition of good to allow:

- Multiset orderings.
- Cycles of higher order. Example:

```
zip []       l = l
 |  (x::xs) l = x :: zip l xs;
```

Slide 22

References

[AA99] Andreas Abel and Thorsten Altenkirch. A predicative analysis of structural recursion. Submitted to the Journal of Functional Programming, December 1999.

[Abe00] Andreas Abel. Specification and verification of a formal system for structurally recursive functions. Submitted to TYPES'99, January 2000.

[PS98] Frank Pfenning and Carsten Schürmann. Twelf user's guide. Technical report, Carnegie Mellon University, 1998.