Formal and Constructive Theory of Computation

Second Bachelor Seminar Talk

Fabian Kunze

Advisor: Prof. Dr. Gert Smolka

SAARLAND UNIVERSITY

COMPUTER SCIENCE

19.06.2015

Overview

1. Review
   - L: weak, call-by-value λ-Calculus
   - SK: call-by-value Combinator Calculus
2. connecting L and SK
   - reduction-respecting $L \to SK$-homomorphism
   - L-decidable $\Rightarrow$ SK-decidable
   - reduction-respecting $SK \to L$-homomorphism
   - SK-decidable $\Rightarrow$ L-decidable
3. L_C: L with closures
   - L-normalization tactic in Coq
4. Outlook

L: weak, call-by-value λ-Calculus

$$s, t \ ::= \ x \ \mid \ \lambda s \ \mid \ s \ t \qquad (x \in \mathbb{N})$$

$$x^x_u = u \qquad\qquad (\lambda s)^y_u = \lambda(s^{y+1}_u)$$

$$x^y_u = x \qquad\qquad (s \ t)^x_u = s^x_u \ t^x_u$$

$$\frac{}{(\lambda s)\ (\lambda t) \succ s^0_{\lambda t}} \qquad \frac{s \succ s'}{s \ t \succ s' \ t} \qquad \frac{t \succ t'}{s \ t \succ s \ t'}$$

switched back from named variables to de Bruijn indices:

- α-equivalence 'for free'
- simpler inductive definition of closed
- simple representation of enviroments (for L_C)

$\Rightarrow$ simplifies formal proofs

But: named variables more intuitive $\Rightarrow$ used as human-readable description

SK: call-by-value Combinator Calculus

$$N, M \;::=\; x \mid \mathsf{K} \mid \mathsf{S} \mid N\ M \qquad (x \in \mathbb{N})$$

$$\mathit{Val}\!:\ N, M \;::=\; x \mid \mathsf{K} \mid \mathsf{K}\ N \mid \mathsf{S} \mid \mathsf{S}\ N \mid \mathsf{S}\ N\ M \qquad (x \in \mathbb{N})$$

$$\frac{N, M \in \mathit{Val}}{\mathsf{K}\ N\ M \succ N} \qquad \frac{N, M, H \in \mathit{Val}}{\mathsf{S}\ N\ M\ H \succ N\ H\ (M\ H)} \qquad \frac{N \succ N'}{N\ M \succ N'\ M}$$

$$\frac{M \succ M'}{N\ M \succ N\ M'}$$

last talk:

- uniform confluent
- reduction-respecting homomorphism from L into SK

reduction-respecting $L \to SK$-homomorphism

desired properties

- $\underline{\bullet} : L \to SK$ which is:
  - homomorphic: by definition

$$\underline{s\ t} = \underline{s}\ \underline{t}$$

  - reduction-respecting: proven

$$\frac{s \succ_L t}{\underline{s} \succ_{SK}^{*} \underline{t}}$$

  - value preserving: proven

$$s \text{ irreduzible } \Leftrightarrow \underline{s} \in Val$$

  - tight: not proven yet (but for old version)

$$\frac{\underline{s} \succ_{SK} N}{\exists t, s \succ_L t \wedge N \succ_{SK}^{*} \underline{t}}$$

reduction-respecting $L \to SK$-homomorphism

improved homomorphism

$$
\begin{aligned}
[x].x &:= \mathsf{I} & & \\
[x].N &:= \mathsf{K}\ N & & \text{if } x \notin \mathrm{FV}(N) \text{ and } N \in \mathit{Val} \\
[x].(N\ M) &:= \mathsf{S}\ ([x].N)\ ([x].M) & & \text{otherwise}
\end{aligned}
$$

- $\cancel{N \in \mathit{Val} \Longrightarrow} [x].N \in \mathit{Val}$
- $\cancel{N}, M \in \mathit{Val} \Longrightarrow ([x].N)\ M \succ^* N^x_M$
- $x \neq y \land y \notin \mathrm{FV}(M) \Longrightarrow ([y].N)^x_M = [y].(N^x_M)$

$$
\begin{aligned}
\underline{x} &:= x & \cancel{\mathsf{H}'x} &:= \cancel{\mathsf{K}\ x} \\
\underline{s\ t} &:= \underline{s}\ \underline{t} & \cancel{\mathsf{H}'(s\ t)} &:= \cancel{\mathsf{S}\ (\mathsf{H}'s)\ (\mathsf{H}'t)} \\
\underline{\lambda x.s} &:= \cancel{\mathsf{S}}\ ([x].\cancel{\mathsf{H}'}\underline{s})\ \cancel{\mathsf{I}} & \cancel{\mathsf{H}'(\lambda x.s)} &:= \cancel{\mathsf{K}\ (\mathsf{S}\ ([x].\mathsf{H}'s)\ \mathsf{I})}
\end{aligned}
$$

reduction-respecting $L \to SK$-homomorphism

desired properties

$\underline{\bullet} : L \to SK$ which is:

- homomorphic: by definition

$$\underline{s\ t} = \underline{s}\ \underline{t}$$

- reduction-respecting: proven

$$\frac{s \succ_L t}{\underline{s} \succ^*_{SK} \underline{t}}$$

- value preserving: proven

$$s \text{ irreduzible } \Leftrightarrow \underline{s} \in Val$$

- tight: not proven yet (but for old version)

$$\frac{\underline{s} \succ_{SK} N}{\exists t, s \succ_L t \wedge N \succ^*_{SK} \underline{t}}$$

L-decidable $\Rightarrow$ SK-decidable

u is L-decider for predicate p:

$$\forall s, p\ s \land u \ulcorner s \urcorner \succ^* \ulcorner \text{true} \urcorner$$
$$\lor \neg p\ s \land u \ulcorner s \urcorner \succ^* \ulcorner \text{false} \urcorner$$

$\overset{?}{\Rightarrow}$

N is SK-decider for predicate p:

$$\forall s, p\ s \land N\ \underline{\ulcorner s \urcorner} \succ^* \underline{\ulcorner \text{true} \urcorner}$$
$$\lor \neg p\ s \land N\ \underline{\ulcorner s \urcorner} \succ^* \underline{\ulcorner \text{false} \urcorner}$$

For L-decider u, $\underline{u}$ is SK-decider:

- $\forall s, \underline{u}\ \underline{\ulcorner s \urcorner} = \underline{u \ulcorner s \urcorner}$
- $\underline{\bullet}$ is reduction-respecting

$\to$ L-decidable $\Rightarrow$ SK-decidable

reduction-respecting $SK \to L$-homomorphism

desired properties

$\overline{\cdot} : SK \to L$ which is:

- homomorphic: conflicts with value preservation (but $\mathcal{L}$ is)

$$\overline{N\ M} = \overline{N}\ \overline{M}$$

- reduction-respecting: proven

$$\frac{N \succ_{SK} M}{\overline{N} \succ_{L}^{*} \overline{M}}$$

- value preserving: proven

$$N \in Val \Leftrightarrow \overline{N} \text{ irreduzible}$$

- tight: not proven yet

$$\frac{\overline{N} \succ_{L} s}{\exists M, N \succ_{SK} M \wedge s \succ_{L}^{*} \overline{t}}$$

reduction-respecting $SK \to L$-homomorphism

$$\begin{aligned}\mathcal{L}\mathsf{K} &:= \lambda x.\lambda y.x \\ \mathcal{L}\mathsf{S} &:= \lambda x.\lambda y.\lambda z.(x\ z)\ (y\ z) \\ \mathcal{L}(N\ M) &:= \mathcal{L}N\ \mathcal{L}M\end{aligned}$$

homomorphism, but for partly applied combinators:
not value-preserving (e.g. $\mathcal{L}(K\ K)$)

$$\begin{aligned}\overline{\mathsf{K}} &:= \lambda x.\lambda y.x \\ \overline{(\mathsf{K}\ N)} &:= \lambda y.\overline{N} && (N \in Val) \\ &\vdots && \vdots \\ \overline{(\mathsf{S}\ N\ M)} &:= \lambda z.(\overline{N}\ z)\ (\overline{M}\ z) && (N, M \in Val) \\ \overline{(N\ M)} &:= \overline{N}\ \overline{M} && \text{otherwise}\end{aligned}$$

- $\mathcal{L}s \succ^* \overline{s}$

reduction-respecting $SK \to L$-homomorphism

desired properties

$\overline{\cdot} : SK \to L$ which is:

- homomorphic: conflicts with value preservation (but $\mathcal{L}$ is)

$$\overline{N\ M} = \overline{N}\ \overline{M}$$

- reduction-respecting: proven

$$\frac{N \succ_{SK} M}{\overline{N} \succ_L^* \overline{M}}$$

- value preserving: proven

$$N \in Val \Leftrightarrow \overline{N} \text{ irreduzible}$$

- tight: not proven yet

$$\frac{\overline{N} \succ_L s}{\exists M, N \succ_{SK} M \land s \succ_L^* \overline{t}}$$

SK-decidable $\Rightarrow$ L-decidable

N is SK-decider for predicate p:

$$\forall s, p\ s \land N\ \underline{\ulcorner s \urcorner} \succ^* \underline{\ulcorner \text{true} \urcorner}$$
$$\lor\ \neg p\ s \land N\ \underline{\ulcorner s \urcorner} \succ^* \underline{\ulcorner \text{false} \urcorner}$$

$\overset{?}{\Rightarrow}$

u is L-decider for predicate p:

$$\forall s, p\ s \land u\ \ulcorner s \urcorner \succ^* \ulcorner \text{true} \urcorner$$
$$\lor\ \neg p\ s \land u\ \ulcorner s \urcorner \succ^* \ulcorner \text{false} \urcorner$$

Just $\overline{\bullet}$ does not suffice:
We know about $\overline{N\ \underline{\ulcorner s \urcorner}}$ and $\overline{N}\ \overline{(\underline{\ulcorner s \urcorner})}$, not about $\overline{N}\ \ulcorner s \urcorner$

- internalize LH $\in L$ with LH $\ulcorner s \urcorner \succ^* \overline{(\underline{\ulcorner s \urcorner})}$
- $\overline{N}$ (LH $\ulcorner s \urcorner$) $\succ^* \overline{N}\ \overline{(\underline{\ulcorner s \urcorner})}$ ($\succ^* \overline{(\underline{\ulcorner b \urcorner})}$)
- $\overline{(\underline{\ulcorner b \urcorner})}\ \ulcorner \text{true} \urcorner\ \ulcorner \text{false} \urcorner \succ^* \ulcorner b \urcorner$ for booleans

$\rightarrow$ N is SK-decider $\Rightarrow$ $\lambda x.(\overline{N}\ (\text{LH}\ x)\ \ulcorner \text{true} \urcorner\ \ulcorner \text{false} \urcorner)$ is L-decider:
$\rightarrow$ SK-decidable $\Rightarrow$ L-decidable

L_C: L with closures

$$p, q \ ::= \ s\,[C] \ \mid \ p\ q \qquad (s \in L,\ C \in L_C \text{ list})$$

Intuition: Computation tree of L-interpreter using stack as environments

$$\frac{}{(\lambda s)[C]\ (\lambda t)[D] \succ s[(\lambda t)[D]\ :: C]} \qquad \frac{}{(s\ t)[C] \succ s[C]\ t[C]}$$

$$\frac{}{x[C] \succ C!!x} \qquad \frac{p \succ p'}{p\ q \succ p'\ q} \qquad \frac{q \succ q'}{p\ q \succ p\ q'}$$

- $p \in \mathit{ValComp} :\Leftrightarrow p$ closed and environments contain only lambdas
- $\ulcorner \bullet \urcorner : L_C \to L$ (substitutes the enviroments)
- uniform confluent
- Sound: If $p \in \mathit{ValComp}$ and $p \succ q$, then $\ulcorner p \urcorner \succ_L \ulcorner q \urcorner$ (easy induction)
- Complete: If $p \in \mathit{ValComp}$ and $\ulcorner p \urcorner \succ_L s$,
  then there is q s.t. $s = \ulcorner q \urcorner$ and $p \succ^* q$ (induction)

step-indexed interpreter for L_C

$$
\begin{array}{lll}
\text{Eval } 0 \quad p & = & p \\
\text{Eval } (\text{S } n)\ ((\lambda s[A])\ (\lambda t)[B]) & = & \text{Eval } n\ (s[(\lambda t[B]) :: A]) \\
\text{Eval } (\text{S } n)\ (s[A]\ t[B]) & = & \text{Eval } n\ ((\text{Eval } n\ s[A])\ (\text{Eval } n\ t[B])) \\
\text{Eval } (\text{S } n)\ ((s\ t)[A]) & = & \text{Eval } n\ (s[A]\ t[A]) \\
\text{Eval } (\text{S } n)\ (x[A]) & = & A!!x
\end{array}
$$

- Correct: $s \succ^* (\text{Eval } n\ s)$
- Efficient: linear in length of reduction (fixed depth of binding)

L-normalization tactic in Coq

Example (Old Proof terms)

$(\lambda x.\lambda y.x)\ \mathsf{I}\ \mathsf{I} \succ (\lambda y.x)^x_{\mathsf{I}}\ \mathsf{I} = (\lambda y.\mathsf{I})\ \mathsf{I} \succ \mathsf{I}^y_{\mathsf{I}} = \mathsf{I}$

- rewrite step by step
- executes substitutions

$\Rightarrow O(n^2)$ for n-long reduction (e.g. $(\lambda x_1.\lambda x_2.\ldots\lambda x_n.x_1)\ \mathsf{I} \ldots\ \mathsf{I}$)

Example (New Proof terms)

$(\lambda x.\lambda y.x)\ \mathsf{I}\ \mathsf{I} \succ^* \ulcorner \mathrm{Eval}\ n\ ((\lambda x.\lambda y.x)\ \mathsf{I}\ \mathsf{I})[\,] \urcorner = \mathsf{I}$

$\Rightarrow O(n)$ for n-long reduction

Benchmark

$$\mathsf{Q}\ \ulcorner \lambda x.x \urcorner \succ^* \ulcorner\ulcorner \lambda x.x \urcorner\urcorner$$

Old: 27 sec

New: 0.3 sec

L-normalization tactic in Coq

Improved usability for intermediate reductions:

- crush_old: proofs goal or diverges(or unfolds everything)
- crush: unfolds everything and normalizes as far as possible
- wcrush: normalizes, do not unfold
- ccrush: normalizes, do not unfold, use correctness lemmas

Demo

Outlook

- are the reduction homomorphisms:
  - tight?
  - injective?
- L-undecidable properties of SK-Terms:
  - term-equivalence
  - halting problem
- Multivariate λ-Calculus

If there is enough time left:

- undecidability of intuitionistic first-order-logic