Type Preservation as a Confluence Problem – Aaron Stump et al.

Outline

- “Type Preservation as a Confluence Problem” by Aaron Stump, Garrin Kimmell, and Roba El Haj Omar
- “Confluence by decreasing diagrams” by Vincent van Oostrom

Definitions

(Global) Confluence

$$\begin{tikzcd} t \arrow[r, "*"] \arrow[d, "*"] & s_1 \arrow[d, dashed, "*"] \\ s_2 \arrow[r, dashed, "*"'] & t' \end{tikzcd}$$

Local Confluence

$$\begin{tikzcd} t \arrow[r] \arrow[d] & s_1 \arrow[d, dashed, "*"] \\ s_2 \arrow[r, dashed, "*"'] & t' \end{tikzcd}$$

Confluence via decreasing diagrams

by Vincent van Oostrom

Reduces proving confluence to finding a labeling on the transitions of a transition system and a well-founded partial-order on those labels such that the local confluence is “compatible” with that partial order. We will talk about this “compatiblity” in detail later.

This method is complete for countable transition systems. However, Finding this labeling can be “hard” since confluence is undecidable.

The following are corollaries of the theorem of decreasing diagrams:

- Newman's lemma: If a RS is terminating and locally confluent, it is globally confluent
- Hindley-Rosen lemma: if $\to_\alpha$ and $\to_\beta$ are confluent and $\to_\alpha^*$ and $\to_\beta^*$ commute, then their union is confluent
- ...

Theory of decreasing diagrams

- $\rightarrow_\alpha$ rewrite one-step with transition of label alpha
- ${}_\alpha\leftarrow$ inverse
- $\rightarrow_\alpha^=$ reflexive closure: zero or one steps
- $\rightarrow_\alpha^*$ reflexive transitive closure: zero or more steps
- $\curlyvee x$ is set of the labels **strictly** less than x.

Theory of decreasing diagrams

If every local peak can be completed to a locally decreasing diagram w.r.t. a fixed well-founded partial ordering on labeled steps in the diagram, then the reduction system is confluent.

A locally decreasing diagram has peaks and valleys of the form:

$$
\begin{tikzcd}
& & t \arrow[dll, "\to_{\beta}"'] \arrow[drr, "\to_{\alpha}"] & & \\
s_1 \arrow[d, dashed, "\to^{*}_{\curlyvee\beta}"'] & & & & s_2 \arrow[d, dashed, "\to^{*}_{\curlyvee\alpha}"] \\
{} \arrow[d, dashed, "\to^{=}_{\beta}"'] & & & & {} \arrow[d, dashed, "\to^{=}_{\alpha}"] \\
{} \arrow[drr, dashed, "\to^{*}_{\curlyvee\alpha\cup\curlyvee\beta}"'] & & & & {} \arrow[dll, dashed, "\to^{*}_{\curlyvee\alpha\cup\curlyvee\beta}"] \\
& & t' & &
\end{tikzcd}
$$

Newman's Lemma / Diamond Lemma

Every strongly normalizing/terminating abstract rewriting system $\mathcal{A}$ is confluent iff it is locally confluent.

- label every transition $a \rightarrow b$ with a

Type preservation of STLC as a Confluence Problem

- Traditionally typing is viewed as a big-step semantics.
- if, instead, we view typing as an small-steps operational semantics we can use our standard toolkit of rewriting tools.

In particular, we can phrase type-preservation as confluence on well-typed terms.

Syntax of Simply Typed Lambda Calculus

$$
\begin{array}{lcl}
\textit{types } T & ::= & A \mid T_1 \Rightarrow T_2 \\
\textit{standard terms } t & ::= & x \mid \lambda x : T.\, t \mid t\; t' \mid a \mid f \\
\textit{mixed terms } m & ::= & x \mid \lambda x : T.\, m \mid m\; m' \mid a \mid f \mid \\
 & & A \mid T \Rightarrow m \\
\textit{standard values } v & ::= & \lambda x : T.t \mid a \mid f \\
\textit{mixed values } u & ::= & \lambda x : T.m \mid T \Rightarrow m \mid A \mid a \mid f
\end{array}
$$

Figure 1: STLC syntax

Abstract and Concrete operational Semantics

$$\frac{}{E_c[f\ a]\ \rightarrow_c\ E_c[a]}\ c(f\text{-}\beta) \qquad \frac{}{E_c[(\lambda x:T.\,m)\ u]\ \rightarrow_c\ E_c[[u/x]m]}\ c(\beta)$$

$$\frac{}{E_a[(T \Rightarrow m)\ T]\ \rightarrow_a\ E_a[m]}\ a(\beta) \qquad \frac{}{E_a[\lambda x:T.\,m]\ \rightarrow_a\ E_a[T \Rightarrow [T/x]m]}\ a(\lambda)$$

$$\frac{}{E_a[f]\ \rightarrow_a\ E_a[A \Rightarrow A]}\ a(f) \qquad \frac{}{E_a[a]\ \rightarrow_a\ E_a[A]}\ a(a)$$

$$\begin{array}{lcl} \textit{mixed evaluation contexts } E_c & ::= & * \mid (E_c\ t) \mid (u\ E_c) \\ \textit{abstract evaluation contexts } E_a & ::= & * \mid (E_a\ m) \mid (m\ E_a) \mid \lambda x:T.\,E_a \mid T \Rightarrow E_a \end{array}$$

Figure 2: STLC Abstract and Concrete semantics

We want to show that on the set of *well-typed terms*, the combined reduction system ac is confluent.

We choose labels:

- c for concrete reduction rules
- a for abstract reduction rules
- and the partial order $a < c$

Instantiating the theory of decreasing diagrams, we need to show:

For every typable term t:

1. every c - c peak $s_1 {}_c\!\leftarrow t \rightarrow_c s_2$,
   can be completed with a valley: $s_1 \rightarrow_a^* \rightarrow_c^= \rightarrow_a^* t' {}_a^*\!\leftarrow {}_c^=\!\leftarrow {}_a^*\!\leftarrow s_2$
2. every a - a peak $s_1 {}_a\!\leftarrow t \rightarrow_a s_2$,
   can be completed with a valley $s_1 \rightarrow_a^= t' {}_a^=\!\leftarrow s_2$ (i.e. a has the diamond property)
3. every a - c peak $s_1 {}_a\!\leftarrow t \rightarrow_c s_2$,
   can be completed with a valley: $s_1 \rightarrow_c^= \rightarrow_a^* t' {}_a^*\!\leftarrow {}_c^=\!\leftarrow {}_a^*\!\leftarrow s_2$

$$
\begin{tikzcd}
& E_c[(\lambda x : T_1.m)\ u] \arrow[dl, "a"'] \arrow[dr, "c"] & \\
E_c[(T_1 \Rightarrow [T_1/x]m)\ u] \arrow[d, "a\ *", "since\ u \to_a^* T_1"'] & & E_c[[u/x]m] \arrow[ddl, "a\ *"', "since\ u \to_a^* T_1"] \\
E_c[(T_1 \Rightarrow [T_1/x]m)\ T_1] \arrow[dr, "a"'] & & \\
& E_c[[T_1/x]m] &
\end{tikzcd}
$$

Figure 3: a-c peak 5

Conclusion

- Stating typing as an abstract small-step semantics gives us the power of term-rewriting for proving meta properties of the type system.
- The proof is simple and intuitive