Example 1: Addition of ordinal numbers in SML

datatype Nat = \ldots

datatype Ord = O
  \mid S \text{ of Ord}
  \mid \text{Lim of Nat} \rightarrow \text{Ord};

fun addord x O = x
  \mid addord x (S \text{ y}') = S (addord x \text{ y}')
  \mid addord x (\text{Lim f}) = \text{Lim (fn z => addord x (f z))}
Example 2: A pattern matching proof in LEGO

\[\text{leRefl } v\text{Unit }\Rightarrow \text{leUnit} \]
\[\text{leRefl } (v\text{Inl } S v) \Rightarrow \text{leInl } S S (\text{leRefl } v) \]
\[\text{leRefl } (v\text{Inr } S v) \Rightarrow \text{leInr } S S (\text{leRefl } v) \]
\[\text{leRefl } (v\text{Pair } v w) \Rightarrow \text{lePair } (\text{leRefl } v) (\text{leRefl } w) \]
\[\text{leRefl } (v\text{Fold } R x) \Rightarrow \text{leFoldl } R (\text{leFoldr } R (\text{leRefl } x)) \]

Goal: From structural recursiveness ...
\[\forall v. (\forall w < v. f(w) \Downarrow) \rightarrow f(v) \Downarrow \]
... infer termination
\[\forall v. f(v) \Downarrow \]

Outline:
1. Def. of the foetus system: types \(\sigma \in \text{Ty}(\vec{X})\), terms \(t \in \text{Tm}^\sigma[\Gamma]\)
2. Def. of the evaluation strategy: syntactic values \(v \in \text{Val}^\sigma\), closures \(\langle t; e \rangle \in \text{Cl}^\sigma\), op. sem. \(\Downarrow \subseteq \text{Cl}^\sigma \times \text{Val}^\sigma\)
3. Def. of the semantics: “good” values \(v \in \llbracket \sigma \rrbracket\)
4. Def. of the structural ordering \(\prec_{\sigma,\tau} \subseteq \llbracket \sigma \rrbracket \times \llbracket \tau \rrbracket\)
5. Proof of the wellfoundedness \(\llbracket \sigma \rrbracket\) w.r.t. \(<\)
6. Def. of the good terms \(\text{TM}^\sigma[\Gamma]\)
7. Proof of the normalization: \(\forall t \in \text{TM}(\langle t; e \rangle \Downarrow)\)
### The foetus system

<table>
<thead>
<tr>
<th>Type</th>
<th>Terms / Values</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Unit)</td>
<td>1</td>
<td>unit set</td>
</tr>
<tr>
<td>(Var)</td>
<td>X, Y, Z, ...</td>
<td>type variables</td>
</tr>
<tr>
<td>(Sum)</td>
<td>(\sigma + \tau)</td>
<td>disjoint sum</td>
</tr>
<tr>
<td>(Prod)</td>
<td>(\sigma \times \tau)</td>
<td>product</td>
</tr>
<tr>
<td>(Arr)</td>
<td>(\sigma \rightarrow (\vec{X}))</td>
<td>function space</td>
</tr>
<tr>
<td>(Rec)</td>
<td>(\text{Rec}\ X.\sigma(X))</td>
<td>recursive (fixed-point) type</td>
</tr>
</tbody>
</table>

\[
\sigma(\text{Rec}\ X.\sigma(X)) \xrightarrow{\text{fold}} \text{Rec}\ X.\sigma(X) \xleftarrow{\text{unfold}}
\]

### Example 3: Recursor for Nat in foetus

\[
\begin{align*}
\text{Nat} & \equiv \text{Rec}\ X.1 + X \\
\text{O} & \equiv \text{fold}\text{(\text{inl}())} \\
\text{S}(v) & \equiv \text{fold}\text{(\text{inr}(v))} \\
\end{align*}
\]

\[
\text{R}^\sigma \equiv \text{rec}\text{R}^\sigma\rightarrow(\text{Nat} \rightarrow \sigma \rightarrow \sigma)\rightarrow\text{Nat} \rightarrow \sigma. \lambda f_0^\sigma. \lambda f_S^\text{Nat} \rightarrow \sigma. \lambda n^\text{Nat}.
\]

\[
\begin{align*}
\text{case}(&\text{unfold}(n), \\
\text{\text{\_1}.f_0}, \\
\text{n}^\text{Nat}. f_S n' (\text{R} f_0 f_S n'))
\end{align*}
\]
Example 4: addord in foetus

\[ \text{Ord} \equiv \text{Rec} X.(1 + X) + (\text{Nat} \to X) \]

\[ O \equiv \text{fold}([\text{inl}()]) \]
\[ S(v) \equiv \text{fold}([\text{inr}(v)]) \]
\[ \text{Lim}(f) \equiv \text{fold}(\text{inr}(f)) \]

\[ \text{addOrd} \equiv \text{rec addOrd} \to \text{Ord} \to \text{Ord} \to \text{Ord}. \lambda x. \text{case}(\text{unfold}(y), \]
\[ n \to \text{Ord}. \text{case}(n, \]
\[ 1. x, \]
\[ y \to \text{Ord}. S(\text{addOrd} x y') \]
\[ f \to \text{Nat} \to \text{Ord}. \text{Lim}(\lambda z. \text{addOrd} x (f z)) \]

Operational semantics

Closures \( \text{Cl}^\sigma \):

\[ \langle t^\sigma ; e \rangle \quad t \text{ term, } e \text{ environment} \]
\[ f^\sigma \to \sigma \@ u^\rho \quad f \text{ function value, } u \text{ argument value} \]

Evaluation relation \( \Downarrow^\sigma \subseteq \text{Cl}^\sigma \times \text{Val}^\sigma \):

- big step
- call-by-value
- fixed evaluation strategy
Semantics

Let $\tilde{V} \subseteq \text{Val}$. Define $[[\sigma(\tilde{X})]]_{\tilde{V}}$ inductively:

(Unit) $[1] := \{()\}$

(Var) $[X_n]_{\tilde{V}} := V_n$

(Sum) $[((\sigma + \tau)(\tilde{X}))]_{\tilde{V}} := \{\text{inl}(v) : v \in [[\sigma(\tilde{X})]]_{\tilde{V}}\} \cup \{\text{inr}(v) : v \in [[\tau(\tilde{X})]]_{\tilde{V}}\}$

(Arr) $[[\sigma \rightarrow \tau(\tilde{X})]]_{\tilde{V}} := \{f \in \text{Val}^{\sigma \rightarrow \tau(\tilde{X})} : \forall u \in [\sigma]. \exists v \in [[\tau(\tilde{X})]]_{\tilde{V}}. f@u \Downarrow v\}$

(Rec) $[[\text{Rec} \cdot \sigma(\tilde{X}, Y)]]_{\tilde{V}} := \text{lfp } F$, where we define $F$ as

$$F : \mathcal{P}(\text{Val}^{\text{Rec} \cdot \sigma(\tilde{X}, Y)}) \rightarrow \mathcal{P}(\text{Val}^{\text{Rec} \cdot \sigma(\tilde{X}, Y)})$$

$$W \mapsto \text{fold } \left( [[\sigma(\tilde{X}, Y)]_{\tilde{V}}, W \right)$$

Fixed-point

Let $(\mathcal{U}, \subseteq)$ be a complete lattice, $F : \mathcal{U} \rightarrow \mathcal{U}$ an operator. The least fixed-point $F = \text{lfp } F$ is characterized by:

$$(\text{ispfp}) \quad F(F) \subseteq F$$

$$(\text{ismpfp}) \quad \forall A \in \mathcal{U}. F(A) \subseteq A \rightarrow F \subseteq A$$

Monotonicity

$$\forall \sigma(X). A \subseteq B \rightarrow [[\sigma(X)]]_A \subseteq [[\sigma(X)]]_B$$

Proof: Induction on $\sigma$:

(Arr) Show: For all $f \in [[\sigma \rightarrow \tau(X)]]_A$ and $u \in [\sigma]$ there is a $v \in [[\tau(X)]]_B$ satisfying $f@u \Downarrow v$.

(Rec) Show: $[[\text{Rec} \cdot \sigma(\tilde{X}, Z)]]_A \subseteq [[\text{Rec} \cdot \sigma(\tilde{X}, Z)]]_B.$
Substitution

\[ \sigma(X)_{[\tau]} = \sigma(\tau) \]

For \( V \subseteq [\tau] \) we get

\[ \sigma(X)_V \subseteq \sigma(\tau) \]

---

Example 5: All numerals are good

\[ \text{Val}^{\text{Nat}} = \{(\text{fold} \circ \text{inr})^n(\text{inl}()) : n \in \mathbb{N}\} = [\text{Nat}] \]

Show: \( \text{Val}^{\text{Nat}} \) is smallest fixed-point of

\[ \mathcal{F} : \mathcal{P}(\text{Val}^{\text{Nat}}) \rightarrow \mathcal{P}(\text{Val}^{\text{Nat}}) \]

\[ \mathcal{F}(W) := \{\text{fold}(v) : v \in [1 + X]_W\} \]

\[ = \{\text{fold}(\text{inl}()), \text{fold}(\text{inr}(v)) : v \in W\} \]

We must only show (ismfp):

(i) \( \bigcup_{n \in \mathbb{N}} \mathcal{F}^n(\emptyset) \subseteq W \)

(ii) \( \text{Val}^{\text{Nat}} \subseteq \bigcup_{n \in \mathbb{N}} \mathcal{F}^n(\emptyset) \)
Structural ordering

Idea: Values are trees, \(<\) is subtree relation.

Definition of \(\leq_{\sigma,\tau} \subseteq [\sigma] \times [\tau]\):

1. (lefl) \(v \leq_{\sigma,\sigma} v\)
2. (left) \(w <_{\sigma,\tau} v, w \leq_{\sigma,\tau} v\)
3. (ltinl) \(w \leq_{\rho,\sigma} v, w <_{\rho,\sigma+\tau} \text{inl}(v)\)
4. (ltinr) \(w \leq_{\rho,\sigma} v, w <_{\rho,\sigma+\tau} \text{inr}(v)\)
5. (ltarr) \(\exists v \in \text{CoDom}(f). w <_{\rho,\tau} \text{inl}(v)\supseteq \text{inr}(v)\)
6. (ltfold) \(w <_{\sigma,\tau} \text{fold}(v)\supseteq \text{rec}(X.\tau(\text{X}))\)

Wellfoundedness

Def. of the accessible set \(\text{Acc}^{\sigma} \subseteq [\sigma]\):

1. (acc) \(\forall \tau, [\tau] \ni w < v, w \in \text{Acc}^{\tau} \Rightarrow v \in \text{Acc}^{\tau}\)

All semantic values are accessible.

\(\forall \sigma(X), \rho, \sigma(X) \subseteq \text{Acc}^{\rho}\)

Proof by induction on generation of \(\sigma\).

1. (Arr) Let \(f \in [\sigma \to \tau(X)]\). By monotonicity \(f \in [\sigma \to \tau(\rho)]\), and from the induction hypothesis \(\text{CoDom}(f) \subseteq [\tau(X)]\). We infer \(f \in \text{Acc}^{\sigma \to \tau(\rho)}\) (we need a small lemma).

2. (Rec) Show \([\text{Rec} \sigma(X), Y, \sigma(\text{Rec}Y)]\). We use the induction hypothesis \([\sigma(X, Y)]\) and the fixed-point properties.
Structural recursive terms

\[
\text{SR}^{\sigma \rightarrow \tau}[\Gamma] := \{ \text{rec } g.t \in \text{Tm}^{\sigma \rightarrow \tau}[\Gamma] : \forall e \in [\Gamma], v \in [\sigma]. \\
(\forall [\sigma] \ni w < v. (\text{rec } g.t; e)@w \downarrow) \rightarrow (\text{rec } g.t; e)@v \downarrow \}
\]

Induction principle for wellfounded sets:

\[
(\text{accind}) \quad \forall v \in [\sigma], (\forall [\sigma] \ni w < v. P(w)) \rightarrow P(v) \\
\forall v \in \text{Acc}^\sigma . P(v)
\]

All structural recursive terms are good:

\[
t \in \text{SR}^{\sigma \rightarrow \tau}[\Gamma], e \in [\Gamma] \rightarrow (t; e) \in [\sigma \rightarrow \tau]
\]

Example 6: addord is structural recursive

Recursive calls:

\[
\ldots \text{S}(\text{addOrd } x' y) \ldots
\]

\[
\begin{align*}
& v' \leq v' \quad \text{lerfl} \\
& v' \leq \text{inr}(v') \quad \text{ltinr} \\
& v' \leq \text{inr}(v') \quad \text{l elt} \\
& v' \leq \text{inr}(v') \quad \text{ltlnl} \\
& v' < \text{S}(v') \equiv \text{fold}(\text{inl}(\text{inr}(v'))) \quad \text{ltfold}
\end{align*}
\]

\[
\ldots \text{Lim}(\lambda z^{\text{Nat}}. \text{addOrd } (f z) y) \ldots
\]

\[
\begin{align*}
& w \leq w \quad \text{lerfl} \\
& \exists v' \in \text{CoDom}(f). w \leq v' \quad \exists \text{learr} \\
& w \leq f \quad \text{ltinr} \\
& w < \text{Lim}(f) \equiv \text{fold}(\text{inr}(f)) \quad \text{ltfold}
\end{align*}
\]
Normalization

Define the good terms $\text{Tm}^\sigma[\Gamma] \subset \text{Tm}^\sigma[\Gamma]$ inductively in the same way as $\text{Tm}$ with the exception

\[
\begin{align*}
\text{(REC)} & \quad \frac{t \in \text{Tm}^\sigma[\Gamma], g^\sigma \rightarrow \tau \quad \text{rec } g.t \in \text{SR}^\sigma[\Gamma]}{\text{rec } g.t \in \text{Tm}^\sigma[\Gamma]}
\end{align*}
\]

Show normalization

\[\forall \sigma, \Gamma, t \in \text{Tm}^\sigma[\Gamma], e \in \llbracket \Gamma \rrbracket. \langle t; e \rangle \Downarrow\]

by induction on $t$ using the operational semantics.

Extensions and open questions

- positive types (?)
- polymorphic types \(\surd\)
- dependent types
- coinductive types