CS 671 Automated Reasoning

Universes and Set Types

Methodology for building types

- **Syntax:**
  - Define canonical type
  - Define canonical members of the type
  - Define noncanonical expressions corresponding to the type
- **Semantics**
  - Introduce evaluation rules for non-canonical expressions
  - Define type equality judgment for the type

    The typehood judgment is a special case of type equality
  - Define member equality judgment for canonical members

    The membership judgment is a special case of member equality

  Define judgments only in terms of the new expressions $\mapsto$ Consistency
- **Proof Theory**
  - Introduce proof rules that are consistent with the semantics

Methodology for defining proof rules

- **Type Formation rules:**
  - When are two types equal? (*type*`Equality`) $\Gamma \vdash S = T \in \mathbb{U}_j$
  - How to build the type? (*type*`Formation`) $\Gamma \vdash \mathbb{U}_j \;\lfloor \mathsf{ext}\ T \rfloor$
- **Canonical rules:**
  - When are two members equal? (*member*`Equality`) $\Gamma \vdash s = t \in T$
  - How to build members? (*member*`Formation`) $\Gamma \vdash T \;\lfloor \mathsf{ext}\ t \rfloor$
- **Noncanonical rules:**
  - When does a term inhabit a type? (*noncanonical*`Equality`) $\Gamma \vdash s = t \in T$
  - How to use a variable of the type (*type*`Elimination`) $\Gamma, x{:}S, \Delta \vdash T \;\lfloor \mathsf{ext}\ t \rfloor$
- **Computation rules:**
  - Reduction of redices in an equality (*noncanonical*`Reduce*`) $\Gamma \vdash \mathit{redex} = t \in T$
- **Special purpose rules**

Universes

- **Syntactical representation of typehood**
  - T type expressed as $T \in \mathbb{U}$ — $S{=}T$ expressed as $S{=}T \in \mathbb{U}$
- **Universes are object-level terms**
  - $\mathbb{U}$ is a type and a universe
  - Girard's Paradox: a theory with dependent types and $\mathbb{U} \in \mathbb{U}$ is inconsistent
  - $\mapsto$ *No single universe can capture the notion of typehood*
  - Typehood $\hat{=}$ cumulative hierarchy of universes $\mathbb{U}{=}\mathbb{U}_1 \overset{\in}{\subset} \mathbb{U}_2 \overset{\in}{\subset} \mathbb{U}_3 \overset{\in}{\subset} \ldots$

Syntax:

Canonical: $\mathbb{U}_j$

Noncanonical: —

Semantics:

- $\mathbb{U}_{j_1} = \mathbb{U}_{j_2}$ if $j_1{=}j_2$
- $\mathbb{U}_{j_1} = \mathbb{U}_{j_2}$ in $\mathbb{U}_j$ if $j_1{=}j_2{<}j$
- $S = T$ in $\mathbb{U}_j$ if … mimic semantics for $S = T$ as types…

See Appendix A.3.4 for further details

Subset Types: Hiding Computational Content

- **Representation of mathematical concept of subsets**
  - $\{x:S \mid T[x]\}$ formally similar to dependent product $x:S\times T[x]$

    ...but ...
  - Members are elements of $s\in S$, not pairs <s,t>
  - Only implicit evidence for $T[s]$ but no explicit proof component

---

Syntax:

Canonical: $\{x:S \mid T\}$, $\{S \mid T\}$

Noncanonical: —

Semantics:

· $\{x_1:S_1 \mid T_1\} = \{x_2:S_2 \mid T_2\}$ if $S_1=S_2$ and there are terms p_1, p_2 and a variable x, which occurs neither in T_1 nor in T_2, such that

p_1 in $\forall x:S_1.\ T_1[x/x_1] \Rightarrow T_2[x/x_2]$

and p_2 in $\forall x:S_1.\ T_2[x/x_2] \Rightarrow T_1[x/x_1]$. (violates separation principle)

· $s = t$ in $\{x:S \mid T\}$ if $\{x:S \mid T\}$ type,

$s{=}t$ in S, and there is some p in $T[s/x]$.

Subset Types: Proof theory

Proof rules must manage implicit information

– We "know" $T[s]$ if s in $\{x{:}S \mid T\}$

– We cannot use the proof term for $T[s]$ computationally

– Proof term for $T[s]$ must be available in non-computational proof parts

– Some refinement rules generate hidden assumptions

$$\Gamma,\ z{:}\{x{:}S \mid T\},\ \Delta \vdash C \quad \lfloor \texttt{ext}\ (\lambda y.t)\ z \rfloor$$
$$\textbf{by}\ \texttt{setElimination}\ i\ y\ v$$
$$\Gamma,\ z{:}\{x{:}S \mid T\},\ y{:}S,\ [\![v]\!]{:}T[y/x],\ \Delta[y/z] \vdash C[y/z] \quad \lfloor \texttt{ext}\ t \rfloor$$

– Hidden assumptions made visible by refinement rules with extract term Ax

See Appendix A.3.12 for further details