

A Guided Tour of a Twelf Proof

Matthew Fluet

fluet@cs.cornell.edu

Introduction

- The POPLMARK Challenge and Twelf solution has suggested that Twelf could be *the* way of formalizing programming language meta-theory, particularly type-safety proofs.
- Having been persuaded, I'm naturally led to try out Twelf for my own work, which has recently focused on applying substructural type systems to “tame” state and effects.

Introduction

- A type-safety proof for a substructural lambda calculus:
 - does not come close to a feature-complete language
 - does highlight some of the strengths and weaknesses of Twelf in this domain.
- Code and slides at:
<http://www.cs.cornell.edu/People/fluet/twelf>

The POPLMARK Challenge

<http://www.cis.upenn.edu/group/proj/plclub/mmm/>

Many proofs about programming languages are long, straightforward, and tedious, with just a few interesting cases. . . . Automated proof assistants offer the hope of significantly easing these problems.

We believe that the time is ripe to join the efforts of the two communities, bringing developers of automated proof assistants together with a large pool of eager potential clients – programming language designers and researchers.

The POPLMARK Challenge

- Problems from the basic metatheory of System $F_{<:}$:
 - 1a Transitivity of Subtyping
 - 1b Transitivity of Subtyping with Records
 - 2a Type Safety for Pure $F_{<:}$
 - 2b Type Safety for with Records and Pattern Matching
 - 3 Testing and Animating with Respect to the Semantics
- Paper proofs approximately 9 pages, allowing
Proof: By induction on the structure of T . □

The POPLMARK Challenge: Twelf Solution

- Problems from the basic metatheory of System F $<:$:

1a	Transitivity	(559 lines)
1b	with Records	(1227 lines)
2a	Type Safety	(745 lines)
2b	with Records and Pattern Matching	(4176 lines)
3	Testing and Animating	(n/a)

- Evaluation

drag B

accessibility C+

transparency *strong disagreement*

LF/Twelf Methodology

Slides cribbed from

- *Mechanizing Language Definitions*
Bob Harper

<http://www-2.cs.cmu.edu/~rwh/talks>

See also

- *How to Believe a Twelf Proof*
Bob Harper and Karl Crary

<http://www-2.cs.cmu.edu/~rwh/papers.htm#how>

Twelf is an implementation of the logical framework LF.

<http://www.twelf.org>

LF/Twelf Methodology

- Formalize language definition in LF.
- State meta-theorems relationally in LF.
- Use Twelf to prove “totality”.

- Remarkably, this approach works well both “in the small” and “in the large”!

Encoding

- Establish a compositional bijection between
 - objects of each syntactic category of object language
 - canonical forms of associated types of the LF lambda calculus
- “Compositional” means “commutes with substitution” (aka “natural”).

Encoding

- Here the syntactic categories include
 - abstract syntax, usually including binding and scoping conventions
 - typing derivations
 - evaluation derivations
- The latter two cases give rise to the slogan “judgements as types”.

Meta-Reasoning

- Adequacy ensures that we can reason about the object language by analyzing canonical forms of appropriate LF type.
 - Canonical forms are long $\beta\eta$ normal forms.
 - Structural induction, parallel and lexicographic extension to tuples.
- Applies to informal and formal reasoning!

Meta-Reasoning

- Twelf supports checking of proofs of Π_2 ($\forall\exists$) propositions over canonical forms in a specified class of contexts (world).
 - Enough for preservation, progress, ...
- These are totality assertions for a relation between inputs ($\forall/+$) and outputs ($\exists/-$)!
 - Polarity notation is an unfortunate relic.

Relational Meta-Theory

- Ask Twelf to verify the totality of the relation representing the theorem.
 - Specify the worlds to consider.
 - Specify mode of the relation.
 - Specify induction principle to use.
- Checks that all cases are covered, and induction is used appropriately.

Some Examples

- We use Twelf daily in our work at CMU!
 - TALT, a full-scale certified object code format with a generic safety policy.
 - Compilation through closure conversion, type safety for Classical S5 for dist'd prog'ing.
 - First, and only, solution to the POPLmark Challenge to verify meta-theory of $F_{<:}$.
 - Type safety (almost), regularity for HS semantics of Standard ML.

Who I Am (and Who I Am Not)

- A Twelf convert, not a Twelf zealot
 - I see warts, not beauty marks
- Read the *Twelf User's Guide*, not Schuermann's thesis
 - I understand (some of) the how, not (all of) the why
- Multiple type-safety proofs under my belt
 - I know how the paper proof should look

A Substructural Lambda Calculus

Based on

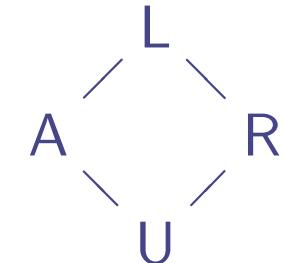
- *Substructural Type Systems*
David Walker
Advanced Topics in Type and Programming Languages

Want the type of a value to describe

- its “shape” (e.g., function, pair)
- the number of times it may be used (e.g., applied, projected)

Syntax

Qualifiers $q \in Quals = \{U, R, A, L\}$



Locations $l \in Locs$

Expressions $e ::=$

$l \mid x \mid {}^q \lambda x : \tau . e \mid e_1 e_2 \mid$

${}^q \langle \rangle \mid \text{let } \langle \rangle = e_1 \text{ in } e_2 \mid {}^q \langle e_1, e_2 \rangle \mid \text{let } \langle x_1, x_2 \rangle = e_1 \text{ in } e_2 \mid$

${}^q \langle \rangle \mid {}^q \langle e_1, e_2 \rangle \mid \text{fst } e \mid \text{snd } e$

$\text{abort } e \mid {}^q \text{inl } e \mid {}^q \text{inr } e \mid \text{case } e_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_r$

Syntax

PreValues $w ::= \lambda x:\tau. e \mid \langle \rangle \mid \langle l_1, l_2 \rangle \mid (\emptyset) \mid (e_1, e_2) \mid \text{inl } l \mid \text{inr } l$

Values $v ::= {}^q w$

Stores $\sigma ::= \emptyset \mid \sigma, l \mapsto (v, i)$

Dynamics: Stores

$$\frac{l \notin \text{dom}(\sigma)}{(\sigma; v) \xrightarrow{\text{alloc}} (\sigma, l \mapsto (v, 0); l)}$$

Dynamics: Stores

$$\frac{q \sqsubseteq R}{(\sigma, l \mapsto (^q w, i); l) \xrightarrow{\text{fetch}} (\sigma, l \mapsto (^q w, i + 1); ^q w)}$$

$$\frac{A \sqsubseteq q}{(\sigma, l \mapsto (^q w, i); l) \xrightarrow{\text{fetch}} (\sigma; ^q w)}$$

$$\frac{l \neq l' \quad (\sigma_1; l) \xrightarrow{\text{fetch}} (\sigma_2; v)}{(\sigma_1, l' \mapsto (v', i'); l) \xrightarrow{\text{fetch}} (\sigma_2, l' \mapsto (v', i'); v)}$$

Dynamics: Expressions

$$\frac{(\sigma; {}^q \lambda x:\tau. e) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q \lambda x:\tau. e) \longmapsto (\sigma', l)}$$

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{(\sigma, e_1 e_2) \longmapsto (\sigma', e'_1 e_2)} \quad \frac{(\sigma, e_2) \longmapsto (\sigma', e'_2)}{(\sigma, l_1 e_2) \longmapsto (\sigma', l_1 e'_2)}$$

$$\frac{(\sigma; l_1) \xrightarrow{\text{fetch}} (\sigma'; {}^q \lambda x:\tau. e)}{(\sigma, l_1 l_2) \longmapsto (\sigma', e[l_2/x])}$$

Dynamics: Expressions

$$\frac{(\sigma; {}^q \langle \rangle) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q \langle \rangle) \longmapsto (\sigma', l)}$$

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{(\sigma, \text{let } \langle \rangle = e_1 \text{ in } e_2) \longmapsto (\sigma', \text{let } \langle \rangle = e'_1 \text{ in } e_2)}$$

$$\frac{(\sigma; l_1) \xrightarrow{\text{fetch}} (\sigma'; {}^q \langle \rangle)}{(\sigma, \text{let } \langle \rangle = l_1 \text{ in } e_2) \longmapsto (\sigma', e_2)}$$

Dynamics: Expressions

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{(\sigma, {}^q\langle e_1, e_2 \rangle) \longmapsto (\sigma', {}^q\langle e'_1, e_2 \rangle)}$$

$$\frac{(\sigma, e_2) \longmapsto (\sigma', e'_2)}{(\sigma, {}^q\langle l_1, e_2 \rangle) \longmapsto (\sigma', {}^q\langle l_1, e'_2 \rangle)}$$

$$\frac{(\sigma; {}^q\langle l_1, l_2 \rangle) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q\langle l_1, l_2 \rangle) \longmapsto (\sigma', l)}$$

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{(\sigma, \text{let } \langle x, y \rangle = e_1 \text{ in } e_2) \longmapsto (\sigma', \text{let } \langle x, y \rangle = e'_1 \text{ in } e_2)}$$

$$\frac{(\sigma; l_1) \xrightarrow{\text{fetch}} (\sigma'; {}^q\langle l_x, l_y \rangle)}{(\sigma, \text{let } \langle x, y \rangle = l_1 \text{ in } e_2) \longmapsto (\sigma', e_2[l_x/x][l_y/y])}$$

Dynamics: Expressions

$$\frac{(\sigma; {}^q \langle \rangle) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q \langle \rangle) \longmapsto (\sigma', l)}$$

Dynamics: Expressions

$$\frac{(\sigma; {}^q \langle e_1, e_2 \rangle) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q \langle e_1, e_2 \rangle) \longmapsto (\sigma', l)}$$

$$\frac{(\sigma, e) \longmapsto (\sigma', e')}{(\sigma, \mathbf{fst} e) \longmapsto (\sigma', \mathbf{fst} e')}$$

$$\frac{(\sigma; l) \xrightarrow{\text{fetch}} (\sigma'; {}^q \langle e_1, e_2 \rangle)}{(\sigma, \mathbf{fst} l) \longmapsto (\sigma', e_1)}$$

$$\frac{(\sigma, e) \longmapsto (\sigma', e')}{(\sigma, \mathbf{snd} e) \longmapsto (\sigma', \mathbf{snd} e')}$$

$$\frac{(\sigma; l) \xrightarrow{\text{fetch}} (\sigma'; {}^q \langle e_1, e_2 \rangle)}{(\sigma, \mathbf{snd} l) \longmapsto (\sigma', e_2)}$$

Dynamics: Expressions

$$\frac{(\sigma, e) \longmapsto (\sigma', e')}{(\sigma, \text{abort } e) \longmapsto (\sigma', \text{abort } e')}$$

Dynamics: Expressions

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{(\sigma, {}^q\text{inl } e_1) \longmapsto (\sigma', {}^q\text{inl } e'_1)}$$

$$\frac{(\sigma; {}^q\text{inl } l_1) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q\text{inl } l_1) \longmapsto (\sigma', l)}$$

$$\frac{(\sigma, e_2) \longmapsto (\sigma', e'_2)}{(\sigma, {}^q\text{inr } e_2) \longmapsto (\sigma', {}^q\text{inr } e'_2)}$$

$$\frac{(\sigma; {}^q\text{inr } l_2) \xrightarrow{\text{alloc}} (\sigma'; l)}{(\sigma, {}^q\text{inr } l_2) \longmapsto (\sigma', l)}$$

Dynamics: Expressions

$$\frac{(\sigma, e_1) \longmapsto (\sigma', e'_1)}{} \quad \text{(rule for } \text{case}$$

$$\frac{(\sigma, \text{case } e_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_y) \longmapsto (\sigma', \text{case } e_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_y)}{}$$

$$(\sigma; l_1) \xrightarrow{\text{fetch}} (\sigma'; {}^q\text{inl } l_x)$$

$$\frac{(\sigma, \text{case } l_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_y) \longmapsto (\sigma', e_l[l_x/x])}{}$$

$$(\sigma; l_1) \xrightarrow{\text{fetch}} (\sigma'; {}^q\text{inr } l_x)$$

$$\frac{(\sigma, \text{case } l_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_y) \longmapsto (\sigma', e_r[l_y/y])}{}$$

Dynamics: Multi-step

$$\overline{(\sigma, e) \longmapsto^* (\sigma, e)}$$

$$\frac{(\sigma_1, e_1) \longmapsto^* (\sigma_2, e_2) \quad (\sigma_2, e_2) \longmapsto^* (\sigma_3, e_3)}{(\sigma_1, e_1) \longmapsto^* (\sigma_3, e_3)}$$

$$\frac{(\sigma_1, e_1) \longmapsto (\sigma_2, e_2)}{(\sigma_1, e_1) \longmapsto^* (\sigma_2, e_2)}$$

Syntax

<i>PreTypes</i>	$\bar{\tau} ::= \tau_1 \multimap \tau_2 \mid \mathbf{1}_\otimes \mid \tau_1 \otimes \tau_2 \mid \mathbf{1}_\circledast \mid \tau_1 \circledast \tau_2 \mid \mathbf{0} \mid \tau_1 \oplus \tau_2$
<i>Types</i>	$\tau ::= {}^q\bar{\tau}$
<i>Store Types</i>	$\Sigma ::= \bullet \mid \Sigma, l \mapsto \tau$
<i>Contexts</i>	$\Gamma ::= \bullet \mid \Gamma, x:\tau$

Statics

$$q \preceq q'$$

$$\tau \preceq q'$$

$$\Gamma \preceq q'$$

$$\Sigma \preceq q'$$

$$\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma$$

$$\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma$$

$$\Sigma \vdash l : \tau$$

$$\Gamma; \Sigma \vdash e : \tau$$

$$\Sigma \vdash v : \tau$$

$$\vdash \sigma : \Sigma$$

Statics ($q \preceq q'$)

$$\frac{}{\mathsf{U} \preceq q}$$

$$\frac{q_1 \sqsubseteq q_2}{q_1 \preceq q_2}$$

$$\frac{}{q \preceq \mathsf{L}}$$

$$\frac{}{q \preceq q}$$

$$\frac{q_1 \preceq q_2 \quad q_2 \preceq q_3}{q_1 \preceq q_3}$$

Statics ($\tau \preceq q'$ and $\Gamma \preceq q'$ and $\Sigma \preceq q'$)

$$\frac{}{\tau \preceq \mathsf{L}}$$

$$\frac{q \preceq q'}{^q\bar{\tau} \preceq q'}$$

$$\frac{}{\bullet \preceq q'}$$

$$\frac{\Gamma \preceq q' \quad \tau \preceq q'}{\Gamma, x:\tau \preceq q'}$$

$$\frac{}{\emptyset \preceq q'}$$

$$\frac{\Sigma \preceq q' \quad \tau \preceq q'}{\Sigma, l \mapsto \tau \preceq q'}$$

Statics ($\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma$)

$$\frac{}{\bullet \boxdot \bullet \rightsquigarrow \bullet}$$

$$\frac{\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma}{\Gamma_1, x:\tau \boxdot \Gamma_2 \rightsquigarrow \Gamma, x:\tau}$$

$$\frac{\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma}{\Gamma_1 \boxdot \Gamma_2, x:\tau \rightsquigarrow \Gamma, x:\tau}$$

$$\frac{\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma \quad \tau \preceq R}{\Gamma_1, x:\tau \boxdot \Gamma_2, x:\tau \rightsquigarrow \Gamma, x:\tau}$$

Statics ($\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma$)

$$\overline{\emptyset \odot \emptyset \rightsquigarrow \emptyset}$$

$$\frac{\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma}{\Sigma_1, l \mapsto \tau \odot \Sigma_2 \rightsquigarrow \Sigma, l \mapsto \tau} \qquad \frac{\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma}{\Sigma_1 \odot \Sigma_2, l \mapsto \tau \rightsquigarrow \Sigma, l \mapsto \tau}$$

$$\frac{\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Gamma \quad \tau \preceq R}{\Sigma_1, l \mapsto \tau \odot \Sigma_2, l \mapsto \tau \rightsquigarrow \Sigma, l \mapsto \tau}$$

Statics ($\Gamma; \Sigma \vdash l : \tau$)

$$\frac{}{\bullet; \emptyset, l \mapsto \tau \vdash x : \tau}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{}{\bullet, x:\tau; \emptyset \vdash x : \tau}$$

$$\frac{\begin{array}{c} \Gamma \preceq q \quad \Sigma \preceq q \\ \Gamma, x:\tau_x; \Sigma \vdash e : \tau \end{array}}{\Gamma; \Sigma \vdash {}^q\lambda x:\tau_x. e : {}^q(\tau_x \multimap \tau)}$$

$$\frac{\begin{array}{c} \Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma \quad \Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \\ \Gamma_1; \Sigma_1 \vdash e_1 : {}^q(\tau_x \multimap \tau) \quad \Gamma_2; \Sigma_2 \vdash e_2 : \tau_x \end{array}}{\Gamma; \Sigma \vdash e_1 e_2 : \tau}$$

Statics $(\Gamma; \Sigma \vdash e : \tau)$

$$\frac{}{\bullet; \emptyset \vdash {}^q \langle \rangle : {}^q \mathbf{1}_\otimes}$$

$$\frac{\begin{array}{c} \Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma & \Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \\ \Gamma_1; \Sigma_1 \vdash e_1 : {}^q \mathbf{1}_\otimes & \Gamma_2; \Sigma_2 \vdash e_2 : \tau \end{array}}{\Gamma; \Sigma \vdash \text{let } \langle \rangle = e_1 \text{ in } e_2 : \tau}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{\begin{array}{c} \Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma & \Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \\ \Gamma_1; \Sigma_1 \vdash e_1 : \tau_1 & \tau_1 \preceq q & \Gamma_2; \Sigma_2 \vdash e_2 : \tau_2 & \tau_2 \preceq q \end{array}}{\Gamma; \Sigma \vdash {}^q\langle e_1, e_2 \rangle : {}^q(\tau_1 \otimes \tau_2)}$$

$$\frac{\begin{array}{c} \Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma & \Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \\ \Gamma_1; \Sigma_1 \vdash e_1 : {}^q(\tau_x \otimes \tau_y) & \Gamma_2, x:\tau_x, y:\tau_y; \Sigma_2 \vdash e_2 : \tau \end{array}}{\Gamma; \Sigma \vdash \text{let } \langle x, y \rangle = e_1 \text{ in } e_2 : \tau}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{\Gamma \preceq q \quad \Sigma \preceq q}{\Gamma; \Sigma \vdash {}^q(\emptyset) : {}^q \mathbf{1}_\otimes}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{\Gamma \preceq q \quad \Sigma \preceq q \quad \Gamma; \Sigma \vdash e_1 : \tau_1 \quad \Gamma; \Sigma \vdash e_2 : \tau_2}{\Gamma; \Sigma \vdash {}^q(\!(e_1, e_2)\!) : {}^q(\tau_1 \circledast \tau_2)}$$
$$\frac{\Gamma; \Sigma \vdash e : {}^q(\tau_1 \circledast \tau_2)}{\Gamma; \Sigma \vdash \mathbf{fst}\, e : \tau_1}$$
$$\frac{\Gamma; \Sigma \vdash e : {}^q(\tau_1 \circledast \tau_2)}{\Gamma; \Sigma \vdash \mathbf{snd}\, e : \tau_2}$$

Statics $(\Gamma; \Sigma \vdash e : \tau)$

$$\frac{\Gamma; \Sigma \vdash e : {}^q \mathbf{0}}{\Gamma; \Sigma \vdash \text{abort } e : \tau}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{\Gamma; \Sigma \vdash e_1 : \tau_1 \quad \tau_1 \preceq q}{\Gamma; \Sigma \vdash {}^q\text{inl } e_1 : {}^q(\tau_1 \oplus \tau_2)}$$

$$\frac{\Gamma; \Sigma \vdash e_2 : \tau_2 \quad \tau_2 \preceq q}{\Gamma; \Sigma \vdash {}^q\text{inl } e_2 : {}^q(\tau_1 \oplus \tau_2)}$$

$$\frac{\begin{array}{c} \Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma \quad \Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \\ \Gamma_1; \Sigma_1 \vdash e_1 : {}^q(\tau_1 \oplus \tau_2) \\ \Gamma_2, x:\tau_1 \vdash e_l \vdash \tau \quad \Gamma_2, y:\tau_2 \vdash e_r \vdash \tau \end{array}}{\Gamma; \Sigma \vdash \text{case } e_1 \text{ of inl } x \Rightarrow e_l \parallel \text{inr } y \Rightarrow e_r : \tau}$$

Statics ($\Gamma; \Sigma \vdash e : \tau$)

$$\frac{\Sigma \vdash l : \tau}{\bullet; \Sigma \vdash l : \tau}$$

$$\frac{\Gamma_1 \boxdot \Gamma_2 \rightsquigarrow \Gamma \quad \Gamma_1 \preceq A \quad \Gamma_2; \Sigma \vdash e : \tau}{\Gamma; \Sigma \vdash e : \tau}$$

$$\frac{\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \quad \Sigma_1 \preceq A \quad \Gamma; \Sigma_2 \vdash e : \tau}{\Gamma; \Sigma \vdash e : \tau}$$

Statics ($\Sigma \vdash v : \tau$)

$$\frac{\Sigma \preceq q \quad \bullet, x:\tau_x; \Sigma \vdash e : \tau}{\Sigma \vdash {}^q\lambda x:\tau_x. e : {}^q(\tau_x \multimap \tau)}$$

$$\frac{}{\emptyset \vdash {}^q\langle\rangle : {}^q\mathbf{1}_\otimes}$$

$$\frac{\Sigma_1 \odot \Sigma_2 \rightsquigarrow \Sigma \quad \Sigma_1 \vdash l_1 : \tau_1 \quad \Sigma_2 \vdash l_2 : \tau_2}{\Sigma \vdash {}^q\langle l_1, l_2 \rangle : {}^q(\tau_1 \otimes \tau_2)}$$

$$\frac{\Sigma \preceq q}{\Sigma \vdash {}^q\langle\rangle : {}^q\mathbf{1}_\otimes}$$

$$\frac{\Sigma \preceq q \quad \bullet; \Sigma \vdash e_1 : \tau_1 \quad \bullet; \Sigma \vdash e_2 : \tau_2}{\Sigma \vdash {}^q\langle e_1, e_2 \rangle : {}^q(\tau_1 \circledast \tau_2)}$$

$$\frac{\Sigma \vdash l : \tau_1 \quad \tau_1 \preceq q}{\Sigma \vdash {}^q\text{inl } l : {}^q(\tau_1 \oplus \tau_2)}$$

$$\frac{\Sigma \vdash l : \tau_2 \quad \tau_2 \preceq q}{\Sigma \vdash {}^q\text{inr } l : {}^q(\tau_1 \oplus \tau_2)}$$

Statics ($\vdash \sigma : \Sigma$)

$$\frac{}{\vdash \emptyset : \emptyset} \quad \frac{\vdash \sigma : \Sigma_\star \quad \Sigma_v \odot \Sigma \rightsquigarrow \Sigma_\star \quad \Sigma_v \vdash v : \tau}{\vdash \sigma, l \mapsto (v, i) : \Sigma, l \mapsto \tau}$$

$$\frac{\vdash \sigma : \Sigma \quad q \preceq A}{\vdash \sigma, l \mapsto (^q w, i) : \Sigma}$$

$$\frac{\vdash \sigma : \Sigma \quad q \preceq R \quad i > 0}{\vdash \sigma, l \mapsto (^q w, i) : \Sigma}$$

Type Safety

Theorem 1 (Preservation)

*If $(\sigma_1, e_1) \longmapsto (\sigma_2, e_2)$ and $\vdash \sigma_1 : \Sigma_1$ and $\bullet; \Sigma_1 \vdash e_1 : \tau$,
then there exists Σ_2 such that $\vdash \sigma_2 : \Sigma_2$ and $\bullet; \Sigma_2 \vdash e_2 : \tau$.*

Theorem 2 (Progress)

*If $\vdash \sigma_1 : \Sigma_1$ and $\bullet; \Sigma_1 \vdash e_1 : \tau$,
then either there exists l such that $e_1 \equiv l$
or there exists σ_2 and e_2 such that $(\sigma_1, e_1) \longmapsto (\sigma_2, e_2)$.*

Theorem 3 (Safety)

If $\vdash \sigma_1 : \Sigma_1$ and $\bullet; \Sigma_1 \vdash e_1 : \tau$ and $(\sigma_1, e_1) \longmapsto^ (\sigma_2, e_2)$,
then either there exists l such that $e_2 \equiv l$
or there exists σ_3 and e_3 such that $(\sigma_2, e_2) \longmapsto (\sigma_3, e_3)$.*

Twelf Proof: Statistics

- 32 files
- 5432 lines
- 44 “relations”
- 121 “theorems”

Twelf Proof: Relations

- 44 “relations”
 - 10 syntactic classes (but not Γ)
 - 12 judgements (but not $\Gamma_1 \sqdot \Gamma_2 \rightsquigarrow \Gamma$ or $\Gamma \preceq q$)
 - 22 supporting relations
 - 1 empty relation (falsity)
 - 5 devoted to well-formedness of σ and Σ as finite maps

Twelf Proof: Theorems

- 121 “theorems”
 - 83 by direct proof
 - 36 by (simple) structural induction
 - 2 by simultaneous structural induction
 - 57 cases by contradiction

Twelf Proof: Challenges

- contexts
 - LF/Twelf admits all the structural rules
 - HOAS makes context implicit
- store and store typing
 - encoding a partial map from (alpha-varying) locations to values (which may have free occurrences of those locations) or types is awkward.
 - stores and store typings should not contain multiple entries for the same location
- Twelfisms