Institutions - Part 1

Liam O'Reilly

09.05.07

Outline

Why Do We Need Institutions?

- There are many different logics in the world, for instance: EL, FOL, HOL, $SubPCFOL^{=}$, temporal logic, Horn clause logic, etc.
- Each program / prover tends to use its own logic.
- Many general results are actually completely independent of what logic system is used.

Institutions allow:

- Translation of sentences from logic to logic whilst preserving soundness.
- Forces us to write down logics in a standard way.
- Allows us to use tools from one logic on another logic.

An institution captures how truth can be preserved under change of symbols.

Outline

Definition - Institutions

Definition

An institution is a quadruple $\langle \textbf{SIGN}, \textbf{gram}, \textbf{mod}, \models \rangle$ where:

- **SIGN** is a category.
- $\textbf{gram} : \textbf{SIGN} \rightarrow \textbf{SET}$ is a functor.
- $\textbf{mod} : \textbf{SIGN}^{\textbf{op}} \rightarrow \textbf{CAT}$ is a functor.
- For every $\Sigma : \textbf{SIGN}$, $\models_\Sigma : \textbf{mod}(\Sigma) \times \textbf{gram}(\Sigma)$ which satisfies the satisfaction condition: for every $\sigma : \Sigma \rightarrow \Sigma', p \in \textbf{gram}(\Sigma)$ and $M' \in \textbf{mod}(\Sigma')$, $\textbf{mod}(\sigma)(M') \models_\Sigma p$ iff $M' \models_{\Sigma'} \textbf{gram}(\sigma)(p)$.

Outline

Signatures for Many-Sorted Equational Logic

A signature $\Sigma = (S, \Omega)$ is a pair of sets, where

- S is a set of sorts
- Ω is a set of total functions symbols, of the form $n : s_1 \times \ldots \times s_k \to s$ with $s_1, \ldots, s_k, s \in S$ and $k \geqslant 0$.

Signature Morphisms for Many-Sorted Equational Logic

Given two signatures $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$, a signature morphism $\sigma : \Sigma \rightarrow \Sigma'$ is a pair $(\sigma_S, \sigma_\Omega)$ where

- $\sigma_S : S \rightarrow S'$
- $\sigma_\Omega : \Omega \rightarrow \Omega'$

such that for each function symbol $n : s_1 \times \ldots s_k \rightarrow s \in \Omega, k \geq 0$, there exists a function name m with $\sigma_\Omega(n : s_1 \times \ldots s_k \rightarrow s) = (m : \sigma_S(s_1) \times \ldots \sigma_S(s_k) \rightarrow \sigma_S(s))$.

Models for Many-Sorted Equational Logic

Given a signature $\Sigma = (S, \Omega)$ a total algebra(model) for Σ assigns :

- A carrier set $A(s)$ to each sort $s \in S$.
- A total function $A(n : s_1 \times s_k \rightarrow s) : A(s_1) \times \ldots \times A(s_k) \rightarrow A(s)$ to each operation $(n : s_1 \times \ldots \times s_k \rightarrow s) \in \Omega, k \geq 0$.

Models Morphisms for Many-Sorted Equational Logic

Given two models $A, B \in \mathbf{mod}(\Sigma)$, a model morphism $h : A \to B$ is a family $(h_s)_{s \in S}$ of functions $h_s : A(s) \to B(s)$ such that for any function $f \in \Omega$ say $f = (n : s_1 \times \ldots \times s_k \to s), k \geq 0$, the following condition holds:

$$h_s(A(f)(a_1, \ldots, a_k)) = B(f)(h_{s1}(a_1), \ldots, h_{sk}(a_k))$$

for all $(a_1, \ldots, a_k) \in A(s_1) \times \ldots \times A(s_k)$.

Reducts

Given two signatures $\Sigma = (S, \Omega)$ and $\Sigma' = (S', \Omega')$, and a signature morphism $\sigma : \Sigma \to \Sigma'$ then for a Σ'-algebra A' the σ-reduct of A' is defined by:

- $(A' \mid_\sigma)(s) = A'(\sigma(s))$ for all $s \in S$
- $(A' \mid_\sigma)(f) = A'(\sigma(f))$ for all $f \in \Omega$

Terms in Many-Sorted Equational Logic

Given a signature $\Sigma = (S, \Omega)$ and a family of variables $X = (X_s)_{s \in S}$ of disjoint infinite sets, then $T_{\Sigma(X),s}$ is defined by

1. $X_s \subseteq T_{\Sigma(X),s}$,
2. if $n :\rightarrow s$ is an operation of Ω then $n \in T_{\Sigma(X),s}$,
3. if $n : s_1 \times \ldots \times s_k \rightarrow s, k \geqslant 1$ is an operation of Ω and if $t_i \in T_{\Sigma(X),s_i}$, for $1 \leqslant i \leqslant k$, then $n(t_1, \ldots, t_k) \in T_{\Sigma(X),s}$.

Sentences Many-Sorted Equational Logic

For each signature Σ the set of formulae of EL is

$$\mathbf{gram}(\Sigma) = \{\forall X.t = u \mid t, u \in T_{\Sigma(X),s}\}$$

Translation of Sentences

Translation of Variables

Given $\sigma : \Sigma \rightarrow \Sigma'$ the variable translation is defined as

$$\sigma((X_s)_{s \in S}) = ((\bigcup_{\sigma(s)=s'} X_s)_{s' \in S'})$$

Translation of Terms

Given $\sigma : \Sigma \rightarrow \Sigma'$ the term translation is defined as

$$\sigma(x : s) = x : \sigma_S(s)$$

$$\sigma(f(t_1, \ldots, t_k)) = \sigma_\Omega(f)(\sigma_S(t_1), \ldots, \sigma_S(t_k))$$

Translation of Sentences

Given $\sigma : \Sigma \rightarrow \Sigma'$ the sentence translation is defined as

$$\sigma(\forall X.t = u) = \forall \sigma(X).\sigma(t) = \sigma(u)$$

Assignments

Given $\Sigma = (S, \Omega)$ then an assignment of X for A is a family $\alpha = (\alpha_s)_{s \in S}$ of functions $\alpha_s : X_s \to A(s)$.

We can just write $\alpha : X \to A$.

Evaluation of Terms

Given a signature $\Sigma = (S, \Omega)$, a Σ-Algebra A, a set of variables X, a term $t \in T_{\Sigma(X)}$ and an assignment $\alpha : X \rightarrow A$ then $A(\alpha)(t)$ is defined by:

1. $A(\alpha)(t) = \alpha_s(x)$ if $t = x$ with $x \in X_s, s \in S$,
2. $A(\alpha)(t) = A(w)$ if $t = n$ and $w = (n :\rightarrow s) \in \Omega$,
3. $A(\alpha)(t) = A(w)(A(\alpha)(t_1), \ldots, A(\alpha)(t_k))$
   if $t = n(t_1, \ldots, t_k)$,
   $w = (n : s_1, \ldots, s_k \rightarrow s) \in \Omega$,
   $k \geqslant 1$ and
   $t_i \in T_{\Sigma(X), s_i}, 1 \leqslant i \leqslant k$.

The Satisfaction Relation for Many-Sorted Equational Logic

Let Σ be a signature.

$$A \models_\Sigma \forall X.t = u$$

:iff

for all assignments $\alpha : X \rightarrow A$, $A(\alpha)(t) = A(\alpha)(u)$

for each Σ-Algebra A and
for each equation $\forall X.t = u \in EL(\Sigma)$.

Outline

CASL Signatures

A many-sorted signature $\Sigma = (S, TF, PF, P)$ consists of:

- A set S of sort symbols.
- Sets $TF_{w,s}$ and $PF_{w,s}$ of total function symbols and partial function symbols such that $TF_{w,s} \cap PF_{w,s} = \emptyset$, for each function profile (w, s) consisting of a sequence or argument sorts $w \in S^*$ and a result sort $s \in S$ (constants are treated as functions with no arguments).
- A set P_w of predicate symbols for each predicate profile consisting of a sequence of argument sorts $w \in S^*$.

CASL Signatures Morphisms

A many-sorted signature morphism $\sigma : (S, TF, PF, P) \rightarrow (S', TF', PF', P')$ consists of:

- A mapping from S to S'.
- For each $w \in S^*, s \in S$, a mapping between the corresponding sets of functions, resp. predicate symbols. A partial function symbol may be mapped to a total function symbol, but not vice versa.

CASL Models

For a many sorted signature $\Sigma = (S, TF, PF, P)$ a many-sorted model $M \in \textbf{mod}(\Sigma)$ is a many-sorted first-order structure consisting of a many-sorted partial algebra:

- A non-empty carrier set s^M for each sort $s \in S$ (let w^M denote the Cartesian product $s_1^M \times \ldots \times s_n^M$ where $w = s_1 \ldots s_n$).
- A partial function f^M from w^M to s^M for each function symbol $f \in TF_{w,s}$ or $f \in PF_{w,s}$, the function being required to be total in the former case.
- A predicate $p^M \subseteq w^M$ for each predicate symbol $p \in P_w$.

CASL Model Morphisms

A (weak) many-sorted homomorphism h from M_1 to M_2, with $M_1, M_2 \in \textbf{mod}(S, TF, PF, P)$, consist of a function $h_s : s^{M_1} \to s^{M_2}$ for each $s \in S$ preserving not only the values of functions but also their definedness, and preserving the truth of predicates.
Any signature morphism $\sigma : \Sigma \to \Sigma'$ determines the many-sorted reduct of each model $M' \in \textbf{mod}(\Sigma')$ to a model $M \in \textbf{mod}(\Sigma)$, defined by interpreting symbols of Σ in M in the same way that their images under σ are interpreted in M'.

CASL Sentences

The many-sorted terms on a signature $\Sigma = (S, TF, PF, P)$ and a set of sorted, non-overloaded variables X are build from:

- Variables from X.
- application of qualified function symbols in $TF \cup PF$ to arguments terms of appropriate sorts.

gram(Σ) are the usual closed many-sorted first-order logic formulae, built from atomic formulae using quantification (over sorted variables) and logical connectives.

CASL Satisfaction Relation

The satisfaction of a sentence in a structure M is determined as usual by the holding of its atomic formulae w.r.t. assignments of (defined) values to all the variables that occur in them, the values assigned to variables of sort s being in s^M.

Outline

Closure Systems

Definition of Closure Systems

A closure system s a pair $\langle L, c \rangle$ where L is a set and $c : 2^L \to 2^L$ is a total function satisfying the following properties:

- Reflexivity: for every $\Phi \subseteq L$, $\Phi \subseteq c(\Phi)$.
- Monotonicity: for every $\Phi, \Gamma \subseteq L$, $\Phi \subseteq \Gamma$ implies $c(\Phi) \subseteq c(\Gamma)$.
- Idempotence: for every $\Phi \subseteq L$, $c(c(\Phi)) \subseteq c(\Phi)$.

Category **CLOS**

The category **CLOS** has as objects closure systems and morphisms $f : \langle L, c \rangle \to \langle L', c' \rangle$ are the maps $f : L \to L'$ such that $f(c(\Phi)) \subseteq c'(f(\Phi))$ for all $\Phi \subseteq L$.

Outline

Definition - Π-Institutions

Definition

A π-institutions consists of a pair $\langle \textbf{SIGN}, \textbf{clos} \rangle$, where **SIGN** is a category (of signatures) and **clos** : **SIGN** $\rightarrow$ **CLOS** is a functor.

Alternative Definition - Π-Institutions

Definition

A π-institutions consists of :

- A category **SIGN**.
- A functor **gram** : **SIGN** $\rightarrow$ **SET**.
- for every Σ : **SIGN**, a consequence relation $\rightsquigarrow_\Sigma$: $2^{\mathbf{gram}(\Sigma)} \times \mathbf{gram}(\Sigma)$ satisfying the following properties:
  - For every $p \in \mathbf{gram}(\Sigma)$, $p \rightsquigarrow_\Sigma p$.
  - For every $p \in \mathbf{gram}(\Sigma)$ and $\Phi_1, \Phi_2 \subseteq \mathbf{gram}(\Sigma)$, if $\Phi_1 \subseteq \Phi_2$ and $\Phi_1 \rightsquigarrow_\Sigma p$ then $\Phi_2 \rightsquigarrow_\Sigma p$.
  - For every $p \in \mathbf{gram}(\Sigma)$ and $\Phi_1, \Phi_2 \subseteq \mathbf{gram}(\Sigma)$, if $\Phi_1 \rightsquigarrow_\Sigma p$ and for every $p' \in \Phi_1$, $\Phi_2 \rightsquigarrow_\Sigma p'$ then $\Phi_2 \rightsquigarrow_\Sigma p$.
  - For every $\sigma : \Sigma \rightarrow \Sigma', p \in \mathbf{gram}(\Sigma)$ and $\Phi \subseteq \mathbf{gram}(\Sigma)$, $\Phi \rightsquigarrow_\Sigma p$ implies $\mathbf{gram}(\sigma)(\Phi) \rightsquigarrow_{\Sigma'} \mathbf{gram}(\sigma)(p)$.

Both Π-Institutions are Equivalent

Definition 1 to Definition 2

- The **gram** functor is the composition of **clos** with the forgetful functor that maps closure systems to the underlying languages.
- Every closure operator defines a consequence relation: $\Phi \leadsto_\Sigma p$ iff $p \in c_\Sigma(\Phi)$.

Definition 2 to Definition 1

- The closure operator itself is derived from the consequence relation: for every $\Phi \subseteq \mathbf{gram}(\Sigma)$, $c_\Sigma(\Phi) = \{p \in \mathbf{gram}(\Sigma) : \Phi \leadsto_\Sigma p\}$

Outline

Every Institution Presents A Π-Institution

Theorem

Every institution $\langle \textbf{SIGN}, \textbf{gram}, \textbf{mod}, \models \rangle$ *presents the* π*-institution* $\langle \textbf{SIGN}, \textbf{gram}, \leadsto \rangle$*, where for every signature* $\Sigma, p \in \textbf{gram}(\Sigma)$ *and* $\Phi \subseteq \textbf{gram}(\Sigma), \Phi \leadsto_\Sigma p$ *iff for every* $M \in \textbf{mod}(\Sigma), M \models_\Sigma \Phi$ *implies* $M \models_\Sigma p$.

Proof Outline

In order to prove this we must prove:

1. For every $p \in \textbf{gram}(\Sigma)$, $p \rightsquigarrow_\Sigma p$.
2. For every $p \in \textbf{gram}(\Sigma)$ and $\Phi_1, \Phi_2 \subseteq \textbf{gram}(\Sigma)$, if $\Phi_1 \subseteq \Phi_2$ and $\Phi_1 \rightsquigarrow_\Sigma p$ then $\Phi_2 \rightsquigarrow_\Sigma p$.
3. For every $p \in \textbf{gram}(\Sigma)$ and $\Phi_1, \Phi_2 \subseteq \textbf{gram}(\Sigma)$, if $\Phi_1 \rightsquigarrow_\Sigma p$ and for every $p' \in \Phi_1$, $\Phi_2 \rightsquigarrow_\Sigma p'$ then $\Phi_2 \rightsquigarrow_\Sigma p$.
4. For every $\sigma : \Sigma \to \Sigma', p \in \textbf{gram}(\Sigma)$ and $\Phi \subseteq \textbf{gram}(\Sigma)$, $\Phi \rightsquigarrow_\Sigma p$ implies $\textbf{gram}(\sigma)(\Phi) \rightsquigarrow_{\Sigma'} \textbf{gram}(\sigma)(p)$.

Proof - Part 3

To show:

For every $p \in \mathbf{gram}(\Sigma)$ and $\Phi_1, \Phi_2 \subseteq \mathbf{gram}(\Sigma)$, if $\Phi_1 \leadsto_\Sigma p$ and for every $p' \in \Phi_1$, $\Phi_2 \leadsto_\Sigma p'$ then $\Phi_2 \leadsto_\Sigma p$.

Definition of $\leadsto$

$\Phi \leadsto_\Sigma p$ iff for every $M \in \mathbf{mod}(\Sigma)$, $M \models_\Sigma \Phi$ implies $M \models_\Sigma p$.

Proof - Part 4

To show:

For every $\sigma : \Sigma \to \Sigma'$, $p \in \mathbf{gram}(\Sigma)$ and $\Phi \subseteq \mathbf{gram}(\Sigma)$, $\Phi \rightsquigarrow_\Sigma p$ implies $\mathbf{gram}(\sigma)(\Phi) \rightsquigarrow_{\Sigma'} \mathbf{gram}(\sigma)(p)$.

Definition of $\rightsquigarrow$

$\Phi \rightsquigarrow_\Sigma p$ iff for every $M \in \mathbf{mod}(\Sigma)$, $M \models_\Sigma \Phi$ implies $M \models_\Sigma p$.

Satisfaction Condition

For every $\sigma : \Sigma \to \Sigma'$, $p \in \mathbf{gram}(\Sigma)$ and $M' \in \mathbf{mod}(\Sigma')$, $\mathbf{mod}(\sigma)(M') \models_\Sigma p$ iff $M' \models_{\Sigma'} \mathbf{gram}(\sigma)(p)$.

Summary

- Institutions provide a frame work for dealing with logics, that capture the notions of translations and models.
- Π-Institutions and Institutions are equivalent, but useful for different purposes.

The Atomic Formulae

The atomic formulae are:

- Application of qualified predicate symbols $p \in P$ to argument terms of appropriate sorts.
- Assertions about the definedness of fully-qualified terms.
- Existential and strong equations between fully-quantified terms of the same sort.