An Institution-Independent Approach to Logic Programming

Ionuț Țuțu[1,2] José Luiz Fiadeiro[1]

[1]Department of Computer Science, Royal Holloway University of London

[2]Simion Stoilow Institute of Mathematics of the Romanian Academy

IFIP WG 1.3

Theddingworth near Leicester, 2014

Goal

to describe an axiomatic approach to logic programming based on which the conventional variant of the paradigm [Lloyd, 1987] can be explored for other formalisms

- order-sorted [Goguen and Meseguer, 1986]
- higher-order [Meseguer, 1989]
- constraint [Diaconescu, 2000]
- behavioural [Goguen et al., 2002]
- service-oriented [Țuțu and Fiadeiro, 2013]

Goal

to describe an axiomatic approach to logic programming based on which the conventional variant of the paradigm [Lloyd, 1987] can be explored for other formalisms

founded on a three-level hierarchy of concepts meant to capture

- the denotational semantics of logic programming, based on a notion of generalised substitution system
- the operational semantics of logic programming, supported by a notion of logic-programming framework
- the various constructions of logic programs through a notion of logic-programming language

Goal

to describe an axiomatic approach to logic programming based on which the conventional variant of the paradigm [Lloyd, 1987] can be explored for other formalisms

founded on a three-level hierarchy of concepts meant to capture

- the denotational semantics of logic programming, based on a notion of generalised substitution system
- the operational semantics of logic programming, supported by a notion of logic-programming framework
- the various constructions of logic programs through a notion of logic-programming language

Goal

> to describe an axiomatic approach to logic programming based on which the conventional variant of the paradigm [Lloyd, 1987] can be explored for other formalisms

founded on a three-level hierarchy of concepts meant to capture

- the denotational semantics of logic programming, based on a notion of generalised substitution system
- the operational semantics of logic programming, supported by a notion of logic-programming framework
- the various constructions of logic programs through a notion of logic-programming language

Goal

to describe an axiomatic approach to logic programming based on which the conventional variant of the paradigm [Lloyd, 1987] can be explored for other formalisms

suitable for investigating

- properties concerning the satisfaction of quantified sentences
- an abstract variant of Herbrand's theorem
- a resolution-based procedure for computing solutions to queries

Conventional Logic Programming

signature

  op $\mathtt{0}\colon 0$

  op $\mathtt{s\_}\colon 1$

  pred $\mathtt{add}\colon 3$

axioms

  $\mathtt{add}(\mathtt{0}, M, M) \xleftarrow[M]{}$

  $\mathtt{add}(\mathtt{s}\,M, N, \mathtt{s}\,P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$\xmapsto[X_1]{} \mathtt{add}(\mathtt{s}\,\mathtt{0}, \mathtt{s}\,\mathtt{0}, X_1)$

module $\mathtt{NAT}$ = **free**

  sort $\mathtt{Nat}$

  op $\mathtt{0}\colon \ \to \mathtt{Nat}$

  op $\mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}$

module $\mathtt{ADD}$ = $\mathtt{NAT}$ **then**

  op $\_ + \_\colon \mathtt{Nat}\,\mathtt{Nat} \to \mathtt{Nat}$

  cl $\mathtt{0} + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$

  cl $(\mathtt{s}\,M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$\xmapsto[X_1\,:\,\mathtt{Nat}]{} \mathtt{s}\,\mathtt{0} + \mathtt{s}\,\mathtt{0} = X_1$

Conventional Logic Programming

signature

op $\mathtt{0}\colon 0$

op $\mathtt{s\_}\colon 1$

pred $\mathtt{add}\colon 3$

axioms

$\mathtt{add}(\mathtt{0}, M, M) \xleftarrow[M]{}$

$\mathtt{add}(\mathtt{s}\, M, N, \mathtt{s}\, P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$\vdash_{X_1} \mathtt{add}(\mathtt{s}\,\mathtt{0}, \mathtt{s}\,\mathtt{0}, X_1)$

module NAT = **free**

sort Nat

op $\mathtt{0}\colon \ \to \mathtt{Nat}$

op $\mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}$

module ADD = NAT **then**

op $\_+\_\colon \mathtt{Nat}\,\mathtt{Nat} \to \mathtt{Nat}$

cl $\mathtt{0} + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$

cl $(\mathtt{s}\, M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{0} + \mathtt{s}\,\mathtt{0} = X_1$

Conventional Logic Programming

signature

  op $\mathtt{0}: 0$

  op $\mathtt{s\_}: 1$

  pred $\mathtt{add}: 3$

axioms

  $\mathtt{add}(\mathtt{0}, M, M) \xleftarrow[M]{}$

  $\mathtt{add}(\mathtt{s}\, M, N, \mathtt{s}\, P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$$\vdash_{X_1} \mathtt{add}(\mathtt{s}\,\mathtt{0}, \mathtt{s}\,\mathtt{0}, X_1)$$

module NAT = **free**

  sort Nat

  op $\mathtt{0}: \ \to \mathtt{Nat}$

  op $\mathtt{s\_}: \mathtt{Nat} \to \mathtt{Nat}$

module ADD = NAT **then**

  op $\_ + \_: \mathtt{Nat}\,\mathtt{Nat} \to \mathtt{Nat}$

  cl $\mathtt{0} + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$

  cl $(\mathtt{s}\, M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{0} + \mathtt{s}\,\mathtt{0} = X_1$$

Conventional Logic Programming

signature

op $\mathtt{0}\colon 0$

op $\mathtt{s\_}\colon 1$

pred $\mathtt{add}\colon 3$

axioms

$\mathtt{add}(\mathtt{0}, M, M) \xleftarrow[M]{}$

$\mathtt{add}(\mathtt{s}\, M, N, \mathtt{s}\, P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$\vdash_{X_1} \mathtt{add}(\mathtt{s}\,\mathtt{0}, \mathtt{s}\,\mathtt{0}, X_1)$

module $\mathtt{NAT}$ = **free**

sort $\mathtt{Nat}$

op $\mathtt{0}\colon \to \mathtt{Nat}$

op $\mathtt{s\_}\colon \mathtt{Nat} \to \mathtt{Nat}$

module $\mathtt{ADD}$ = $\mathtt{NAT}$ **then**

op $\_+\_\colon \mathtt{Nat}\,\mathtt{Nat} \to \mathtt{Nat}$

cl $\mathtt{0} + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$

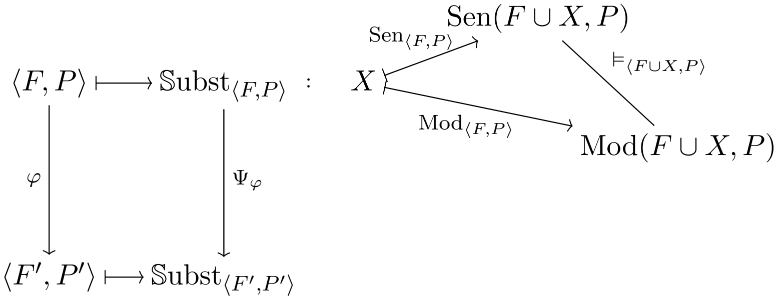
cl $(\mathtt{s}\, M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{0} + \mathtt{s}\,\mathtt{0} = X_1$

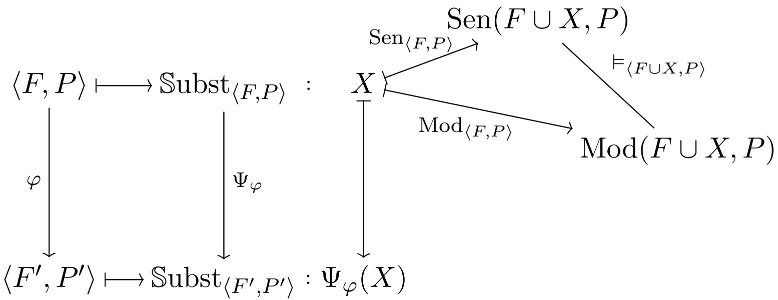
The Relational Setting

$$\langle F, P \rangle \longmapsto \mathbb{S}\mathrm{ubst}_{\langle F,P \rangle} \; : \quad X \xrightarrow{\mathrm{Sen}_{\langle F,P\rangle}} \mathrm{Sen}(F \cup X, P), \qquad X \xrightarrow{\mathrm{Mod}_{\langle F,P\rangle}} \mathrm{Mod}(F \cup X, P), \qquad \mathrm{Sen}(F \cup X, P) \;\overset{\models_{\langle F \cup X, P\rangle}}{—}\; \mathrm{Mod}(F \cup X, P)$$

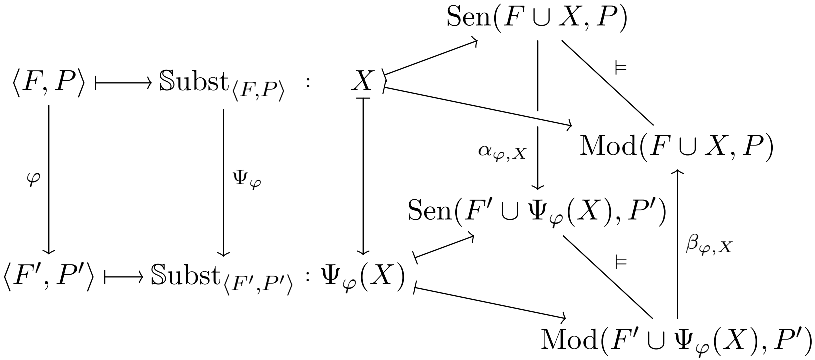
The Relational Setting



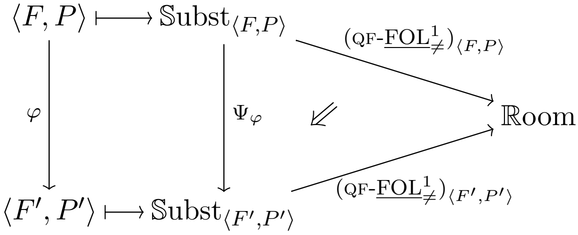
The Relational Setting



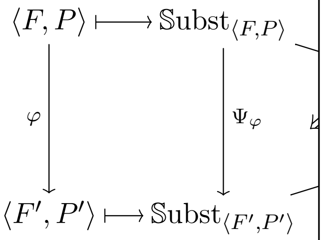
The Relational Setting



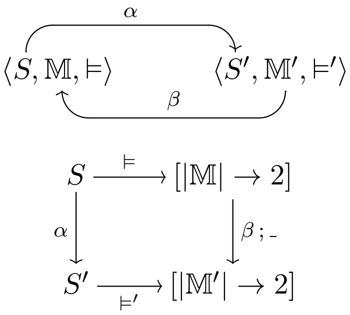
The Relational Setting



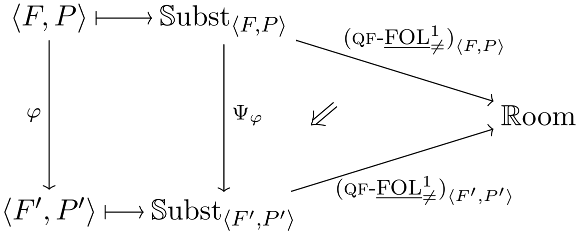
The Relational Setting



Rooms



The Relational Setting



The Relational Setting

$$\begin{array}{ccc} & \mathrm{Sen}(F,P) & \\ {\scriptstyle \alpha_{\langle F,P\rangle,X}}\swarrow & & \searrow{\scriptstyle \alpha_{\langle F,P\rangle,Y}} \\ \mathrm{Sen}_{\langle F,P\rangle}(X) & \xrightarrow[\psi(\_)]{} & \mathrm{Sen}_{\langle F,P\rangle}(Y) \end{array}$$

$$\begin{array}{ccc} & \mathrm{Mod}(F,P) & \\ {\scriptstyle \beta_{\langle F,P\rangle,X}}\nearrow & & \nwarrow{\scriptstyle \beta_{\langle F,P\rangle,Y}} \\ \mathrm{Mod}_{\langle F,P\rangle}(X) & \xleftarrow[\_\restriction_{\psi}]{} & \mathrm{Mod}_{\langle F,P\rangle}(Y) \end{array}$$

The Relational Setting

$$
\begin{array}{ccc}
 & \mathrm{Sen}(F,P) & \\
 \swarrow^{\alpha_{\langle F,P\rangle,X}} & & \searrow^{\alpha_{\langle F,P\rangle,Y}} \\
\mathrm{Sen}_{\langle F,P\rangle}(X) & \xrightarrow[\psi(\_)]{} & \mathrm{Sen}_{\langle F,P\rangle}(Y)
\end{array}
\qquad
\begin{array}{ccc}
 & \mathrm{Mod}(F,P) & \\
 \nearrow^{\beta_{\langle F,P\rangle,X}} & & \nwarrow^{\beta_{\langle F,P\rangle,Y}} \\
\mathrm{Mod}_{\langle F,P\rangle}(X) & \xleftarrow[\_\restriction_{\psi}]{} & \mathrm{Mod}_{\langle F,P\rangle}(Y)
\end{array}
$$

Substitution systems

$$\langle F,P\rangle \longmapsto \mathbb{S}\mathrm{ubst}_{\langle F,P\rangle} \longrightarrow G_{\langle F,P\rangle} \,/\, \mathbb{R}\mathrm{oom}$$

The Relational Setting

$$
\begin{array}{ccc}
 & \mathrm{Sen}(F,P) & \\
\swarrow^{\alpha_{\langle F,P\rangle,X}} & & \searrow^{\alpha_{\langle F,P\rangle,Y}} \\
\mathrm{Sen}_{\langle F,P\rangle}(X) & \xrightarrow[\psi(\_)]{} & \mathrm{Sen}_{\langle F,P\rangle}(Y)
\end{array}
\qquad
\begin{array}{ccc}
 & \mathrm{Mod}(F,P) & \\
\nearrow^{\beta_{\langle F,P\rangle,X}} & & \nwarrow^{\beta_{\langle F,P\rangle,Y}} \\
\mathrm{Mod}_{\langle F,P\rangle}(X) & \xleftarrow[\_\upharpoonright_{\psi}]{} & \mathrm{Mod}_{\langle F,P\rangle}(Y)
\end{array}
$$

Substitution systems

$$
\begin{array}{ccccccc}
\langle F,P\rangle & \longmapsto & \mathbb{S}\mathrm{ubst}_{\langle F,P\rangle} & \longrightarrow & G_{\langle F,P\rangle} \,/\, \mathbb{R}\mathrm{oom} & & G_{\langle F,P\rangle} \\
\varphi \big\downarrow & & \Psi_{\varphi} \big\downarrow & \Downarrow \tau_{\varphi} & \big\uparrow \kappa_{\varphi}/\mathbb{R}\mathrm{oom} & & \big\downarrow \kappa_{\varphi} \\
\langle F',P'\rangle & \longmapsto & \mathbb{S}\mathrm{ubst}_{\langle F',P'\rangle} & \longrightarrow & G_{\langle F',P'\rangle} \,/\, \mathbb{R}\mathrm{oom} & & G_{\langle F',P'\rangle}
\end{array}
$$

Quantified Sentences

Consider a generalised substitution system $\mathcal{GS} : \mathbb{S}\mathrm{ig} \to \mathbb{S}\mathrm{ubst}\mathbb{S}\mathrm{ys}$.

Universally-quantified sentence (for a signature Σ):

$$(\forall X)\,\rho$$

where $X \in |\mathbb{S}\mathrm{ubst}_\Sigma|$ and $\rho \in \mathrm{Sen}_\Sigma(X)$.

$M \vDash^{\mathrm{qs}}_\Sigma (\forall X)\,\rho$ if and only if all X-expansions of M satisfy ρ

Existentially-quantified sentences are defined in a similar manner.

Quantified Sentences

Proposition. Every generalised substitution system $\mathcal{GS}\colon \mathbb{S}\mathrm{ig} \to \mathbb{S}\mathrm{ubst}\mathbb{S}\mathrm{ys}$ that has weak model amalgamation, i.e.

$$\begin{array}{ccc} |\mathrm{Mod}(\Sigma)| & \xleftarrow{\ \ -\upharpoonright_{\varphi}\ \ } & |\mathrm{Mod}(\Sigma')| \\ {\scriptstyle -\upharpoonright_{\Sigma}}\uparrow & & \uparrow{\scriptstyle -\upharpoonright_{\Sigma'}} \\ |\mathrm{Mod}_{\Sigma}(X)| & \xleftarrow[\beta_{\varphi,X}]{} & |\mathrm{Mod}_{\Sigma'}(\Psi_{\varphi}(X))| \end{array}$$

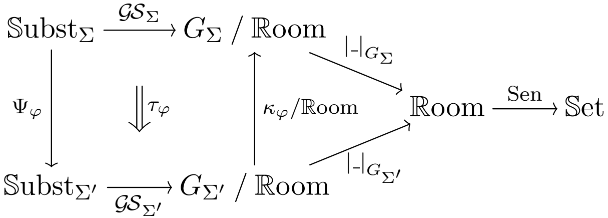
is a weak pullback for every signature morphism $\varphi\colon \Sigma \to \Sigma'$, gives rise to an institution of quantified sentences

$$\mathcal{GS}^{\mathrm{qs}} = \langle \mathbb{S}\mathrm{ig}, \mathrm{QSen}, \mathrm{Mod}, \models^{\mathrm{qs}} \rangle.$$

Local Sentences

Generalised substitution systems $\mathcal{GS}\colon \mathbb{S}\mathrm{ig} \to \mathbb{S}\mathrm{ubst}\mathbb{S}\mathrm{ys}$ determine local-sentence functors $\mathrm{LSen}\colon \mathbb{S}\mathrm{ig} \to [\_ \to \mathbb{S}\mathrm{et}]^\sharp$ that map

- signatures Σ to $\mathcal{GS}_\Sigma \,;\, |\_|_{G_\Sigma} \,;\, \mathrm{Sen}\colon \mathbb{S}\mathrm{ubst}_\Sigma \to \mathbb{S}\mathrm{et}$, and
- signature morphisms $\varphi\colon \Sigma \to \Sigma'$ to $\langle \Psi_\varphi, \tau_\varphi \cdot (|\_|_{G_\Sigma} \,;\, \mathrm{Sen}) \rangle$.



Abstract Logic Programming

Logic-programming framework: $\mathcal{F} = \langle \mathcal{GS}, \mathrm{C}, \mathrm{Q}, \Vdash \rangle$, where

- $\mathcal{GS}$ is a gen. subst. sys. that has weak model amalgamation,
- C and Q are generalised subfunctors of LSen, and
- $\Vdash$ is a generalised subfunctor of $(\mathrm{Q} \times \mathrm{C}) \times \mathrm{Q}$,

such that

1. preservation of queries:

$$N_1 \models_{\Sigma,X} \rho \quad \text{implies} \quad N_2 \models_{\Sigma,X} \rho,$$

   for every $\rho \in \mathrm{Q}_\Sigma(X)$ and $h \colon N_1 \to N_2$ in $\mathrm{Mod}_\Sigma(X)$,

2. soundness of goal-directed rules:

$$\rho_1, \gamma \Vdash_{\Sigma,X} \rho_2 \quad \text{implies} \quad \{\rho_2, \gamma\} \models_{\Sigma,X} \rho_1.$$

Abstract Logic Programming

Example. Relational logic programming is defined over the generalised substitution system $\textsc{qf}\text{-}\mathrm{FOL}^1_{\neq}$ as follows:

- $\mathrm{C}_{\langle F,P\rangle}(X)$ consists of all implications $\bigwedge H \Rightarrow C$, where H is a finite set of relational atoms and C is an atom,
- $\mathrm{Q}_{\langle F,P\rangle}(X)$ consists of all finite conjunctions of atoms $\bigwedge Q$,
- the goal-directed rules are given by

$$\bigwedge(\{C\} \cup Q), \bigwedge H \Rightarrow C \Vdash_{\langle F,P\rangle,X} \bigwedge(Q \cup H).$$

We obtain in this way a logic-programming framework $\mathcal{FOL}^1_{\neq}$.

Abstract Logic Programming

Example. Equational logic programming is defined over the generalised substitution system $\text{QF-FOL}_=$ as follows:

- $\mathrm{C}_{\langle S,F\rangle}(X)$ consists of implications $\bigwedge H \Rightarrow (l = r)$, where H is a finite set of equational atoms and $l = r$ is an atom,
- $\mathrm{Q}_{\langle F,P\rangle}(X)$ consists of finite conjunctions of atoms $\bigwedge Q$,
- the goal-directed rules are given by

$$\bigwedge(\{c[l] = t\} \cup Q), \bigwedge H \Rightarrow (l = r) \Vdash_{\langle S,F\rangle,X} \bigwedge(\{c[r] = t\} \cup Q \cup H)$$

or

$$\bigwedge(\{t = c[l]\} \cup Q), \bigwedge H \Rightarrow (l = r) \Vdash_{\langle S,F\rangle,X} \bigwedge(\{t = c[r]\} \cup Q \cup H).$$

This determines a logic-programming framework $\mathcal{FOL}_=$.

Abstract Logic Programming

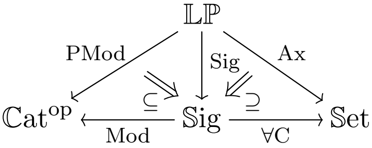
Given a signature Σ in a framework $\mathcal{F} = \langle \mathcal{GS}, \mathrm{C}, \mathrm{Q}, \Vdash \rangle$,

Σ-clause: quantified sentence $(\forall X)\,\gamma$ over Σ such that $\gamma \in \mathrm{C}_\Sigma(X)$.

Σ-query: quantified sentence $(\exists X)\,\rho$ over Σ such that $\rho \in \mathrm{Q}_\Sigma(X)$.

This yields two subfunctors $\forall\mathrm{C}, \exists\mathrm{Q} \subseteq \mathrm{QSen}$.

Logic-programming language $\mathcal{L} = \langle \mathbb{LP}, \mathrm{Sig}, \mathrm{PMod}, \mathrm{Ax} \rangle$ over $\mathcal{F}$:



- a category $\mathbb{LP}$ of logic programs,
- a signature functor $\mathrm{Sig} \colon \mathbb{LP} \to \mathbb{S}\mathrm{ig}$, and
- subfunctors $\mathrm{PMod} \subseteq \mathrm{Sig}\,;\mathrm{Mod}$ and $\mathrm{Ax} \subseteq \mathrm{Sig}\,;\forall\mathrm{C}$,

such that for every program P and model M of P, $M \models^{\mathrm{qs}} \mathrm{Ax}(P)$.

Abstract Logic Programming

Consider a logic program $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$. A Σ-substitution $\psi\colon X \to Y$ is a $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$-solution to a Σ-query $(\exists X)\,\rho$ if

- Y is conservative, i.e. $\beta_{\Sigma,Y}\colon \mathrm{Mod}_\Sigma(Y) \to \mathrm{Mod}(\Sigma)$ is surjective on objects, and
- $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle \models^{\mathrm{lp}}_{\Sigma} (\forall Y)\,\psi(\rho)$.

Theorem (Herbrand's theorem). For every program $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$ and Σ-query $(\exists X)\,\rho$ such that $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$ has an X-reachable initial model $0_{\langle\!\langle \Sigma, \Gamma \rangle\!\rangle}$, the following statements are equivalent:

1. $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle \models^{\mathrm{lp}}_{\Sigma} (\exists X)\,\rho$,
2. $0_{\langle\!\langle \Sigma, \Gamma \rangle\!\rangle} \models^{\mathrm{qs}}_{\Sigma} (\exists X)\,\rho$,
3. $(\exists X)\,\rho$ admits a $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$-solution.

Operational Semantics

- $\texttt{add}(0, M, M) \xleftarrow[M]{}$
- $\texttt{add}(\texttt{s}\, M, N, \texttt{s}\, P) \xleftarrow[M,N,P]{} \texttt{add}(M, N, P)$

$$\xmapsto[X_1]{} \texttt{add}(\texttt{s}\, 0, \texttt{s}\, 0, X_1)$$

Operational Semantics

- $\mathtt{add}(0, M, M) \xleftarrow[M]{}$
- $\mathtt{add}(\mathtt{s}\,M, N, \mathtt{s}\,P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$\xmapsto[X_1]{} \mathtt{add}(\mathtt{s}\,0, \mathtt{s}\,0, X_1) \qquad \leftarrow$

Operational Semantics

- $\mathtt{add}(0, M, M) \xleftarrow[M]{}$
- $\mathtt{add}(\mathtt{s}\, M, N, \mathtt{s}\, P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$$\frac{\vdash_{X_1} \mathtt{add}(\mathtt{s}\,0, \mathtt{s}\,0, X_1) \qquad \leftarrow}{\vdash_{X_2} \mathtt{add}(0, \mathtt{s}\,0, X_2)} \; X_1 \mapsto \mathtt{s}\, X_2$$

Operational Semantics

- $\mathtt{add}(0, M, M) \xleftarrow[M]{}$
- $\mathtt{add}(\mathtt{s}\,M, N, \mathtt{s}\,P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$$\frac{\vdash_{X_1} \mathtt{add}(\mathtt{s}\,0, \mathtt{s}\,0, X_1) \quad \leftarrow}{\vdash_{X_2} \mathtt{add}(0, \mathtt{s}\,0, X_2) \quad \leftarrow} \; X_1 \mapsto \mathtt{s}\,X_2$$

Operational Semantics

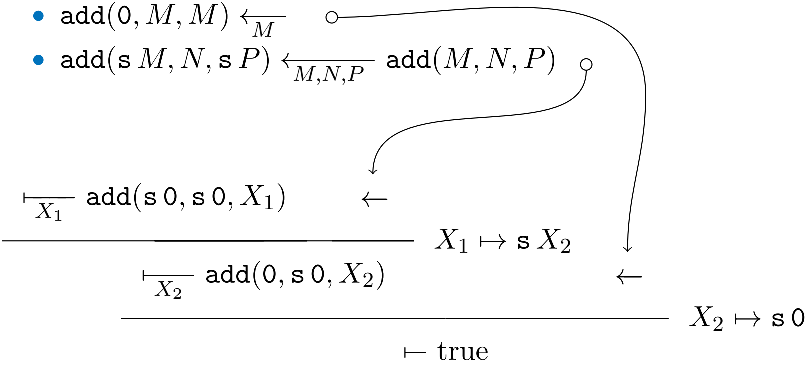
- $\mathtt{add}(0, M, M) \xleftarrow[M]{}$
- $\mathtt{add}(\mathtt{s}\,M, N, \mathtt{s}\,P) \xleftarrow[M,N,P]{} \mathtt{add}(M, N, P)$

$$\dfrac{\vdash_{X_1} \mathtt{add}(\mathtt{s}\,0, \mathtt{s}\,0, X_1) \leftarrow}{\vdash_{X_2} \mathtt{add}(0, \mathtt{s}\,0, X_2) \leftarrow}\; X_1 \mapsto \mathtt{s}\,X_2$$

$$\dfrac{\vdash_{X_2} \mathtt{add}(0, \mathtt{s}\,0, X_2) \leftarrow}{\vdash \mathrm{true}}\; X_2 \mapsto \mathtt{s}\,0$$

Operational Semantics

- $\texttt{add}(0, M, M) \xleftarrow[M]{}$
- $\texttt{add}(\texttt{s}\, M, N, \texttt{s}\, P) \xleftarrow[M,N,P]{} \texttt{add}(M, N, P)$



$$X_1 \mapsto \texttt{s}\, X_2 \mapsto \texttt{s}\,\texttt{s}\,0$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M\,:\,\texttt{Nat}]{}$
- $(\texttt{s}\, M) + N = \texttt{s}\,(M + N) \xleftarrow[M,N\,:\,\texttt{Nat}]{}$

$$\vdash_{X_1\,:\,\texttt{Nat}} \texttt{s}\,0 + \texttt{s}\,0 = X_1$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$
- $(\mathtt{s}\,M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,0 + \mathtt{s}\,0 = X_1 \quad \leftarrow$$

Operational Semantics

- $\mathsf{0} = \mathsf{0} \leftarrow$
- $\mathsf{0} + M = M \xleftarrow[M : \mathtt{Nat}]{}$
- $(\mathtt{s}\, M) + N = \mathtt{s}\, (M + N) \xleftarrow[M, N : \mathtt{Nat}]{}$

$$\dfrac{\vdash_{X_1 : \mathtt{Nat}} \; \mathtt{s}\, \mathsf{0} + \mathtt{s}\, \mathsf{0} = X_1 \quad \leftarrow}{\vdash_{X_2 : \mathtt{Nat}} \; \mathtt{s}\, (\mathsf{0} + \mathtt{s}\, \mathsf{0}) = X_2} \; X_1 \mapsto X_2$$

Operational Semantics

- $\mathtt{0} = \mathtt{0} \leftarrow$
- $\mathtt{0} + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{\quad}$
- $(\mathtt{s}\,M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{\quad}$

$$\frac{\vdash_{X_1\,:\,\mathtt{Nat}} \; \mathtt{s}\,\mathtt{0} + \mathtt{s}\,\mathtt{0} = X_1 \quad \leftarrow}{\vdash_{X_2\,:\,\mathtt{Nat}} \; \mathtt{s}\,(\mathtt{0} + \mathtt{s}\,\mathtt{0}) = X_2 \quad \leftarrow} \; X_1 \mapsto X_2$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M:\,\mathtt{Nat}]{}$
- $(\mathtt{s}\,M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N:\,\mathtt{Nat}]{}$

$$\vdash_{X_1:\,\mathtt{Nat}} \mathtt{s}\,0 + \mathtt{s}\,0 = X_1 \quad \leftarrow$$

$$\overline{}\; X_1 \mapsto X_2$$

$$\vdash_{X_2:\,\mathtt{Nat}} \mathtt{s}\,(0 + \mathtt{s}\,0) = X_2 \quad \leftarrow$$

$$\overline{}\; X_2 \mapsto X_3$$

$$\vdash_{X_3:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{s}\,0 = X_3$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$
- $(\mathtt{s}\,M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,0 + \mathtt{s}\,0 = X_1 \quad \leftarrow$$

$$X_1 \mapsto X_2$$

$$\vdash_{X_2\,:\,\mathtt{Nat}} \mathtt{s}\,(0 + \mathtt{s}\,0) = X_2 \quad \leftarrow$$

$$X_2 \mapsto X_3$$

$$\vdash_{X_3\,:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{s}\,0 = X_3 \quad \leftarrow$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M\,:\,\mathtt{Nat}]{}$
- $(\mathtt{s}\, M) + N = \mathtt{s}\,(M + N) \xleftarrow[M,N\,:\,\mathtt{Nat}]{}$

$$\vdash_{X_1\,:\,\mathtt{Nat}} \mathtt{s}\,0 + \mathtt{s}\,0 = X_1 \quad \leftarrow$$

$$X_1 \mapsto X_2$$

$$\vdash_{X_2\,:\,\mathtt{Nat}} \mathtt{s}\,(0 + \mathtt{s}\,0) = X_2 \quad \leftarrow$$

$$X_2 \mapsto X_3$$

$$\vdash_{X_3\,:\,\mathtt{Nat}} \mathtt{s}\,\mathtt{s}\,0 = X_3 \quad \leftarrow$$

$$\vdash \mathtt{s}\,\mathtt{s}\,0 = \mathtt{s}\,\mathtt{s}\,0 \qquad X_3 \mapsto \mathtt{s}\,\mathtt{s}\,0$$

Operational Semantics

- $0 = 0 \leftarrow$
- $0 + M = M \xleftarrow[M : \mathtt{Nat}]{}$
- $(\mathsf{s}\, M) + N = \mathsf{s}\,(M + N) \xleftarrow[M,N : \mathtt{Nat}]{}$

$$\vdash_{X_1 : \mathtt{Nat}} \mathsf{s}\,0 + \mathsf{s}\,0 = X_1 \quad \leftarrow$$

$$X_1 \mapsto X_2$$

$$\vdash_{X_2 : \mathtt{Nat}} \mathsf{s}\,(0 + \mathsf{s}\,0) = X_2 \quad \leftarrow$$

$$X_2 \mapsto X_3$$

$$\vdash_{X_3 : \mathtt{Nat}} \mathsf{s}\,\mathsf{s}\,0 = X_3 \quad \leftarrow$$

$$X_3 \mapsto \mathsf{s}\,\mathsf{s}\,0$$

$$\vdash \mathsf{s}\,\mathsf{s}\,0 = \mathsf{s}\,\mathsf{s}\,0$$

$$\boxed{X_1 \mapsto X_2 \mapsto X_3 \mapsto \mathsf{s}\,\mathsf{s}\,0}$$

Operational Semantics

Resolution: Let $(\exists X_1)\,\rho_1$ be a query and $(\forall Y_1)\,\gamma_1$ a clause over Σ. A query $(\exists X_2)\,\rho_2$ is derived by resolution from $(\exists X_1)\,\rho_1$ and $(\forall Y_1)\,\gamma_1$ using the computed substitution $\theta_1 \colon X_1 \to X_2$ if there exists a substitution $\psi_1 \colon Y_1 \to X_2$ such that

$$\theta_1(\rho_1), \psi_1(\gamma_1) \Vdash_{\Sigma, X_2} \rho_2.$$

Proposition. For every step $(\exists X_1)\,\rho_1 \longrightarrow\!\!\!\!\!\rightarrow_{\Gamma,\theta_1} (\exists X_2)\,\rho_2$ and every solution $\psi \colon X_2 \to Y$ to the query $(\exists X_2)\,\rho_2$, the substitution $\theta_1\,;\psi$ is a solution to $(\exists X_1)\,\rho_1$.

Operational Semantics

Trivial query: a query $(\exists Y)\,\top$ such that

- Y is conservative and
- any Y-model satisfies $\top$.

Computed answer: (to a query $(\exists X)\,\rho$, with respect to $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$)
a substitution $\theta \colon X \to Y$ such that

$$(\exists X)\,\rho \twoheadrightarrow^{*}_{\Gamma,\theta} (\exists Y)\,\top$$

for some trivial query $(\exists Y)\,\top$.

Theorem (Soundness of resolution).
For any program $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$ and any Σ-query $(\exists X)\,\rho$,
every computed answer to $(\exists X)\,\rho$ is a solution to $(\exists X)\,\rho$.

Operational Semantics

Query-completeness: $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$ is query-complete if for every conservative signature of Σ-variables X, and every X-query ρ,

$$\langle\!\langle \Sigma, \Gamma \rangle\!\rangle \vDash^{\mathrm{lp}}_{\Sigma} (\forall X)\, \rho \quad \text{implies} \quad \rho, X(\Gamma) \Vdash^{*}_{\Sigma,X} \top$$

for some trivial X-query $\top$, where

$$X(\Gamma) = \big\{ \psi(\gamma) \in \mathrm{Sen}_{\Sigma}(X) \;\big|\; (\forall Y)\, \gamma \in \Gamma \text{ and } \psi \colon Y \to X \big\}.$$

Identity clause: (of a query $(\exists X)\, \rho$) a clause $(\forall X)\, \gamma$ such that γ is a tautology and $\rho, \gamma \Vdash_{\Sigma,X} \rho$.

Theorem (Completeness of resolution).
If $\langle\!\langle \Sigma, \Gamma \rangle\!\rangle$ is a query-complete and $(\exists X)\, \rho$ admits an identity $(\forall X)\, \gamma \in \Gamma$ then every solution to $(\exists X)\, \rho$ can be computed.

Conclusions

Summary

- axiomatic theory of logic programming
- properties related to the satisfaction of quantified sentences
- a generalisation of Herbrand's theorem to abstract logic-programming languages
- a sound and (conditionally) complete procedure – but not complete with respect to the implementation – for computing solutions to queries

Outlook

- other forms of logic programming (higher-order, behavioural, service-oriented)
- preservation and reflection of answers along morphisms
- maps of logic-programming languages

Thank you!

References

Răzvan Diaconescu. Category-based constraint logic. *Mathematical Structures in Computer Science*, 10:373–407, 5 2000.

Joseph A. Goguen and José Meseguer. Eqlog: Equality, types, and generic modules for logic programming. In *Logic Programming: Functions, Relations, and Equations*, pages 295–363. Prentice Hall, 1986.

Joseph A. Goguen, Grant Malcolm, and Tom Kemp. A hidden herbrand theorem: combining the object and logic paradigms. *Journal of Logic and Algebraic Programming*, 51(1):1–41, 2002.

John W. Lloyd. *Foundations of logic programming*. Symbolic computation: Artificial intelligence. Springer, 1987.

José Meseguer. General logics. In Heinz-Dieter Ebbinghaus, José Fernández-Prida, Manuel Garrido, Daniel Lascar, and Mario Rodriquez-Artalejo, editors, *Logic Colloquium '87*, volume 129 of *Studies in Logic and the Foundations of Mathematics Series*, pages 275–329. Elsevier, 1989.

Ionuț Țuțu and José L. Fiadeiro. A logic-programming semantics of services. In Reiko Heckel and Stefan Milius, editors, *Algebra and Coalgebra in Computer Science*, volume 8089 of *Lecture Notes in Computer Science*, pages 299–313. Springer, 2013.