Constraint Programming V: Logical Semantics

François Fages
INRIA Rocquencourt,
Francois.Fages@inria.fr

1. Introduction
2. Logical Background
3. Constraint Logic Programs
4. Operational and Fixpoint Semantics
5. Logical Semantics
6. Concurrent Constraint Languages
7. Operational and Denotational Semantics
8. Semantics in Linear Logic

Part V: Logical Semantics

1. Logical semantics of CLP($\mathcal{X}$)
2. Automated deduction
   proofs in group theory
3. CLP(λ)
4. Negation as Failure
   completeness w.r.t. Clark's completion

5. Logical Semantics of CLP($\mathcal{X}$) Programs

- Proper logical semantics

$$(1)\quad P, \mathcal{T} \models \exists(G) \qquad (4)\quad P, \mathcal{T} \models c \supset G,$$

- Logical semantics in a fixed pre-interpretation

$$(2)\quad P \models_{\mathcal{X}} \exists(G) \qquad (5)\quad P \models_{\mathcal{X}} c \supset G,$$

- Algebraic semantics

$$(3)\quad M_P^{\mathcal{X}} \models \exists(G) \qquad (6)\quad M_P^{\mathcal{X}} \models c \supset G.$$

We show (1) $\Leftrightarrow$ (2) $\Leftrightarrow$ (3) and (4) $\Rightarrow$ (5) $\Leftrightarrow$ (6).

Soundness of CSLD Resolution

Theorem 1 *[JL87] If c is a computed answer for the goal G then $M_P^{\mathcal{X}} \models c \supset G$, $P \models_{\mathcal{X}} c \supset G$ and $P, \mathcal{T} \models c \supset G$.*

Proof: If $G = (d|A_1, ..., A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, ..., c_n$ for the goals $A_1, ..., A_n$ such that $c = d \wedge \bigwedge_{i=1}^{n} c_i$ is satisfiable. For every $1 \leq i \leq n$

$c_i|A_i \in S_P^{\mathcal{X}} \uparrow \omega$, by **??**,

$[c_i|A_i]_{\mathcal{X}} \subseteq M_P^{\mathcal{X}}$, by **??**, and **??**, hence $M_P^{\mathcal{X}} \models \forall(c_i \supset A_i)$,

$P \models_{\mathcal{X}} \forall(c_i \supset A_i)$ as $M_P^{\mathcal{X}}$ is the least $\mathcal{X}$-model of P,

$P \models_{\mathcal{X}} \forall(c \supset A_i)$ as $\mathcal{X} \models \forall(c \supset c_i)$ for all i, $1 \leq i \leq n$.

Therefore we have $P \models_{\mathcal{X}} \forall(c \supset (d \wedge A_1 \wedge ... \wedge A_n))$,

and as the same reasoning applies to any model $\mathcal{X}$ of $\mathcal{T}$,
$P, \mathcal{T} \models \forall(c \supset (d \wedge A_1 \wedge ... \wedge A_n))$ $\square$

Completeness of CSLD resolution

Theorem 2 *[Maher87] If $M_P^{\mathcal{X}} \models_{\mathcal{X}} c \supset G$ then there exists a set $\{c_i\}_{i\geq 0}$ of computed answers for G, such that:*

$$\mathcal{X} \models \forall(c \supset \bigvee_{i\geq 0} \exists Y_i c_i).$$

PROOF: For every solution ρ of c, for every atom A_j in G,

$M_P^{\mathcal{X}} \models A_j\rho$ iff $A_j\rho \in T_P^{\mathcal{X}} \uparrow \omega$, by **??**, iff $A_j\rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$, by **??**,

iff $c_{j,\rho}|A_j \in S_P^{\mathcal{X}} \uparrow \omega$, for some constraint $c_{j,\rho}$ s.t. ρ is solution of $\exists Y_{j,\rho} c_{j,\rho}$, where $Y_{j,\rho} = V(c_{j,\rho}) \setminus V(A_j)$,

iff $c_{j,\rho}$ is a computed answer for A_j (by **??**) and $\mathcal{X} \models \exists Y_{j,\rho} c_{j,\rho}\rho$.

Let c_ρ be the conjunction of $c_{j,\rho}$ for all j. c_ρ is a computed answer for G.

By taking the collection of c_ρ for all ρ we get $\mathcal{X} \models \forall(c \supset \bigvee_{c_\rho} \exists Y_\rho c_\rho)$ $\square$

Completeness w.r.t. the theory of the structure

Theorem 3 *[Maher87] If $P, \mathcal{T} \models c \supset G$ then there exists a finite set $\{c_1, ..., c_n\}$ of computed answers to G, such that:*

$$\mathcal{T} \models \forall(c \supset \exists Y_1 c_1 \vee ... \vee \exists Y_n c_n).$$

Proof: If $P, \mathcal{T} \models c \supset G$ then for every model $\mathcal{X}$ of $\mathcal{T}$, for every $\mathcal{X}$-solution ρ of c, there exists a computed constraint $c_{\mathcal{X},\rho}$ for G s.t. $\mathcal{X} \models c_{\mathcal{X},\rho}\rho$. Let $\{c_i\}_{i\geq 0}$ be the set of these computed answers. Then for every model $\mathcal{X}$ and for every $\mathcal{X}$-valuation ρ, $\mathcal{X} \models c \supset \bigvee_{i\geq 1} \exists Y_i c_i$, therefore $\mathcal{T} \models c \supset \bigvee_{i\geq 1} \exists Y_i c_i$,

As $\mathcal{T} \cup \{\exists(c \wedge \neg\exists Y_i c_i)\}_i$ is unsatisfiable, by applying the compactness theorem of first-order logic there exists a finite part $\{c_i\}_{1\leq i\leq n}$,
s.t. $\mathcal{T} \models c \supset \bigvee_{i=1}^{n} \exists Y_i c_i$. $\square$

5. First-order theorem proving in CLP($\mathcal{H}$)

Prolog can be used to find proofs by refutation of Horn clauses (with a complete search meta-interpreter).

$P, \forall(\neg A)$ is unsatisfiable iff $P \models \exists(A)$ iff $A \longrightarrow^* \Box$.

Groups can be axiomatized with Horn clauses with a ternary predicate $p(x, y, z)$ meaning $x * y = z$.

```
clause(p(e,X,X)).
clause(p(i(X),X,e)).
clause((p(U,Z,W) :- p(X,Y,U), p(Y,Z,V), p(X,V,W))).
clause((p(X,V,W) :- p(X,Y,U), p(Y,Z,V), p(U,Z,W))).
```

Theorem proving in groups

To show $i(i(x)) = x$ by refutation,

we show that the formula $\neg\forall x \; p(i(i(X)), e, X)$ is unsatisfiable

By Skolemization we get the goal clause $\neg p(i(i(a)), e, a)$

```
| ?- solve(p(i(i(a)),e,a)).
depth 2
yes
| ?- solve(p(a,e,a)).
depth 4
yes
| ?- solve(p(a,i(a),e)).
depth 3
yes
```

Theorem proving in groups (cont.)

To show that any non empty subset of a group, stable by division, is a subgroup we add two clauses

```
clause(s(a)).
clause((s(Z) :- s(X), s(Y), p(X,i(Y),Z))).
```

and prove that s contains e and $i(a)$.

```
| ?- res(s(e)).
depth 4
yes
| ?- res(s(i(a))).
depth 5
yes
```

Higher-order theorem proving in CLP(λ)

Church's simply typed λ-calculus

$t \ ::= \ v \mid t_1 \to t_2$

$e : t \ ::= \ x : t \mid (\lambda x : t_1 . e : t_2) : t_1 \to t_2 \mid (e_1 : t_1 \to t_2 (e_2 : t_1)) : t_2$

Theory of functionality

$\lambda x . e_1 =_\alpha \lambda y . e_1[y/x]$ if $y \notin V(e_1)$,

$(\lambda x . e_1) e_2 \ \to_\beta \ e_1[e_2/x]$

$=_\alpha . \to_\beta$ is terminating and confluent

$$e_1 \ =_{\alpha,\beta} \ e_2 \ iff \ \downarrow_\beta e_1 \ =_\alpha \downarrow_\beta e_2 .$$

Equality is decidable, but not unification...

Theorem proving in CLP(λ)

Theorem 4 (Cantor's Theorem) $\mathbf{N}^{\mathbf{N}}$ *is not countable.*

Proof: By two steps of CSLD resolution!

Let us suppose $\exists h : \mathbf{N} \to (\mathbf{N} \to \mathbf{N})\ \forall f : \mathbf{N} \to \mathbf{N}\ \exists n : \mathbf{N}\ h(n) = f$

After Skolemisation we get $\forall F\ h(n(F)) = F$, i.e. $\forall F\ \neg h(n(F)) \neq F$.

Let us consider the following program $G \neq H \ \leftarrow\ G(N) \neq H(N).$

$$N \neq s(N).$$

We have $h(n\ F) \neq F \ \longrightarrow^{\sigma_1}\ (h(n\ F))(I) \neq F(I) \ \longrightarrow^{\sigma_2}\ \square$

where the unifier $\sigma_2 = \{G = h\ I\ I,\ I = n(F),\ F = \lambda i.s(h\ i\ i),\ H = F\}$ is Cantor's diagonal argument! $\square$

6. Negation by Failure

A derivation CSLD is fair if every atom which appears in a goal of the derivation is selected after a finite number of resolution steps.

A fair CSLD tree for a goal G is a CSLD derivation tree for G in which all derivations are fair.

A goal G i finitely failed if G has a fair CSLD derivation tree to G, which is finite and which contains no success.

```
p :- p.

| ?- member(a,[b,c,d]).
no

| ?- p, member(a,[b,c,d]).
...
```

Logical semantics of finite failure?

Horn clauses entail no negative information: the Herbrand's base $\mathcal{B}_{\mathcal{X}}$ is a model.

On the other hand, the complement of the least $\mathcal{X}$-model M_P^X is not recursively enumerable.

Indeed let us suppose the opposite. We could define in Prolog the predicates

`success(P,B)` which succeeds iff $M_P \models \exists B$, i.e. if the goal B has a successful SLD derivation with the program P

`fail(P,B)` which succeeds iff $M_P \models \neg\exists B$

Undecidability of $M_P^{\mathcal{X}}$

```
loop:- loop.
contr(P):- success(P,P), loop.
contr(P):- fail(P,P).
?- contr(contr).
```

If `contr(contr)` has a success,

then `success(contr,contr)` succeeds,

and `fail(contr,contr)` doesn't succeed,

hence `contr(contr)` doesn't succeed: contradiction.

if `contr(contr)` doesn't succeed,

then `fail(contr,contr)` succeeds,

hence `contr(contr)` succeeds: contradiction.

Therefore programs `success` and `fail` cannot exist.

Clark's completion

The Clark's completion of P is the set P^* of formulas of the form

$\forall X p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A_1^1 \wedge \ldots \wedge A_{n_1}^1) \vee \ldots \vee (\exists Y_k c_k \wedge A_1^k \wedge \ldots \wedge A_{n_k}^k)$ where the $p(X) \leftarrow c_i | A_1^i, \ldots, A_{n_i}^i$ are the rules in P and Y_i's the local variables,

$\forall X \neg p(X)$ if p is not defined in P

Example: CLP($\mathcal{H}$) program `p(s(X):- p(X)`

Clark's completion $P^* = \{\forall x\ p(x) \leftrightarrow \exists y\ x = s(y) \wedge p(y)$.

The goal `p(0)` finitely fails, we have $P^*, CET \models \neg p(0)$.

The goal `p(X)` doesn't finitely fail,

we have $P^*, CET \not\models \neg \exists X\ p(X)$ although $P^* \models_{\mathcal{H}} \neg \exists X\ p(X)$

Supported $\mathcal{X}$-models

Proposition 5 *i) I is a supported $\mathcal{X}$-model of P iff ii) I is a $\mathcal{X}$-model of P^* iff iii) I is a fixed point of $T_P^{\mathcal{X}}$.*

PROOF: I is a $\mathcal{X}$-model of P

iff I is a $\mathcal{X}$-model of $\forall X\ p(X) \leftarrow \phi_1 \vee ... \vee \phi_k$ for every formula $\forall X\ p(X) \leftrightarrow \phi_1 \vee ... \vee \phi_k$ in P^*,

iff I is a post-fixed point of $T_P^{\mathcal{X}}$, i.e. .$T_P^{\mathcal{X}}(I) \subseteq I$.

I is a supported $\mathcal{X}$-interpretation of P,

iff I is a $\mathcal{X}$-model of $\forall X\ p(X) \rightarrow \phi_1 \vee ... \vee \phi_k$ for every formula $\forall X\ p(X) \leftrightarrow \phi_1 \vee ... \vee \phi_k$ in P^*,

iff I is a pre-fixed point of $T_P^{\mathcal{X}}$, i.e. $I \subseteq T_P^{\mathcal{X}}(I)$.

Thus *i)* I is a supported $\mathcal{X}$-model of P iff *ii)* I is a $\mathcal{X}$-model of P^* iff *iii)* I is a fixed point of $T_P^{\mathcal{X}}$. $\square$

Models of the Clark's completion

Theorem 6 *i) P^* has the same least $\mathcal{X}$-model than P, $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$*

ii) $P \models_{\mathcal{X}} c \rightarrow A$ iff $P^* \models_{\mathcal{X}} c \rightarrow A$, for all c and A,

iii) $P, \mathcal{T} \models c \rightarrow A$ iff $P^*, \mathcal{T} \models c \rightarrow A$.

Proof: i) is an immediate corollary of **??** and **??**.

For iii) we clearly have $(P, \mathcal{T} \models c \rightarrow A) \Rightarrow (P^*, \mathcal{T} \models c \rightarrow A)$. We show the contrapositive of the opposite, $(P, \mathcal{T} \not\models c \rightarrow A) \Rightarrow (P^*, \mathcal{T} \not\models c \rightarrow A)$.

Let I be a model of P and T, based on a structure $\mathcal{X}$, let ρ be a valuation such that $I \models \neg A\rho$ and $\mathcal{X} \models c\rho$.

We have $M_P^{\mathcal{X}} \models \neg A\rho$, thus $M_{P^*}^{\mathcal{X}} \models \neg A\rho$, and as $\mathcal{T} \models c\rho$, we conclude that $P^*, \mathcal{T} \not\models c \rightarrow A$.

The proof of ii) is identical, the structure $\mathcal{X}$ being fixed. $\square$

Soundness of Negation as Finite Failure

Theorem 7 *If G is finitely failed then $P^*, \mathcal{T} \models \neg G$.*

Proof: By induction on the height h of the tree in finite failure for $G = c|A, \alpha$ where A is the selected atom at the root of the tree.

In the base case $h = 1$, the constrained atom $c|A$ has no CSLD transition, we can deduce that $P^*, \mathcal{T} \models \neg(c \wedge A)$ hence that $P^*, \mathcal{T} \models \neg G$.

For the induction step, let us suppose $h > 1$. Let $G_1, ..., G_n$ be the sons of the root and $Y_1, ..., Y_n$ be the respective sets of introduced variables. We have $P^*, \mathcal{T} \models G \leftrightarrow \exists Y_1\ G_1 \vee ... \vee \exists_n\ G_n$. By induction hypothesis, $P^*, \mathcal{T} \models \neg G_i$ for every $1 \leq i \leq n$, therefore $P^*, \mathcal{T} \models \neg G$. $\square$

Completeness of Negation as Failure

Theorem 8 *[JL87] If $P^*, \mathcal{T} \models \neg G$ then G is finitely failed.*

Proof: We show that if G is not finitely failed then $P^*, \mathcal{T}, \exists(G)$ is satisfiable. If G has a success then by 1, $P^*, \mathcal{T} \models \exists G$. Else G has a fair infinite derivation $G = c_0|G_0 \longrightarrow c_1|G1 \longrightarrow \ldots$

For every $i \geq 0$, c_i is $\mathcal{T}$-satisfiable, thus by the compactness theorem, $c_\omega = \bigcup_{i \geq 0} c_i$ is $\mathcal{T}$-satisfiable. Let $\mathcal{X}$ be a model of $\mathcal{T}$ s.t. $\mathcal{X} \models \exists(c_\omega)$. Let $I_0 = \{A\rho \mid A \in G_i$ for some $i \geq 0$ and $\mathcal{X} \models c_\omega \rho\}$. As the derivation is fair, every atom A in I_0 is selected, thus $c_\omega|A \longrightarrow c_\omega|A_1, \ldots, A_n$ with $[c_\omega|A] \cup \ldots \cup [c_\omega|A_n] \subseteq I_0$. We deduce that $I_0 \subseteq T_P^{\mathcal{X}}(I_0)$. By Knaster-Tarski's theorem, the iterated application up to ordinal ω of the operator T_P^S from I_0 leads to a fixed point I s.t. $I_0 \subseteq I$, thus $[c_\omega|G_0] \in I$. Hence $P^*, \exists(G)$ is $\mathcal{X}$-satisfiable, and $P^*, \mathcal{T}, \exists(G)$ is satisfiable. $\square$