

# Constraint Logic Programming

Sylvain Soliman and François Fages  
{Sylvain.Soliman,Francois.Fages}@inria.fr

INRIA – Project-Team CONTRAINTES

MPRI C-2-4-1 Course – September 2009 - February 2010

## Part III

# CLP - Operational and Fixpoint Semantics

## Part III: CLP - Operational and Fixpoint Semantics

- 1 Operational Semantics
  - CSLD resolution
  - Observables
- 2 Fixpoint Semantics
  - Fixpoint Preliminaries
  - Fixpoint Semantics of Successes
  - Fixpoint Semantics of Computed Answers
- 3 Program Analysis
  - Abstract Interpretation
  - Constraint-based Model Checking

# Full abstraction

## Theorem 1 ([JL87])

$$T_P^{\mathcal{X}} \uparrow \omega = O_{gs}(P).$$

$T_P^{\mathcal{X}} \uparrow \omega \subseteq O_{gs}(P)$  is proved by induction on the powers  $n$  of  $T_P^{\mathcal{X}}$ .  $n = 0$ , i.e.  $\emptyset$ , is trivial. Let  $A\rho \in T_P^{\mathcal{X}} \uparrow n$ , there exists a rule  $(A \leftarrow c | A_1, \dots, A_n) \in P$ , s.t.  $\{A_1\rho, \dots, A_n\rho\} \subseteq T_P^{\mathcal{X}} \uparrow n - 1$  and  $\mathcal{X} \models c\rho$ . By induction  $\{A_1\rho, \dots, A_n\rho\} \subseteq O_{gs}(P)$ . By definition of  $O_{gs}$  and  $\wedge$ -compositionality. we get  $A\rho \in O_{gs}(P)$ .

$O_{gs}(P) \subseteq T_P^{\mathcal{X}} \uparrow \omega$  is proved by induction on the length of derivations. Successes with derivation of length 0 are ground facts in  $T_P^{\mathcal{X}} \uparrow 1$ .

Let  $A\rho \in O_{gs}(P)$  with a derivation of length  $n$ . By definition of  $O_{gs}$  there exists  $(A \leftarrow c | A_1, \dots, A_n) \in P$  s.t.  $\{A_1\rho, \dots, A_n\rho\} \subseteq O_{gs}(P)$  and  $\mathcal{X} \models c\rho$ . By induction  $\{A_1\rho, \dots, A_n\rho\} \subseteq T_P^{\mathcal{X}} \uparrow \omega$ . Hence by definition of  $T_P^{\mathcal{X}}$  we get  $A\rho \in T_P^{\mathcal{X}} \uparrow \omega$ .

$T_P^{\mathcal{X}}$  and  $\mathcal{X}$ -models

## Proposition 2

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

## Proof.

*$I$  is a  $\mathcal{X}$ -model of  $P$ ,  
iff for each clause  $A \leftarrow c \mid A_1, \dots, A_n \in P$  and for each  $\mathcal{X}$ -valuation  $\rho$ , if  $\mathcal{X} \models c\rho$  and  $\{A_1\rho, \dots, A_n\rho\} \subseteq I$  then  $A\rho \in I$ ,  
iff  $T_P^{\mathcal{X}}(I) \subseteq I$ .* □

$T_P^{\mathcal{X}}$  and  $\mathcal{X}$ -modelsTheorem 3 (Least  $\mathcal{X}$ -model [JL87])

Let  $P$  be a constraint logic program on  $\mathcal{X}$ .  $P$  has a *least  $\mathcal{X}$ -model*, denoted by  $M_P^{\mathcal{X}}$  satisfying:

$$M_P^{\mathcal{X}} = T_P^{\mathcal{X}} \uparrow \omega$$

Proof.

$T_P^{\mathcal{X}} \uparrow \omega = \text{lfp}(T_P^{\mathcal{X}})$  is also the least post-fixed point of  $T_P^{\mathcal{X}}$ , thus by Prop. 2,  $\text{lfp}(T_P^{\mathcal{X}})$  is the least  $\mathcal{X}$ -model of  $P$ . □

## Part IV

# Logical Semantics

## Part IV: Logical Semantics

- 4 Logical Semantics of CLP( $\mathcal{X}$ )
  - Soundness
  - Completeness
- 5 Automated Deduction
  - Proofs in Group Theory
- 6 CLP( $\lambda$ )
  - $\lambda$ -calculus
  - Proofs in  $\lambda$ -calculus
- 7 Negation as Failure
  - Finite Failure
  - Clark's Completion
  - Soundness w.r.t. Clark's Completion
  - Completeness w.r.t. Clark's Completion

## Soundness of CSLD Resolution

### Theorem 4 ([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  $P, \mathcal{T} \models c \supset G$ .*

If  $G = (d | A_1, \dots, A_n)$ , we deduce from the  $\wedge$ -compositionality lemma, that there exist computed answers  $c_1, \dots, c_n$  for the goals  $A_1, \dots, A_n$  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 the full abstraction Thm  $[c_i | A_i]_{\mathcal{X}} \subseteq M_P^{\mathcal{X}}$ , 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  $P \models_{\mathcal{X}} \forall (c \supset (d \wedge A_1 \wedge \dots \wedge A_n))$ , and as the same reasoning applies to any model  $\mathcal{X}$  of  $\mathcal{T}$ ,  $P, \mathcal{T} \models \forall (c \supset (d \wedge A_1 \wedge \dots \wedge A_n))$

# Completeness of CSLD resolution

## Theorem 5 ([Mah87])

*If  $M_P^{\mathcal{X}} \models c \supset G$  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  $\rho$  of  $c$ , for every atom  $A_j$  in  $G$ ,  
 $M_P^{\mathcal{X}} \models A_j \rho$  iff  $A_j \rho \in T_P^{\mathcal{X}} \uparrow \omega$ , iff  $A_j \rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$   
 iff  $c_{j,\rho} | A_j \in S_P^{\mathcal{X}} \uparrow \omega$ , for some constraint  $c_{j,\rho}$  s.t.  $\rho$  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  $A_j$  and  $\mathcal{X} \models \exists Y_{j,\rho} c_{j,\rho}$ .  
 Let  $c_\rho$  be the conjunction of  $c_{j,\rho}$  for all  $j$ .  $c_\rho$  is a computed answer for  $G$ .  
 By taking the collection of  $c_\rho$  for all  $\rho$  we get  $\mathcal{X} \models \forall(c \supset \bigvee_{c_\rho} \exists Y_\rho c_\rho)$



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

### Theorem 6 ([Mah87])

If  $P, \mathcal{T} \models c \supset G$  then there exists a finite set  $\{c_1, \dots, c_n\}$  of computed answers to  $G$ , such that:  
 $\mathcal{T} \models \forall (c \supset \exists Y_1 c_1 \vee \dots \vee \exists Y_n c_n)$ .

### Proof.

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

## Negation as 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  $G$  is **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  $M_P^{\mathcal{X}}$  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 CSLD derivation with the program  $P$
- `fail(P,B)` which succeeds iff  $M_P \models \neg \exists B$

## Undecidability of $M_P^{\lambda}$

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

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 both exist.**

## Clark's completion

The **Clark's completion** of  $P$  is the set  $P^*$  of formulas of the form  $\forall X p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A_1^1 \wedge \dots \wedge A_{n_1}^1) \vee \dots \vee (\exists Y_k c_k \wedge A_1^k \wedge \dots \wedge A_{n_k}^k)$  where the  $p(X) \leftarrow c_i | A_1^i, \dots, A_{n_i}^i$  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 7

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

Clark's completion  $P^* =$

## Clark's completion

The **Clark's completion** of  $P$  is the set  $P^*$  of formulas of the form  $\forall X p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A_1^1 \wedge \dots \wedge A_{n_1}^1) \vee \dots \vee (\exists Y_k c_k \wedge A_1^k \wedge \dots \wedge A_{n_k}^k)$  where the  $p(X) \leftarrow c_i | A_1^i, \dots, A_{n_i}^i$  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 7

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

## Clark's completion

The **Clark's completion** of  $P$  is the set  $P^*$  of formulas of the form  $\forall X p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A_1^1 \wedge \dots \wedge A_{n_1}^1) \vee \dots \vee (\exists Y_k c_k \wedge A_1^k \wedge \dots \wedge A_{n_k}^k)$  where the  $p(X) \leftarrow c_j | A_1^j, \dots, A_{n_j}^j$  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 7

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 8

*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.

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $P$

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $\forall X p(X) \leftarrow \phi_1 \vee \dots \vee \phi_k$  for every formula

$\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  in  $P^*$ ,

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $\forall X p(X) \leftarrow \phi_1 \vee \dots \vee \phi_k$  for every formula

$\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  in  $P^*$ ,

iff  $I$  is a post-fixed point of  $T_P^{\mathcal{X}}$ , i.e.  $T_P^{\mathcal{X}}(I) \subseteq I$  (by Prop. 2).

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $\forall X p(X) \leftarrow \phi_1 \vee \dots \vee \phi_k$  for every formula  $\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  in  $P^*$ ,

iff  $I$  is a post-fixed point of  $T_P^{\mathcal{X}}$ , i.e.  $T_P^{\mathcal{X}}(I) \subseteq I$  (by Prop. 2).

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

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

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $\forall X p(X) \leftarrow \phi_1 \vee \dots \vee \phi_k$  for every formula  $\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  in  $P^*$ ,

iff  $I$  is a post-fixed point of  $T_P^{\mathcal{X}}$ , i.e.  $T_P^{\mathcal{X}}(I) \subseteq I$  (by Prop. 2).

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

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

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

## Supported $\mathcal{X}$ -models

### Proposition 8

*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  $\forall X p(X) \leftarrow \phi_1 \vee \dots \vee \phi_k$  for every formula  $\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  in  $P^*$ ,

iff  $I$  is a post-fixed point of  $T_P^{\mathcal{X}}$ , i.e.  $T_P^{\mathcal{X}}(I) \subseteq I$  (by Prop. 2).

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

iff  $I$  is a  $\mathcal{X}$ -model of  $\forall X p(X) \rightarrow \phi_1 \vee \dots \vee \phi_k$  for every formula  $\forall X p(X) \leftrightarrow \phi_1 \vee \dots \vee \phi_k$  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  $T_P^{\mathcal{X}}$ . □

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$*
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,*
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .*

Proof.

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$*
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,*
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .*

### Proof.

i) is an immediate corollary of full abstraction and least  $\mathcal{X}$ -model theorems (1 and 3).

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$*
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,*
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .*

### Proof.

i) is an immediate corollary of full abstraction and least  $\mathcal{X}$ -model theorems (1 and 3).

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

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$*
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,*
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .*

### Proof.

i) is an immediate corollary of full abstraction and least  $\mathcal{X}$ -model theorems (1 and 3).

For iii) we clearly have  $(P, \mathcal{T} \models c \supset A) \Rightarrow (P^*, \mathcal{T} \models c \supset A)$ . We show the contrapositive of the opposite,  $(P, \mathcal{T} \not\models c \supset A) \Rightarrow (P^*, \mathcal{T} \not\models c \supset A)$ . Let  $I$  be a model of  $P$  and  $\mathcal{T}$ , based on a structure  $\mathcal{X}$ , let  $\rho$  be a valuation such that  $I \models \neg A\rho$  and  $\mathcal{X} \models c\rho$ .

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .

### Proof.

i) is an immediate corollary of full abstraction and least  $\mathcal{X}$ -model theorems (1 and 3).

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

Let  $I$  be a model of  $P$  and  $\mathcal{T}$ , based on a structure  $\mathcal{X}$ , let  $\rho$  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  $P^*, \mathcal{T} \not\models c \supset A$ .

## Models of the Clark's completion

### Theorem 9

- i)  $P^*$  has the same least  $\mathcal{X}$ -model than  $P$ ,  $M_P^{\mathcal{X}} = M_{P^*}^{\mathcal{X}}$
- ii)  $P \models_{\mathcal{X}} c \supset A$  iff  $P^* \models_{\mathcal{X}} c \supset A$ , for all  $c$  and  $A$ ,
- iii)  $P, \mathcal{T} \models c \supset A$  iff  $P^*, \mathcal{T} \models c \supset A$ .

### Proof.

i) is an immediate corollary of full abstraction and least  $\mathcal{X}$ -model theorems (1 and 3).

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

Let  $I$  be a model of  $P$  and  $\mathcal{T}$ , based on a structure  $\mathcal{X}$ , let  $\rho$  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  $P^*, \mathcal{T} \not\models c \supset A$ .

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

## Soundness of Negation as Finite Failure

### Theorem 10

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

Proof.

## Soundness of Negation as Finite Failure

### Theorem 10

*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.

## Soundness of Negation as Finite Failure

### Theorem 10

*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  $P^*, \mathcal{T} \models \neg G$ .

## Soundness of Negation as Finite Failure

### Theorem 10

*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  $G_1, \dots, G_n$  be the sons of the root and  $Y_1, \dots, Y_n$  be the respective sets of introduced variables.

We have  $P^*, \mathcal{T} \models G \leftrightarrow \exists Y_1 G_1 \vee \dots \vee \exists Y_n G_n$ . 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 11 ([JL87])

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

## Completeness of Negation as Failure

### Theorem 11 ([JL87])

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

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable.

## Completeness of Negation as Failure

### Theorem 11 ([JL87])

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

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable. If  $G$  has a success then by the soundness of CSLD resolution,  $P^*, \mathcal{T} \models \exists G$ .

## Completeness of Negation as Failure

### Theorem 11 ([JL87])

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

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable. If  $G$  has a success then by the soundness of CSLD resolution,  $P^*, \mathcal{T} \models \exists G$ . Else  $G$  has a fair infinite derivation  $G = c_0 | G_0 \longrightarrow c_1 | G_1 \longrightarrow \dots$   
For every  $i \geq 0$ ,  $c_i$  is  $\mathcal{T}$ -satisfiable,

## Completeness of Negation as Failure

### Theorem 11 ([JL87])

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

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable. If  $G$  has a success then by the soundness of CSLD resolution,  $P^*, \mathcal{T} \models \exists G$ . Else  $G$  has a fair infinite derivation  $G = c_0 | G_0 \longrightarrow c_1 | G_1 \longrightarrow \dots$

For every  $i \geq 0$ ,  $c_i$  is  $\mathcal{T}$ -satisfiable, thus by the **compactness theorem**,  $c_\omega = \bigwedge_{i \geq 0} c_i$  is  $\mathcal{T}$ -satisfiable.

# Completeness of Negation as Failure

## Theorem 11 ([JL87])

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

We show that if  $G$  is not finitely failed then  $P^*, \mathcal{T}, \exists(G)$  is satisfiable. If  $G$  has a success then by the soundness of CSLD resolution,  $P^*, \mathcal{T} \models \exists G$ . Else  $G$  has a fair infinite derivation  $G = c_0 | G_0 \longrightarrow c_1 | G_1 \longrightarrow \dots$

For every  $i \geq 0$ ,  $c_i$  is  $\mathcal{T}$ -satisfiable, thus by the **compactness theorem**,  $c_\omega = \bigwedge_{i \geq 0} c_i$  is  $\mathcal{T}$ -satisfiable. Let  $\mathcal{X}$  be a model of  $\mathcal{T}$  s.t.  $\mathcal{X} \models \exists(c_\omega)$ . Let  $l_0 = \{A\rho \mid A \in G_i \text{ for some } i \geq 0 \text{ and } \mathcal{X} \models c_\omega\rho\}$ . As the derivation is fair, every atom  $A$  in  $l_0$  is selected, thus  $c_\omega | A \longrightarrow c_\omega | A_1, \dots, A_n$  with  $[c_\omega | A] \cup \dots \cup [c_\omega | A_n] \subseteq l_0$ . We deduce that  $l_0 \subseteq T_P^{\mathcal{X}}(l_0)$ . By Knaster-Tarski's theorem, the iterated application up to ordinal  $\omega$  of the operator  $T_P^{\mathcal{X}}$  from  $l_0$  leads to a fixed point  $l$  s.t.  $l_0 \subseteq l$ , thus  $[c_\omega | G_0] \in l$ . Hence  $P^*, \exists(G)$  is  $\mathcal{X}$ -satisfiable, and  $P^*, \mathcal{T}, \exists(G)$  is satisfiable.

## Part V

# Practical CLP Programming

# The Warren Abstract Machine

First Prolog implementation in the early 70's (by Colmerauer et al.).

In 1983, David H. Warren creates the **Warren Abstract Machine**.

Remains the state of the art (for term representation, basic instructions, ...)

Slightly extended for CLP

(C)SLD resolution seen as a call stack (with marks for choice points)

# The Warren Abstract Machine

First Prolog implementation in the early 70's (by Colmerauer et al.).

In 1983, David H. Warren creates the **Warren Abstract Machine**.

Remains the state of the art (for term representation, basic instructions, . . . )

Slightly extended for CLP (**constraints instead of substitutions**)

(C)SLD resolution seen as a call stack (with marks for choice points)

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity,

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

**Tail recursion** can be optimized,

## Optimizations from the WAM

Search for predicates should be almost in constant time

Use a hash table - **indexing** - for the predicate name/arity, and the functor of the first argument

Each call normally adds a frame to the call stack (removed on backtracking)

As for other programming paradigms, not always necessary

**Tail recursion** can be optimized, when calling and called contexts are **deterministic**.

## Putting it all together

### Naive sum

```
sum([], 0).  
sum([H | T], S) :-  
    sum(T, S1),  
    S is S1 + H.
```

## Putting it all together

### Naive sum

```
sum([], 0).  
sum([H | T], S) :-  
    sum(T, S1),  
    S is S1 + H.
```

### Much better

```
sum(L, S) :-  
    sum_aux(L, 0, S).  
  
sum_aux([], S, S).  
sum_aux([H | T], S0, S) :-  
    S1 is S0 + H,  
    sum_aux(T, S1, S).
```

## Putting it all together

If numbers are coded as the fact `number(X)`?

```
sum(S) :- findall(X, number(X), L), sum(L, S).
```

## Putting it all together

If numbers are coded as the fact number(X)?

```
sum(S) :- findall(X, number(X), L), sum(L, S).
```

```
sum(S) :-  
    g_assign(sum, 0),  
    (  
        number(N),  
        g_read(sum, S1),  
        S2 is S1 + N,  
        g_assign(sum, S2),  
        fail  
    );  
    g_read(sum, S)  
).
```

# Bibliography I



Maurizio Gabbrielli and Giorgio Levi.

Modeling answer constraints in constraint logic programs.

In K. Furukawa, editor, *Proceedings of ICLP'91, International Conference on Logic Programming*, pages 238–252, Cambridge, 1991. MIT Press.



Joxan Jaffar and Jean-Louis Lassez.

Constraint logic programming.

In *Proceedings of the 14th ACM Symposium on Principles of Programming Languages, Munich, Germany*, pages 111–119. ACM, January 1987.



Michael J. Maher.

Logic semantics for a class of committed-choice programs.

In *Proceedings of ICLP'87, International Conference on Logic Programming*. MIT Press, 1987.