Constraint Logic Programming

Sylvain Soliman

Sylvain.Soliman@inria.fr

Project-Team LIFEWARE

MPRI 2.35.1 Course – September–November 2017

Part I: CLP - Introduction and Logical Background

1. The Constraint Programming paradigm
2. Examples and Applications
3. First Order Logic
4. Models
5. Logical Theories

Part II: Constraint Logic Programs

6. Constraint Languages
7. CLP($\mathcal{X}$)
8. CLP($\mathcal{H}$)
9. CLP($\mathcal{R}, \mathcal{FD}, \mathcal{B}$)

Part III: CLP - Operational and Fixpoint Semantics

Full abstraction

Theorem 1 ([JL87popl])

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

$T_P^{\mathcal{X}} \uparrow \omega \subset O_{gs}(P)$ 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 $(A \leftarrow c|A_1, \ldots, A_n) \in P$, s.t. $\{A_1\rho, \ldots, A_n\rho\} \subset T_P^{\mathcal{X}} \uparrow n-1$ and $\mathcal{X} \models c\rho$. By induction $\{A_1\rho, \ldots, A_n\rho\} \subset O_{gs}(P)$. By definition of O_{gs} and $\wedge$-compositionality. we get $A\rho \in O_{gs}(P)$.
$O_{gs}(P) \subset T_P^{\mathcal{X}} \uparrow \omega$ 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 $(A \leftarrow c|A_1, \ldots, A_n) \in P$ s.t. $\{A_1\rho, \ldots, A_n\rho\} \subset O_{gs}(P)$ and $\mathcal{X} \models c\rho$. By induction $\{A_1\rho, \ldots, A_n\rho\} \subset T_P^{\mathcal{X}} \uparrow \omega$. 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 $T_P^{\mathcal{X}}$, $T_P^{\mathcal{X}}(I) \subset I$

Proof.

I is a $\mathcal{X}$-model of P,
iff for each clause $A \leftarrow c|A_1, \ldots, A_n \in P$ and for each
$\mathcal{X}$-valuation ρ, if $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I$ then $A\rho \in I$,
iff $T_P^{\mathcal{X}}(I) \subset I$ □

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

Theorem 3 (Least $\mathcal{X}$-model [JL87popl])

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.

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

Relating $S_P^{\mathcal{X}}$ and $T_P^{\mathcal{X}}$ operators

Theorem 4 ([JL87popl])

For every ordinal α, $T_P^{\mathcal{X}} \uparrow \alpha = [S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}}$

Proof.

The base case $\alpha = 0$ is trivial. For a successor ordinal, we have

$$
\begin{aligned}
[S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} &= [S_P^{\mathcal{X}}(S_P^{\mathcal{X}} \uparrow \alpha - 1)]_{\mathcal{X}} \\
&= T_P^{\mathcal{X}}([S_P^{\mathcal{X}} \uparrow \alpha - 1]_{\mathcal{X}}) \\
&= T_P^{\mathcal{X}}(T_P^{\mathcal{X}} \uparrow \alpha - 1) \text{ by induction} \\
&= T_P^{\mathcal{X}} \uparrow \alpha
\end{aligned}
$$

For a limit ordinal, we have

$$
\begin{aligned}
[S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} &= [\bigcup_{\beta<\alpha} S_P^{\mathcal{X}} \uparrow \beta]_{\mathcal{X}} \\
&= \bigcup_{\beta<\alpha} [S_P^{\mathcal{X}} \uparrow \beta]_{\mathcal{X}} \\
&= \bigcup_{\beta<\alpha} T_P^{\mathcal{X}} \uparrow \beta \text{ by induction} \\
&= T_P^{\mathcal{X}} \uparrow \alpha
\end{aligned}
$$

□

Full abstraction w.r.t. computed answers

Theorem 5 (Theorem of full abstraction [GL91iclp])

$O_{ca}(P) = S_P^{\mathcal{X}} \uparrow \omega$

$S_P^{\mathcal{X}} \uparrow \omega \subset O_{ca}(P)$ is proved by induction on the powers n of $S_P^{\mathcal{X}}$. $n = 0$ is trivial. Let $c|A \in S_P^{\mathcal{X}} \uparrow n$, there exists a rule $(A \leftarrow d|A_1, \ldots, A_n) \in P$, s.t. $\{c_1|A_1, \ldots, c_n|A_n\} \subset S_P^{\mathcal{X}} \uparrow n-1$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$ and $\mathcal{X} \models \exists c$. By induction $\{c_1|A_1, \ldots, c_n|A_n\} \subset O_{ca}(P)$. By definition of O_{ca} we get $c|A \in O_{ca}(P)$.
$O_{ca}(P) \subset S_P^{\mathcal{X}} \uparrow \omega$ is proved by induction on the length of derivations. Successes with derivation of length 0 are facts in $S_P^{\mathcal{X}} \uparrow 1$. Let $c|A \in O_{ca}(P)$ with a derivation of length n. By definition of O_{ca} there exists $(A \leftarrow d|A_1, \ldots, A_n) \in P$ s.t. $\{c_1|A_1, \ldots, c_n|A_n\} \subset O_{ca}(P)$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$ and $\mathcal{X} \models \exists c$. By induction $\{c_1|A_1, \ldots, c_n|A_n\} \subset S_P^{\mathcal{X}} \uparrow \omega$. Hence by definition of $S_P^{\mathcal{X}}$ we get $c|A \in S_P^{\mathcal{X}} \uparrow \omega$.

Constraint-based Model Checking [DP99tacas]

Analysis of unbounded states concurrent systems by CLP programs.

Concurrent transition systems defined by condition-action rules [Shankar93acm]:

$$\text{condition } \phi(\vec{x}) \text{ action } \vec{x}' = \psi(\vec{x})$$

Translation into CLP clauses over one predicate p (for states)

$$p(\vec{x}) \leftarrow \phi(\vec{x}),\ \psi(\vec{x}', \vec{x}),\ p(\vec{x}').$$

The transitions of the concurrent system are in one-to-one correspondance to the CSLD derivations of the CLP program.

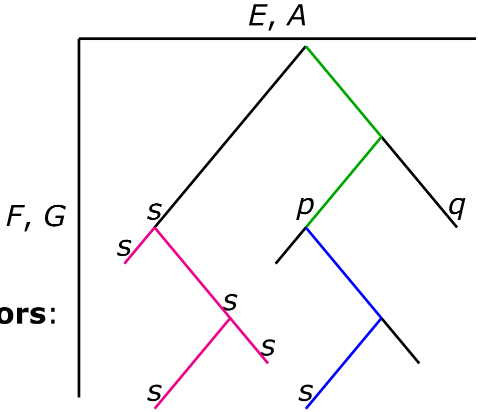
Proposition 6

The set of states from which a set of states defined by a constraint c is reachable is the set $lfp(T_P)$ where P is the CLP program plus the clause $p(\vec{x}) \leftarrow c(\vec{x})$.

Computation Tree Logic CTL

Temporal logic for branching time:

- States described by propositional or first-order formulas
- Two **path quantifiers** for non-determinism:
  - *A* “for all paths”
  - *E* “for some path”
- Several **temporal operators**:
  - *X* “next time”,
  - *F* “eventually”,
  - *G* “always”,
  - *U* “until”.



Model Checking

Two types of interesting properties:
$AG\neg\phi$ "Safety" property.
$AF\psi$ "Liveness" property.

Duality: for any formula ϕ we have
$EF\phi = \neg AG\neg\phi$ and
$EG\phi = \neg AF\neg\phi$.

Model checking is an algorithm for computing, in a given Kripke structure $K = (S, I, R)$, $I \subset S, R \subset S \times S$ (S is the set of states, I the initial states and R the transition relation), the set of states which satisfy a given CTL formula ϕ, i.e., the set $\{s \in S | K, s \models \phi\}$.

(Symbolic) Model Checking

Basic algorithm

When S is finite, represent K as a graph, and iteratively label the nodes with the subformulas of ϕ which are true in that node.

Add A to the states satisfying A ($\neg A$, $A \wedge B$,...)

Add $EF\phi$ ($EX\phi$) to the (immediate) predecessors of states labeled by ϕ

Add $E(\phi U \psi)$ to the predecessor states of ψ while they satisfy ϕ

Add $EG\phi$ to the states for which there exists a path leading to a non trivial strongly connected components of the subgraph restricted to the states satisfying ϕ

Symbolic model checking

Use OBDD's to represent states and transitions as boolean formulas (S is finite).

Constraint-based Model Checking

Constraint-based model checking [DP99tacas] applies to Kripke structures with an infinite set of states.
Numerical constraints provide a finite representation for an infinite set of states.

Constraint logic programming theory:

$$EF(\phi) = lfp(T_{R \cup \{p(\vec{x}) \leftarrow \phi\}})$$

$$EG(\phi) = gfp(T_{R \wedge \phi})$$

Prototype implementation *DMC* in Sicstus Prolog + Simplex, CLP($\mathcal{H}, \mathcal{FD}, \mathcal{R}, \mathcal{B}$)

Part IV

Logical Semantics

Part IV: Logical Semantics

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

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

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $P, \mathcal{T} \models c \supset G$.

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $G = (d|A_1, \ldots, A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, \ldots, c_n$ for the goals $A_1, \ldots, A_n$ 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 5,

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $G = (d|A_1, \ldots, A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, \ldots, c_n$ for the goals $A_1, \ldots, A_n$ 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 5,
$[c_i|A_i]_{\mathcal{X}} \subset M_P^{\mathcal{X}}$, by Thm. 4, and Thm. 3, hence $M_P^{\mathcal{X}} \models \forall(c_i \supset A_i)$,

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $G = (d|A_1, \ldots, A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, \ldots, c_n$ for the goals $A_1, \ldots, A_n$ 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 5,
$[c_i|A_i]_{\mathcal{X}} \subset M_P^{\mathcal{X}}$, by Thm. 4, and Thm. 3, 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,

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $G = (d|A_1, \ldots, A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, \ldots, c_n$ for the goals $A_1, \ldots, A_n$ 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 5,
$[c_i|A_i]_{\mathcal{X}} \subset M_P^{\mathcal{X}}$, by Thm. 4, and Thm. 3, 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, $1 \leq i \leq n$.

Soundness of CSLD Resolution

Theorem 7 ([JL87popl])

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 $G = (d|A_1, \ldots, A_n)$, we deduce from the $\wedge$-compositionality lemma, that there exist computed answers $c_1, \ldots, c_n$ for the goals $A_1, \ldots, A_n$ 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 5,
$[c_i|A_i]_{\mathcal{X}} \subset M_P^{\mathcal{X}}$, by Thm. 4, and Thm. 3, 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 $P \models_{\mathcal{X}} \forall(c \supset (d \wedge A_1 \wedge \cdots \wedge A_n))$,
and as the same reasoning applies to any model $\mathcal{X}$ of $\mathcal{T}$,
$P, \mathcal{T} \models \forall(c \supset (d \wedge A_1 \wedge \cdots \wedge A_n))$

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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.

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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 ρ of c, for every atom A_j in G,

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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 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 Thm. 3, iff $A_j \rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$ by Thm. 4,

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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 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 Thm. 3, iff $A_j\rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$ by Thm. 4,
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 $Y_{j,\rho} = V(c_{j,\rho}) \setminus V(A_j)$,

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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 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 Thm. 3, iff $A_j\rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$ by Thm. 4,
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 5) and $\mathcal{X} \models \exists Y_{j,\rho} c_{j,\rho} \rho$.

Completeness of CSLD resolution

Theorem 8 ([Maher87iclp])

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 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 Thm. 3, iff $A_j\rho \in [S_P^{\mathcal{X}} \uparrow \omega]_{\mathcal{X}}$ by Thm. 4,
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 5) 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 $\mathcal{X} \models \forall(c \supset \bigvee_{c_\rho} \exists Y_\rho c_\rho)$ □

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

Theorem 9 ([Maher87iclp])

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

Proof.

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

Theorem 9 ([Maher87iclp])

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

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 $\{c_i\}_{i\geq 1}$ be the set of these computed answers. Then for every model $\mathcal{X}$ and for every $\mathcal{X}$-valuation ρ, $\mathcal{X} \models c \supset \bigvee_{i\geq 1} \exists Y_i c_i$,

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

Theorem 9 ([Maher87iclp])

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

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 $\{c_i\}_{i\geq 1}$ 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 $\mathcal{T} \models c \supset \bigvee_{i\geq 1} \exists Y_i c_i$,

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

Theorem 9 ([Maher87iclp])

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

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 $\{c_i\}_{i\geq 1}$ 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. $\mathcal{T} \models c \supset \bigvee_{i=1}^{n} \exists Y_i c_i$. □

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 $x * y = z$.

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 $A \longrightarrow^* \square$.

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

```
clause(p(e,X,X)).
```

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)).
```

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,

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 $\neg p(i(i(a)), e, a)$

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
```

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
```

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

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 $i(a)$.

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 $i(a)$.

```
| ?- solve(s(e)).
depth 4
yes
| ?- solve(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

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

Equality is decidable, but not unification…

Theorem proving in CLP(λ)

Theorem 10 (Cantor's Theorem)

$\mathbb{N}^{\mathbb{N}}$ *is not countable.*

Proof.

By two steps of CSLD resolution!

Theorem proving in CLP(λ)

Theorem 10 (Cantor's Theorem)

$\mathbb{N}^{\mathbb{N}}$ is not countable.

Proof.

By two steps of CSLD resolution!
Let us suppose $\exists h : \mathbb{N} \to (\mathbb{N} \to \mathbb{N})\ \forall f : \mathbb{N} \to \mathbb{N}\ \exists n : \mathbb{N}\ h(n) = f$
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 $\quad G \neq H \leftarrow G(N) \neq H(N).$

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

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! □

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 $M_P^{\mathcal{X}}$

```
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
$\forall X\, p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A^1_1 \wedge \cdots \wedge A^1_{n_1}) \vee \cdots \vee (\exists Y_k c_k \wedge A^k_1 \wedge \cdots \wedge A^k_{n_k})$
where the $p(X) \leftarrow c_i | A^i_1, \ldots, A^i_{n_i}$ 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 11

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
$\forall X\, p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A^1_1 \wedge \cdots \wedge A^1_{n_1}) \vee \cdots \vee (\exists Y_k c_k \wedge A^k_1 \wedge \cdots \wedge A^k_{n_k})$
where the $p(X) \leftarrow c_i | A^i_1, \ldots, A^i_{n_i}$ 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 11

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

$\forall X\ p(X) \leftrightarrow (\exists Y_1 c_1 \wedge A^1_1 \wedge \cdots \wedge A^1_{n_1}) \vee \cdots \vee (\exists Y_k c_k \wedge A^k_1 \wedge \cdots \wedge A^k_{n_k})$

where the $p(X) \leftarrow c_i | A^i_1, \ldots, A^i_{n_i}$ 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 11

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 12

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 12

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 12

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

Supported $\mathcal{X}$-models

Proposition 12

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 $\forall X\, p(X) \leftarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula $\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a post-fixed point of $T_P^{\mathcal{X}}$, i.e., $.T_P^{\mathcal{X}}(I) \subset I$ (by Prop. 2).

Supported $\mathcal{X}$-models

Proposition 12

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 $\forall X\, p(X) \leftarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula $\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a post-fixed point of $T_P^{\mathcal{X}}$, i.e., .$T_P^{\mathcal{X}}(I) \subset I$ (by Prop. 2).
I is a supported $\mathcal{X}$-interpretation of P,
iff I is a $\mathcal{X}$-model of $\forall X\, p(X) \rightarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula $\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,

Supported $\mathcal{X}$-models

Proposition 12

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 $\forall X\, p(X) \leftarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula
$\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a post-fixed point of $T_P^{\mathcal{X}}$, i.e., $.T_P^{\mathcal{X}}(I) \subset I$ (by Prop. 2).
I is a supported $\mathcal{X}$-interpretation of P,
iff I is a $\mathcal{X}$-model of $\forall X\, p(X) \rightarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula
$\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a pre-fixed point of $T_P^{\mathcal{X}}$, i.e., $I \subset T_P^{\mathcal{X}}(I)$.

Supported $\mathcal{X}$-models

Proposition 12

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 $\forall X\, p(X) \leftarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula $\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a post-fixed point of $T_P^{\mathcal{X}}$, i.e., $.T_P^{\mathcal{X}}(I) \subset I$ (by Prop. 2).
I is a supported $\mathcal{X}$-interpretation of P,
iff I is a $\mathcal{X}$-model of $\forall X\, p(X) \rightarrow \phi_1 \vee \cdots \vee \phi_k$ for every formula $\forall X\, p(X) \leftrightarrow \phi_1 \vee \cdots \vee \phi_k$ in P^*,
iff I is a pre-fixed point of $T_P^{\mathcal{X}}$, i.e., $I \subset T_P^{\mathcal{X}}(I)$.
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 13

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 13

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 13

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 13

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 ρ be a valuation such that $I \models \neg A\rho$ and $\mathcal{X} \models c\rho$.

Models of the Clark's completion

Theorem 13

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 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 $P^*, \mathcal{T} \not\models c \supset A$.

Models of the Clark's completion

Theorem 13

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 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 $P^*, \mathcal{T} \not\models c \supset A$.
The proof of ii) is identical, the structure $\mathcal{X}$ being fixed. □

Soundness of Negation as Finite Failure

Theorem 14

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

Proof.

Soundness of Negation as Finite Failure

Theorem 14

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 14

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 14

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 $P^*, \mathcal{T} \models \neg G$. □

Completeness of Negation as Failure

Theorem 15 ([JL87popl])

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

Completeness of Negation as Failure

Theorem 15 ([JL87popl])

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 15 ([JL87popl])

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 7 ,
$P^*, \mathcal{T} \models \exists G$.

Completeness of Negation as Failure

Theorem 15 ([JL87popl])

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 7 ,
$P^*, \mathcal{T} \models \exists G$. 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,

Completeness of Negation as Failure

Theorem 15 ([JL87popl])

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 7 , $P^*, \mathcal{T} \models \exists G$. Else G has a fair infinite derivation
$G = c_0|G_0 \longrightarrow c_1|G1 \longrightarrow \dots$
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 15 ([JL87popl])

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 7 , $P^*, \mathcal{T} \models \exists G$. Else G has a fair infinite derivation
$G = c_0|G_0 \longrightarrow c_1|G1 \longrightarrow \dots$
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 $I_0 = \{A\rho \mid A \in G_i$ 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 $c_\omega|A \longrightarrow c_\omega|A_1, \dots, A_n$ with $[c_\omega|A] \cup \dots \cup [c_\omega|A_n] \subset I_0$. We deduce that $I_0 \subset T_P^{\mathcal{X}}(I_0)$. By Knaster-Tarski's theorem, the iterated application up to ordinal ω of the operator $T_P^{\mathcal{X}}$ from I_0 leads to a fixed point I s.t. $I_0 \subset I$, thus $[c_\omega|G_0] \subset I$. Hence $P^*, \exists(G)$ is $\mathcal{X}$-satisfiable, and $P^*, \mathcal{T}, \exists(G)$ is satisfiable.

Part V

Constraint Solving

Part V: Constraint Solving

17 Solving by Rewriting

18 Solving by Domain Reduction

Solving Equality Constraints in $\mathcal{H}$ by Rewriting

Systems of equations Γ:

$$M_1 = N_1 \wedge \cdots \wedge M_n = N_n$$

A system is in solved form if it is of the form

$$x_1 = M_1 \wedge \cdots \wedge x_n = M_n$$

with $n \geq 0$ and $\{x_1, \ldots, x_n\} \cap (V(M_1) \cup \cdots \cup V(M_n)) = \emptyset$

Proposition 16

If Γ is in solved form then $\mathcal{H} \models \exists(\Gamma)$

Idea of the unification algorithm: try to simplify Γ into either a solved form or $\bot$

Herbrand-Robinson's Unification Algorithm

Dec $f(M_1, \ldots, M_n) = f(N_1, \ldots, N_n) \wedge \Gamma$
$\rightarrow M_1 = N_1 \wedge \cdots \wedge M_n = N_n \wedge \Gamma$,

D$\perp$ $f(M_1, \ldots, M_n) = g(N_1, \ldots, N_m) \wedge \Gamma \rightarrow \perp$ if $f \neq g$,

Triv $x = x \wedge \Gamma \rightarrow \Gamma$,

Var $x = M \wedge \Gamma \rightarrow x = M \wedge \Gamma\sigma$
if $x \notin V(M)$, $x \in V(\Gamma)$, $\sigma = \{x \leftarrow M\}$,

V$\perp$ $x = M \wedge \Gamma \rightarrow \perp$
if $x \in V(M)$ and $x \neq M$

Lemma 17 (Validity)

If $\Gamma \longrightarrow \Gamma'$ *then* $CET_{\mathcal{H}} \models \Gamma \supset \Gamma'$

Proof.

Simple application of the axioms for each rule □

Herbrand-Robinson's Unification Algorithm

Lemma 18 (Termination)

The rules terminate

Proof.

Herbrand-Robinson's Unification Algorithm

Lemma 18 (Termination)

The rules terminate

Proof.

Take as complexity measure of Γ, the number of variables in non-solved form, and the size of Γ, ordered lexicographically □

Proposition 19 (Decidability of unification)

CET $\models \exists(\Gamma)$ *iff the irreducible form of* Γ *is a solved form*

Proof.

Herbrand-Robinson's Unification Algorithm

Lemma 18 (Termination)

The rules terminate

Proof.

Take as complexity measure of Γ, the number of variables in non-solved form, and the size of Γ, ordered lexicographically □

Proposition 19 (Decidability of unification)

CET $\models \exists(\Gamma)$ *iff the irreducible form of* Γ *is a solved form*

Proof.

An irreducible form is either $\perp$, in which case Γ is unsatisfiable, or, by case analysis, a solved form, in which case Γ is satisfiable □

Herbrand-Robinson's Unification Algorithm

Corollary 20 (Completeness of CET)

For any equation system Γ*, either CET* $\vdash \exists(\Gamma)$*, or CET* $\vdash \neg\exists(\Gamma)$

Corollary 21

$\mathcal{H} \models \exists(\Gamma)$ *iff CET* $\models \exists(\Gamma)$

Fourier's Alg. for Lin. Ineq. Constraints over $\mathcal{R}$

Check the satisfiability of a system of linear inequalities
$\Sigma_{i=1}^{m} a_i x_i + c \leq \Sigma_{j=1}^{n} b_j y_j + d$
Normal forms: $t \leq x$, $x \leq t$, or $t \leq 0$, where t is linear and $x \notin V(t)$
The normal form of $s \leq t$ w.r.t. x is noted $\overline{s \leq t}^{x}$

- $\Gamma \rightarrow \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} s_i \leq t_j \wedge \Gamma'$
  if $\overline{\Gamma}^{x} = \bigwedge_{i=1}^{n} s_i \leq x \wedge x \leq \bigwedge_{j=1}^{m} t_j \wedge \Gamma'$ where $x \notin V(\Gamma')$,
- $s \leq t \wedge \Gamma \rightarrow \Gamma$ if $s, t \in \mathcal{R}$ and $s \leq t$,
- $s \leq t \wedge \Gamma \rightarrow \bot$ if $s, t \in \mathcal{R}$ and $s > t$

The rules terminate

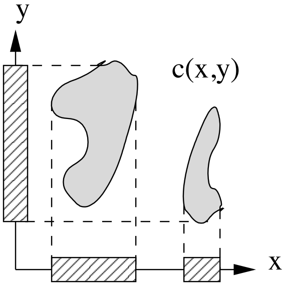
Theorem 22

A system of linear inequalities Γ is satisfiable over $\mathcal{R}$ iff it reduces to the empty system

Constraint Solving by Domain Reduction

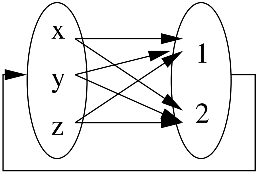
Simple reasoning on the domain of variables for each constraint independently

"Arc consistency": for each constraint c,
for each variable x in c,
for each value e of the domain of x,
there exists a solution of c with $x = e$



Example: $x, y, z \in \{1, 2\}$
System $x \neq y \wedge x \neq z \wedge y \neq z$ arc-consistent

Global constraint `all-different([x,y,z])`
non arc-consistent



Domain Reduction over Finite Domains

$$Sol(\Gamma, \mathcal{FD}) = \{\sigma \mid \sigma = \{x^d \leftarrow v \mid x^d \in V(\Gamma),\ v \in d\},\ \mathcal{FD} \models \Gamma\sigma\}$$

The reduced domain of a variable x^d w.r.t. a basic constraint c is the domain

$$DR(x^d, c) = \{v \in d \mid \mathcal{FD} \models \exists(c[v/x^d])\}$$

A constraint system Γ is arc-consistent if

$$\forall c \in \Gamma\ \forall x^d \in V(c)\ DR(x^d, c) = d$$

Idea of constraint propagation: reduce the domain of variables independently to make the system arc-consistent

Example $a * X \geq b * Y + d$

Simple interval reasoning:

$$aX^{[k,l]} \geq bY^{[m,n]} + d \qquad a, b > 0, d \geq 0$$

Example $a * X \geq b * Y + d$

Simple interval reasoning:

$$aX^{[k,l]} \geq bY^{[m,n]} + d \qquad a, b > 0, d \geq 0$$

we have

$$DR(X^{[k,l]}, c) = [max(k, k'), l]$$

$$DR(Y^{[m,n]}, c) = [m, min(n, n')]$$

where $k' = \lceil \frac{bm+d}{a} \rceil$ and $n' = \lfloor \frac{al-d}{b} \rfloor$

Domain Reduction Algorithm

Fail: $c \wedge \Gamma \rightarrow \perp$ if $x^d \in V(c)$ and $DR(x^d, c) = \emptyset$.

FC: $c \wedge \Gamma \rightarrow \Gamma\sigma$
if $V(c) = \{x^d\}$, $d' = DR(x^d, c)$, $d' \neq \emptyset$, and $\sigma = \{x^d \leftarrow y^{d'}\}$

LA: $c \wedge \Gamma \rightarrow c\sigma \wedge \Gamma\sigma$
if $|V(c)| > 1$,
$x^d \in V(c),\ d' = DR(x^d, c), d' \neq \emptyset, d' \neq d, \sigma = \{x^d \leftarrow y^{d'}\}$

PLA: $c \wedge \Gamma \rightarrow c\sigma \wedge \Gamma\sigma$
if $|V(c)| > 1$, $x^d \in V(c)$, $DR(x^d, c) \subset d' \subsetneq d$, $d' \neq \emptyset$, $\sigma = \{x^d \leftarrow y^{d'}\}$

EL: $c \wedge \Gamma \rightarrow \Gamma$
if $\mathcal{FD} \models c\sigma$ for every valuation σ of the variables in c by values of their domain

Domain Reduction Algorithm (continued)

Lemma 23 (Validity)

If $\Gamma \longrightarrow^*_\sigma \Gamma'$ *then* $Sol(\Gamma, \mathcal{FD}) = \{\sigma\theta \mid \theta \in Sol(\Gamma', \mathcal{FD})\}$.

Proposition 24 (Completeness of LA for 2 var. ineq.)

Let Γ *be a constraint system of the form*

$$aX \geq bY + d \qquad a, b > 0, d \geq 0.$$

Let $\Gamma \longrightarrow^*_\sigma \Gamma' \not\rightarrow$ *Then* Γ *is satisfiable if and only if* $\Gamma' \neq \bot$

Proof.

If $\Gamma' \neq \bot$ is an irreducible form of Γ then for all $c \in \Gamma'$ and $x \in V(c)$ we have $DR(x^d, c) = d$ and $\{x^{[k,l]} \leftarrow k \mid x \in V(\Gamma')\}$ is a solution of Γ' □

CLP($\mathcal{FD}$) scheduling

Simple PERT problem

```
| ?- fd_minimize((B#>=A+5, C#>=B+2, D#>=B+3,
                  E#>=C+5, E#>=D+5), E).
A = 0, B = 5, D = 8, E = 13, C = _#1(7..8) ?
yes
```

Disjunctive scheduling is NP-hard

```
| ?- fd_minimize((B#>=A+5, C#>=B+2, D#>=B+3, E#>=C+5,
                  E#>=D+5, (C#>=D+5 ; D#>=C+5)), E).
A = 0, B = 5, C = 7, D = 12, E = 17 ? ;
no
```

Disjunctive scheduling: bridge problem (4000 nodes)