Constraint Logic Programming

Sylvain Soliman

Sylvain.Soliman@inria.fr

informatics mathematics Inria

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

Compactness theorem

Theorem 1

Corollary 2

$\mathcal{T}$ is consistent iff every finite part of $\mathcal{T}$ is consistent.

$\mathcal{T}$ is inconsistent iff $\mathcal{T} \vdash$ false,
iff for some finite part $\mathcal{T}'$ of $\mathcal{T}$, $\mathcal{T}' \vdash$ false,
iff some finite part of $\mathcal{T}$ is inconsistent $\square$

Compactness theorem

Theorem 1

$\mathcal{T} \models \phi$ iff $\mathcal{T}' \models \phi$ for some finite part $\mathcal{T}'$ of $\mathcal{T}$

By Gödel's completeness theorem, $\mathcal{T} \models \phi$ iff $\mathcal{T} \vdash \phi$.
As the proofs are finite, they use only a finite part of non logical axioms $\mathcal{T}$.
Therefore $\mathcal{T} \models \phi$ iff $\mathcal{T}' \models \phi$ for some finite part $\mathcal{T}'$ of $\mathcal{T}$ □

Corollary 2

$\mathcal{T}$ is consistent iff every finite part of $\mathcal{T}$ is consistent.

$\mathcal{T}$ is inconsistent iff $\mathcal{T} \vdash$ false,
iff for some finite part $\mathcal{T}'$ of $\mathcal{T}$, $\mathcal{T}' \vdash$ false,
iff some finite part of $\mathcal{T}$ is inconsistent □

Part II

Constraint Logic Programs

Part II: Constraint Logic Programs

Linear Programming

- Variables with a continuous domain $\mathbb{R}$

$$A.x \leq B$$

Satisfiability and optimization has polynomial complexity (Simplex algorithm, interior point method)

- Mixed Integer Linear Programming
Variables with a continuous or a discrete domain $\mathbb{Z}$

$$x \in \mathbb{Z} \qquad A.x \leq B$$

NP-hard (Branch and bound, Gomory's cuts,...)

CLP($\mathcal{R}$) mortgage program

```
int(P,T,I,B,M) :- T > 0, T <= 1, B + M = P * (1 + I)
int(P,T,I,B,M) :-
   T > 1, int(P * (1 + I) - M, T - 1, I, B, M).

| ?- int(120000, 120, 0.01, 0, M).
M = 1721.651381 ?
yes
| ?- int(P, 120, 0.01, 0, 1721.651381).
P = 120000 ?
yes
| ?- int(P, 120, 0.01, 0, M).
P = 69.700522*M ?
yes
| ?- int(P, 120, 0.01, B, M).
P = 0.302995*B + 69.700522*M ?
yes
| ?- int(999, 3, Int, 0, 400).
400 = (-400 + (599 + 999*Int) * (1 + Int)) * (1 + Int) ?
```

CLP($\mathcal{R}$) heat equation

```
| ?- X=[[  0,0,0,0,0,0,0,0,0,0,  0],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,_,_,_,_,_,_,_,_,_,100],
        [100,100,100,100,100,100,100,100,100,100,100]],
     laplace(X).

X=[[0,0,0,0,0,0,0,0,0,0,0],
[100,51.11,32.52,24.56,21.11,20.12,21.11,24.56,32.52,51.11,100],
[100,71.91,54.41,44.63,39.74,38.26,39.74,44.63,54.41,71.91,100],
[100,82.12,68.59,59.80,54.97,53.44,54.97,59.80,68.59,82.12,100],
[100,87.97,78.03,71.00,66.90,65.56,66.90,71.00,78.03,87.97,100],
[100,91.71,84.58,79.28,76.07,75.00,76.07,79.28,84.58,91.71,100],
[100,94.30,89.29,85.47,83.10,82.30,83.10,85.47,89.29,94.30,100],
[100,96.20,92.82,90.20,88.56,88.00,88.56,90.20,92.82,96.20,100],
[100,97.67,95.59,93.96,92.93,92.58,92.93,93.96,95.59,97.67,100],
[100,98.89,97.90,97.12,96.63,96.46,96.63,97.12,97.90,98.89,100],
[100,100,100,100,100,100,100,100,100,100,100]] ?
```

CLP($\mathcal{R}$) heat equation

```
laplace([H1, H2, H3 | T]) :-
   laplace_vec(H1, H2, H3), laplace([H2, H3 | T]).
laplace([_, _]).

laplace_vec([TL, T, TR | T1], [ML, M, MR | T2], [BL, B, BR | T3]) :-
   B + T + ML + MR - 4 * M = 0,
   laplace_vec([T, TR | T1], [M, MR | T2], [B, BR | T3]).
laplace_vec([_, _],[_, _],[_, _]).

| ?- laplace([[B11, B12, B13, B14],
              [B21, M22, M23, B24],
              [B31, M32, M33, B34],
              [B41, B42, B43, B44]]).
B12 = -B21 - 4*B31 + 16*M32 - 8*M33 + B34 - 4*B42 + B43,
B13 = -B24 + B31 - 8*M32 + 16*M33 - 4*B34 + B42 - 4*B43,
M22 = -B31 + 4*M32 - M33 - B42,
M23 = -M32 + 4*M33 - B34 - B43 ?
```

CLP($\mathcal{FD}$)= over Finite Domains

Variables $\{x_1, \ldots, x_v\}$
over a finite domain $D = \{e_1, \ldots, e_d\}$

Constraints to satisfy:

- unary constraints of domains $x \in \{e_i, e_j, e_k\}$
- binary constraints: $c(x, y)$
  defined intentionally, $x > y + 2$,
  or extentionally, $\{c(a, b), c(d, c), c(a, d)\}$
- n-ary *global constraints*: $c(x_1, \ldots, x_n)$

CLP($\mathcal{FD}$) send+more=money

```
:- use_module(library(clpfd)).

send(L) :-
   sendc(L),
   label(L).

sendc([S, E, N, D, M, O, R, Y]) :-
   [S, E, N, D, M, O, R, Y] ins 0..9,
   all_different([S, E, N, D, M, O, R, Y]),
   S #\= 0, M #\= 0,
              1000*S + 100*E + 10*N + D
            + 1000*M + 100*O + 10*R + E
   #= 10000*M+1000*O + 100*N + 10*E + Y.
```

```
| ?- send(L).
L = [9, 5, 6, 7, 1, 0, 8, 2] ;
false.
```

CLP($\mathcal{FD}$) send+more=money

```
| ?- sendc([S,E,N,D,M,O,R,Y]).
S = 9,
D = 1,
O = 0,
E = 4..7,
all_different([9, E, N, D, 1, 0, R, Y]),
91*E+D+10*R#=90*N+Y,
N = 5..8,
D = 2..8,
R = 2..8,
Y = 2..8.
```

Part III

CLP - Operational and Fixpoint Semantics

Part III: CLP - Operational and Fixpoint Semantics

Operational semantics: CSLD Resolution

A CLP($\mathcal{X}$) program P is a set of clauses representing inductive definitions of constraints. Taking the solver as a black-box a Constraint Selective Linear Definite clause resolution step is:

A successful derivation is a derivation of the form

$$G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \ldots \longrightarrow c|\Box$$

c is called a for G

Operational semantics: CSLD Resolution

A CLP($\mathcal{X}$) program P is a set of clauses representing inductive definitions of constraints. Taking the solver as a black-box a Constraint Selective Linear Definite clause resolution step is:

$$\frac{}{(c|\alpha, p(s_1, s_2), \alpha') \longrightarrow}$$

A successful derivation is a derivation of the form

$$G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \ldots \longrightarrow c|\Box$$

c is called a for G

Operational semantics: CSLD Resolution

A CLP($\mathcal{X}$) program P is a set of clauses representing inductive definitions of constraints. Taking the solver as a black-box a Constraint Selective Linear Definite clause resolution step is:

$$\frac{(p(t_1, t_2) \leftarrow c' | A_1, \ldots, A_n)\theta \in P}{(c | \alpha, p(s_1, s_2), \alpha') \longrightarrow}$$

where θ is a renaming substitution of the program clause with new variables

A successful derivation is a derivation of the form

$$G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \ldots \longrightarrow c | \Box$$

c is called a for G

Operational semantics: CSLD Resolution

A CLP($\mathcal{X}$) program P is a set of clauses representing inductive definitions of constraints. Taking the solver as a black-box a Constraint Selective Linear Definite clause resolution step is:

$$\frac{(p(t_1,t_2) \leftarrow c'|A_1,\ldots,A_n)\theta \in P \quad \mathcal{X} \models \exists(c \wedge s_1 = t_1 \wedge s_2 = t_2 \wedge c')}{(c|\alpha, p(s_1,s_2), \alpha') \longrightarrow (c, s_1 = t_1, s_2 = t_2, c' \mid \alpha, A_1, \ldots, A_n, \alpha')}$$

where θ is a renaming substitution of the program clause with new variables

A successful derivation is a derivation of the form

$$G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \ldots \longrightarrow c|\square$$

c is called a for G

Operational semantics: CSLD Resolution

A CLP($\mathcal{X}$) program P is a set of clauses representing inductive definitions of constraints. Taking the solver as a black-box a Constraint Selective Linear Definite clause resolution step is:

$$\frac{(p(t_1,t_2) \leftarrow c'|A_1,\ldots,A_n)\theta \in P \quad \mathcal{X} \models \exists(c \wedge s_1 = t_1 \wedge s_2 = t_2 \wedge c')}{(c|\alpha, p(s_1,s_2), \alpha') \longrightarrow (c, s_1 = t_1, s_2 = t_2, c' \mid \alpha, A_1, \ldots, A_n, \alpha')}$$

where θ is a renaming substitution of the program clause with new variables

A successful derivation is a derivation of the form

$$G \longrightarrow G_1 \longrightarrow G_2 \longrightarrow \ldots \longrightarrow c|\Box$$

c is called a computed answer constraint for G

$\wedge$-Compositionality of CSLD-derivations

Lemma 3 ($\wedge$-compositionality)

c is a computed answer for the goal $(d|A_1, \ldots, A_n)$
iff
there exist computed answers $c_1, \ldots, c_n$ *for the goals* $true|A_1, \ldots, true|A_n$*, such that* $c = d \wedge \bigwedge_{i=1}^{n} c_i$ *is satisfiable.*

Corollary 4

Independence of the selection strategy

$\wedge$-Compositionality of CSLD-derivations

Proof.

($\Leftarrow$) $d|A_1,\ldots,A_n \rightarrow^*$

$\wedge$-Compositionality of CSLD-derivations

Proof.

$(\Leftarrow)$ $d|A_1, \ldots, A_n \rightarrow^* d \wedge c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \wedge c_1 \wedge \cdots \wedge c_n|\Box$.

$\wedge$-Compositionality of CSLD-derivations

Proof.

($\Leftarrow$) $d|A_1, \ldots, A_n \rightarrow^* d \wedge c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \wedge c_1 \wedge \cdots \wedge c_n|\Box$.
($\Rightarrow$) By induction on the length l of the derivation

$\wedge$-Compositionality of CSLD-derivations

Proof.

($\Leftarrow$) $d|A_1, \ldots, A_n \rightarrow^* d \wedge c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \wedge c_1 \wedge \cdots \wedge c_n|\Box$.
($\Rightarrow$) By induction on the length l of the derivation
If $l = 1$ we have $true|A_1 \rightarrow c_1|\Box$

$\wedge$-Compositionality of CSLD-derivations

Proof.

($\Leftarrow$) $d|A_1, \ldots, A_n \rightarrow^* d \wedge c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \wedge c_1 \wedge \cdots \wedge c_n|\Box$.
($\Rightarrow$) By induction on the length l of the derivation
If $l = 1$ we have $true|A_1 \rightarrow c_1|\Box$
Otherwise, suppose A_1 is the selected atom, there exists a rule $(A_1 \leftarrow d_1|B_1, \ldots, B_k) \in P$ such that
$d|A_1, \ldots, A_n \rightarrow d \wedge d_1|B_1, \ldots, B_k, A_2, \ldots, A_n \rightarrow^* c|\Box$
By induction, there exist computed answers $e_1, \ldots, e_k, c_2, \ldots, c_n$ for the goals $B_1, \ldots, B_k, A_2, \ldots, A_n$ such that $c = d \wedge d_1 \wedge \bigwedge_{i=1}^{k} e_i \wedge \bigwedge_{j=2}^{n} c_j$. Now let $c_1 = d_1 \wedge \bigwedge_{i=1}^{k} e_i$, c_1 is a computed answer for $true|A_1$ $\Box$

Operational Semantics of CLP($\mathcal{X}$) Programs

Observation of the sets of projected computed answer constraints

$$O(P) = \{(\exists X\ c)|A\ :\ true|A \longrightarrow^* c|\Box,\ \mathcal{X} \models \exists(c),\ X = V(c) \setminus V(A)\}$$

Program equivalence: $P \equiv P'$ iff $O(P) = O(P')$ iff for every goal G, P and P' have same sets of computed answer constraints

Finer observables:
multisets of computed answer constraints
sets of successful CSLD derivations (equivalence of traces)

More abstract observable:
sets of goals having a success
(theorem proving versus programming point of view)

Operational Semantics of CLP($\mathcal{X}$) Programs

Observation of computed answer constraints

$$O_{ca}(P) = \{c|A \ : \ true|A \longrightarrow^* c|\Box, \ \mathcal{X} \models \exists(c)\}$$

$P \equiv_{ca} P'$ iff for every goal G, P and P' have the same sets of computed answer constraints

Observation of ground successes

$$O_{gs}(P) = \{A\rho \in B_{\mathcal{X}} \ : \ true|A \longrightarrow^* c|\Box, \ \mathcal{X} \models c\rho\}$$

$P \equiv_{gs} P'$ iff P and P' have the same ground success sets, iff for every goal G, G has a CSLD refutation in P iff G has one in P'

Some definitions

Let $(S, \leq)$ be a partial order Let $X \subset S$ be a subset of S

- An upper bound of X is an element $a \in S$ such that $\forall x \in X \; x \leq a$
- The maximum element of X, if it exists, is the unique upper bound of X belonging to X
- The least upper bound (lub) of X, if it exists, is the minimum of the upper bounds of X
- A sup-semi-lattice is a partial order such that every finite part admits a lub
- A lattice is a sup-semi-lattice and an inf-semi-lattice
- A chain is an increasing sequence $x_1 \leq x_2 \leq$ ...
- A partial order is complete if every chain admits a lub
- A function $f : S \rightarrow S$ is monotonic if $x \leq y \Rightarrow f(x) \leq f(y)$
- f is continuous if $f(lub(X)) = lub(f(X))$ for every chain X

Fixpoint theorems

Theorem 5 (Knaster-Tarski)

Let $(S, \leq)$ *be a* *complete partial order*, *and* $f : S \to S$ *a continuous operator over* S
Then f *admits a least fixed point* $lfp(f) = f \uparrow \omega$

Proof.

First,

$a \qquad\qquad\qquad = f \uparrow \omega.$

a is a fixed point of f

Let e be any fixed point of f.

hence $a \leq e$ □

Fixpoint theorems

Theorem 5 (Knaster-Tarski)

Let $(S, \leq)$ *be a complete partial order, and* $f : S \to S$ *a continuous operator over* S
Then f *admits a least fixed point* $lfp(f) = f \uparrow \omega$

Proof.

First, as f is continuous, f is monotonic, hence
$\bot \leq f(\bot) \leq f(f(\bot)) \leq \ldots$ forms an increasing chain.
Let $a = lub(\{f^n(\bot) \mid n \in \mathbb{N}\}) = f \uparrow \omega$. By continuity
$f(a) = lub(\{f^{n+1}(\bot) \mid n \in \mathbb{N}\}) = a$, hence a is a fixed point of f

Let e be any fixed point of f.

hence $a \leq e$ □

Fixpoint theorems

Theorem 5 (Knaster-Tarski)

Let $(S, \leq)$ be a complete partial order, and $f : S \to S$ a continuous operator over S
Then f admits a least fixed point $lfp(f) = f \uparrow \omega$

Proof.

First, as f is continuous, f is monotonic, hence $\bot \leq f(\bot) \leq f(f(\bot)) \leq \dots$ forms an increasing chain.
Let $a = lub(\{f^n(\bot) \mid n \in \mathbb{N}\}) = f \uparrow \omega$. By continuity $f(a) = lub(\{f^{n+1}(\bot) \mid n \in \mathbb{N}\}) = a$, hence a is a fixed point of f

Let e be any fixed point of f. We show that for all integer n, $f^n(\bot) \leq e$, by induction on n.

hence $a \leq e$ □

Fixpoint theorems

Theorem 5 (Knaster-Tarski)

Let $(S, \leq)$ *be a complete partial order, and* $f : S \rightarrow S$ *a continuous operator over* S
Then f *admits a least fixed point* $lfp(f) = f \uparrow \omega$

Proof.

First, as f is continuous, f is monotonic, hence
$\perp \leq f(\perp) \leq f(f(\perp)) \leq \ldots$ forms an increasing chain.
Let $a = lub(\{f^n(\perp) \mid n \in \mathbb{N}\}) = f \uparrow \omega$. By continuity
$f(a) = lub(\{f^{n+1}(\perp) \mid n \in \mathbb{N}\}) = a$, hence a is a fixed point of f

Let e be any fixed point of f. We show that for all integer n, $f^n(\perp) \leq e$, by induction on n. Clearly $\perp \leq e$. Furthermore if $f^n(\perp) \leq e$ then by monotonicity, $f^{n+1}(\perp) \leq f(e) = e$.
Thus $f^n(\perp) \leq e$ for all n, hence $a \leq e$ □

Least Post-Fixed Point

Theorem 6

Let $(S, \leq)$ be a complete sup-semi-lattice. Let f be a continuous operator over S. Then f admits a least post-fixed point (i.e., an element e satisfying $f(e) \leq e$) which is equal to $lfp(f)$.

Proof.

Least Post-Fixed Point

Theorem 6

Let $(S, \leq)$ be a complete sup-semi-lattice. Let f be a continuous operator over S. Then f admits a least post-fixed point (i.e., an element e satisfying $f(e) \leq e$) which is equal to $lfp(f)$.

Proof.

Let $g(x) = lub(x, f(x))$.

Least Post-Fixed Point

Theorem 6

Let $(S, \leq)$ be a complete sup-semi-lattice. Let f be a continuous operator over S. Then f admits a least post-fixed point (i.e., an element e satisfying $f(e) \leq e$) which is equal to $lfp(f)$.

Proof.

Let $g(x) = lub(x, f(x))$.
An element e is a post fixed point of f, i.e., $f(e) \leq e$, iff e is a fixed point of g, $g(e) = e$.
Now g is continuous, hence $lfp(g)$ is the least fixed point of g and the least post-fixed point of f.
Furthermore, $lfp(g) = lub\{f^n(\perp)\} = lfp(f)$. □

Fixpoint semantics of O_{gs}

Consider the complete lattice of $\mathcal{X}$-interpretations $(2^{\mathcal{B}_\mathcal{X}}, \subset)$
The bottom element is the empty $\mathcal{X}$-interpretation (all atoms false)
The top element is $\mathcal{B}_\mathcal{X}$ (all atoms true)

A chain X is an increasing sequence $I_1 \subset I_2 \subset \ldots$
$lub(X) = \bigcup_{i \geq 1} I_i$

Let us define the semantics $O_{gs}(P)$ as the least solution of a fixpoint equation over $2^{\mathcal{B}_\mathcal{X}}$: $I = T(I)$

$T_P^{\mathcal{X}}$ immediate consequence operator

$T_P^{\mathcal{X}} : 2^{\mathcal{B}_{\mathcal{X}}} \rightarrow 2^{\mathcal{B}_{\mathcal{X}}}$ is defined by:

$T_P^{\mathcal{X}}(I)$ = $\{A\rho \in \mathcal{B}_{\mathcal{X}} |$ there exists a renamed clause in normal form $(A \leftarrow c | A_1, \ldots, A_n) \in P$, and a valuation ρ s.t. $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I\}$

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 7

$T_P^{\mathcal{H}}(\emptyset)$ =

$T_P^{\mathcal{X}}$ immediate consequence operator

$T_P^{\mathcal{X}} : 2^{\mathcal{B}_\mathcal{X}} \rightarrow 2^{\mathcal{B}_\mathcal{X}}$ is defined by:

$T_P^{\mathcal{X}}(I) = \{A\rho \in \mathcal{B}_\mathcal{X} |$ there exists a renamed clause in normal form $(A \leftarrow c|A_1, \ldots, A_n) \in P$, and a valuation ρ s.t. $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I\}$

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 7

$T_P^{\mathcal{H}}(\emptyset) = \{append([], B, B) \mid B \in \mathcal{H}\}$

$T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset)) =$

$T_P^{\mathcal{X}}$ immediate consequence operator

$T_P^{\mathcal{X}} : 2^{\mathcal{B}_{\mathcal{X}}} \to 2^{\mathcal{B}_{\mathcal{X}}}$ is defined by:

$T_P^{\mathcal{X}}(I) = \{A\rho \in \mathcal{B}_{\mathcal{X}} |$ there exists a renamed clause in normal form $(A \leftarrow c | A_1, \ldots, A_n) \in P$, and a valuation ρ s.t. $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I\}$

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 7

$$T_P^{\mathcal{H}}(\emptyset) = \{append([], B, B) \mid B \in \mathcal{H}\}$$
$$T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset)) = T_P^{\mathcal{H}}(\emptyset) \cup \{append([X], B, [X|B]) \mid X, B \in \mathcal{H}\}$$
$$T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset))) =$$

$T_P^{\mathcal{X}}$ immediate consequence operator

$T_P^{\mathcal{X}} : 2^{\mathcal{B}_{\mathcal{X}}} \rightarrow 2^{\mathcal{B}_{\mathcal{X}}}$ is defined by:

$T_P^{\mathcal{X}}(I) =$ $\{A\rho \in \mathcal{B}_{\mathcal{X}} |$ there exists a renamed clause in normal form $(A \leftarrow c | A_1, \ldots, A_n) \in P$, and a valuation ρ s.t. $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I\}$

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 7

$$
\begin{aligned}
T_P^{\mathcal{H}}(\emptyset) &= \{append([], B, B) \mid B \in \mathcal{H}\} \\
T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset)) &= T_P^{\mathcal{H}}(\emptyset) \cup \{append([X], B, [X|B]) \mid X, B \in \mathcal{H}\} \\
T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset))) &= T_P^{\mathcal{H}}(T_P^{\mathcal{H}}(\emptyset)) \cup \\
&\quad \{append([X, Y], B, [X, Y|B]) \mid X, Y, B \in \mathcal{H}\}
\end{aligned}
$$

Continuity of $T_P^{\mathcal{X}}$ operator

Proposition 8

$T_P^{\mathcal{X}}$ is a continuous operator on the complete lattice of $\mathcal{X}$-interpretations

Proof.

□

Corollary 9

$T_P^{\mathcal{X}}$ admits a least (post) fixed point $T_P^{\mathcal{X}} \uparrow \omega$

Continuity of $T_P^{\mathcal{X}}$ operator

Proposition 8

$T_P^{\mathcal{X}}$ is a continuous operator on the complete lattice of $\mathcal{X}$-interpretations

Proof.

Let X be a chain of $\mathcal{X}$-interpretations. $A\rho \in T_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow c|A_1, \ldots, A_n) \in P$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset lub(X)$,

iff $A\rho \in lub(T_P^{\mathcal{X}}(X))$. □

Corollary 9

$T_P^{\mathcal{X}}$ admits a least (post) fixed point $T_P^{\mathcal{X}} \uparrow \omega$

Continuity of $T_P^{\mathcal{X}}$ operator

Proposition 8

$T_P^{\mathcal{X}}$ is a continuous operator on the complete lattice of $\mathcal{X}$-interpretations

Proof.

Let X be a chain of $\mathcal{X}$-interpretations. $A\rho \in T_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow c|A_1, \ldots, A_n) \in P$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset lub(X)$,
iff $(A \leftarrow c|A_1, \ldots, A_n) \in P$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I$,
for some $I \in X$ (as X is a chain)
iff $A\rho \in T_P^{\mathcal{X}}(I)$ for some $I \in X$, iff $A\rho \in lub(T_P^{\mathcal{X}}(X))$. □

Corollary 9

$T_P^{\mathcal{X}}$ admits a least (post) fixed point $T_P^{\mathcal{X}} \uparrow \omega$

Full abstraction

Theorem 10 ([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 $T_P^{\mathcal{X}}$.

Full abstraction

Theorem 10 ([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.

Full abstraction

Theorem 10 ([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 $A\rho \in T_P^{\mathcal{X}} \uparrow n$,

Full abstraction

Theorem 10 ([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 $\mathcal{X} \models c\rho$.

Full abstraction

Theorem 10 ([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 $(A \leftarrow c | A_1, \dots, A_n) \in P$, s.t. $\{A_1\rho, \dots, A_n\rho\} \subset T_P^{\mathcal{X}} \uparrow n - 1$ and $\mathcal{X} \models c\rho$. By induction $\{A_1\rho, \dots, A_n\rho\} \subset O_{gs}(P)$.

Full abstraction

Theorem 10 ([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 $A\rho \in O_{gs}(P)$.

Full abstraction

Theorem 10 ([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.

Full abstraction

Theorem 10 ([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 $T_P^{\mathcal{X}} \uparrow 1$.

Full abstraction

Theorem 10 ([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

Full abstraction

Theorem 10 ([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 $\mathcal{X} \models c\rho$.

Full abstraction

Theorem 10 ([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 $\{A_1\rho, \ldots, A_n\rho\} \subset T_P^{\mathcal{X}} \uparrow \omega$.

Full abstraction

Theorem 10 ([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 11

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

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

Proposition 11

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 12 (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. 11, $lfp(T_P^{\mathcal{X}})$ is the least $\mathcal{X}$-model of P. □

Fixpoint semantics of O_{ca}

Consider the set of constrained atoms
$\mathcal{B}'_{\mathcal{X}} = \{c|A \ : \ A$ is an atom and $\mathcal{X} \models \exists(c)\}$ modulo renaming

Consider the lattice of constrained interpretations $(2^{\mathcal{B}'_{\mathcal{X}}}, \subset)$

For a constrained interpretation I, let us define the closed $\mathcal{X}$-interpretation:
$[I]_{\mathcal{X}} = \{A\rho \ :$ there exists a valuation ρ and $c|A \in I$ s.t. $\mathcal{X} \models c\rho\}$

Let us define the semantics $O_{ca}(P)$ as the least solution of a fixpoint equation over $2^{\mathcal{B}'_{\mathcal{X}}}$

Non-ground immediate consequence operator

$S_P^{\mathcal{X}} : 2^{\mathcal{B}'_{\mathcal{X}}} \to 2^{\mathcal{B}'_{\mathcal{X}}}$ is defined as:

$S_P^{\mathcal{X}}(I) =$ $\{c|A \in \mathcal{B}'_{\mathcal{X}}$ | there exists a renamed clause in normal form $(A \leftarrow d|A_1, \ldots, A_n) \in P$, and constrained atoms $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, s.t. $c = d \wedge \bigwedge_{i=1}^{n} c_i$ is $\mathcal{X}$-satisfiable$\}$

Proposition 13

For any $\mathcal{B}'_{\mathcal{X}}$-interpretation I, $[S_P^{\mathcal{X}}(I)]_{\mathcal{X}} = T_P^{\mathcal{X}}([I]_{\mathcal{X}})$

Proof.

$A\rho \in [S_P^{\mathcal{X}}(I)]_{\mathcal{X}}$

Non-ground immediate consequence operator

$S_P^{\mathcal{X}} : 2^{\mathcal{B}'_{\mathcal{X}}} \rightarrow 2^{\mathcal{B}'_{\mathcal{X}}}$ is defined as:

$S_P^{\mathcal{X}}(I) =$ $\{c|A \in \mathcal{B}'_{\mathcal{X}}$ | there exists a renamed clause in normal form $(A \leftarrow d|A_1, \ldots, A_n) \in P$, and constrained atoms $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, s.t. $c = d \wedge \bigwedge_{i=1}^{n} c_i$ is $\mathcal{X}$-satisfiable$\}$

Proposition 13

For any $\mathcal{B}'_{\mathcal{X}}$-interpretation I, $[S_P^{\mathcal{X}}(I)]_{\mathcal{X}} = T_P^{\mathcal{X}}([I]_{\mathcal{X}})$

Proof.

$A\rho \in [S_P^{\mathcal{X}}(I)]_{\mathcal{X}}$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models c\rho$ and
$\{c_1|A_1, \ldots, c_n|A_n\} \subset I$

Non-ground immediate consequence operator

$S_P^{\mathcal{X}} : 2^{\mathcal{B}'_{\mathcal{X}}} \to 2^{\mathcal{B}'_{\mathcal{X}}}$ is defined as:

$S_P^{\mathcal{X}}(I) =$ $\{c|A \in \mathcal{B}'_{\mathcal{X}} \mid$ there exists a renamed clause in normal form $(A \leftarrow d|A_1, \ldots, A_n) \in P$, and constrained atoms $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, s.t. $c = d \wedge \bigwedge_{i=1}^n c_i$ is $\mathcal{X}$-satisfiable$\}$

Proposition 13

For any $\mathcal{B}'_{\mathcal{X}}$-interpretation I, $[S_P^{\mathcal{X}}(I)]_{\mathcal{X}} = T_P^{\mathcal{X}}([I]_{\mathcal{X}})$

Proof.

$A\rho \in [S_P^{\mathcal{X}}(I)]_{\mathcal{X}}$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^n c_i$, $\mathcal{X} \models c\rho$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^n c_i$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset [I]_{\mathcal{X}}$

Non-ground immediate consequence operator

$S_P^{\mathcal{X}} : 2^{\mathcal{B}'_{\mathcal{X}}} \rightarrow 2^{\mathcal{B}'_{\mathcal{X}}}$ is defined as:

$S_P^{\mathcal{X}}(I) =$ $\{c|A \in \mathcal{B}'_{\mathcal{X}}$ | there exists a renamed clause in normal form $(A \leftarrow d|A_1, \ldots, A_n) \in P$, and constrained atoms $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, s.t. $c = d \wedge \bigwedge_{i=1}^{n} c_i$ is $\mathcal{X}$-satisfiable$\}$

Proposition 13

For any $\mathcal{B}'_{\mathcal{X}}$-interpretation I, $[S_P^{\mathcal{X}}(I)]_{\mathcal{X}} = T_P^{\mathcal{X}}([I]_{\mathcal{X}})$

Proof.

$A\rho \in [S_P^{\mathcal{X}}(I)]_{\mathcal{X}}$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models c\rho$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset [I]_{\mathcal{X}}$
iff $A\rho \in T_P^{\mathcal{X}}([I]_{\mathcal{X}})$ □

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Let X be a chain of constrained interpretations. $c|A \in S_P^{\mathcal{X}}(lub(X))$, iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)$

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Let X be a chain of constrained interpretations. $c|A \in S_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, for some $I \in X$ (as X is a chain)

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Let X be a chain of constrained interpretations. $c|A \in S_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, for some $I \in X$ (as X is a chain)
iff $c|A \in S_P^{\mathcal{X}}(I)$ for some $I \in X$,

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Let X be a chain of constrained interpretations. $c|A \in S_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and
$\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and
$\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, for some $I \in X$ (as X is a chain)
iff $c|A \in S_P^{\mathcal{X}}(I)$ for some $I \in X$,
iff $c|A \in lub(S_P^{\mathcal{X}}(X))$ □

Corollary 15

Continuity of $S_P^{\mathcal{X}}$ operator

Proposition 14

$S_P^{\mathcal{X}}$ *is continuous*

Proof.

Let X be a chain of constrained interpretations. $c|A \in S_P^{\mathcal{X}}(lub(X))$,
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)$
iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, for some $I \in X$ (as X is a chain)
iff $c|A \in S_P^{\mathcal{X}}(I)$ for some $I \in X$,
iff $c|A \in lub(S_P^{\mathcal{X}}(X))$ □

Corollary 15

$S_P^{\mathcal{X}}$ *admits a least (post) fixed point* $lfp(S_P^{\mathcal{X}}) = S_P^{\mathcal{X}} \uparrow \omega$

Example CLP($\mathcal{H}$)

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 16

$$
\begin{aligned}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &=
\end{aligned}
$$

Example CLP($\mathcal{H}$)

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 16

$$
\begin{aligned}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &= \{A = [], B = C \mid append(A, B, C)\} \\
S_P^{\mathcal{H}} \uparrow 2 &= S_P^{\mathcal{H}} \uparrow 1 \ \cup
\end{aligned}
$$

Example CLP($\mathcal{H}$)

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 16

$$
\begin{aligned}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &= \{A = [], B = C \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 2 &= S_P^{\mathcal{H}} \uparrow 1 \ \cup \\
& \quad \{A = [X|L], C = [X|R], L = [], B = R \mid append(A,B,C)\} \\
&= S_P^{\mathcal{H}} \uparrow 1 \cup \{A = [X], C = [X|B] \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 3 &= S_P^{\mathcal{H}} \uparrow 2 \ \cup
\end{aligned}
$$

Example CLP($\mathcal{H}$)

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 16

$$
\begin{aligned}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &= \{A = [], B = C \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 2 &= S_P^{\mathcal{H}} \uparrow 1 \ \cup \\
&\quad \{A = [X|L], C = [X|R], L = [], B = R \mid append(A,B,C)\} \\
&= S_P^{\mathcal{H}} \uparrow 1 \cup \{A = [X], C = [X|B] \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 3 &= S_P^{\mathcal{H}} \uparrow 2 \ \cup \\
&\quad \{A = [X,Y], C = [X,Y|B] \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 4 &= S_P^{\mathcal{H}} \uparrow 3 \ \cup
\end{aligned}
$$

Example CLP($\mathcal{H}$)

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

Example 16

$$
\begin{aligned}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &= \{A = [], B = C \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 2 &= S_P^{\mathcal{H}} \uparrow 1 \ \cup \\
& \quad \{A = [X|L], C = [X|R], L = [], B = R \mid append(A,B,C)\} \\
&= S_P^{\mathcal{H}} \uparrow 1 \cup \{A = [X], C = [X|B] \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 3 &= S_P^{\mathcal{H}} \uparrow 2 \ \cup \\
& \quad \{A = [X,Y], C = [X,Y|B] \mid append(A,B,C)\} \\
S_P^{\mathcal{H}} \uparrow 4 &= S_P^{\mathcal{H}} \uparrow 3 \ \cup \\
& \quad \{A = [X,Y,Z], C = [X,Y,Z|B] \mid append(A,B,C)\} \\
\ldots &= \ldots
\end{aligned}
$$

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

Theorem 17 ([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

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

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

Theorem 17 ([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

$$
\begin{aligned}
[S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} &= [S_P^{\mathcal{X}}(S_P^{\mathcal{X}} \uparrow \alpha - 1)]_{\mathcal{X}} \\
&= T_P^{\mathcal{X}}([S_P^{\mathcal{X}} \uparrow \alpha - 1]_{\mathcal{X}}) \text{ by Prop. 13} \\
&= T_P^{\mathcal{X}}(T_P^{\mathcal{X}} \uparrow \alpha - 1) \text{ by induction} \\
&= T_P^{\mathcal{X}} \uparrow \alpha
\end{aligned}
$$

For a limit ordinal, we have

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

Theorem 17 ([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

$$\begin{array}{lcl} [S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} & = & [S_P^{\mathcal{X}}(S_P^{\mathcal{X}} \uparrow \alpha - 1)]_{\mathcal{X}} \\ & = & T_P^{\mathcal{X}}([S_P^{\mathcal{X}} \uparrow \alpha - 1]_{\mathcal{X}}) \text{ by Prop. 13} \\ & = & T_P^{\mathcal{X}}(T_P^{\mathcal{X}} \uparrow \alpha - 1) \text{ by induction} \\ & = & T_P^{\mathcal{X}} \uparrow \alpha \end{array}$$

For a limit ordinal, we have

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

□

Full abstraction w.r.t. computed answers

Theorem 18 (Theorem of full abstraction [GL91iclp])

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

Full abstraction w.r.t. computed answers

Theorem 18 (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 $S_P^{\mathcal{X}}$.

Full abstraction w.r.t. computed answers

Theorem 18 (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 $c|A \in O_{ca}(P)$.

Full abstraction w.r.t. computed answers

Theorem 18 (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.

Full abstraction w.r.t. computed answers

Theorem 18 (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 $c|A \in S_P^{\mathcal{X}} \uparrow \omega$.

Program analysis by abstract interpretation

$S_P^{\mathcal{H}} \uparrow \omega$ captures the set of computed answer constraints
nevertheless this set may be infinite and
may contain too much information for proving some properties of the computed constraints

Abstract interpretation [CC77popl] is a method for proving properties of programs without handling irrelevant information

The idea is to replace the real computation domain by an abstract computation domain which retains sufficient information w.r.t. the property to prove

Groundness analysis by abstract interpretation

Consider the CLP($\mathcal{H}$) append program

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

What is the groundness relation between arguments after a success?

The term structure can be abstracted by a boolean structure which expresses the groundness of the arguments.
We thus associate a CLP($\mathcal{B}$) abstract program:

Groundness analysis by abstract interpretation

Consider the CLP($\mathcal{H}$) append program

```
append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
```

What is the groundness relation between arguments after a success?

The term structure can be abstracted by a boolean structure which expresses the groundness of the arguments.
We thus associate a CLP($\mathcal{B}$) abstract program:

```
append(A,B,C):- A=true, B=C.
append(A,B,C):- A=X/\L, C=X/\R, append(L,B,R).
```

Its least fixed point computed in at most 2^3 steps will express the groundness relation between arguments of the concrete program.

Groundness analysis (continued)

$$\begin{aligned} S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\ S_P^{\mathcal{B}} \uparrow 1 &= \end{aligned}$$

Groundness analysis (continued)

$$
\begin{aligned}
S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{B}} \uparrow 1 &= \{A = true, B = C \mid append(A, B, C)\} \\
S_P^{\mathcal{B}} \uparrow 2 &= S_P^{\mathcal{B}} \uparrow 1 \ \cup
\end{aligned}
$$

Groundness analysis (continued)

$$
\begin{aligned}
S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{B}} \uparrow 1 &= \{A = \mathit{true}, B = C \mid \mathit{append}(A, B, C)\} \\
S_P^{\mathcal{B}} \uparrow 2 &= S_P^{\mathcal{B}} \uparrow 1 \ \cup \\
&\quad \{A = X \wedge L, C = X \wedge R, L = \mathit{true}, B = R \mid \mathit{append}(A, B, C)\} \\
&= S_P^{\mathcal{B}} \uparrow 1 \cup \{C = A \wedge B \mid \mathit{append}(A, B, C)\} \\
S_P^{\mathcal{B}} \uparrow 3 &= S_P^{\mathcal{B}} \uparrow 2 \ \cup
\end{aligned}
$$

Groundness analysis (continued)

$$
\begin{aligned}
S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{B}} \uparrow 1 &= \{A = true, B = C \mid append(A, B, C)\} \\
S_P^{\mathcal{B}} \uparrow 2 &= S_P^{\mathcal{B}} \uparrow 1 \ \cup \\
&\quad \{A = X \wedge L, C = X \wedge R, L = true, B = R \mid append(A, B, C)\} \\
&= S_P^{\mathcal{B}} \uparrow 1 \cup \{C = A \wedge B \mid append(A, B, C)\} \\
S_P^{\mathcal{B}} \uparrow 3 &= S_P^{\mathcal{B}} \uparrow 2 \ \cup \\
&\quad \{A = X \wedge L, C = X \wedge R, R = L \wedge B \mid append(A, B, C)\} \\
&= S_P^{\mathcal{B}} \uparrow 2 \cup \{C = A \wedge B \mid append(A, B, C)\} \\
&= S_P^{\mathcal{B}} \uparrow 2 = S_P^{\mathcal{B}} \uparrow \omega
\end{aligned}
$$

Groundness analysis (continued)

$$\begin{aligned}
S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{B}} \uparrow 1 &= \{A = \mathit{true}, B = C \mid \mathit{append}(A,B,C)\} \\
S_P^{\mathcal{B}} \uparrow 2 &= S_P^{\mathcal{B}} \uparrow 1 \ \cup \\
& \quad \{A = X \wedge L, C = X \wedge R, L = \mathit{true}, B = R \mid \mathit{append}(A,B,C)\} \\
&= S_P^{\mathcal{B}} \uparrow 1 \cup \{C = A \wedge B \mid \mathit{append}(A,B,C)\} \\
S_P^{\mathcal{B}} \uparrow 3 &= S_P^{\mathcal{B}} \uparrow 2 \ \cup \\
& \quad \{A = X \wedge L, C = X \wedge R, R = L \wedge B \mid \mathit{append}(A,B,C)\} \\
&= S_P^{\mathcal{B}} \uparrow 2 \cup \{C = A \wedge B \mid \mathit{append}(A,B,C)\} \\
&= S_P^{\mathcal{B}} \uparrow 2 = S_P^{\mathcal{B}} \uparrow \omega
\end{aligned}$$

In a success of $\mathit{append}(A,B,C)$,
C is ground iff A and B are ground.

Groundness analysis of reverse

Concrete CLP($\mathcal{H}$) program:

```
rev(A,B) :- A=[], B=[].
rev(A,B) :- A=[X|L], rev(L,K), append(K,[X],B).
```

Abstract CLP($\mathcal{B}$) program:

Groundness analysis of reverse

Concrete CLP($\mathcal{H}$) program:

```
rev(A,B) :- A=[], B=[].
rev(A,B) :- A=[X|L], rev(L,K), append(K,[X],B).
```

Abstract CLP($\mathcal{B}$) program:

```
rev(A,B) :- A=true, B=true.
rev(A,B) :- A=X/\L, rev(L,K), append(K,X,B).
```

Groundness analysis of reverse

Concrete CLP($\mathcal{H}$) program:

```
rev(A,B) :- A=[], B=[].
rev(A,B) :- A=[X|L], rev(L,K), append(K,[X],B).
```

Abstract CLP($\mathcal{B}$) program:

```
rev(A,B) :- A=true, B=true.
rev(A,B) :- A=X/\L, rev(L,K), append(K,X,B).
```

$$
\begin{aligned}
S_P^{\mathcal{B}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{B}} \uparrow 1 &= \{A = \mathit{true}, B = \mathit{true} \mid \mathit{rev}(A,B)\} \\
S_P^{\mathcal{B}} \uparrow 2 &= S_P^{\mathcal{B}} \uparrow 1 \ \cup \{A = X, B = X \mid \mathit{rev}(A,B)\} \\
&= S_P^{\mathcal{B}} \uparrow 1 \cup \{A = B \mid \mathit{rev}(A,B)\} \\
S_P^{\mathcal{B}} \uparrow 3 &= S_P^{\mathcal{B}} \uparrow 2 \ \cup \{A = X \wedge L, L = K, B = K \wedge X \mid \mathit{rev}(A,B)\} \\
&= S_P^{\mathcal{B}} \uparrow 2 \cup \{A = B \mid \mathit{rev}(A,B)\} = S_P^{\mathcal{B}} \uparrow 2 = S_P^{\mathcal{B}} \uparrow \omega
\end{aligned}
$$