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
Compactness theorem

Theorem 1

$T_j = \varphi$ iff $T'_j = \varphi$ for some finite part $T'$ of $T$.

By Gödel's completeness theorem, $T_j = \varphi$ iff $T \vdash \varphi$.

As the proofs are finite, they use only a finite part of non-logical axioms $T$.

Therefore $T_j = \varphi$ iff $T'_j = \varphi$ for some finite part $T'$ of $T$.

Corollary 2

$T$ is consistent iff every finite part of $T$ is consistent.

$T$ is inconsistent iff $T \vdash \text{false}$,

iff for some finite part $T'$ of $T$, $T' \vdash \text{false}$,

iff some finite part of $T$ is inconsistent.
Compactness theorem

**Theorem 1**

\[ \mathcal{T} \models \phi \iff \mathcal{T}' \models \phi \text{ for some finite part } \mathcal{T}' \text{ of } \mathcal{T} \]

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 \( \mathcal{T} \models \phi \iff \mathcal{T}' \models \phi \text{ for some finite part } \mathcal{T}' \text{ of } \mathcal{T} \)

**Corollary 2**

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

\( \mathcal{T} \) is inconsistent iff \( \mathcal{T} \vdash \text{false} \),

iff for some finite part \( \mathcal{T}' \) of \( \mathcal{T} \), \( \mathcal{T}' \vdash \text{false} \),

iff some finite part of \( \mathcal{T} \) is inconsistent
Part II

Constraint Logic Programs
Part II: Constraint Logic Programs

6 Constraint Languages

7 CLP(\(\mathcal{X}\))

8 CLP(\(\mathcal{H}\))

9 CLP(\(\mathcal{R}, FD, B\))
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** $\mathbb{Z}$

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

  **NP-hard** (Branch and bound, Gomory’s cuts,...)
CLP(\(\mathcal{R}\)) mortgage program

\[
\text{int}(P, T, I, B, M) :\ - \ T > 0, \ T \leq 1, \ B + M = P * (1 + I)
\]
\[
\text{int}(P, T, I, B, M) : - \\
- \ T > 1, \ \text{int}(P * (1 + I) - M, T - 1, I, B, M).
\]

?- \text{int}(120000, 120, 0.01, 0, M).
M = 1721.651381 ?
yes

?- \text{int}(P, 120, 0.01, 0, 1721.651381).
P = 120000 ?
yes

?- \text{int}(P, 120, 0.01, 0, M).
P = 69.700522*M ?
yes

?- \text{int}(P, 120, 0.01, B, M).
P = 0.302995*B + 69.700522*M ?
yes

?- \text{int}(999, 3, \text{Int}, 0, 400).
400 = (-400 + (599 + 999*\text{Int}) * (1 + \text{Int})) * (1 + \text{Int}) ?
CLP(\(\mathcal{R}\)) heat equation

?- X=[ [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]],

laplace(X).

X=[ [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],
      [100,97.76,95.38,93.96,92.93,92.58,92.93,93.96,95.38,97.76,100],
      [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

\[
\text{laplace}([H_1, H_2, H_3 \mid T]) :-
\]
\[
\text{laplace_vec}(H_1, H_2, H_3), \text{laplace}([H_2, H_3 \mid T]).
\]
\[
\text{laplace}([\_, \_]).
\]

\[
\text{laplace_vec}([TL, T, TR \mid T_1], [ML, M, MR \mid T_2], [BL, B, BR \mid T_3]) :-
\]
\[
B + T + ML + MR - 4 \times M = 0,
\]
\[
\text{laplace_vec}([T, TR \mid T_1], [M, MR \mid T_2], [B, BR \mid T_3]).
\]
\[
\text{laplace_vec}([\_, \_],[\_, \_],[\_, \_]).
\]

\| ?- \text{laplace}([[B_{11}, B_{12}, B_{13}, B_{14}],
[B_{21}, M_{22}, M_{23}, B_{24}],
[B_{31}, M_{32}, M_{33}, B_{34}],
[B_{41}, B_{42}, B_{43}, B_{44}]]).
\]

\[
B_{12} = -B_{21} - 4\times B_{31} + 16\times M_{32} - 8\times M_{33} + B_{34} - 4\times B_{42} + B_{43},
\]
\[
B_{13} = -B_{24} + B_{31} - 8\times M_{32} + 16\times M_{33} - 4\times B_{34} + B_{42} - 4\times B_{43},
\]
\[
M_{22} = -B_{31} + 4\times M_{32} - M_{33} - B_{42},
\]
\[
M_{23} = -M_{32} + 4\times M_{33} - B_{34} - B_{43}?
\]
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 extensionally, $\{c(a, b), c(d, c), c(a, d)\}$
- n-ary global constraints: $c(x_1, \ldots, x_n)$
CLP(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(\textit{FD}) send+more=money

| ?- sendc([S,E,N,D,M,O,R,Y]).
S = 9,
D = 1,
O = 0,
E = 4..7,
\texttt{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

10 Operational Semantics

11 Fixpoint Semantics

12 Program Analysis
Operational semantics: CSLD Resolution

A CLP(λ) 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:

\[
\begin{aligned}
&\text{successful derivation is a derivation of the form} \\
&G \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow c \vdash \square
\end{aligned}
\]

$c$ is called a computed answer constraint 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:

\[
(c|\alpha, p(s_1, s_2), \alpha') \rightarrow
\]

A successful derivation is a derivation of the form

\[
G \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow c|\square
\]

c is called a computed answer constraint 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:

\[
\begin{align*}
(p(t_1, t_2) \leftarrow c'|A_1, \ldots, A_n)\theta \in P \\
\hline
(c|\alpha, p(s_1, s_2), \alpha') \rightarrow
\end{align*}
\]

where \(\theta\) is a renaming substitution of the program clause with new variables.

A successful derivation is a derivation of the form

\[
G \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow c|\square
\]

c is called a \(\text{computed answer constraint}\) 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:

\[
(p(t_1, t_2) \leftarrow c'|A_1, \ldots, A_n)\theta \in P \quad \mathcal{X} \models \exists(c \land s_1 = t_1 \land s_2 = t_2 \land c')
\]

\[
(c|\alpha, p(s_1, s_2), \alpha') \rightarrow (c, s_1 = t_1, s_2 = t_2, c' \mid \alpha, A_1, \ldots, A_n, \alpha')
\]

where \(\theta\) is a renaming substitution of the program clause with new variables

A successful derivation is a derivation of the form

\[
G \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow c|\Box
\]

c is called a computed answer constraint for \(G\)
Operational semantics: CSLD Resolution

A CLP(\(X\)) 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:

\[
\frac{(p(t_1, t_2) \leftarrow c'|A_1, \ldots, A_n)\theta \in P \quad \chi \models \exists(c \land s_1 = t_1 \land s_2 = t_2 \land c')}{(c|\alpha, p(s_1, s_2), \alpha') \rightarrow (c, s_1 = t_1, s_2 = t_2, c' \mid \alpha, A_1, \ldots, A_n, \alpha')}
\]

where \(\theta\) is a renaming substitution of the program clause with new variables

A successful derivation is a derivation of the form

\[
G \rightarrow G_1 \rightarrow G_2 \rightarrow \ldots \rightarrow c|\square
\]

\(c\) 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*
Proof.

(\Leftrightarrow) \quad d \mid A_1, \ldots, A_n \rightarrow^{*}
\textbf{\^\text{-}Compositionality of CSLD-derivations}

\textbf{Proof.}

\((\Leftarrow)\ d \mid A_1, \ldots, A_n \rightarrow^* d \land c_1 \mid A_2, \ldots, A_n \cdots \rightarrow^* d \land c_1 \land \cdots \land c_n \mid \Box.\)
-Compositionality of CSLD-derivations

Proof.

\[(\Leftrightarrow)\ d|A_1, \ldots, A_n \rightarrow^* d \land c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \land c_1 \land \cdots \land c_n |\Box.\]

\[(\Rightarrow)\ By\ induction\ on\ the\ length\ l\ of\ the\ derivation\]
∧-Compositionality of CSLD-derivations

Proof.

\( \iff d|A_1, \ldots, A_n \rightarrow^* d \land c_1|A_2, \ldots, A_n \cdots \rightarrow^* d \land c_1 \land \cdots \land c_n|\Box. \)

\( \implies \) By induction on the length \( l \) of the derivation

If \( l = 1 \) we have \( true|A_1 \rightarrow c_1|\Box \)
Proof.

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

($\Rightarrow$) By induction on the length $l$ of the derivation

If $l = 1$ we have $true | A_1 \rightarrow c_1 | \square$

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

$d|A_1, \ldots, A_n \rightarrow d \land d_1|B_1, \ldots, B_k, A_2, \ldots, A_n \rightarrow^* c | \square$

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 $c = d \land d_1 \land \land_{i=1}^{k} e_i \land \land_{j=2}^{n} c_j$. Now let $c_1 = d_1 \land \land_{i=1}^{k} e_i$, $c_1$ is a computed answer for $true | A_1$
Operational Semantics of CLP(\(\mathcal{X}\)) Programs

Observation of the sets of projected computed answer constraints

\[
O(P) = \{(\exists X\ c)|A : true|A \rightarrow^* c\dashv, \mathcal{X} \models \exists(c),\ X = V(c) \setminus V(A)\}
\]

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

\[ O_{ca}(P) = \{ c|A : \text{true}|A \rightarrow^* c|\Box, \mathcal{X} \models \exists(c) \} \]

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

Observation of ground successes

\[ O_{gs}(P) = \{ A_\rho \in B_\mathcal{X} : \text{true}|A \rightarrow^* c|\Box, \mathcal{X} \models c_\rho \} \]

\(P \equiv_{gs} P'\) 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 \(x_1 \leq x_2 \leq \ldots\).
- A partial order is complete if every chain admits a lub.
- A function \(f : S \to S\) is monotonic if \(x \leq y \implies f(x) \leq f(y)\).
- \(f\) is continuous if \(f(\text{lub}(X)) = \text{lub}(f(X))\) for every chain \(X\).
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 \(\text{lfp}(f) = f^{\uparrow \omega}\)

Proof.

First,

\[ a = f^{\uparrow \omega}. \]

\(a\) is a fixed point of \(f\)

Let \(e\) be any fixed point of \(f\).

\[ \text{hence } a \leq e \]
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 \(\text{lfp}(f) = f^\uparrow \omega\).

**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 \(a = \text{lub}(\{f^n(\bot) \mid n \in \mathbb{N}\}) = f^\uparrow \omega\). By continuity \(f(a) = \text{lub}(\{f^{n+1}(\bot) \mid n \in \mathbb{N}\}) = a\), hence \(a\) is a fixed point of \(f\).

Let \(e\) be any fixed point of \(f\).

\[\text{hence } a \leq e\]
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
\(
\bot \leq f(\bot) \leq f(f(\bot)) \leq \ldots \text{ forms an increasing chain.}
\)

Let \(a = \text{lub}(\{f^n(\bot) \mid n \in \mathbb{N}\}) = f \uparrow \omega\). By continuity
\(f(a) = \text{lub}(\{f^{n+1}(\bot) \mid n \in \mathbb{N}\}) = a\), 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 \(n\).

\[
\text{hence } a \leq e
\]
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 \(\text{lfp}(f) = f \uparrow \omega\).

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 \(a = \text{lub}(\{f^n(\bot) \mid n \in \mathbb{N}\}) = f \uparrow \omega\). By continuity \(f(a) = \text{lub}(\{f^{n+1}(\bot) \mid n \in \mathbb{N}\}) = a\), 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 \(n\). Clearly \(\bot \leq e\). Furthermore if \(f^n(\bot) \leq e\) then by monotonicity, \(f^{n+1}(\bot) \leq f(e) = e\). Thus \(f^n(\bot) \leq e\) for all \(n\), hence \(a \leq e\).
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 \(\text{lfp}(f)\).

Proof.

Let \(g(x) = \text{lub}(x, f(x))\).
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 \(\text{lfp}(f)\).

**Proof.**

Let \(g(x) = \text{lub}(x, f(x))\). 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 \(\text{lfp}(g)\) is the least fixed point of \(g\) and the least post-fixed point of \(f\).

Furthermore, \(\text{lfp}(g) = \text{lub}\{f^n(\bot)\} = \text{lfp}(f)\).
Consider the complete lattice of $\mathcal{X}$-interpretations $(2^{\mathcal{B}_X}, \subset)$

The bottom element is the empty $\mathcal{X}$-interpretation (all atoms false)

The top element is $\mathcal{B}_X$ (all atoms true)

A chain $\mathcal{X}$ is an increasing sequence $I_1 \subset I_2 \subset \ldots$

$lub(\mathcal{X}) = \bigcup_{i \geq 1} I_i$

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

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

$T^\mathcal{X}_P (I) = \{ A_\rho \in \mathcal{B}_\mathcal{X} \mid \text{there exists a renamed clause in normal form } (A \leftarrow c|A_1, \ldots, A_n) \in P, \text{ and a valuation } \rho \text{ s.t. } \mathcal{X} \models c_\rho \text{ and } \{ A_1\rho, \ldots, A_n\rho \} \subseteq I \}$

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

Example 7

$T^\mathcal{H}_P (\emptyset)$
$T^\mathcal{X}_P$ immediate consequence operator

$T^\mathcal{X}_P : 2^{B_\mathcal{X}} \rightarrow 2^{B_\mathcal{X}}$ is defined by:

$T^\mathcal{X}_P(I) = \{ A_\rho \in B_\mathcal{X} \mid \text{there exists a renamed clause in normal form } (A \leftarrow c|A_1, \ldots, A_n) \in P, \text{ and a valuation } \rho \text{ s.t. } \mathcal{X} \models c_\rho \text{ and } \{A_1\rho, \ldots, A_n\rho\} \subset I\}$

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

Example 7

$T^\mathcal{H}_P(\emptyset)$

$T^\mathcal{H}_P(T^\mathcal{H}_P(\emptyset))$ =

$\{append([], B, B) \mid B \in \mathcal{H}\}$
\( T_P^\chi \) immediate consequence operator

\( T_P^\chi : 2^{B_\chi} \to 2^{B_\chi} \) is defined by:
\[
T_P^\chi(I) = \{ A_\rho \in B_\chi \mid \text{there exists a renamed clause in normal form } (A \leftarrow c|A_1, \ldots, A_n) \in P, \text{ and a valuation } \rho \text{ s.t. } \chi | = c_\rho \text{ and } \{A_1\rho, \ldots, A_n\rho\} \subset I \}
\]

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

Example 7

\[
\begin{align*}
T_P^H(\emptyset) &= \{\text{append}([], B, B) \mid B \in \mathcal{H}\} \\
T_P^H(T_P^H(\emptyset)) &= T_P^H(\emptyset) \cup \{\text{append}([X], B, [X|B]) \mid X, B \in \mathcal{H}\} \\
T_P^H(T_P^H(T_P^H(\emptyset))) &=
\end{align*}
\]
$T^X_P$ immediate consequence operator

$T^X_P : 2^{2^X} \rightarrow 2^{2^X}$ is defined by:

$T^X_P(I) = \{A\rho \in B_X| \text{ there exists a renamed clause in normal form}\ (A \leftarrow c|A_1,\ldots,A_n) \in P, \text{ and a valuation } \rho \text{ s.t. }\ x \models c\rho \text{ and } \{A_1\rho,\ldots,A_n\rho\} \subseteq I\}$

Example 7

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

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

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

append(A,B,C):- A=[], B=C.
append(A,B,C):- A=[X|L], C=[X|R], append(L,B,R).
Continuity of $T^X_P$ operator

**Proposition 8**

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

**Proof.**

**Corollary 9**

$T^X_P$ admits a **least (post) fixed point** $T^X_P \uparrow \omega$
Continuity of \( T^X_P \) operator

**Proposition 8**

\( T^X_P \) is a **continuous** operator on the complete lattice of \( X \)-interpretations

**Proof.**

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

**Corollary 9**

\( T^X_P \) admits a **least (post) fixed point** \( T^X_P \uparrow \omega \)
Continuity of $T^X_P$ operator

**Proposition 8**

$T^X_P$ is a *continuous* operator on the complete lattice of $X$-interpretations.

**Proof.**

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

**Corollary 9**

$T^X_P$ admits a *least (post) fixed point* $T^X_P \uparrow \omega$. 
Full abstraction

Theorem 10 ([JL87popl])

\[ T_P^\chi \uparrow \omega = O_{gs}(P) \]

\( T_P^\chi \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T_P^\chi \).
Full abstraction

Theorem 10 ([JL87popl])

\[ T_P^\omega \uparrow \omega = O_{gs}(P) \]

\( T_P^\omega \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T_P^\omega \). \( n = 0 \), i.e., \( \emptyset \), is trivial.
Full abstraction

Theorem 10 ([JL87popl])

\[ T^x_P \uparrow \omega = O_{gs}(P) \]

\[ T^x_P \uparrow \omega \subset O_{gs}(P) \] is proved by induction on the powers \( n \) of \( T^x_P \). 

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T^x_P \uparrow n \),
Theorem 10 ([JL87popl])

\[ T^\chi_P \uparrow \omega = O_{gs}(P) \]

\( T^\chi_P \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T^\chi_P \). \( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T^\chi_P \uparrow n \), there exists a rule \((A \leftarrow c|A_1,\ldots,A_n) \in P\), s.t. \( \{A_1\rho,\ldots,A_n\rho\} \subset T^\chi_P \uparrow n - 1 \) and \( \mathcal{X} \models c_\rho \).
Full abstraction

Theorem 10 ([JL87popl])

\[ T_P^X \uparrow \omega = O_{gs}(P) \]

\[ T_P^X \uparrow \omega \subset O_{gs}(P) \] is proved by induction on the powers \( n \) of \( T_P^X \).

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T_P^X \uparrow n \), there exists a rule

\( (A \leftarrow c|A_1, \ldots, A_n) \in P \), s.t. \( \{A_1^\rho, \ldots, A_n^\rho\} \subset T_P^X \uparrow n - 1 \) and \( X \models c_\rho \). By induction \( \{A_1^\rho, \ldots, A_n^\rho\} \subset O_{gs}(P) \).
Full abstraction

Theorem 10 ([JL87popl])

\[ T^\chi_P \uparrow \omega = O_{gs}(P) \]

\( T^\chi_P \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T^\chi_P \).

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T^\chi_P \uparrow n \), there exists a rule

\( (A \leftarrow c | A_1, \ldots, A_n) \in P \), s.t. \( \{A_1, \ldots, A_n\} \subset T^\chi_P \uparrow n - 1 \) and \( \chi \models c_\rho \). By induction \( \{A_1, \ldots, A_n\} \subset O_{gs}(P) \).

By definition of \( O_{gs} \) and \( \wedge \)-compositionality. we get \( A_\rho \in O_{gs}(P) \).
Full abstraction

Theorem 10 ([JL87popl])

\[ T_P^X \uparrow \omega = O_{gs}(P) \]

\[ T_P^X \uparrow \omega \subset O_{gs}(P) \] is proved by induction on the powers \( n \) of \( T_P^X \).

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T_P^X \uparrow n \), there exists a rule 
\((A \leftarrow c | A_1, \ldots, A_n) \in P\), s.t. \( \{A_1\rho, \ldots, A_n\rho\} \subset T_P^X \uparrow n - 1 \) and \( X \models c_\rho \). By induction \( \{A_1\rho, \ldots, A_n\rho\} \subset O_{gs}(P) \). By definition of \( O_{gs} \) and \( \wedge \)-compositionality, we get \( A_\rho \in O_{gs}(P) \).

\( O_{gs}(P) \subset T_P^X \uparrow \omega \) 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 \( \land \)-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 \( T_P^{\mathcal{X}} \uparrow 1 \).
Theorem 10 ([JL87popl])

\( T_P^\chi \uparrow \omega = O_{gs}(P) \)

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

\( O_{gs}(P) \subset T_P^\chi \uparrow \omega \) is proved by induction on the length of derivations. Successes with derivation of length 0 are ground facts in \( T_P^\chi \uparrow 1 \). 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])**

\[ T_P^\chi \uparrow \omega = O_{gs}(P) \]

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

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

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 \( \chi \models \ c_\rho \).
Full abstraction

Theorem 10 ([JL87popl])

\( T_P^{\chi} \uparrow \omega = O_{gs}(P) \)

\( T_P^{\chi} \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T_P^{\chi} \).

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T_P^{\chi} \uparrow n \), there exists a rule 
\( (A \leftarrow c | A_1, \ldots, A_n) \in P \), s.t. \( \{A_1_\rho, \ldots, A_n_\rho\} \subset T_P^{\chi} \uparrow n - 1 \) and \( \chi \vdash c_\rho \). By induction \( \{A_1_\rho, \ldots, A_n_\rho\} \subset O_{gs}(P) \). By definition of \( O_{gs} \) and \( ^\wedge \)-compositionality. we get \( A_\rho \in O_{gs}(P) \).

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

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 
\( \chi \vdash c_\rho \). By induction \( \{A_1_\rho, \ldots, A_n_\rho\} \subset T_P^{\chi} \uparrow \omega \).
Theorem 10 ([JL87popl])

\[ T^X_P \uparrow \omega = O_{gs}(P) \]

\( T^X_P \uparrow \omega \subset O_{gs}(P) \) is proved by induction on the powers \( n \) of \( T^X_P \).

\( n = 0 \), i.e., \( \emptyset \), is trivial. Let \( A_\rho \in T^X_P \uparrow n \), there exists a rule

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

\( O_{gs}(P) \subset T^X_P \uparrow \omega \) is proved by induction on the length of derivations. Successes with derivation of length 0 are ground facts in \( T^X_P \uparrow 1 \).

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 \( X \models c_\rho \). By induction \( \{A_1 \rho, \ldots, A_n \rho\} \subset T^X_P \uparrow \omega \). Hence by definition of \( T^X_P \) we get \( A_\rho \in T^X_P \uparrow \omega \).
$T^X_P$ and $X$-models

Proposition 11

$I$ is a $X$-model of $P$ iff $I$ is a post-fixed point of $T^X_P$, $T^X_P(I) \subseteq I$

Proof.

$I$ is a $X$-model of $P$, iff
Proposition 11

$I$ is a $\kappa$-model of $P$ iff $I$ is a post-fixed point of $T^\kappa_P$, $T^\kappa_P(I) \subset I$

Proof.

$I$ is a $\kappa$-model of $P$, iff for each clause $A \leftarrow c|A_1, \ldots, A_n \in P$ and for each $\kappa$-valuation $\rho$, if $\kappa \models c\rho$ and $\{A_1\rho, \ldots, A_n\rho\} \subset I$ then $A\rho \in I$, iff $T^\kappa_P(I) \subset I$. 
Theorem 12 (Least $\chi$-model [JL87popl])

Let $P$ be a constraint logic program on $\chi$. $P$ has a least $\chi$-model, denoted by $M_P^\chi$ satisfying:

$$M_P^\chi = T_P^\chi \uparrow \omega$$

Proof.

$T_P^\chi \uparrow \omega = \text{lfp}(T_P^\chi)$ is also the least post-fixed point of $T_P^\chi$, thus by Prop. 11, $\text{lfp}(T_P^\chi)$ is the least $\chi$-model of $P$. □
Fixpoint semantics of $O_{ca}$

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

Consider the lattice of constrained interpretations $(2^{\mathcal{B}'_X}, \subseteq)$

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

Let us define the semantics $O_{ca}(P)$ as the least solution of a fixpoint equation over $2^{\mathcal{B}'_X}$
Non-ground immediate consequence operator

\[ S^\mathcal{X}_P : 2^{B'_\mathcal{X}} \rightarrow 2^{B'_\mathcal{X}} \] is defined as:

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

**Proposition 13**

For any \( B'_{\mathcal{X}} \)-interpretation \( I \), \[ [S^\mathcal{X}_P (I)]_\mathcal{X} = T^\mathcal{X}_P ([I]_\mathcal{X}) \]

**Proof.**

\( A_\rho \in [S^\mathcal{X}_P (I)]_\mathcal{X} \)
Non-ground immediate consequence operator

\( S_{\mathcal{P}}^{\mathcal{X}} : 2^{B'_{\mathcal{X}}} \rightarrow 2^{B'_{\mathcal{X}}} \) is defined as:

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

Proposition 13

For any \( B'_{\mathcal{X}} \)-interpretation \( I \), \( [S_{\mathcal{P}}^{\mathcal{X}}(I)]_{\mathcal{X}} = T_{\mathcal{P}}^{\mathcal{X}}([I]_{\mathcal{X}}) \)

Proof.

\( A_{\rho} \in [S_{\mathcal{P}}^{\mathcal{X}}(I)]_{\mathcal{X}} \)

iff \( (A \leftarrow d \mid A_1, \ldots, A_n) \in P, c = d \land \bigwedge_{i=1}^{n} c_i, \mathcal{X} \models c_{\rho} \) and

\( \{c_1 \mid A_1, \ldots, c_n \mid A_n\} \subset I \)
Non-ground immediate consequence operator

\[ S^X_p : 2^{B'_X} \rightarrow 2^{B'_X} \] is defined as:
\[ S^X_p(I) = \{ c|A \in B'_X \mid \text{there exists a renamed clause in normal form } (A \leftarrow d|A_1, \ldots, A_n) \in P, \text{ and constrained atoms } \{c_1|A_1, \ldots, c_n|A_n\} \subset I, \text{ s.t. } c = d \land \land_{i=1}^n c_i \text{ is } X\text{-satisfiable} \} \]

**Proposition 13**

For any \( B'_X \)-interpretation \( I \),
\[ [S^X_p(I)]_X = T^X_p([I]_X) \]

**Proof.**

\[ A_\rho \in [S^X_p(I)]_X \]
iff \((A \leftarrow d|A_1, \ldots, A_n) \in P, \ c = d \land \land_{i=1}^n c_i, \ X \models c_\rho \text{ and} \)
\[ \{c_1|A_1, \ldots, c_n|A_n\} \subset I \]
iff \((A \leftarrow d|A_1, \ldots, A_n) \in P, \ c = d \land \land_{i=1}^n c_i, \ X \models c_\rho \text{ and} \)
\[ \{A_1\rho, \ldots, A_n\rho\} \subset [I]_X \]
Non-ground immediate consequence operator

\[ S^X_P : 2^{B'_X} \rightarrow 2^{B'_X} \] is defined as:

\[ S^X_P(I) = \{ c | A \in B'_X \mid \text{there exists a renamed clause in normal form } (A \leftarrow d|A_1, \ldots, A_n) \in P, \text{ and constrained atoms } \{c_1|A_1, \ldots, c_n|A_n\} \subset I, \text{ s.t. } c = d \land \bigwedge_{i=1}^n c_i \text{ is } X\text{-satisfiable} \} \]

**Proposition 13**

For any \( B'_X \)-interpretation \( I \), \[ [S^X_P(I)]_X = T^X_P([I]_X) \]

**Proof.**

\( A_\rho \in [S^X_P(I)]_X \)

iff \( (A \leftarrow d|A_1, \ldots, A_n) \in P, c = d \land \bigwedge_{i=1}^n c_i, X \models c_\rho \) and \( \{c_1|A_1, \ldots, c_n|A_n\} \subset I \)

iff \( (A \leftarrow d|A_1, \ldots, A_n) \in P, c = d \land \bigwedge_{i=1}^n c_i, X \models c_\rho \) and \( \{A_1\rho, \ldots, A_n\rho\} \subset [I]_X \)

iff \( A_\rho \in T^X_P([I]_X) \)
**Continuity of \( S^\chi_P \) operator**

**Proposition 14**

\[ S^\chi_P \text{ is continuous} \]

**Proof.**

\[ \text{Let } X \text{ be a chain of constrained interpretations.} \]

\[ c_j: A_2 \ldots A_n \in S^\chi_P (\text{lub}(X)), \iff (A_d: A_1; \ldots ; A_n) \in P, c = d \land \bigwedge_{i=1}^n c_i, \]

\[ X_j = \exists (c) \text{ and } f(c_1: A_1; \ldots ; c_n: A_n) \in \text{lub}(X) \iff (A_d: A_1; \ldots ; A_n) \in P, c = d \land \bigwedge_{i=1}^n c_i, \]

\[ \text{for some } I \in X \text{ (as } X \text{ is a chain)} \]

\[ \text{iff } c_j: A_2 \in S^\chi_P (\text{lub}(X)) \text{ for some } I \in X, \]

\[ \text{Corollary 15} \]

\( S^\chi_P \) admits a least (post) fixed point \( \text{lfp}(S^\chi_P) = S^\chi_P \)
Proposition 14

\( S^\chi_P \) is continuous

Proof.

Let \( X \) be a chain of constrained interpretations. \( c \models A \in S^\chi_P(lub(X)), \)
iff \( (A \leftarrow d|A_1, \ldots, A_n) \in P, \) \( c = d \land \bigwedge_{i=1}^{n} c_i, \) \( \chi \models \exists(c) \) and
\( \{c_1|A_1, \ldots, c_n|A_n\} \subseteq lub(X) \)
Proposition 14

\( S^\chi_P \) is continuous

Proof.

Let \( X \) be a chain of constrained interpretations. \( c|A \in S^\chi_P(lub(X)) \), iff \( (A \leftarrow d|A_1, \ldots, A_n) \in P, \ c = d \land \bigwedge_{i=1}^{n} c_i, \ \chi \models \exists(c) \) and \( \{c_1|A_1, \ldots, c_n|A_n\} \subseteq lub(X) \)

iff \( (A \leftarrow d|A_1, \ldots, A_n) \in P, \ c = d \land \bigwedge_{i=1}^{n} c_i, \ \chi \models \exists(c) \) and \( \{c_1|A_1, \ldots, c_n|A_n\} \subseteq I, \) for some \( I \in X \) (as \( X \) is a chain)
Continuity of $S^\chi_P$ operator

**Proposition 14**

$S^\chi_P$ is **continuous**

**Proof.**

Let $X$ be a chain of constrained interpretations. $c|A \in S^\chi_P(lub(X))$, iff $(A \leftarrow d|A_1,\ldots,A_n) \in P$, $c = d \land \bigwedge_{i=1}^n c_i$, $\chi \models \exists(c)$ and

$\{c_1|A_1,\ldots,c_n|A_n\} \subset lub(X)$

iff $(A \leftarrow d|A_1,\ldots,A_n) \in P$, $c = d \land \bigwedge_{i=1}^n c_i$, $\chi \models \exists(c)$ and

$\{c_1|A_1,\ldots,c_n|A_n\} \subset I$, for some $I \in X$ (as $X$ is a chain)

iff $c|A \in S^\chi_P(I)$ for some $I \in X$, for some $I \in X$, for some $I \in X$.
Continuity of $S^x_P$ operator

Proposition 14

$S^x_P$ is continuous

Proof.

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

\{c_1|A_1, \ldots, c_n|A_n\} \subset lub(X)

iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \land \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and

\{c_1|A_1, \ldots, c_n|A_n\} \subset I$, for some $I \in X$ (as $X$ is a chain)

iff $c|A \in S^x_P(I)$ for some $I \in X$

iff $c|A \in lub(S^x_P(X))$

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 $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \land \bigwedge_{i=1}^n c_i$, $\mathcal{X} \models \exists(c)$ and \{c_1|A_1, \ldots, c_n|A_n\} $\subseteq$ lub($X$)

iff $(A \leftarrow d|A_1, \ldots, A_n) \in P$, $c = d \land \bigwedge_{i=1}^n c_i$, $\mathcal{X} \models \exists(c)$ and \{c_1|A_1, \ldots, c_n|A_n\} $\subseteq$ 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

\[
\begin{align*}
S^\mathcal{H}_P & \uparrow 0 = \emptyset \\
S^\mathcal{H}_P & \uparrow 1 =
\end{align*}
\]
Example CLP($\mathcal{H}$)

\[
\text{append}(A, B, C) :- A = [], B = C.
\]

\[
\text{append}(A, B, C) :- A = [X \mid L], C = [X \mid R], \text{append}(L, B, R).
\]

Example 16

\[
\begin{align*}
S_P^{\mathcal{H}} \uparrow 0 & = \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 & = \{A = [], B = C \mid \text{append}(A, B, C)\} \\
S_P^{\mathcal{H}} \uparrow 2 & = S_P^{\mathcal{H}} \uparrow 1 \cup \\
\end{align*}
\]
Example CLP(\(\mathcal{H}\))

\[
\text{append}(A, B, C) :- A=[], B=C.
\]

\[
\text{append}(A, B, C) :- A=[X|L], C=[X|R], \text{append}(L, B, R).
\]

Example 16

\[
\begin{align*}
S_P^{\mathcal{H}} \uparrow 0 &= \emptyset \\
S_P^{\mathcal{H}} \uparrow 1 &= \{A=[], B=C \mid \text{append}(A, B, C)\}
\end{align*}
\]

\[
\begin{align*}
S_P^{\mathcal{H}} \uparrow 2 &= S_P^{\mathcal{H}} \uparrow 1 \cup \\
&\quad \{A=[X|L], C=[X|R], L=[], B=R \mid \text{append}(A, B, C)\} \\
&= S_P^{\mathcal{H}} \uparrow 1 \cup \{A=[X], C=[X|B] \mid \text{append}(A, B, C)\}
\end{align*}
\]

\[
\begin{align*}
S_P^{\mathcal{H}} \uparrow 3 &= S_P^{\mathcal{H}} \uparrow 2 \cup \\
&\quad \{A=[X|L], C=[X|R], L=[], B=R \mid \text{append}(A, B, C)\}
\end{align*}
\]
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

\[
\begin{align*}
S^\mathcal{H}_P \uparrow 0 &= \emptyset \\
S^\mathcal{H}_P \uparrow 1 &= \{ A = [], B = C \mid \text{append}(A, B, C) \} \\
S^\mathcal{H}_P \uparrow 2 &= S^\mathcal{H}_P \uparrow 1 \cup \\
&\quad \{ A = [X|L], C = [X|R], L = [], B = R \mid \text{append}(A, B, C) \} \\
&= S^\mathcal{H}_P \uparrow 1 \cup \{ A = [X], C = [X|B] \mid \text{append}(A, B, C) \} \\
S^\mathcal{H}_P \uparrow 3 &= S^\mathcal{H}_P \uparrow 2 \cup \\
&\quad \{ A = [X, Y], C = [X, Y|B] \mid \text{append}(A, B, C) \} \\
S^\mathcal{H}_P \uparrow 4 &= S^\mathcal{H}_P \uparrow 3 \cup
\end{align*}
\]
### Example CLP($\mathcal{H}$)

append($A, B, C$) :- $A=[]$, $B=C$.

### Example 16

<table>
<thead>
<tr>
<th>$S^H_P \uparrow 0$</th>
<th>$= \emptyset$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S^H_P \uparrow 1$</td>
<td>$= {A = [], B = C \mid \text{append}(A, B, C)}$</td>
</tr>
<tr>
<td>$S^H_P \uparrow 2$</td>
<td>$= S^H_P \uparrow 1 \cup {A = [X</td>
</tr>
<tr>
<td></td>
<td>$= S^H_P \uparrow 1 \cup {A = [X], C = [X</td>
</tr>
<tr>
<td>$S^H_P \uparrow 3$</td>
<td>$= S^H_P \uparrow 2 \cup {A = [X, Y], C = [X, Y</td>
</tr>
<tr>
<td>$S^H_P \uparrow 4$</td>
<td>$= S^H_P \uparrow 3 \cup {A = [X, Y, Z], C = [X, Y, Z</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Relating $S_\alpha^\chi$ and $T_\alpha^\chi$ operators

**Theorem 17 ([JL87popl])**

For every ordinal $\alpha$, $T_\alpha^\chi \uparrow \alpha = [S_\alpha^\chi \uparrow \alpha]^\chi$

**Proof.**

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

$$[S_\alpha^\chi \uparrow \alpha]^\chi = [S_\alpha^\chi (S_\alpha^\chi \uparrow \alpha - 1)]^\chi$$
Theorem 17 ([JL87popl])

For every ordinal $\alpha$, $T^x_P \uparrow \alpha = [S^x_P \uparrow \alpha]_x$

Proof.

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

$[S^x_P \uparrow \alpha]_x = [S^x_P (S^x_P \uparrow \alpha - 1)]_x$

$= T^x_P ([S^x_P \uparrow \alpha - 1]_x)$ by Prop. 13

$= T^x_P (T^x_P \uparrow \alpha - 1)$ by induction

$= T^x_P \uparrow \alpha$

For a limit ordinal, we have
Relating $S^\chi_P$ and $T^\chi_P$ operators

**Theorem 17 ([JL87popl])**

For every ordinal $\alpha$, $T^\chi_P \uparrow \alpha = [S^\chi_P \uparrow \alpha] \chi$

**Proof.**

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

$$[S^\chi_P \uparrow \alpha] \chi = [S^\chi_P (S^\chi_P \uparrow \alpha - 1)] \chi$$

$$= T^\chi_P ([S^\chi_P \uparrow \alpha - 1] \chi) \text{ by Prop. 13}$$

$$= T^\chi_P (T^\chi_P \uparrow \alpha - 1) \text{ by induction}$$

$$= T^\chi_P \uparrow \alpha$$

For a limit ordinal, we have

$$[S^\chi_P \uparrow \alpha] \chi = [\bigcup_{\beta < \alpha} S^\chi_P \uparrow \beta] \chi$$

$$= \bigcup_{\beta < \alpha} [S^\chi_P \uparrow \beta] \chi$$

$$= \bigcup_{\beta < \alpha} T^\chi_P \uparrow \beta \text{ by induction}$$

$$= T^\chi_P \uparrow \alpha$$
Full abstraction w.r.t. computed answers

Theorem 18 (Theorem of full abstraction [GL91iclp])

\[ O_{ca}(P) = S_P^X \uparrow \omega \]
Full abstraction w.r.t. computed answers

Theorem 18 (Theorem of full abstraction [GL91iclp])

\[ O_{\text{ca}}(P) = S_P^X \uparrow \omega \]

\( S_P^X \uparrow \omega \subset O_{\text{ca}}(P) \) is proved by induction on the powers \( n \) of \( S_P^X \).
**Theorem 18 (Theorem of full abstraction [GL91iclp])**

\[ O_{ca}(P) = S_{P}^{\chi} \uparrow \omega \]

\( S_{P}^{\chi} \uparrow \omega \subset O_{ca}(P) \) is proved by induction on the powers \( n \) of \( S_{P}^{\chi} \). \( n = 0 \) is trivial. Let \( c|A \in S_{P}^{\chi} \uparrow n \), there exists a rule \( (A \leftarrow d|A_1, \ldots, A_n) \in P \), s.t. \( \{c_1|A_1, \ldots, c_n|A_n\} \subset S_{P}^{\chi} \uparrow n - 1 \), \( c = d \land \bigwedge_{i=1}^{n} c_i \) and \( \chi \models \exists c \). 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])

\[ O_{ca}(P) = S_P^X \uparrow \omega \]

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

\( O_{ca}(P) \subset S_P^X \uparrow \omega \) is proved by induction on the length of derivations.
Full abstraction w.r.t. computed answers

Theorem 18 (Theorem of full abstraction [GL91icl])

\[ O_{ca}(P) = S_P^X \uparrow \omega \]

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

\( O_{ca}(P) \subset S_P^X \uparrow \omega \) is proved by induction on the length of derivations. Successes with derivation of length 0 are facts in \( S_P^X \uparrow 1 \). 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. \( \{c_1|A_1, \ldots, c_n|A_n\} \subset O_{ca}(P) \), \( c = d \land \bigwedge_{i=1}^{n} c_i \) and \( X \models \exists c \). By induction \( \{c_1|A_1, \ldots, c_n|A_n\} \subset S_P^X \uparrow \omega \). Hence by definition of \( S_P^X \) we get \( c|A \in S_P^X \uparrow \omega \).
Program analysis by abstract interpretation

$S^H_P \uparrow \omega$ 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.
Consider the CLP(\(H\)) append program

\[
\text{append}(A, B, C) :- A=[], B=C.
\]

\[
\text{append}(A, B, C) :- A=[X|L], C=[X|R], \text{append}(L, B, R).
\]

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(\(B\)) abstract program:
Groundness analysis by abstract interpretation

Consider the CLP($H$) append program

\[
\begin{align*}
\text{append}(A, B, C) :& \quad A = [], \ B = C. \\
\text{append}(A, B, C) :& \quad A = [X | L], \ C = [X | R], \ \text{append}(L, B, R).
\end{align*}
\]

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($B$) abstract program:

\[
\begin{align*}
\text{append}(A, B, C) :& \quad A = \text{true}, \ B = C. \\
\text{append}(A, B, C) :& \quad A = X / \backslash L, \ C = X / \backslash R, \ \text{append}(L, B, R).
\end{align*}
\]

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)

\[ S_P^B \uparrow 0 = \emptyset \]
\[ S_P^B \uparrow 1 = \]
Groundness analysis (continued)

\[ S_P^B \uparrow 0 = \emptyset \]
\[ S_P^B \uparrow 1 = \{ A = \text{true}, B = C \mid \text{append}(A, B, C) \} \]
\[ S_P^B \uparrow 2 = S_P^B \uparrow 1 \cup \]
Groundness analysis (continued)

\[ S_P^{\mathcal{B}} \uparrow 0 \ = \ \emptyset \]
\[ S_P^{\mathcal{B}} \uparrow 1 \ = \ \{ A = \text{true}, B = C \mid \text{append}(A, B, C) \} \]
\[ S_P^{\mathcal{B}} \uparrow 2 \ = \ S_P^{\mathcal{B}} \uparrow 1 \ \cup \]
\[ \{ A = X \land L, C = X \land R, L = \text{true}, B = R \mid \text{append}(A, B, C) \} \]
\[ \ = \ S_P^{\mathcal{B}} \uparrow 1 \ \cup \ \{ C = A \land B \mid \text{append}(A, B, C) \} \]
\[ S_P^{\mathcal{B}} \uparrow 3 \ = \ S_P^{\mathcal{B}} \uparrow 2 \ \cup \]
Groundness analysis (continued)

\[ S_P^B \uparrow 0 = \emptyset \]
\[ S_P^B \uparrow 1 = \{ A = \text{true}, B = C \mid \text{append}(A, B, C) \} \]
\[ S_P^B \uparrow 2 = S_P^B \uparrow 1 \cup \]
\[ \{ A = X \land L, C = X \land R, L = \text{true}, B = R \mid \text{append}(A, B, C) \} \]
\[ = S_P^B \uparrow 1 \cup \{ C = A \land B \mid \text{append}(A, B, C) \} \]
\[ S_P^B \uparrow 3 = S_P^B \uparrow 2 \cup \]
\[ \{ A = X \land L, C = X \land R, R = L \land B \mid \text{append}(A, B, C) \} \]
\[ = S_P^B \uparrow 2 \cup \{ C = A \land B \mid \text{append}(A, B, C) \} \]
\[ = S_P^B \uparrow 2 = S_P^B \uparrow \omega \]
Groundness analysis (continued)

\[ S^B_P \uparrow 0 = \emptyset \]
\[ S^B_P \uparrow 1 = \{ A = true, B = C \mid append(A, B, C) \} \]
\[ S^B_P \uparrow 2 = S^B_P \uparrow 1 \cup \]
\[ \{ A = X \land L, C = X \land R, L = true, B = R \mid append(A, B, C) \} \]
\[ = S^B_P \uparrow 1 \cup \{ C = A \land B \mid append(A, B, C) \} \]
\[ S^B_P \uparrow 3 = S^B_P \uparrow 2 \cup \]
\[ \{ A = X \land L, C = X \land R, R = L \land B \mid append(A, B, C) \} \]
\[ = S^B_P \uparrow 2 \cup \{ C = A \land B \mid append(A, B, C) \} \]
\[ = S^B_P \uparrow 2 = S^B_P \uparrow \omega \]

In a success of \( append(A, B, C) \),
\( C \) is ground iff \( A \) and \( B \) are ground.
Groundness analysis of reverse

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

\[
\text{rev}(A,B) :- A=[], B=[].
\]
\[
\text{rev}(A,B) :- A=[X|L], \text{rev}(L,K), \text{append}(K,[X],B).
\]

Abstract CLP(\(\mathcal{B}\)) program:
Groundness analysis of reverse

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

\[
\text{rev}(A,B) :- A=[], B=[].
\]
\[
\text{rev}(A,B) :- A=[X|L], \text{rev}(L,K), \text{append}(K,[X],B).
\]

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

\[
\text{rev}(A,B) :- A=\text{true}, B=\text{true}.
\]
\[
\text{rev}(A,B) :- A=X/\text{\backslash L}, \text{rev}(L,K), \text{append}(K,X,B).
\]
Groundness analysis of reverse

Concrete CLP(∀H) program:

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

Abstract CLP(∀B) program:

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

\[
S^B_P \uparrow 0 = \emptyset \\
S^B_P \uparrow 1 = \{A = true, B = true \mid rev(A,B)\} \\
S^B_P \uparrow 2 = S^B_P \uparrow 1 \cup \{A = X, B = X \mid rev(A,B)\} \\
\quad = S^B_P \uparrow 1 \cup \{A = B \mid rev(A,B)\} \\
S^B_P \uparrow 3 = S^B_P \uparrow 2 \cup \{A = X \land L, L = K, B = K \land X \mid rev(A,B)\} \\
\quad = S^B_P \uparrow 2 \cup \{A = B \mid rev(A,B)\} = S^B_P \uparrow 2 = S^B_P \uparrow \omega
\]