Constraint Programming III: Constraint Logic Programs

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

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

Part IV: Operational and Fixpoint Semantics

1. Operational semantics

   compositionality

2. Fixpoint semantics of successes

   lattice of $\mathcal{X}$-interpretations

3. Fixpoint semantics of computed answer constraints

   constrained interpretations

4. Program analysis by abstract interpretation

5. Constraint-based model checking of infinite states concurrent systems

Operational semantics: CSLD Resolution

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

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

A successful derivation is a derivation of the form

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

c is called a computed answer constraint for G.

$\wedge$-Compositionality of CSLD-derivations

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

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

$(\Rightarrow)$ 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 $(A_1 \leftarrow d_1|B_1, ..., B_k) \in P$ such that
$d|A_1, ..., A_n \rightarrow d \wedge d_1|B_1, ..., B_k, A_2, ..., A_n \rightarrow^* c|\Box$. By induction, there exist computed answers $e_1, ..., e_l, c_2, ..., c_n$ for the goals $B_1, ..., B_l, A_2, ..., A_n$ such that $c = d \wedge d_1 \wedge \bigwedge_{i=1}^{l} e_i \wedge \bigwedge_{j=2}^{n} c_j$. Now let $c_1 = d_1 \wedge \bigwedge_{i=1}^{l} e_i$, c_1 is a computed answer for $true|A_1$.

Corollary 2 *Independance of the selection strategy.*

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 the same sets of computed answer constraints.

Finer observables: the multisets of computed answer constraints

or the sets of succesful CSLD derivations (equivalence of traces)

More abstract observable: the set of goals having a success

(theorem proving versus programming point of view).

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

Observation of computed answer constraints

$$O_2(P) = \{c|A \ : \ true|A \longrightarrow^* c|\Box, \ \mathcal{X} \models \exists(c)\}$$

$P \equiv_2 P'$ iff for every goal G, P and P' have the same sets of computed answer constraints.

Observation of ground successes

$$O_1(P) = \{A\rho \in B_{\mathcal{X}} \ : \ true|A \longrightarrow^* c|\Box, \ \mathcal{X} \models c\rho\}$$

$P \equiv_1 P'$ 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'.

1. Fixed Point Semantics

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

A upper bound of a 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 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 least upper bound.

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 least upper bound.

A function $f : S \to S$ is monotonic if $x \leq y \Rightarrow f(x) \leq f(y)$.

continuous if $f(lub(X)) = lub(f(X))$ for every chain X.

Fixpoint theorems

Theorem 3 (Knaster-Tarski) *Let S be a complete partial order. Let $f : S \to S$ be 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 $\perp \leq f(\perp) \leq f(f(\perp)) \leq ...$ forms an increasing chain. Let $a = lub(\{f^n(\perp) | n \in \mathbf{N}\}) = f \uparrow \omega$. By continuity $f(a) = lub(\{f^{n+1}(\perp) \mid n \in \mathbf{N}\}) = a$, 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 $a \leq e$. $\square$

Least Post-Fixed Point

Theorem 4 *Let S 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. $f(e) \leq e$, if and only if 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_1

Consider the complete lattice of $\mathcal{X}$-interpretations $(2^{\mathcal{B}_{\mathcal{X}}}, \subseteq)$

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 $I_1 \subseteq I_2 \subseteq \ldots$

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

Define the semantics $O_1(P)$ as the least solution of a fixpoint equation over $2^{\mathcal{B}_{\mathcal{X}}}$: $I = T(I)$.

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

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

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

Example:

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

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

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

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

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

Proposition 5 *$T_P^{\mathcal{X}}$ 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 $(A \leftarrow c|A_1, ..., A_n) \in P$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, ..., A_n\rho\} \subset lub(X)$,

iff $(A \leftarrow c|A_1, ..., A_n) \in P$, $\mathcal{X} \models c\rho$ and $\{A_1\rho, ..., A_n\rho\} \subset I$, 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 $A\rho \in lub(T_P^{\mathcal{X}}(X))$. . $\square$

Corollary 6 *$T_P^{\mathcal{X}}$ admits a least (post) fixed point $T_P^{\mathcal{X}} \uparrow \omega$.*

Full abstraction

Let $F_1(P) = lfp(T_P^{\mathcal{X}}) = T_P^{\mathcal{X}} \uparrow \omega = ...T_P^{\mathcal{X}}(T_P^{\mathcal{X}}(\emptyset))...$

Theorem 7 *[JL87]* $F_1(P) = O_1(P)$.

$F_1(P) \subseteq O_1(P)$ is proved by induction on the powers n of $T_P^{\mathcal{X}}$. $n = 0$ is trivial. Let $A\rho \in T_P^{\mathcal{X}} \uparrow n$, there exists a rule $(A \leftarrow c|A_1, ..., A_n) \in P$, s.t. $\{A_1\rho, ..., A_n\rho\} \subseteq T_P^{\mathcal{X}} \uparrow n - 1$ and $\mathcal{X} \models c\rho$. By induction $\{A_1\rho, ..., A_n\rho\} \subseteq O_1(P)$. By definition of O_1 we get $A\rho \in O_1(P)$.

$O_1(P) \subseteq F_1(P)$ 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 $A\rho \in O_1(P)$ with a derivation of length n. By definition of O_1 there exists $(A \leftarrow c|A_1, ..., A_n) \in P$ s.t. $\{A_1\rho, ..., A_n\rho\} \subseteq O_1(P)$ and $\mathcal{X} \models c\rho$. By induction $\{A_1\rho, ..., A_n\rho\} \subseteq F_1(P)$. Hence by definition of $T_P^{\mathcal{X}}$ we get $A\rho \in F_1(P)$.

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

Proposition 8 *I is a $\mathcal{X}$-model of P iff I is a post-fixed point of $T_P^{\mathcal{X}}$, $T_P^{\mathcal{X}}(I) \subseteq I$.*

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

iff for each clause $A \leftarrow c|A_1, ..., A_n \in P$ and for each $\mathcal{X}$-valuation ρ, if $\mathcal{X} \models c\rho$ and $\{A_1\rho, ..., A_n\rho\} \subseteq I$ then $A\rho \in I$,

iff $T_P^{\mathcal{X}}(I) \subseteq I$. □

Theorem 9 (Least $\mathcal{X}$-model) *[JL87] Let P a constraint logic program on $\mathcal{X}$. P has a least $\mathcal{X}$-model, denoted by $M_P^{\mathcal{X}}$ satisfying:*

$$M_P^{\mathcal{X}} = F_1(P)$$

PROOF: $F_1(P) = lfp(T_P^{\mathcal{X}})$ is also the least post-fixed point of $T_P^{\mathcal{X}}$, thus by 8, $lfp(T_P^{\mathcal{X}})$ is the least $\mathcal{X}$-model of P. □

2. Fixpoint semantics of O_2

Consider the set of constrained atoms

$\mathcal{B} = \{c|A \ : \ A$ is an atom and $\mathcal{X} \models \exists(c)\}$ modulo renaming.

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

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. $\mathcal{X} \models c\rho\}$.

Define the semantics $O_2(P)$ as the least solution of a fixpoint equation over $2^{\mathcal{B}}$.

Non-ground immediate consequence operator

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

$$S_P^{\mathcal{X}}(I) = \{c|A \in \mathcal{B} \mid \text{there exists a renamed clause in normal form } (A \leftarrow d|A_1, ..., A_n) \in P, \text{ and constrained atoms } \{c_1|A_1, ..., c_n|A_n\} \subseteq I, \text{ s.t. } c = d \wedge \bigwedge_{i=1}^n c_i \text{ is } \mathcal{X}\text{-satisfiable}\}.$$

Proposition 10 *For any $\mathcal{B}$-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 $(A \leftarrow d|A_1, ..., A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^n c_i$, $\mathcal{X} \models c\rho$ and $\{c_1|A_1, ..., c_n|A_n\} \subset I$

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

iff $A\rho \in T_P^{\mathcal{X}}([I]_{\mathcal{X}})$. $\square$

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

Proposition 11 *$S_P^{\mathcal{X}}$ is continuous.*

PROOF: Let X be a chain of constrained interpretations.

$c|A \in S_P^{\mathcal{X}}(lub(X))$,

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

iff $(A \leftarrow d|A_1, ..., A_n) \in P$, $c = d \wedge \bigwedge_{i=1}^{n} c_i$, $\mathcal{X} \models \exists(c)$ and $\{c_1|A_1, ..., c_n|A_n\} \subset I$, 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 $c|A \in lub(S_P^{\mathcal{X}}(X))$. . $\square$

Corollary 12 *$S_P^{\mathcal{X}}$ admits a least (post) fixed point*

$$F_2(P) = lfp(S_P^{\mathcal{X}}) = S_P^{\mathcal{X}} \uparrow \omega.$$

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).
```

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

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

Theorem 13 *[JL87] 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

$[S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} = [S_P^{\mathcal{X}}(S_P^{\mathcal{X}} \uparrow \alpha - 1)]_{\mathcal{X}}$,

$= T_P^{\mathcal{X}}([S_P^{\mathcal{X}} \uparrow \alpha - 1]_{\mathcal{X}})$ by 10,

$= T_P^{\mathcal{X}}(T_P^{\mathcal{X}} \uparrow \alpha - 1)$ by induction,

$= T_P^{\mathcal{X}} \uparrow \alpha$.

For a limit ordinal, we have

$[S_P^{\mathcal{X}} \uparrow \alpha]_{\mathcal{X}} = [\bigcup_{\beta<\alpha} S_P^{\mathcal{X}} \uparrow \beta]_{\mathcal{X}}$

$= \bigcup_{\beta<\alpha} [S_P^{\mathcal{X}} \uparrow \beta]_{\mathcal{X}}$,

$= \bigcup_{\beta<\alpha} T_P^{\mathcal{X}} \uparrow \beta$ by induction,

$= T_P^{\mathcal{X}} \uparrow \alpha$. $\square$

Full abstraction w.r.t. computed constraints

Theorem 14 (Theorem of full abstraction) *[GL91]* $O_2(P) = F_2(P)$.

PROOF: $F_2(P) \subseteq O_2(P)$ 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 $(A \leftarrow d|A_1, ..., A_n) \in P$, s.t. $\{c_1|A_1, ..., c_n|A_n\} \subseteq S_P^{\mathcal{X}} \uparrow n-1$, $c = d \wedge \bigwedge_{i=1}^{n}$ and $\mathcal{X} \models \exists c$. By induction $\{c_1|A_1, ..., c_n|A_n\} \subseteq O_2(P)$. By definition of O_2 we get $c|A \in O_2(P)$.

$O_2(P) \subseteq F_2(P)$ 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 $c|A \in O_2(P)$ with a derivation of length n. By definition of O_2 there exists $(A \leftarrow d|A_1, ..., A_n) \in P$ s.t. $\{c_1|A_1, ..., c_n|A_n\} \subseteq O_2(P)$, $c = d \wedge \bigwedge_{i=1}^{n}$ and $\mathcal{X} \models \exists c$. By induction $\{c_1|A_1, ..., c_n|A_n\} \subseteq F_2(P)$. Hence by definition of $S_P^{\mathcal{X}}$ we get $A\rho \in F_2(P)$. $\square$

3. Program analysis by abstract interpretation

$S_P^{\mathcal{H}} \uparrow \omega$ captures the set of computed answer constraints with P,

nevertheless this set may be infinite and

it may contain too much information for proving some properties of the computed constraints.

Abstract interpretation [Cousot78] 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 in append?

The term structure can be abstracted by a boolean structure which expresses the groundness of the arguments.

We thus associate a CLP($Bool$) 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)

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

In a success of $append(A, B, C)$ C is ground if and only if 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($Bool$) program:

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

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

Constraint-based Model Checking

Analysis of unbounded states concurrent systems by CLP programs [Delzanno Podelski 99]

Concurrent transition systems defined by condition-action rules [Shankar]

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

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

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

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

Proposition 15 *The set of states from which a set of states defined by a constraint c is reachable is the set*

$lfp(T_P)$ where P is the CLP program plus the clause $p(\vec{x}) \leftarrow c(\vec{x})$.

Computation Tree Logic CTL

Temporal logic for branching time:

States described by propositional or first-order formulas

Two path quantifiers for non-determinism: A "for all transition paths"

E "for some transition path"

Several temporal operators: X "next time", F "eventually",

G "always", U "until".

$AG\neg\phi$ "Safety" property.

$AF\psi$ "Liveness" property.

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

Symbolic Model Checking

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

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

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

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

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

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

Symbolic model checking: uses OBDD's to represent states and transitions as boolean formulas (S is finite).

Constraint-based Model Checking

Constraint-based model checking [Delzanno Podelski 99] applies to Kripke structures with an infinite set of states

Numerical constraints provide a finite representation for an infinite set of states.

Constraint logic programming theory:

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

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

Prototype implementation in Sicstus Prolog + Simplex, CLP(H,FD,R,B)