

Constraint Logic Programming

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

INRIA – Project-Team CONTRAINTES

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

Part III: CLP - Operational and Fixpoint Semantics

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

Part IV: Logical Semantics

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

Part V: Practical CLP Programming

- 8 CLP implementation, the WAM
- 9 Optimizing CLP
- 10 Summing up

Part VI: Concurrent Constraint Programming

11 Introduction

- Syntax
- CC vs. CLP

12 Operational Semantics

- Transitions
- Properties
- Observables

13 Examples

- append
- merge
- $CC(\mathcal{FD})$

Part VII

CC - Denotational Semantics

Part VII: CC - Denotational Semantics

- 14 Deterministic Case
 - Syntax
 - I/O Function
 - Terminal Stores
- 15 Constraint Propagation
 - Closure Operators
 - Chaotic Iteration
- 16 Non-deterministic Case
 - Problems
 - Blind Choice
 - Example: merge
- 17 Sequentiality

Deterministic CC

Agents:

$$A ::= \text{tell}(c) \mid c \rightarrow A \mid A \parallel A \mid \exists xA \mid p(\vec{x})$$

- No choice operator
- Deterministic ask.

Replace non-deterministic pattern matching

$$\forall \vec{x}(c \rightarrow A)$$

by deterministic ask and tell:

$$(\exists \vec{x}c) \rightarrow \exists \vec{x}(\text{tell}(c) \parallel A)$$

Denotational semantics: input/output function

Input: **initial store** c_0

Output: **terminal store** c or *false* for infinite computations

Order the lattice of constraints (\mathcal{C}, \leq) by the information ordering:

$\forall c, d \in \mathcal{C} \ c \leq d$ iff $d \vdash_{\mathcal{X}} c$ iff $\uparrow d \subseteq \uparrow c$ where

$\uparrow c = \{d \in \mathcal{C} \mid c \leq d\}$.

$\llbracket \mathcal{D}.A \rrbracket : \mathcal{C} \rightarrow \mathcal{C}$ is

- ① Extensive: $\forall c \ c \leq \llbracket \mathcal{D}.A \rrbracket c$
- ② Monotone: $\forall c, d \ c \leq d \Rightarrow \llbracket \mathcal{D}.A \rrbracket c \leq \llbracket \mathcal{D}.A \rrbracket d$
- ③ Idempotent: $\forall c \ \llbracket \mathcal{D}.A \rrbracket c = \llbracket \mathcal{D}.A \rrbracket (\llbracket \mathcal{D}.A \rrbracket c)$

i.e. $\llbracket \mathcal{D}.A \rrbracket$ is a **closure operator** over (\mathcal{C}, \leq) .

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\begin{aligned} &\llbracket \mathcal{D}.tell(c) \rrbracket \\ &\llbracket \mathcal{D}.c \rightarrow A \rrbracket \end{aligned}$$

$$\begin{aligned} &\llbracket \mathcal{D}.A \parallel B \rrbracket \\ &\llbracket \mathcal{D}.\exists x A \rrbracket \\ &\llbracket \mathcal{D}.p(\vec{x}) \rrbracket \end{aligned}$$

$$\text{if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s. \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\begin{aligned} \llbracket \mathcal{D}.tell(c) \rrbracket &= \uparrow c && (\simeq \lambda s.s \wedge c) \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket & && \end{aligned}$$

$$\begin{aligned} \llbracket \mathcal{D}.A \parallel B \rrbracket \\ \llbracket \mathcal{D}.\exists x A \rrbracket \\ \llbracket \mathcal{D}.p(\vec{x}) \rrbracket \end{aligned} \quad \text{if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s.\llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\llbracket \mathcal{D}.tell(c) \rrbracket = \uparrow c \quad (\simeq \lambda s. s \wedge c)$$

$$\llbracket \mathcal{D}.c \rightarrow A \rrbracket = (\mathcal{C} \setminus \uparrow c) \cup (\uparrow c \cap \llbracket \mathcal{D}.A \rrbracket)$$

$(\simeq \lambda s. \text{if } s \vdash_{\mathcal{C}} c \text{ then } \llbracket \mathcal{D}.A \rrbracket s \text{ else } s)$

$$\llbracket \mathcal{D}.A \parallel B \rrbracket$$

$$\llbracket \mathcal{D}.\exists x A \rrbracket$$

$$\llbracket \mathcal{D}.p(\vec{x}) \rrbracket$$

$$\text{if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s. \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\llbracket \mathcal{D}.tell(c) \rrbracket = \uparrow c \quad (\simeq \lambda s. s \wedge c)$$

$$\llbracket \mathcal{D}.c \rightarrow A \rrbracket = (\mathcal{C} \setminus \uparrow c) \cup (\uparrow c \cap \llbracket \mathcal{D}.A \rrbracket)$$

$(\simeq \lambda s. \text{if } s \vdash_{\mathcal{C}} c \text{ then } \llbracket \mathcal{D}.A \rrbracket s \text{ else } s)$

$$\llbracket \mathcal{D}.A \parallel B \rrbracket = \llbracket \mathcal{D}.A \rrbracket \cap \llbracket \mathcal{D}.B \rrbracket \quad (\simeq \bigvee (\lambda s. \llbracket \mathcal{D}.A \rrbracket \llbracket \mathcal{D}.B \rrbracket s))$$

$$\llbracket \mathcal{D}.\exists x A \rrbracket$$

$$\llbracket \mathcal{D}.p(\vec{x}) \rrbracket \quad \text{if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s. \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\llbracket \mathcal{D}.tell(c) \rrbracket = \uparrow c \quad (\simeq \lambda s. s \wedge c)$$

$$\llbracket \mathcal{D}.c \rightarrow A \rrbracket = (\mathcal{C} \setminus \uparrow c) \cup (\uparrow c \cap \llbracket \mathcal{D}.A \rrbracket)$$

$(\simeq \lambda s. \text{if } s \vdash_{\mathcal{C}} c \text{ then } \llbracket \mathcal{D}.A \rrbracket s \text{ else } s)$

$$\llbracket \mathcal{D}.A \parallel B \rrbracket = \llbracket \mathcal{D}.A \rrbracket \cap \llbracket \mathcal{D}.B \rrbracket \quad (\simeq \forall (\lambda s. \llbracket \mathcal{D}.A \rrbracket \llbracket \mathcal{D}.B \rrbracket s))$$

$$\llbracket \mathcal{D}.\exists x A \rrbracket = \{d \mid c \in \llbracket \mathcal{D}.A \rrbracket, \exists xc = \exists xd\} \quad (\simeq \lambda s. \exists x \llbracket \mathcal{D}.A \rrbracket \exists xs)$$

$$\llbracket \mathcal{D}.p(\vec{x}) \rrbracket \quad \text{if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s. \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Semantic Equations

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{C})$ be a **closure operator** presented by the set of its fixpoints, and defined as **the least fixpoint set** of:

$$\llbracket \mathcal{D}.tell(c) \rrbracket = \uparrow c \quad (\simeq \lambda s. s \wedge c)$$

$$\llbracket \mathcal{D}.c \rightarrow A \rrbracket = (\mathcal{C} \setminus \uparrow c) \cup (\uparrow c \cap \llbracket \mathcal{D}.A \rrbracket)$$

$(\simeq \lambda s. \text{if } s \vdash_{\mathcal{C}} c \text{ then } \llbracket \mathcal{D}.A \rrbracket s \text{ else } s)$

$$\llbracket \mathcal{D}.A \parallel B \rrbracket = \llbracket \mathcal{D}.A \rrbracket \cap \llbracket \mathcal{D}.B \rrbracket \quad (\simeq \forall (\lambda s. \llbracket \mathcal{D}.A \rrbracket \llbracket \mathcal{D}.B \rrbracket s))$$

$$\llbracket \mathcal{D}.\exists x A \rrbracket = \{d \mid c \in \llbracket \mathcal{D}.A \rrbracket, \exists xc = \exists xd\} \quad (\simeq \lambda s. \exists x \llbracket \mathcal{D}.A \rrbracket \exists xs)$$

$$\llbracket \mathcal{D}.p(\vec{x}) \rrbracket = \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket \text{ if } p(\vec{y}) = A \in \mathcal{D} \quad (\simeq \lambda s. \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket s)$$

Theorem 1 ([SRP91])

For any deterministic process $\mathcal{D}.A$

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \begin{cases} \{ \min(\llbracket \mathcal{D}.A \rrbracket \cap \uparrow c) \} & \text{if } \llbracket \mathcal{D}.A \rrbracket \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

Chaotic Iteration of Monotone Operators

Let $L(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ be a complete lattice, and $F : L^n \rightarrow L^n$ a monotone operator over L^n with $n > 0$.

The **chaotic iteration** of F from $D \in L^n$ for a fair transfinite choice sequence $\langle J^\delta : \delta \in \text{Ord} \rangle$ is the sequence $\langle X^\delta \rangle$:

$$X^0 = D,$$

$$X_i^{\delta+1} = F_i(X^\delta) \text{ if } i \in J^\delta, X_i^{\delta+1} = X_i^\delta \text{ otherwise,}$$

$$X_i^\delta = \bigsqcup_{\alpha < \delta} X_i^\alpha \text{ for any limit ordinal } \delta.$$

Theorem 2 ([CC77])

Let $D \in L^n$ be a pre fixpoint of F (i.e. $D \sqsubseteq F(D)$). Any chaotic iteration of F starting from D is increasing and has for limit the least fixpoint of F above D .

Constraint Propagation as Chaotic Iteration

Corollary 3 (Correctness of constraint propagation)

Let $c = a_1 \wedge \dots \wedge a_n$, and E be an environment. Then $\bar{c}(E)$ is the limit of any fair iteration of closure operators $\bar{a}_1, \dots, \bar{a}_n$ from E .

Let $F : L^{n+1} \rightarrow L^{n+1}$ be defined by its projections F_i 's:

$$\left\{ \begin{array}{l} E_1 = \bar{a}_1(E) = F_1(E_1, \dots, E_n, E) \\ E_2 = \bar{a}_2(E) = F_2(E_1, \dots, E_n, E) \\ \dots \\ E_n = \bar{a}_n(E) = F_n(E_1, \dots, E_n, E) \\ E = E_1 \cap \dots \cap E_n = F_{n+1}(E_1, \dots, E_n, E) \end{array} \right.$$

The functions F_i 's are obviously monotonic, any fair iteration of $\bar{a}_1, \dots, \bar{a}_n$ is thus a chaotic iteration of F_1, \dots, F_{n+1} therefore its limit is equal to the least fixpoint greater than E , i.e. $\bar{c}(E)$.

Non-deterministic $CC(\mathcal{X})$ with Local Choice (1)

The set of terminal stores of a CC process with **blind choice** can be characterized easily by adding the semantic equation:

$$\llbracket \mathcal{D}.A + B \rrbracket = \llbracket \mathcal{D}.A \rrbracket \cup \llbracket \mathcal{D}.B \rrbracket$$

Theorem 4 ([dBGP96])

$$\llbracket \mathcal{D}.A \rrbracket = \bigcup_{c \in \mathcal{C}} \mathcal{O}_{ts}(\mathcal{D}.A; c)$$

but the input-output relation cannot be recovered from $\llbracket \mathcal{D}.A \rrbracket$:

$$\llbracket tell(true) \rrbracket = \mathcal{C}$$

$$\llbracket tell(true) + tell(c) \rrbracket = \mathcal{C}$$

$$\mathcal{O}_{ts}(tell(true); true) = \{true\}$$

$$\mathcal{O}_{ts}(tell(true) + tell(c); true) = \{true, c\}$$

Idea: define $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ to distinguish between branches.

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\llbracket \mathcal{D}.c \rrbracket =$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \end{aligned}$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \{\mathcal{C} \setminus \uparrow c\} \cup \{\uparrow c \cap X \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.A \parallel B \rrbracket &= \end{aligned}$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \{\mathcal{C} \setminus \uparrow c\} \cup \{\uparrow c \cap X \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.A \parallel B \rrbracket &= \{X \cap Y \mid X \in \llbracket \mathcal{D}.A \rrbracket, Y \in \llbracket \mathcal{D}.B \rrbracket\} \\ \llbracket \mathcal{D}.A + B \rrbracket &= \end{aligned}$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \{\mathcal{C} \setminus \uparrow c\} \cup \{\uparrow c \cap X \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.A \parallel B \rrbracket &= \{X \cap Y \mid X \in \llbracket \mathcal{D}.A \rrbracket, Y \in \llbracket \mathcal{D}.B \rrbracket\} \\ \llbracket \mathcal{D}.A + B \rrbracket &= \llbracket \mathcal{D}.A \rrbracket \cup \llbracket \mathcal{D}.B \rrbracket \\ \llbracket \mathcal{D}.\exists x A \rrbracket &= \end{aligned}$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \{\mathcal{C} \setminus \uparrow c\} \cup \{\uparrow c \cap X \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.A \parallel B \rrbracket &= \{X \cap Y \mid X \in \llbracket \mathcal{D}.A \rrbracket, Y \in \llbracket \mathcal{D}.B \rrbracket\} \\ \llbracket \mathcal{D}.A + B \rrbracket &= \llbracket \mathcal{D}.A \rrbracket \cup \llbracket \mathcal{D}.B \rrbracket \\ \llbracket \mathcal{D}.\exists x A \rrbracket &= \{\{d \mid \exists xc = \exists xd, c \in X\} \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.p(\vec{x}) \rrbracket &= \end{aligned}$$

Non-deterministic $CC(\mathcal{X})$ with Local Choice (2)

Let $\llbracket \cdot \rrbracket : \mathcal{D} \times A \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{C}))$ be the least fixpoint (for \subseteq) of

$$\begin{aligned} \llbracket \mathcal{D}.c \rrbracket &= \{\uparrow c\} \\ \llbracket \mathcal{D}.c \rightarrow A \rrbracket &= \{\mathcal{C} \setminus \uparrow c\} \cup \{\uparrow c \cap X \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.A \parallel B \rrbracket &= \{X \cap Y \mid X \in \llbracket \mathcal{D}.A \rrbracket, Y \in \llbracket \mathcal{D}.B \rrbracket\} \\ \llbracket \mathcal{D}.A + B \rrbracket &= \llbracket \mathcal{D}.A \rrbracket \cup \llbracket \mathcal{D}.B \rrbracket \\ \llbracket \mathcal{D}.\exists x A \rrbracket &= \{\{d \mid \exists xc = \exists xd, c \in X\} \mid X \in \llbracket \mathcal{D}.A \rrbracket\} \\ \llbracket \mathcal{D}.p(\vec{x}) \rrbracket &= \llbracket \mathcal{D}.A[\vec{x}/\vec{y}] \rrbracket \end{aligned}$$

Theorem 5 ([MFP97])

For any process $\mathcal{D}.A$,

$$\mathcal{O}_{ts}(\mathcal{D}.A; c) = \{d \mid \text{there exists } X \in \llbracket \mathcal{D}.A \rrbracket \text{ s.t. } d = \min(\uparrow c \cap X)\}.$$

'merge' Example Revisited

Merging streams

$$\begin{aligned}
 \text{merge}(A, B, C) = & \\
 & (A = [] \rightarrow \text{tell}(C = B)) \parallel \\
 & (B = [] \rightarrow \text{tell}(C = A)) \parallel \\
 & (\forall X, L(A = [X|L] \rightarrow \text{tell}(C = [X|R]) \parallel \text{merge}(L, B, R)) + \\
 & \forall X, L(B = [X|L] \rightarrow \text{tell}(C = [X|R]) \parallel \text{merge}(A, L, R)))
 \end{aligned}$$

Do we have the expected terminal stores?

'merge' Example Revisited

Merging streams

$$\begin{aligned}
 \text{merge}(A, B, C) = & \\
 & (A = [] \rightarrow \text{tell}(C = B)) \parallel \\
 & (B = [] \rightarrow \text{tell}(C = A)) \parallel \\
 & (\forall X, L(A = [X|L] \rightarrow \text{tell}(C = [X|R]) \parallel \text{merge}(L, B, R)) + \\
 & \forall X, L(B = [X|L] \rightarrow \text{tell}(C = [X|R]) \parallel \text{merge}(A, L, R)))
 \end{aligned}$$

Do we have the expected terminal stores?

No!

for $\text{merge}(X, [1|Y], Z)$ we don't necessarily get 1 in Z , the merging is not *greedy*...

Sequentiality

Let us define a new operator, \bullet , as follows:

$$\frac{(X; c; A) \longrightarrow (Y; d; B)}{(X; c; A \bullet C, \Gamma) \longrightarrow (Y; d; B \bullet C, \Gamma)} \quad (X; c; \emptyset \bullet A) \longrightarrow (X; c; A)$$

We can characterize completely the observables of any CC_{seq} program, $\mathcal{D}.A$, by those of a new CC (without \bullet) program, $\mathcal{D}^\bullet.A^\bullet$, in a new constraint system, \mathcal{C}^\bullet .

Idea

Let ok be a **new** relation symbol of arity one. \mathcal{C}^\bullet is the constraint system \mathcal{C} to which ok is added, without any non-logical axiom. The program $\mathcal{D}^\bullet.A^\bullet$ is defined inductively as follows:

$$(p(\vec{y}) = A)^\bullet = p^\bullet(x, \vec{y}) = A_x^\bullet$$

$$A^\bullet = \exists x A_x^\bullet$$

$$tell(c)_x^\bullet = tell(c \wedge ok(x))$$

$$p(\vec{y})_x^\bullet = p^\bullet(x, \vec{y})$$

$$(A \parallel B)_x^\bullet = \exists y, z (A_y^\bullet \parallel B_z^\bullet \parallel (ok(y) \wedge ok(z)) \rightarrow ok(x))$$

$$(A + B)_x^\bullet = A_x^\bullet + B_x^\bullet$$

$$(\forall \vec{y} (c \rightarrow A))_x^\bullet = \forall \vec{z} (c[\vec{z}/\vec{y}] \rightarrow A[\vec{z}/\vec{y}]_x^\bullet) \text{ with } x \notin \vec{z}$$

$$(\exists y A)_x^\bullet = \exists z A[z/y]_x^\bullet \text{ with } z \neq x$$

$$(A \bullet B)_x^\bullet =$$

Idea

Let ok be a **new** relation symbol of arity one. \mathcal{C}^\bullet is the constraint system \mathcal{C} to which ok is added, without any non-logical axiom. The program $\mathcal{D}^\bullet.A^\bullet$ is defined inductively as follows:

$$(p(\vec{y}) = A)^\bullet = p^\bullet(x, \vec{y}) = A_x^\bullet$$

$$A^\bullet = \exists x A_x^\bullet$$

$$tell(c)_x^\bullet = tell(c \wedge ok(x))$$

$$p(\vec{y})_x^\bullet = p^\bullet(x, \vec{y})$$

$$(A \parallel B)_x^\bullet = \exists y, z (A_y^\bullet \parallel B_z^\bullet \parallel (ok(y) \wedge ok(z)) \rightarrow ok(x))$$

$$(A + B)_x^\bullet = A_x^\bullet + B_x^\bullet$$

$$(\forall \vec{y} (c \rightarrow A))_x^\bullet = \forall \vec{z} (c[\vec{z}/\vec{y}] \rightarrow A[\vec{z}/\vec{y}]_x^\bullet) \text{ with } x \notin \vec{z}$$

$$(\exists y A)_x^\bullet = \exists z A[z/y]_x^\bullet \text{ with } z \neq x$$

$$(A \bullet B)_x^\bullet = \exists y (A_y^\bullet \parallel ok(y) \rightarrow B_x^\bullet)$$

Part VIII

CC and Linear Logic

Part VIII: CC and Linear Logic

- 18 CC - Logical Semantics
 - Intuitionistic
 - Linear
 - Soundness
 - Completeness

- 19 Must Properties
 - Definition
 - Soundness
 - Completeness

- 20 Program Analysis
 - Equivalence
 - Phase Semantics and Theorem Proving

Logical Semantics of CC?

- CC calculus is sound but not complete w.r.t. CLP logical semantics interpreting *asks* as *tells*
- Interpreting $ask(c \rightarrow A)$ as logical implication leads to identify **CC transitions** with **logical deductions**:

$$left \rightarrow \frac{c \vdash_c d}{c \wedge (d \rightarrow A^\dagger) \vdash c \wedge A^\dagger} \quad \frac{p(\vec{x}) \vdash_{\mathcal{D}} A^\dagger}{c \wedge p(\vec{x}) \vdash c \wedge A^\dagger}$$

(reverses the arrow of CLP interpretation...)

- To distinguish between successes and accessible stores agents shouldn't "disappear" by the \rightarrow rule:

Logical Semantics of CC?

- CC calculus is sound but not complete w.r.t. CLP logical semantics interpreting *asks* as *tells*
- Interpreting $ask(c \rightarrow A)$ as logical implication leads to identify **CC transitions** with **logical deductions**:

$$left \rightarrow \frac{c \vdash_c d}{c \wedge (d \rightarrow A^\dagger) \vdash c \wedge A^\dagger} \quad \frac{p(\vec{x}) \vdash_{\mathcal{D}} A^\dagger}{c \wedge p(\vec{x}) \vdash c \wedge A^\dagger}$$

(reverses the arrow of CLP interpretation...)

- To distinguish between successes and accessible stores agents shouldn't "disappear" by the **weakening** rule:

$$leftW \frac{\Gamma \vdash c}{\Gamma, A^\dagger \vdash c}$$

Linear Logic

- Introduced by Jean-Yves Girard in 1986 as a new *constructive* logic without the asymmetry of intuitionistic logic (sequent calculus with symmetric left and right sides)
- Logic of **resource consumption**

$$A \otimes A \not\vdash_{LL} A$$

$$A \otimes (A \multimap B) \vdash_{LL} B$$

$$A \otimes (A \multimap B) \not\vdash_{LL} A \otimes B$$

- $!A$ provides arbitrary duplication (unbounded throwable resource)

$$!A \otimes (A \multimap B) \vdash_{LL} !A \otimes B \vdash_{LL} B$$

- Sequent calculus **without weakening and contraction**

Intuitionistic Linear Logic

Multiplicatives

$$\frac{\Gamma, A, B \vdash C}{\Gamma, A \otimes B \vdash C} \quad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} \quad \frac{\Gamma \vdash A \quad \Delta, B \vdash C}{\Delta, \Gamma, A \multimap B \vdash C} \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}$$

Additives

$$\frac{\Gamma, A \vdash C \quad \Gamma, B \vdash C}{\Gamma, A \oplus B \vdash C} \quad \frac{\Gamma \vdash A}{\Gamma \vdash A \oplus B} \quad \frac{\Gamma \vdash B}{\Gamma \vdash A \oplus B}$$

$$\frac{\Gamma, A \vdash C}{\Gamma, A \& B \vdash C} \quad \frac{\Gamma, B \vdash C}{\Gamma, A \& B \vdash C} \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \& B}$$

Constants

$$\frac{\Gamma \vdash A}{\Gamma, \mathbf{1} \vdash A} \quad \vdash \mathbf{1} \quad \perp \vdash \quad \frac{\Gamma \vdash}{\Gamma \vdash \perp} \quad \Gamma \vdash \top \quad \Gamma, \mathbf{0} \vdash A$$

Intuitionistic Linear Logic (cont.)

Axiom - Cut

$$A \vdash A \quad \frac{\Gamma \vdash A \quad \Delta, A \vdash B}{\Delta, \Gamma \vdash B}$$

Bang

$$\frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B} \quad \frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B} \quad \frac{\Gamma \vdash B}{\Gamma, !A \vdash B} \quad \frac{! \Gamma \vdash A}{! \Gamma \vdash !A}$$

Quantifiers

$$\frac{\Gamma, A[t/x] \vdash B}{\Gamma, \forall x A \vdash B} \quad \frac{\Gamma \vdash A}{\Gamma \vdash \forall x A} \quad x \notin \text{fv}(\Gamma)$$

$$\frac{\Gamma, A \vdash B}{\Gamma, \exists x A \vdash B} \quad x \notin \text{fv}(\Gamma, B) \quad \frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x A}$$

Intuit. Linear Logic = the Logic of CC agents

Translation:

$$\begin{array}{lll} (c \rightarrow A)^\dagger = c \multimap A^\dagger & (A \parallel B)^\dagger = A^\dagger \otimes B^\dagger & \text{tell}(c)^\dagger = !c \\ (A + B)^\dagger = A^\dagger \& B^\dagger & (\exists x A)^\dagger = \exists x A^\dagger & p(\vec{x})^\dagger = p(\vec{x}) \\ (X; c; \Gamma)^\dagger = \exists X(!c \otimes \Gamma^\dagger) & & \end{array}$$

Axioms: $!c \vdash !d$ for all $c \vdash_c d$ $p(\vec{x}) \vdash A^\dagger$ for all $p(\vec{x}) = A \in \mathcal{D}$

Intuit. Linear Logic = the Logic of CC agents

Translation:

$$\begin{array}{lll}
 (c \rightarrow A)^\dagger = c \multimap A^\dagger & (A \parallel B)^\dagger = A^\dagger \otimes B^\dagger & \text{tell}(c)^\dagger = !c \\
 (A + B)^\dagger = A^\dagger \& B^\dagger & (\exists x A)^\dagger = \exists x A^\dagger & p(\vec{x})^\dagger = p(\vec{x}) \\
 (X; c; \Gamma)^\dagger = \exists X(!c \otimes \Gamma^\dagger)
 \end{array}$$

Axioms: $!c \vdash !d$ for all $c \vdash_c d$ $p(\vec{x}) \vdash A^\dagger$ for all $p(\vec{x}) = A \in \mathcal{D}$

Soundness and Completeness

If $(c; \Gamma) \rightarrow_{CC} (d; \Delta)$ then $c^\dagger \otimes \Gamma^\dagger \vdash_{ILL(c, \mathcal{D})} d^\dagger \otimes \Delta^\dagger$.

If $A^\dagger \vdash_{ILL(c, \mathcal{D})} c$ then *there exists a **success store** d such that $(\text{true}; A) \rightarrow_{CC} (d; \emptyset)$ and $d \vdash_c c$.*

If $A^\dagger \vdash_{ILL(c, \mathcal{D})} c \otimes \top$ then *there exists an **accessible store** d such that $(\text{true}; A) \rightarrow_{CC} (d; \Gamma)$ and $d \vdash_c c$.*

Soundness

Theorem 6 (Soundness of transitions)

Let $(X; c; \Gamma)$ and $(Y; d; \Delta)$ be CC configurations.

If $(X; c; \Gamma) \equiv (Y; d; \Delta)$ then $(X; c; \Gamma)^\dagger \dashv\vdash_{ILL(\mathcal{C}, \mathcal{D})} (Y; d; \Delta)^\dagger$.

If $(X; c; \Gamma) \longrightarrow (Y; d; \Delta)$ then $(X; c; \Gamma)^\dagger \vdash_{ILL(\mathcal{C}, \mathcal{D})} (Y; d; \Delta)^\dagger$.

Proof.

By induction on \equiv . Immediate.

By induction on \longrightarrow .

The choice operator $+$ is translated by the additive conjunction $\&$, which expresses “may” properties: $A \& B \vdash A$ and $A \& B \vdash B$. \square

Completeness I

Theorem 7 (Observation of successes)

Let A be a CC agent and c be a constraint.

If $A^\dagger \vdash_{ILL(\mathcal{C}, \mathcal{D})} c$, then there exists a constraint d such that $(\emptyset; 1; A) \longrightarrow (X; d; \emptyset)$ and $\exists X d \vdash_{\mathcal{C}} c$.

Proof.

By induction on a sequent calculus proof π of

$$A_1^\dagger, \dots, A_n^\dagger \vdash_{ILL(\mathcal{C}, \mathcal{D})} \phi$$

where the A_i 's are agents and ϕ is either a constraint or a procedure name. □

Completeness II

Recall that \top is the additive true constant neutral for $\&$.

Theorem 8 (Observation of accessible stores)

Let A be a CC agent and c be a constraint.

*If $A^\dagger \vdash_{ILL(\mathcal{C}, \mathcal{D})} c \otimes \top$, then c is a store accessible from A ,
i.e. there exist a constraint d and a multiset Γ of agents such that
 $(\emptyset; 1; A) \longrightarrow (X; d; \Gamma)$ and $\exists X d \vdash_{\mathcal{C}} c$.*

Proof.

The proof uses the first completeness theorem, and proceeds by induction for the right introduction of the tensor connective in $c \otimes \top$. □

Observing “must” Properties

Properties true on **all branches** on the derivation tree.

Redefine the operational semantics by a rewriting relation on **frontiers**, i.e. multisets of configurations

Blind choice

$$\langle (X; c; A + B), \Phi \rangle \Longrightarrow \langle (X; c; A), (X; c; B), \Phi \rangle$$

Tell

$$\langle (X; c; \text{tell}(d), \Gamma), \Phi \rangle \Longrightarrow \langle (X; c \wedge d; \Gamma), \Phi \rangle$$

Ask

$$\frac{c \vdash_c d}{\langle (X; c; d \rightarrow A, \Gamma), \Phi \rangle \Longrightarrow \langle (X; c; A, \Gamma), \Phi \rangle}$$

Procedure calls

$$\frac{(p(\vec{y}) = A) \in \mathcal{D}}{\langle (X; c; p(\vec{y}), \Gamma), \Phi \rangle \Longrightarrow \langle (X; c; A, \Gamma), \Phi \rangle}$$

Translating the Frontier Calculus in LL with

Translate

$$(A + B)^{\ddagger} =$$

$$\langle (X; c; A), \Phi \rangle^{\ddagger} =$$

same translation for the other operations

Theorem 9 (Soundness of transitions)

Let Φ and Ψ be two frontiers.

If $\Phi \equiv \Psi$ then $(\Phi)^{\ddagger} \Vdash_{ILL(C,D)} (\Psi)^{\ddagger}$.

If $\Phi \Longrightarrow \Psi$ then $\Phi^{\ddagger} \vdash_{ILL(C,D)} \Psi^{\ddagger}$.

Translating the Frontier Calculus in LL with \oplus

Translate

$$(A + B)^{\ddagger} = A^{\ddagger} \oplus B^{\ddagger}$$

$$\langle (X; c; A), \Phi \rangle^{\ddagger} = \exists X (c^{\ddagger} \otimes A^{\ddagger}) \oplus \Phi^{\ddagger}$$

same translation for the other operations

Theorem 9 (Soundness of transitions)

Let Φ and Ψ be two frontiers.

If $\Phi \equiv \Psi$ then $(\Phi)^{\ddagger} \Vdash_{ILL(C,D)} (\Psi)^{\ddagger}$.

If $\Phi \Longrightarrow \Psi$ then $\Phi^{\ddagger} \vdash_{ILL(C,D)} \Psi^{\ddagger}$.

Completeness III for “must” Properties

Theorem 10 (Observation of frontiers' accessible stores)

Let A be a CC agent and c be a constraint.

If $A^\dagger \vdash_{ILL(C,D)} c \otimes \top$

then $\langle (\emptyset; 1; A) \rangle \Longrightarrow \langle (X_1; d_1; \Gamma_1), \dots, (X_n; d_n; \Gamma_n) \rangle$ with

$\forall j \exists X_j d_j \vdash_c c$

Theorem 11 (Observation of frontiers' success stores)

Let A be an CC agent and c be a constraint.

If $A^\dagger \vdash_{ILL(C,D)} c$

then $\langle (\emptyset; 1; A) \rangle \Longrightarrow \langle (X_1; d_1; \emptyset), \dots, (X_n; d_n; \emptyset) \rangle$ with $\forall j \exists X_j d_j \vdash_c c$

Bibliography I



Patrick Cousot and Radhia Cousot.

Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints.

In *POPL'77: Proceedings of the 6th ACM Symposium on Principles of Programming Languages*, pages 238–252, New York, 1977. ACM Press. Los Angeles.



Frank S. de Boer, Maurizio Gabbrielli, and Catuscia Palamidessi.

Proving correctness of constraint logic programming with dynamic scheduling.

In *Proceedings of SAS'96*, LNCS 1145. Springer-Verlag, 1996.



François Fages, Paul Ruet, and Sylvain Soliman.

Phase semantics and verification of concurrent constraint programs.

In *Proceedings of the 13th Annual IEEE Symposium on Logic In Computer Science*, pages 141–152, Indianapolis, 1998. IEEE Computer Society.



Kim Marriott Moreno Falaschi, Maurizio Gabbrielli and Catuscia Palamidessi.

Confluence in concurrent constraint programming.

Theoretical Computer Science, 183(2):281–315, 1997.



Vijay A. Saraswat, Martin C. Rinard, and Prakash Panangaden.

Semantic foundations of concurrent constraint programming.

In *POPL'91: Proceedings of the 18th ACM Symposium on Principles of Programming Languages*, 1991.