From subexponentials in linear logic to concurrent constraint programming and back

Carlos Olarte
Joint work with Elaine Pimentel and Vivek Nigam

ECT, Universidade Federal do Rio Grande do Norte

October 13, 2015
Workshop on Logic, Language and Information
Motivation

Our objective

Languages and reasoning techniques for the specification and verification of concurrent systems where different modalities can be combined.

Potential target applications:

- Multimedia Interactive Systems
- Biochemical Systems
- Mobile systems, Social Networks, distributed systems.
- **Spatial modalities**: locations, places, devices, biochemical interaction domains....
- **Epistemic modalities**: beliefs, opinions, facts, lies...
- **Temporal modalities**: System’s configuration evolves along time-units.
Motivation
Concurrent Constraint Programming (CCP)

A simple and powerful model of concurrency tied to logic:

- Systems are specified by constraints (i.e., formulas in logic) representing partial information on the variables of the system.
- Agents tell and ask constraints on a shared store of constraints.
Motivation

Concurrent Constraint Programming (CCP)

A simple and powerful model of concurrency tied to logic:

- Systems are specified by constraints (i.e., formulas in logic) representing partial information on the variables of the system.
- Agents tell and ask constraints on a shared store of constraints.
Motivation
Concurrent Constraint Programming (CCP)

A simple and powerful model of concurrency tied to logic:

- Systems are specified by **constraints** (i.e., formulas in logic) representing **partial information** on the variables of the system.
- Agents **tell** and **ask** constraints on a shared **store** of constraints.

```plaintext
ask 0<temperature<100 then Q

42 <temperature<70

ask temperature = 50 then P
```
Motivation

Concurrent Constraint Programming (CCP)

A simple and powerful model of concurrency tied to logic:

- Systems are specified by constraints (i.e., formulas in logic) representing partial information on the variables of the system.
- Agents tell and ask constraints on a shared store of constraints.

```
ask temperature = 50

Q

42 < temperature < 70

Remains Blocked

ask temperature = 50 then P
```
Motivation

CCP Calculi

CCP has been extended to deal with different application domains:

- **tcc**: Reactive and timed systems [SJG94].
- **pccp**: Probabilistic choices [GJS97].
- **lccp**: Linearity and resources [FRS01].
- **ntcc**: Time, non-determinism and asynchrony [NPV02].
- **cc-pi**, **utcc**: Mobility [BM07, OV08].
- **soft-cgp**: Soft constraints and preferences [BMR06].
- **eccp** and **sccp**: Epistemic and Spatial reasoning [KPPV12].
Motivation
CCP Calculi

CCP has been extended to deal with different application domains:

- **tcc**: Reactive and timed systems [SJG94].
- **ppcp**: Probabilistic choices [GJS97].
- **lccp**: Linearity and resources [FRS01].
- **ntcc**: Time, non-determinism and asynchrony [NPV02].
- **cc-pi, utcc**: Mobility [BM07, OV08].
- **soft-ccp**: Soft constraints and preferences [BMR06].
- **eccp** and **sccp**: Epistemic and Spatial reasoning [KPPV12].

The idea
Reason about different CCP systems in one logical framework.
Motivation

Subexponentials in Linear Logic (SELL)

Linear logic:

- Formulas are seen as resources, e.g., \( c \otimes c \not\vdash c \).
- Classical reasoning is recovered by the use of exponentials: \( !c \vdash c \otimes c \).

Subexponentials [DJS93]

Intuitively, \( !^a F \) means “\( F \) holds in location \( a \)”.

Such exponentials are not canonical:

\[(\text{in general}) \quad !^a c \not\equiv !^b c \text{ if } a \neq b\]
Motivation

Subexponentials in Linear Logic (SELL)

Linear logic:

- Formulas are seen as resources, e.g., $c \otimes c \nvdash c$.
- Classical reasoning is recovered by the use of exponentials: $!c \vdash c \otimes c$

Subexponentials [DJS93]

Intuitively, $!^a F$ means “$F$ holds in location $a$”. Such exponentials are not canonical:

$$(\text{in general}) \; !^a c \not\equiv !^b c \; \text{if} \; a \neq b$$

The Idea

Quantification on location may allow the specification of interesting behaviours in concurrency.
This work is about

SELL (?!s,?!s)

Quantification on subexp.

SELL^n (?!s,?!s,∪,∩)

SELL^n ⇒ ccp
SELL^n const. sys.

New ccp models: 
- distributed ccp
- linear sccp
- soft constraints
- dynamic shared-spaces

ccp ⇒ SELL^n
∇l, ≤

Proof systems for:
- (linear) ccp
- Epistemic ccp
- Spatial ccp
- Timed ccp

Carlos Olarte (UFRN)
Outline

1 Modalities in CCP

2 SELL interpretation of CCP processes

3 SELL as Constraint System

4 Concluding Remarks
tell(c) adds c to the store (d) leading to d ∧ c.

The process ask c then P evolves into P if c can be deduced from the store. This is a simple and powerful synchronization mechanism.

P || Q: parallel execution of P and Q.

(local x) P: local variables.

Given a definition, p(\bar{y}) \equiv P, the process p(\bar{x}) reduces to P[\bar{x}/\bar{y}].

A simple example: Classical coffee machine

(\begin{align*}
tell(coin) \parallel ask \hfill coin \text{ then } &\text{tell(coffee), } true \hfill \longrightarrow \\
ask \hfill coin \text{ then } &\text{tell(coffee), } coin \hfill \longrightarrow \\
skip, &\text{coin ∧ coffee}
\end{align*})
Outline

1. Modalities in CCP
2. SELL interpretation of CCP processes
3. SELL as Constraint System
4. Concluding Remarks
Linear CCP [FRS01]

Constraints as formulae in (a fragment of) Girard’s ILL:

Ask agents **consume** information when evolving.

The linear coffee machine

\[
\begin{align*}
& (\text{tell}(\text{coin}) \parallel \text{ask} \; \text{coin} \; \text{then} \; \text{coffee}, \text{true}) \rightarrow \\
& (\text{ask} \; \text{coin} \; \text{then} \; \text{coffee}, \text{coin}) \rightarrow (\text{skip}, \text{coffee})
\end{align*}
\]

Declarative Reading of 1cc processes

[FRS01] showed that (L)CCP processes can be read as formulae in ILL:

\[
(P, c) \rightarrow^* (Q, d) \text{ iff } \mathcal{L}[P] \otimes c \vdash \mathcal{L}[Q] \otimes d
\]
Focusing and Adequacy

Logical and operational steps do not correspond (closely) to each other:

- **Process:** $P = \text{tell}(c) \parallel \text{ask } c \text{ then tell}(d) \parallel \text{ask } d \text{ then tell}(e)$
- **Operational side:** $P \Downarrow^e (P \text{ output } e)$.
- **Logical side** $c \otimes (c \dashv o d) \otimes (d \dashv o e) \vdash e$, but:

$$
\begin{align*}
  c \vdash c & \quad d \vdash d \\
  e \vdash e & \quad c, c \dashv o d \vdash d \\
  c, c \dashv o d, d \dashv o e \vdash e
\end{align*}
$$

Andreoli’s focusing system [And92]:

- **negative** connectives $\dashv o, \&, \top, \forall, \ldots$
- **positive** connectives: $\otimes, \oplus, \exists, \ldots$
Focusing and Adequacy

Negative Phase

\[
\frac{[K : \Gamma], \Delta, F, G \rightarrow R}{[K : \Gamma], \Delta, F \otimes G \rightarrow R} \quad \otimes_L \quad \frac{[K : \Gamma], \Delta, F \rightarrow G}{[K : \Gamma], \Delta \rightarrow F \rightarrow G} \quad \circ_R \quad \frac{[K : \Gamma], \Delta \rightarrow G[\chi_e/x]}{[K : \Gamma], \Delta \rightarrow \forall x. G} \quad \forall_R
\]

Positive Phase

\[
\frac{[K_1 : \Gamma_1] \rightarrow F \rightarrow [K_2 : \Gamma_2] \rightarrow G \rightarrow \otimes_R}{[K_1 \otimes K_2 : \Gamma_1, \Gamma_2] \rightarrow F \otimes G \rightarrow} \quad \circ_R \quad \frac{[K_1 : \Gamma_1] \rightarrow F \rightarrow [K_2 : \Gamma_2] \rightarrow H \rightarrow G}{[K_1 \otimes K_2 : \Gamma_1, \Gamma_2] F \rightarrow H \rightarrow G} \quad \rightarrow_L
\]

If we decide to focus on \( c \otimes (c \rightarrow d) \otimes (d \rightarrow e) \vdash e \), the atom \( d \) must be already in the context!

Declarative Reading of lcc processes [OP15]

Focused proofs corresponds, one-to-one, to operational steps in (l)CCP.

\[
(P, c) \rightarrow^* (Q, d) \text{ iff } L[P] \otimes c \vdash L[Q] \otimes d
\]
Modalities in CCP
Epistemic and Spatial behavior in CCP

Assume a set of agents $A=\{i,j,k\ldots\}$,

- $[P]_i$ means $P$ runs in the space-agent $i$.
- $s_i(c)$ means the constraint (information) $c$ holds for agent $i$.

Constraints are of the form $s_i(c)$. Two possible interpretations:

1. **Epistemic:**
   - $s_i(c)$: $i$ knows $c$ (and then, $c$ is true).
   - $s_j(s_i(c))$: $j$ knows that $i$ knows $c$ (and then, $j$ knows $c$).

2. **Spatial**
   - $s_i(c)$: $c$ holds in the space of $i$.
   - $s_j(s_i(c))$: $c$ holds in the space that $j$ conferred to $i$ but $c$ does not necessarily hold in $j$. 
Epistemic CCP

Some properties for $s_i$:

1. $s_i(c) \vdash_{\Delta_e} c$ (believes are facts)
2. $s_i(s_i(c)) = s_i(c)$ (idempotence)

In eccp, knowledge of agents becomes a fact and information propagates to outermost spaces:

\[(\text{ask } \text{coin} \text{ then tell(coffee)} \parallel [\text{tell(coin)}]_i, \text{true}) \rightarrow \]
\[(\text{ask } \text{coin} \text{ then tell(coffee)} \parallel, s_i(\text{coin})) \rightarrow \]
\[(\text{tell(coffee)}, s_i(\text{coin})) \rightarrow \]
\[(\text{skip, } s_i(\text{coin}) \land \text{coffee})\]
Spatial CCP (Information Confinment)

In \textit{sccp}, inconsistency (and information) is confined:

1. \( s_i(0) \not\vdash_{\Delta} s_j(0) \) (false is not propagated outside locations).
2. \( s_i(0) \not\vdash_{\Delta} 0 \) (falsity is not global)

\[
(\text{ask } \textit{coin} \text{ then } \text{tell}(\text{coffee}) \parallel [\text{tell}(\textit{coin})]_i, \text{true}) \rightarrow (\text{ask } \textit{coin} \text{ then } \text{tell}(\text{coffee}), s_i(\textit{coin})) \not\rightarrow
\]

How to give a declarative interpretation of such modalities?
Outline

1. Modalities in CCP
2. SELL interpretation of CCP processes
3. SELL as Constraint System
4. Concluding Remarks
Subexponentials \([\text{DJS93}]\) in Linear Logic

Subexponential Signature

\[ \Sigma = \langle I, \preceq, U \rangle \] where \( I \) is a set of labels, \( U \subseteq I \) set of unbounded subexp and \( \preceq \) is a pre-order among the elements of \( I \).

\[
\frac{\Gamma, F \rightarrow G}{\Gamma, !^a F \rightarrow G} ~ \text{!}^a_L \]
\[
\frac{!^{a_1} F_1, \ldots, !^{a_n} F_n \rightarrow F}{!^{a_1} F_1, \ldots, !^{a_n} F_n \rightarrow !^a F} ~ \text{!}^a_R, \text{ provided } a \preceq a_i
\]

\[
\frac{\Gamma \rightarrow G}{\Gamma, !^b F \rightarrow G} \quad \text{W} \quad \frac{\Gamma, !^b F \rightarrow G}{\Gamma, !^b F \rightarrow G} \quad \text{C}
\]

Assume now two separated rooms \( a \) and \( b \), i.e., \( a \not\preceq b \) and \( b \not\preceq a \).

\[(!^a \text{coin} \rightarrow !^a \text{coffee}) \otimes !^b \text{coin} \not\vdash !^b \text{coffee} \]

- What about a specification like \( \forall l. (!^l \text{coin} \rightarrow !^l \text{coffee}) \)?
- We need a theory for existential/universal quantification on subexponentials.
Quantification on Locations [NOP13]

\[
\begin{align*}
A; L; \Gamma, P[l/x] & \vdash G & \text{\textit{\textsc{R}}}_L \\
A; L; \Gamma, \forall x : a.P & \vdash G & \text{\textit{\textsc{R}}}_R \\
A, l_e : a; L; \Gamma, P[l_e/x] & \vdash G & \text{\textit{\textsc{R}}}_L \\
A; L; \Gamma & \vdash \forall x : a.P & \text{\textit{\textsc{R}}}_R
\end{align*}
\]

- Creating “new” locations: \( \Gamma, \forall l.\left(F\right) \vdash G \)
- Asserting something about all locations: \( \Gamma, \forall l.\left(F\right) \vdash G \)
- Proving that all locations satisfies \( G \): \( \Gamma \vdash \forall l.\left(G\right) \)
- Proving that \( G \) holds in some location: \( \Gamma \vdash \exists l.\left(G\right) \)

Theorem (Cut-elimination) [NOP13]

For any signature \( \Sigma \), the proof system \( \text{SELL}^{\forall} \) admits cut-elimination.
### Epistemic and Spatial Encodings

**The intuition**

<table>
<thead>
<tr>
<th>Connective</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\nabla_s = !^s$</td>
<td>$!^s P$ is <strong>located</strong> at $s$.</td>
</tr>
<tr>
<td>$\nabla_s \equiv !^s ?^s$</td>
<td>$!^s ?^s P$ is <strong>confined</strong> to $s$.</td>
</tr>
<tr>
<td>$\cap l : a P$</td>
<td>$P$ can <strong>move</strong> to locations below (outside) $a$.</td>
</tr>
</tbody>
</table>

### Epistemic Modalities

<table>
<thead>
<tr>
<th>$\triangleleft$</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a.a \sim a$</td>
<td>Modalities are idempotent: $[[P]_a]_a \sim [P]_a$</td>
</tr>
<tr>
<td>$a \triangleleft a.b$</td>
<td>Processes move outside $[[P]_b]_a \rightarrow [P \parallel [P]_b]_a$</td>
</tr>
</tbody>
</table>

### Spatial Modalities

<table>
<thead>
<tr>
<th>$\triangleleft$</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \not\triangleleft b$</td>
<td>$P$ does not communicate with $Q$ in $[P]_a \parallel [Q]_b$</td>
</tr>
<tr>
<td>$a.a \not\triangleright a$</td>
<td>Modalities are <strong>not</strong> necessarily idempotent.</td>
</tr>
<tr>
<td>$a \not\triangleright a.b$</td>
<td>Processes are confined: $[[P]_b]_a \not\triangleright [P \parallel [P]_b]_a$</td>
</tr>
</tbody>
</table>
Adequacy

Take for instance:

\[ P[\text{tell}(c)]_a = !p(a) \cap s : a.(C[\text{c}]_s) \]

We get the following (focused) derivation in SELL\(^\cap\):

\[
\begin{align*}
&[C', D, P] \longrightarrow [G] \\
&[C, D, P], C[\text{c}]_s \longrightarrow [G] \\
&[C, D, P] \quad \cap s : a.C[\text{c}]_s \longrightarrow [G] \\
&[C, D, P + p(a) \cap s : a.C[\text{c}]_s] \longrightarrow [G] \\
&[C, D, P] \quad \cap s : a.C[\text{c}]_s \longrightarrow [G]
\end{align*}
\]

Theorem (Adequacy)

Let \( P \) be an eccp/sccp process, then,

\[ P \downarrow_c \iff P[\text{P}] \longrightarrow C[\text{C}]_\text{nil} \]
Timed Modalities in \textit{SELLF}

The \texttt{tcc} calculus

\[ P, Q, \ldots := \text{tell}(c) \ldots | \circ P | \Box P \]

Theorem (Adequacy)

Let \( P \) be a timed process, \((C_t, \Delta_t)\) be a CS. Then \( P \Downarrow_c \) iff

\[ !^c(\infty)[\Delta_t], \mathcal{P}[P]_1 \rightarrow \biguplus l : 1+!^c(l) ?^c(l) c \otimes T. \]
Outline

1. Modalities in CCP
2. SELL interpretation of CCP processes
3. SELL as Constraint System
4. Concluding Remarks
Subexponential CCP
From SELL\textsuperscript{\textregistered} to CCP

Assume a constraint system where subexponentials are allowed:

\[ F ::= 1 \mid A \mid F \otimes F \mid \exists x. F \mid !^a F \mid !^s?^s F \]

- \( !^a c = (\lfloor c \rfloor)_a \): c holds (is believed) with preference a.
- \( !^s?^{s'} c = [c]^{s'}_s \): c holds in any space in the hierarchy \( s' : s \).

Processes are allowed to create and communicate locations:

\[ P, Q ::= \text{tell}(c) \mid (\text{local } \overline{x}; \overline{l}) Q \mid (\text{abs } \overline{x}; \overline{l}; c) Q \mid P \parallel Q \parallel [P]_s \]

**What do we get?**

A declarative model for concurrency where different modalities can be combined!
Programming Examples
Sharing Information

Assume that $s'' \preceq s' \preceq s$:

1. $[c]_{s'} \vdash \Delta [c]_{s'}$ (information $c$ can be propagated to the inner/lower space $s'$);
2. $[c]_{s''} \vdash \Delta [c]_{s'}$ (information $c$ can be propagated to the intermediate location in the hierarchy);
3. $[c]_s \not\vdash \Delta [c]_{s'}$ (information is confined if sharing is not explicit);

Example (Agent 86’s Coffee Machine)

$(\text{local } l : m/c, l' : m/c) \text{tell}([\text{coin}]_l) \parallel \text{ask } [\text{coin}]_l \text{ then tell}([\text{coffee}]_{l'})$

Example (Nested Locations)

$(\text{local } l : m/c, l' : l) \text{tell}([\text{coin}]_l) \parallel \text{ask } [\text{coin}]_l \text{ then tell}([\text{coffee}]_{l'})$
Programming Examples
Temporal and Spatial Dependencies

Example

\([c_2]_{s_a} \odot [d_{3+}]_{s_a'}\) means that \(c\) holds for agent \(a\) in time-unit 2 while \(d\) holds for \(a'\) in all future time-unit \(t \geq 3\). This is useful for describing sets of biochemical reactions ([CFHO15]).

Mobility

for names: \(\exists x. P \land \forall y. Q \leadsto \exists x. (P \land Q)\)
for locations: \(\uplus l. \downarrow_P \land \bigcap w. \downarrow_Q \leadsto \uplus l. (\downarrow_P \land \downarrow_Q)\)

Example (Service Oriented Computing)

\[
\begin{align*}
\text{request}(a, b) & \overset{\text{def}}{=} (\text{local } x, l : \{a, b\})(\text{tell}([\text{com}(x)]_b) \parallel \text{ask} [\text{com}(x)]_a \text{ then } (\text{tell}([\text{com}(x)]_l) \parallel P)) \\
\text{accept}(a, b) & \overset{\text{def}}{=} (\text{abs } y : b; [\text{com}(y)]_b)(\text{tell}([\text{com}(y)]_a) \parallel (\text{abs } k : b; [\text{com}(y)]_k) Q)
\end{align*}
\]

Carlos Olarte (UFRN)
Preferences and Soft Constraints

Using a **c-semiring** as a subexponential signature, agents can tell/ask preferences:

**Examples of c-semirings** $\langle A, +, \times, \bot_A, \top_A \rangle$

- **Fuzzy**: $S_F = \langle [0, 1], \text{max}, \text{min}, 0, 1 \rangle$ – Preferences
- **Probabilistic**: $S_P = \langle [0, 1], \text{max}, \times, 0, 1 \rangle$
- **Weighted**: $S_w = \langle \mathcal{R}^-, \text{max}, +, -\infty, 0 \rangle$ – Costs

**SELLS System [PON14], Promotion Rule**

$$
\frac{!^{a_1} F_1, \ldots , !^{a_n} F_n \rightarrow G}{!^{a_1} F_1, \ldots , !^{a_n} F_n \rightarrow !^{b} G} \quad b \preceq a_1 \times \ldots \times a_n
$$

1. **Fuzzy**: $(\langle c \rangle_{0.7}) \vdash^\Delta (\langle c \rangle_{0.5})$ (if $c$ is added with a higher preference $a'$, then it can be deduced with a lower preference $a$);
2. **Probabilistic**: $(\langle c \rangle_{0.7} \otimes (\langle d \rangle_{0.3} \vdash^\Delta (\langle c \otimes d \rangle_a (a \leq 0.21))$. 

Carlos Olarte (UFRN)
Outline

1. Modalities in CCP
2. SELL interpretation of CCP processes
3. SELL as Constraint System
4. Concluding Remarks
Concluding Remarks

- We showed that subexponentials can express interesting behaviors in concurrency.
- The resulting system turned out to be a nice proof system for different flavors of CCP:
  - Spatial modalities, where nested locations can be dynamically created and shared.
  - Knowledge
  - Temporal Modalities
  - Soft constraints and preferences
- The logical system guided the design for new (still declarative) constructors for CCP.
- Two concrete applications so far: logic/CCP semantics for:
  - $P$-Systems.
  - Reactive Scores.
Thank you!


Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. The structure of exponentials: Uncovering the dynamics of linear logic proofs.


Vivek Nigam, Carlos Olarte, and Elaine Pimentel.
A general proof system for modalities in concurrent constraint programming.

M. Nielsen, C. Palamidessi, and F.D. Valencia.
Temporal concurrent constraint programming: Denotation, logic and applications.

Carlos Olarte and Elaine Pimentel.
Proving concurrent constraint programming correct, revisited.

Carlos Olarte and Frank D. Valencia.
Universal concurrent constraint programing: Symbolic semantics and applications to security.

Elaine Pimentel, Carlos Olarte, and Vivek Nigam.

Vijay Saraswat, Radha Jagadeesan, and Vineet Gupta.
Foundations of timed concurrent constraint programming.