Deriving Labels and Bisimilarity for Timed Concurrent Constraint Programming

Andres Aristizabal
ERCIM Post-Doctoral Researcher (University of Wroclaw)
Joint work with Camilo Rueda (Universidad Javeriana Cali)
Frank D. Valencia (CNRS École Polytechnique)

30th of January 2015
Concurrent Constraint Programming (ccp) is a well-established formalism for concurrency.

In ccp, processes interact by adding (or telling) and asking information (namely, constraints) in a medium (the global store).

Timed Concurrent Constraint Programming (tcc) extends ccp by adding temporal constructs for modeling time and reactive systems.
Motivation

- **Bisimulation:**
  - Central co-inductive technique to verify *behavioral equivalences* between programs in *concurrency*.
  - A notion far too little considered in timed ccp languages [Tini 2001, Valencia 2003].
  - An adequate notion of bisimilarity for ccp has been introduced recently [Aristizabal et al 2011].
Our Goals

- Define a notion of barbed bisimilarity for tcc.
- Derive labeled semantics for tcc.
- Define a notion of labeled bisimilarity which fully corresponds to the unlabeled one.
Outline

1. Preliminaries
2. Bisimilarity for tcc
3. Labeled Semantics
4. Labeled bisimilarity for tcc
5. Concluding Remarks
Bisimilarity

Definition

Given a labeled transition system \((S, \text{Act}, \rightarrow)\) a bisimulation relation is a binary relation \(\mathcal{R}\) over \(S\) (i.e., \(\mathcal{R} \subseteq S \times S\)) if

\((p, q) \in \mathcal{R}, \ \forall \alpha \in \text{Act} : \)

- \(\forall p' \text{ s.t. } p \xrightarrow{\alpha} p' \exists q' \text{ such that } q \xrightarrow{\alpha} q' \text{ and } (p', q') \in \mathcal{R};\)
- \(\forall q' \text{ s.t. } q \xrightarrow{\alpha} q' \exists p' \text{ such that } p \xrightarrow{\alpha} p' \text{ and } (q', p') \in \mathcal{R}\)
A constraint system \( C \) is a complete algebraic lattice \((Con, Con_0, \sqsubseteq, \sqcup, true, false)\) where \( Con \) is a partially ordered set w.r.t. \( \sqsubseteq \). Where:

- \( Con_0 \) is the subset of finite elements of \( Con \), \( \sqcup \) is the lub operation, and \( true, false \) are the least and greatest elements of \( Con \), respectively.
Motivation
Preliminaries
Bisimilarity for tcc
Labeled Semantics
Labeled bisimilarity for tcc
Concluding Remarks

Concurrent Constraint Programming

If the party is between Thursday and Sunday then it must be in a club, disco or pub!
If the party is in a club or a disco it must start between 8:00pm and 11:00pm!
If the party is at night then it must start between 10:00pm and 11:00pm!
The party must be during the weekend!

Deriving Labels and Bisimilarity for Timed Concurrent Constraint Programming
Motivation
Preliminaries
Bisimilarity for tcc
Labeled Semantics
Labeled bisimilarity for tcc
Concluding Remarks

Concurrent Constraint Programming

Constraint Systems, ccp and tcc
Syntax (tcc)
Operational semantics (tcc)

Any day of the weekend.
If the party is between thursday and sunday then it must be in a club, disco or pub!
If the party is in a club or a disco it must start between 8:00pm and 11:00pm!
If the party is at night then it must start between 10:00pm and 11:00pm!

Deriving Labels and Bisimilarity for Timed Concurrent Constraint Programming
If the party is in a club or a disco it must start between 8:00pm and 11:00pm!

If the party is at night then it must start between 10:00pm and 11:00pm!
Concurrent Constraint Programming

Motivation
Preliminaries
Bisimilarity for tcc
Labeled Semantics
Labeled bisimilarity for tcc
Concluding Remarks

Bisimilarity
Constraint Systems, ccp and tcc
Syntax (tcc)
Operational semantics (tcc)

Deriving Labels and Bisimilarity for Timed Concurrent Constraint Programming
Concurrent Constraint Programming

Deriving Labels and Bisimilarity for Timed Concurrent Constraint Programming
Receives a stimulus (i.e. a constraint) from the environment.
1. Receives a stimulus (i.e. a constraint) from the environment.
2. Computes a ccp process and waits for stability.
Receives a stimulus (i.e. a constraint) from the environment.
Computes a ccp process and waits for stability.
Responds with the resulting store.
1. Receives a stimulus (i.e. a constraint) from the environment.
2. Computes a ccp process and waits for stability.
3. Responds with the resulting store.
4. Executes the residual process in the next time unit.
Syntax

\[ P, Q \ldots ::= \text{skip} \mid \text{tell}(c) \mid \text{when } c \text{ then } P \mid P \parallel Q \mid \exists_x P \mid \text{next } P \mid \text{unless } c \text{ next } P \]

where \( c \in \text{Con}_0, x \in \text{Var} \).
**Operational Semantics**

**Internal reduction (\(\rightarrow\))**

\[\begin{align*}
R1 & \quad \langle \text{tell}(c), d \rangle \rightarrow \langle \text{skip}, d \sqcup c \rangle \\
R2 & \quad \langle \text{when } c \text{ then } P, d \rangle \rightarrow \langle P, d \rangle \\
R3 & \quad \langle P, d \rangle \rightarrow \langle P', d' \rangle \\
R4 & \quad \langle P, e \sqcup \exists x d \rangle \rightarrow \langle P', e' \sqcup \exists x d \rangle \\
R5 & \quad \langle \text{unless } c \text{ next } P, d \rangle \rightarrow \langle \text{skip}, d \rangle
\end{align*}\]

**Observable reduction (\(\Rightarrow\))**

\[\begin{align*}
R6 & \quad \langle P, d \rangle \Rightarrow^* \langle P', d' \rangle \\
& \quad \langle P, d \rangle \xrightarrow{(d, d')} \langle F(P), \text{true} \rangle
\end{align*}\]
Definition

Future Function Let \( F : \text{Proc} \rightarrow \text{Proc} \) be defined by:

\[
F(P) = \begin{cases} 
\text{skip} & \text{if } P = \text{skip} \\
\text{skip} & \text{if } P = \text{when } c \text{ then } Q \\
F(P_1) \parallel F(P_2) & \text{if } P = P_1 \parallel P_2 \\
\exists x F(Q) & \text{if } P = \exists^c_x Q \\
Q & \text{if } P = \text{next } Q \\
Q & \text{if } P = \text{unless } c \text{ next } Q
\end{cases}
\]
(Input-output bisimilarity) An input-output bisimulation is a symmetric relation $\mathcal{R}$ on configurations such that whenever $(\gamma_1, \gamma_2) \in \mathcal{R}$ with $\gamma_1 = \langle P, c \rangle$ and $\gamma_2 = \langle Q, c \rangle$:

1. if $\gamma_1 \xrightarrow{(c,d)} \langle F(P), true \rangle$ then there exists $\langle F(Q), true \rangle$ such that $\gamma_2 \xrightarrow{(c,d)} \langle F(Q), true \rangle$ and $(\langle F(P), true \rangle, \langle F(Q), true \rangle) \in \mathcal{R}$.

We say that $\gamma_1$ and $\gamma_2$ are input-output bisimilar, written $\gamma_1 \sim_{io} \gamma_2$, if there exists an input-output bisimulation $\mathcal{R}$ s.t. $(\gamma_1, \gamma_2) \in \mathcal{R}$. We write $P \sim_{io} Q$ iff $\langle P, true \rangle \sim_{io} \langle Q, true \rangle$. 
Barbs

- **Barbed equivalences** have become the **standard** behavioral equivalences for formalisms equipped with **unlabeled reduction semantics**.
- We have to fix a set of **barbs** i.e., basic observations on the states of processes.
- Barbs for tcc as in ccp are **constraints**.
- A configuration $\gamma = \langle P, d \rangle$ **satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $c \sqsubseteq d$;
- Write $\gamma$ **weakly satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $\gamma \rightarrow^* \gamma'$ and $\gamma' \downarrow_c$. where $\rightarrow^*$ is the transitive and reflexive closure of $\rightarrow$. 

Barbs

- **Barbed equivalences** have become the **standard** behavioral equivalences for formalisms equipped with **unlabeled reduction semantics**.
- We have to fix a set of **barbs** i.e., basic observations on the states of processes.
- Barbs for tcc as in ccp are **constraints**.
- A configuration $\gamma = \langle P, d \rangle$ **satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $c \sqsubseteq d$;
- Write $\gamma$ **weakly satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $\gamma \rightarrow^* \gamma'$ and $\gamma' \downarrow_c$. where $\rightarrow^*$ is the transitive and reflexive closure of $\rightarrow$. 

Barbs

- **Barbed equivalences** have become the **standard** behavioral equivalences for formalisms equipped with **unlabeled reduction semantics**.
- We have to fix a set of **barbs** i.e., basic observations on the states of processes.
- Barbs for tcc as in ccp are **constraints**.
- A configuration $\gamma = \langle P, d \rangle$ **satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $c \sqsubseteq d$;
- Write $\gamma$ **weakly satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $\gamma \rightarrow^* \gamma'$ and $\gamma' \downarrow_c$. where $\rightarrow^*$ is the transitive and reflexive closure of $\rightarrow$. 

Barbs

- **Barbed equivalences** have become the **standard** behavioral equivalences for formalisms equipped with **unlabeled reduction semantics**.
- We have to fix a set of **barbs** i.e., basic observations on the states of processes.
- Barbs for tcc as in ccp are **constraints**.
- A configuration $\gamma = \langle P, d \rangle$ **satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $c \sqsubseteq d$;
- Write $\gamma$ **weakly satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $\gamma \rightarrow^* \gamma'$ and $\gamma' \downarrow_c$. where $\rightarrow^*$ is the transitive and reflexive closure of $\rightarrow$. 

Barbs

- **Barbed equivalences** have become the **standard** behavioral equivalences for formalisms equipped with **unlabeled reduction semantics**.
- We have to fix a set of **barbs** i.e., basic observations on the states of processes.
- Barbs for tcc as in ccp are **constraints**.
- A configuration $\gamma = \langle P, d \rangle$ **satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $c \sqsubseteq d$;
- Write $\gamma$ **weakly satisfies** the barb $c$, written $\gamma \downarrow_c$, iff $\gamma \rightarrow^* \gamma'$ and $\gamma' \downarrow_c$. where $\rightarrow^*$ is the transitive and reflexive closure of $\rightarrow$. 

Barbs
Timed saturated barbed bisimilarity

Definition

(Timed saturated barbed bisimilarity). A timed saturated barbed bisimulation is a symmetric relation $R$ on configurations such that whenever $(\gamma_1, \gamma_2) \in R$ with $\gamma_1 = \langle P, d \rangle$ and $\gamma_2 = \langle Q, e \rangle$:

(i) if $\gamma_1 \Downarrow c$ then $\gamma_2 \Downarrow c$,

(ii) if $\gamma_1 \rightarrow \gamma'_1$ then there exists $\gamma'_2$ such that $\gamma_2 \rightarrow \gamma'_2$ and $(\gamma'_1, \gamma'_2) \in R$,

(iii) if $\gamma_1 \not\rightarrow$ then $\gamma_2 \not\rightarrow$ and $(\langle F(P), true \rangle, \langle F(Q), true \rangle) \in R$,

(iv) for every $a \in Con_0$, $(\langle P, d \sqcup a \rangle, \langle Q, e \sqcup a \rangle) \in R$.

We say that $\gamma_1$ and $\gamma_2$ are timed saturated barbed bisimilar, written $\gamma_1 \sim_{sb} \gamma_2$, if there exists a timed saturated barbed bisimulation $R$ s.t. $(\gamma_1, \gamma_2) \in R$. We write $P \sim_{sb} Q$ iff $\langle P, true \rangle \sim_{sb} \langle Q, true \rangle$. 
Timed weak saturated barbed bisimilarity

Definition

(Timed weak saturated barbed bisimilarity). Timed weak saturated barbed bisimilarity $\approx_{sb}$ is obtained by replacing $\rightarrow$ with $\rightarrow^*$ and $\downarrow_c$ with $\downarrow_c$. 
Labeled Semantics

\[\text{LR1} \quad \langle \text{tell}(c), d \rangle \xrightarrow{\text{true}} \langle \text{skip}, d \cup c \rangle \]

\[\text{LR2} \quad \alpha \in \min \{ a \in \text{Con}_0 \mid c \sqsubseteq d \cup a \} \quad \langle \text{when } c \text{ then } P, d \rangle \xrightarrow{\alpha} \langle P, d \cup \alpha \rangle \]

\[\text{LR3} \quad \langle P, d \rangle \xrightarrow{\alpha} \langle P', d' \rangle \quad \langle P \parallel Q, d \rangle \xrightarrow{\alpha} \langle P' \parallel Q, d' \rangle \]

\[\text{LR4} \quad \langle P[z/x], e[z/x] \cup d \rangle \xrightarrow{\alpha} \langle P', e' \cup d \cup \alpha \rangle \quad \langle \exists_x P, d \rangle \xrightarrow{\alpha} \langle \exists_x^{e'[x/z]} P'[x/z], \exists_x (e'[x/z]) \cup d \cup \alpha \rangle \]

\[x \notin \text{fv}(e'), z \notin \text{fv}(P) \cup \text{fv}(e \cup d \cup \alpha) \]

\[\text{LR5} \quad \alpha \in \min \{ a \in \text{Con}_0 \mid c \sqsubseteq d \cup a \} \quad \langle \text{unless } c \text{ next } P, d \rangle \xrightarrow{\alpha} \langle \text{skip}, d \cup \alpha \rangle \]
Labeled Semantics

Time-unit transition ($\Rightarrow$)

\[ \langle P, d \rangle \xrightarrow{\alpha_i} \ldots \xrightarrow{\alpha_n} \langle P', d' \rangle \xrightarrow{\alpha} \]

\[ \langle P, d \rangle \xrightarrow{\langle d, \xi, d' \rangle} \langle F(P), \text{true} \rangle \]

where $\xi = \langle P, d \rangle \xrightarrow{\alpha_i} \ldots \xrightarrow{\alpha_n} \langle P', d' \rangle$

and $n$ can be 0 and for $i \neq j$ $\alpha_i$ can be equal to $\alpha_j$
**Conjecture (probably proven as in ccp)**

*(Soundness).* If $\langle P, d \rangle \xrightarrow{\alpha} \langle P', d' \rangle$ then $\langle P, d \Box \alpha \rangle \xrightarrow{} \langle P', d' \rangle$

**Conjecture (probably proven as in ccp)**

*(Completeness).* If $\langle P, d \Box a \rangle \xrightarrow{} \langle P', d' \rangle$ then $\exists \alpha, b$ s.t.

$\langle P, d \rangle \xrightarrow{\alpha} \langle P', d'' \rangle$ and $\alpha \Box b = a$, $d'' \Box b = d'$
Timed strong bisimilarity

Definition

*(Timed Strong bisimilarity).* A timed strong bisimulation is a symmetric relation $\mathcal{R}$ on configurations such that whenever $(\gamma_1, \gamma_2) \in \mathcal{R}$ with $\gamma_1 = \langle P, d \rangle$ and $\gamma_2 = \langle Q, e \rangle$:

1. if $\gamma_1 \downarrow_c$ then $\gamma_2 \downarrow_c$,
2. if $\gamma_1 \overset{\alpha}{\rightarrow} \gamma'_1$ then $\exists \gamma'_2$ s.t. $\langle Q, e \sqcup \alpha \rangle \rightarrow \gamma'_2$ and $(\gamma'_1, \gamma'_2) \in \mathcal{R}$,
3. if $\gamma_1 \not\overset{\alpha}{\rightarrow}$ then $\gamma_2 \not\overset{\alpha}{\rightarrow}$ and $(\langle F(P), \text{true} \rangle, \langle F(Q), \text{true} \rangle) \in \mathcal{R}$.

We say that $\gamma_1$ and $\gamma_2$ are timed strongly bisimilar, written $\gamma_1 \sim \gamma_2$, if there exists a timed strong bisimulation $\mathcal{R}$ such that $(\gamma_1, \gamma_2) \in \mathcal{R}$. 
Definition

(Timed weak bisimilarity). A weak bisimulation is a symmetric relation \( R \) on configurations such that whenever \((\gamma_1, \gamma_2) \in R\) with \( \gamma_1 = \langle P, d \rangle \) and \( \gamma_2 = \langle Q, e \rangle \):

(i) if \( \gamma_1 \downarrow_c \) then \( \gamma_2 \downarrow_c \),

(ii) if \( \gamma_1 \xrightarrow{\alpha} \gamma'_1 \) then \( \exists \gamma'_2 \text{ s.t. } \langle Q, e \sqcup \alpha \rangle \rightarrow^* \gamma'_2 \) and \( (\gamma'_1, \gamma'_2) \in R \),

(iii) if \( \gamma_1 \not\rightarrow \) then \( \gamma_2 \not\rightarrow \) and \( (\langle F(P), true \rangle, \langle F(Q), true \rangle) \in R \).

We say that \( \gamma_1 \) and \( \gamma_2 \) are weakly bisimilar, written \( \gamma_1 \approx \gamma_2 \), if there exists a weak bisimulation \( R \) such that \( (\gamma_1, \gamma_2) \in R \).
Correspondence results

**Conjecture**

\[ \sim_{sb} = \sim \text{ and } \approx_{sb} = \approx. \]
Our notions of bisimilarity provides an alternative co-inductive proof method for tcc.

We conjecture that our labeled semantics fully correspond to the unlabeled one.

We hypothesize that our bisimilarity is coherent w.r.t. barbed congruence.
Thank you very much for your attention!
Definition

A timed syntactic bisimulation is a symmetric relation $\mathcal{R}$ on configurations such that whenever $(\gamma_1, \gamma_2) \in \mathcal{R}$ implies that:

(i) if $\gamma_1 \downarrow_c$ then $\gamma_2 \downarrow_c$,

(ii) if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ then $\exists \gamma_2'$ such that $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $(\gamma_1', \gamma_2') \in \mathcal{R}$,

(iii) if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ then $\gamma_2 \xrightarrow{\alpha} \gamma_2'$ and $(\langle F(P), true \rangle, \langle F(Q), true \rangle) \in \mathcal{R}$.

Define $\gamma_1 \sim_S \gamma_2$ iff there exists a timed syntactic bisimulation $\mathcal{R}$ such that $(\gamma_1, \gamma_2) \in \mathcal{R}$.

Write $P \sim_s Q$ iff $\langle P, true \rangle \sim_s \langle Q, true \rangle$. 
Example (over discriminating)

Let \( P = \text{when } x < 10 \text{ then tell}(x < 5) \) and 
\( Q = \text{when } x < 5 \text{ then tell}(5) \) then 
\( \langle P \parallel Q, \text{true} \rangle \not\sim_s \langle P \parallel P, \text{true} \rangle \)
Example (over discriminating)

\[ P \parallel Q \]

\[ P \parallel P \]

- \( x < 10 \)
- \( x < 5 \)
- true

- true
- true
- true
Example (over discriminating)

\[ P \parallel Q \]

\[ P \parallel P \]
Example (over discriminating)
Example (over discriminating)
Example (over discriminating)
Timed strong bisimilarity

Definition

(Timed Strong bisimilarity). A timed strong bisimulation is a symmetric relation $\mathcal{R}$ on configurations such that whenever $(\gamma_1, \gamma_2) \in \mathcal{R}$ with $\gamma_1 = \langle P, d \rangle$ and $\gamma_2 = \langle Q, e \rangle$:

(i) if $\gamma_1 \downarrow_c$ then $\gamma_2 \downarrow_c$,

(ii) if $\gamma_1 \xrightarrow{\alpha} \gamma_1'$ then $\exists \gamma_2'$ s.t. $\langle Q, e \sqcup \alpha \rangle \rightarrow \gamma_2'$ and $(\gamma_1', \gamma_2') \in \mathcal{R}$,

(iii) if $\gamma_1 \not\xrightarrow{\alpha}$ then $\gamma_2 \not\xrightarrow{\alpha}$ and $(\langle F(P), true \rangle, \langle F(Q), true \rangle) \in \mathcal{R}$.

We say that $\gamma_1$ and $\gamma_2$ are timed strongly bisimilar, written $\gamma_1 \sim \gamma_2$, if there exists a timed strong bisimulation $\mathcal{R}$ such that $(\gamma_1, \gamma_2) \in \mathcal{R}$. 