LTSs, revisited

Francesco Zappa Nardelli

francesco.zappa_nardelli@inria.fr

1. INRIA Rocquencourt, MOSCOVA research team.
Plan

Objective:

understand what lies behind the equivalences for process languages.

Plan:

1. A natural contextual equivalence:
   motivations, definition, relationships with bisimilarity in CCS, what an LTS is;

2. from CCS to pi-calculus:
   congruence of bisimulation, full bisimulation.
A historical perspective

**CCS** Milner defined the operational semantics of CCS in term of a *labelled transition system* and associated *bisimilarity*;

...several attempts to handle mobility algebraically led to...

**pi-calculus** Milner, Parrow and Walker introduced the pi-calculus. They defined its semantics along the lines of research on CCS. But...
...lifting CCS techniques was not so smooth

The original paper on pi-calculus defines two LTSs:

**Early LTS**

\[
\overline{x}.P \xrightarrow{x} P \\
\]

\[
x(y).P \xrightarrow{x(v)} \{v/y\} P \\
\]

\[
P \xrightarrow{\overline{x}} P' \quad Q \xrightarrow{x(v)} Q' \\
\]

\[
P || Q \xrightarrow{\tau} P' || Q' \\
\]

**Late LTS**

\[
\overline{x}.P \xrightarrow{x} P \\
\]

\[
x(y).P \xrightarrow{x(y)} P \\
\]

\[
P \xrightarrow{\overline{x}} P' \quad Q \xrightarrow{x(y)} Q' \\
\]

\[
P || Q \xrightarrow{\tau} P' || \{v/y\} Q' \\
\]

These LTSs define the same \(\tau\)-transitions. But the bisimilarity built on top of them observe *all* the labels: do the resulting bisimilarities coincide? No...

Question: which is the right one?
A step backward: defining a language

Recipe:

1. define the *syntax* of the language (that is, specify what a program is);

2. define its *reduction semantics* (that is, specify how programs are executed);

3. define when *two terms are equivalent* (that is, hum...?!).

Share and enjoy the new language...
Suppose that $P$ and $Q$ are equivalent (in symbols: $P \simeq Q$).

Which properties do we expect?

**Preservation under contexts** For all contexts $C[-]$, we have $C[P] \simeq C[Q]$;

**Same observations** If $P \downarrow n$ then $Q \downarrow n$, where $P \downarrow n$ means that we can observe $n$ at $P$ (or $P$ can do $n$);

**Preservation of reductions** $P$ and $Q$ must mimic their reduction steps (that is, they realise the same nondeterministic choices).
What if we apply this recipe to (a subset of) CCS?

Syntax:

\[
P ::= 0 \mid a.P \mid \overline{a}.P \mid P \parallel P \mid (\nu a)P
\]

Reduction semantics:

\[
a.P \parallel \overline{a}.Q \to P \parallel Q
\]

\[
P \equiv P' \implies Q' \equiv Q
\]

\[
P \to Q
\]

where $\equiv$ is defined as:

\[
P \parallel Q \equiv Q \parallel P
\]

\[
(P \parallel Q) \parallel R \equiv P \parallel (Q \parallel R)
\]

\[
(\nu a)P \parallel Q \equiv (\nu a)(P \parallel Q) \text{ if } a \not\in \text{fn}(Q)
\]
The recipe, formally

A relation $\mathcal{R}$ between processes is

*preserved by contexts:* if $P \mathcal{R} Q$ implies $C[P] \mathcal{R} C[Q]$ for all contexts $C[-]$.

*barb preserving:* if $P \mathcal{R} Q$ and $P \Downarrow_n$ imply $Q \Downarrow_n$, where $P \Downarrow_n$ holds if there exists $P'$ such that $P \rightarrow^* P'$ and $P' \Downarrow_n$, while

$$P \Downarrow_n \quad \text{holds if} \quad P \equiv (\nu \tilde{a})(n.P'||P'') \text{ with } n \notin \{\tilde{a}\}.$$

*reduction closed:* if $P \mathcal{R} Q$ and $P \rightarrow P'$, imply that there is a $Q'$ such that $Q \rightarrow^* Q'$ and $P' \mathcal{R} Q'$ ($\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$).
The recipe, formally (ctd.)

**Definition**  *Reduction barbed congruence*, denoted $\simeq$, is the largest symmetric relation over processes which is reduction closed, barb preserving, and preserved by contexts.

*Claim:* reduction barbed congruence is a *natural, intuitive, contextual equivalence.*
Pro and contra of reduction barbed congruence

Reduction barbed congruence is simple (even a programmer will understand it), “natural”, and can be defined over any (process) language: just pick up a reasonable observation $P \Downarrow n$ and you are done. Great!

Great? Hum, the definition of reduction barbed congruence tells you nothing about the language. In particular you have no hints about which terms are equivalent.

And proving that $P \simeq Q$ holds is difficult, because of the universal quantification over all contexts.
The role of bisimilarity

Observation: the definition of bisimilarity does not involve a universal quantification over all contexts!

Question: is there any relationship between (weak) bisimilarity and reduction barbed congruence?

Theorem:

1. \( P \approx Q \) implies \( P \simeq Q \) (soundness of bisimilarity);

2. \( P \simeq Q \) implies \( P \approx Q \) (completeness of bisimilarity).

Point 2. does not hold in general (it does for the subset of CCS we consider). Point 1. ought to hold (otherwise your LTS/bisimilarity is very odd!).
Background: LTS and weak bisimilarity for CCS

\[
\begin{align*}
    a.P \xrightarrow{a} P & \quad \bar{a}.P \xrightarrow{\bar{a}} P \\
    P \xrightarrow{\ell} P' & \quad P \parallel Q \xrightarrow{\tau} P' \parallel Q' \\
    (\nu a)P \xrightarrow{\ell} (\nu a)P' & \quad \text{symmetric rules omitted.}
\end{align*}
\]

Let \( \xrightarrow{\hat{\ell}} \) be \( \tau^* \xrightarrow{\ell} \tau^* \) if \( \ell \neq \tau \), and \( \tau^* \) otherwise.

**Definition:** Weak bisimilarity, denoted \( \approx \), is the largest symmetric relation such that whenever \( P \approx Q \) and \( P \xrightarrow{\ell} P' \) there exists \( Q' \) such that \( Q \xrightarrow{\hat{\ell}} Q' \) and \( P' \approx Q' \).
Soundness of weak bisimilarity: \( P \sim Q \) implies \( P \simeq Q \).

**Proof** We show that \( \sim \) is contextual, barb preserving, and reduction closed.

Contextuality of \( \sim \) can be shown by induction on the structure of the contexts, and by case analysis of the possible interactions between the processes and the contexts. (Omitted).

Suppose that \( P \sim Q \) and \( P \downarrow a \). Then \( P \equiv (\nu \tilde{n})(a.P_1 \parallel P_2) \), with \( a \not\in \tilde{n} \). We derive \( P \xrightarrow{a} (\nu \tilde{n})(P_1 \parallel P_2) \). Since \( P \sim Q \), there exists \( Q' \) such that \( Q \xrightarrow{a} Q' \), that is \( Q \xrightarrow{\tau \ast} Q'' \xrightarrow{a} \ldots \). But \( Q'' \) must be of the form \((\nu \tilde{m})(a.Q_1 \parallel Q_2)\) with \( a \not\in \text{fn}(Q) \). This implies that \( Q'' \downarrow a \), and in turn \( Q \downarrow a \), as required.

Suppose that \( P \sim Q \) and \( P \xrightarrow{\tau} P' \). We have that \( P \xrightarrow{\tau} P'' \equiv P' \). Since \( P \sim Q \), there exists \( Q' \) such that \( Q \xrightarrow{\tau \ast} Q' \) and \( P' \equiv P'' \sim Q' \). Since \( Q \xrightarrow{\tau \ast} Q' \) it holds that \( Q \xrightarrow{\ast} Q' \). Since \( P' \equiv P'' \) implies \( P' \sim P'' \), by transitivity of \( \sim \) we conclude \( P' \sim Q' \), as required. \( \square \)
Completeness of weak bisimilarity: $P \simeq Q$ implies $P \approx Q$.

Proof. We show that $\simeq$ is a bisimulation.

Suppose that $P \simeq Q$ and $P \xrightarrow{a} P'$ (the case $P \simeq Q$ and $P \xrightarrow{\tau} P'$ is easy). Let

$$
C_a[-] = - \parallel \overline{a}.d \\
C_{\overline{a}}[-] = - \parallel a.d
$$

where the names $z, o, f, d$ are fresh for $P$ and $Q$.

**Lemma 1.** $C_a[P] \xrightarrow{*} P' \parallel d$ if and only if $P \xrightarrow{a} P'$. Similarly for $C_{\overline{a}}[-]$.

Since $\simeq$ is contextual, we have $C_a[P] \parallel Flip \simeq C_a[Q] \parallel Flip$. By Lemma 1. we have $C_a[P] \parallel Flip \xrightarrow{*} P_1 \equiv P' \parallel o \parallel (\nu z)(z. -1 \parallel z. -2 \parallel \overline{z})$.

**Lemma 2.** If $P \simeq Q$ and $P \xrightarrow{*} P'$ then there exists $Q'$ such that $Q \xrightarrow{*} Q'$ and $P' \simeq Q'$.
By Lemma 2, there exists $Q_1$ such that $C_a[Q] \parallel Flip \rightarrow^* Q_1$ and $P_1 \simeq Q_1$. Now, $P_1 \downarrow o$ and $P_1 \not\Downarrow f$. Since $\simeq$ is barb preserving, we have $Q_1 \downarrow o$ and $Q_1 \not\Downarrow f$. The absence of the barb $f$ implies that the $\oplus$ operator reduced, and in turn that the $d$ action has been consumed: this can only occur if $Q$ realised the $a$ action. Thus we can conclude $Q_1 \equiv Q' \parallel o \parallel (\nu z) z.f$, and by Lemma 1, we also have $Q \xrightarrow{a} Q'$.

It remains to show that $P' \simeq Q'$.

**Lemma 3.** $(\nu z) z.P \simeq 0$.

Since $P_1 \simeq Q_1$ and $\simeq$ is contextual, we have $(\nu o) P_1 \simeq (\nu o) Q_1$. By Lemma 3., we have

$$P' \equiv P' \parallel (\nu o) o \parallel (\nu z) z.f \equiv (\nu o) P_1 \simeq (\nu o) Q_1 \equiv Q' \parallel (\nu o) o \parallel (\nu z) z.f \simeq Q'.$$

The equivalence $P' \simeq Q'$ follows because $\equiv \subseteq \simeq$ and $\simeq$ is transitive. 

**Exercise:** explain the role of the $Flip$ process.
Both weak labels and weak bisimilarity can be built as done in CCS.

Let \( \hat{\ell} \Rightarrow \tau \Rightarrow^* \ell \Rightarrow^* \) if \( \ell \neq \tau \), and \( \tau \Rightarrow^* \) otherwise.

**Definition:** Weak bisimilarity, denoted \( \approx \), is the largest symmetric relation such that whenever \( P \approx Q \) and \( P \xrightarrow{\ell} P' \) there exists \( Q' \) such that \( Q \xrightarrow{\hat{\ell}} Q' \) and \( P' \approx Q' \).
First, define barbs:

\[ P \downarrow x \text{ iff } P \equiv (\nu \tilde{n})(x(y).P' \parallel P'') \text{ with } x \notin \tilde{n} . \]

Let reduction barbed congruence \( \simeq \) be the largest symmetric relation over pi-calculus processes that is preserved by all contexts, barb preserving, and reduction closed.

**Exercise:** prove that defining \( P \downarrow x \) as \( P \equiv (\nu \tilde{n})(\overline{x}y.P \parallel P'') \) with \( x \notin \tilde{n} \) yields the same equivalence.
Reduction barbed congruence and pi-calculus, ctd.

Exercise: Consider the terms (in a pi-calculus with sums):

\[ P = \overline{xv} \parallel y(z) \]
\[ Q = \overline{xv}.y(z) + y(z).\overline{xv} \]

1. Prove that \( P \approx Q \).

2. Does \( P \simeq Q \)?\(^{12}\)

\(^{1}\)Hint: define a context that equates the names \( x \) and \( y \).
\(^{2}\)Hint: use input prefix.
In pi-calculus, bisimilarity (both strong and weak) is not preserved by input prefixes, that is contexts of the form $C[-] = x(y).-$. 

Question: how to recover the soundness of the bisimilarity with respect to the reduction barbed congruence? Two solutions:

1. close the reduction barbed congruence under all non input prefix contexts; 

2. close the bisimilarity under substitution: let $P \approx^c Q$ ($P$ is fully bisimilar with $Q$) if $P\sigma \approx Q\sigma$ for all substitutions $\sigma$.

Exercise: Show that $P \not\approx^c Q$, where $P$ and $Q$ are defined in the previous slide.
Conclusion: LTSs revisited

Reduction barbed congruence involves a universal quantification over all contexts. Weak bisimilarity does not, yet bisimilarity is *a sound proof technique* for reduction barbed congruence. How is this possible?

An LTS captures all the interactions that a term can have with an arbitrary context. In particular, each label correspond to a minimal context.

For instance, in CCS, $P \xrightarrow{a} P'$ denotes the fact that $P$ can interact with the context $C[\_] = \texttt{\textcolor{red}{-}} || \texttt{\textcolor{red}{a}}$, yielding $P'$.

More interestingly, in pi-calculus (early LTS), $P \xrightarrow{x(v)} P'$ denotes that $P$ can interact with the context $C[\_] = \texttt{\textcolor{red}{-}} || \texttt{\textcolor{red}{xv}}$, yielding $P'$.

And $\tau$ transitions characterises all the interactions with an *empty context*. 