Transitions and Equivalences

Overview
- LTS of concurrent processes
- Reactions vs. transitions
- Strong bisimulation up-to
- Algebraic properties

References
- Robin Milner, “Communication and Concurrency”
- Robin Milner, “Communicating and Mobil Systems”
Labeled Transitions

- Process transitions $P \xrightarrow{\alpha} P'$ extend the reactions $P \rightarrow P'$.

- The case $\alpha = \tau$ corresponds to a reaction, while the case $\alpha = a$ or $\alpha = \bar{a}$ corresponds to the capability of a process $P$ to participate in a reaction provided that another process $Q$, running concurrently, can perform the complementary transition.
LTS of Concurrent Processes

The labeled transition system \((P, T)\) of concurrent processes over the action set \(\Sigma = L \cup \{\tau\}\) has the process expressions \(P\) as its states, and its transitions \(T\) are exactly those, which can be inferred from the rules in the table below:

\[
\begin{align*}
\text{SUM}_T : & \quad M + \alpha.P + N \overset{\alpha}{\rightarrow} P \\
L-\text{PAR}_T : & \quad \frac{P \overset{\alpha}{\rightarrow} P'}{P \mid Q \overset{\alpha}{\rightarrow} P' \mid Q} \\
R-\text{PAR}_T : & \quad \frac{P \overset{\alpha}{\rightarrow} P'}{Q \mid P \overset{\alpha}{\rightarrow} Q \mid P'} \\
\text{RES}_T : & \quad \frac{\text{new } a.P \overset{\alpha}{\rightarrow} \text{new } a.P'}{if \ \alpha \notin \{a, \bar{a}\}} \\
\text{IDENT}_T : & \quad \frac{\{\tilde{b}/\bar{a}\}P_A \overset{\alpha}{\rightarrow} P'}{A\langle\tilde{b}\rangle \overset{\alpha}{\rightarrow} P'} \quad \text{if } A(\bar{a}) = P_A
\end{align*}
\]

\[
\begin{align*}
\text{REACT}_T : & \quad \frac{P \overset{a}{\rightarrow} P'}{Q \overset{\bar{a}}{\rightarrow} Q'} \\
& \quad \frac{P \mid Q \overset{\tau}{\rightarrow} P' \mid Q'}
\end{align*}
\]
Inference Option 1

Consider the processes A and B:

\[ A(a,b) = a.A'(a,b) \quad B(b,c) = b.B'(b,c) \]
\[ A'(a,b) = b.A(a,b) \quad B'(b,c) = \overline{c}.B(b,c) \]

If \( B \mid A' \), then we can infer:

\[
\begin{align*}
\text{SUM}_T & \quad \text{b}.A(a,b) \xrightarrow{b} A(a,b) \\
\text{IDENT}_T & \quad A'(a,b) \xrightarrow{b} A(a,b) \\
\text{L-PAR}_T & \quad A'(a,b) \mid B(b,c) \xrightarrow{b} A(a,b) \mid B(b,c)
\end{align*}
\]

\[ A'(a,b) = \overline{b}.A(a,b) \]
Inference Option 2

Consider the processes A and B:

\[
A(a,b) = a.A'(a,b) \quad B(b,c) = b.B'(b,c) \\
A'(a,b) = b.A(a,b) \quad B'(b,c) = \bar{c}.B(b,c)
\]

If B | A', then we can infer:

\[
\begin{align*}
\text{SUM}_T & \quad b.B'(b,c) \xrightarrow{b} B'(b,c) \\
\text{IDENT}_T & \quad B(b,c) \xrightarrow{b} B'(b,c) \\
\text{R-PAR}_T & \quad A'(a,b) | B(b,c) \xrightarrow{b} A'(a,b) | B'(b,c)
\end{align*}
\]
Consider the processes A and B:

\[ A(a,b) = a.A'(a,b) \quad B(b,c) = b.B'(b,c) \]
\[ A'(a,b) = \overline{b}.A(a,b) \quad B'(b,c) = \overline{c}.B(b,c) \]

If \( B \mid A' \), then we can infer:

\[
\text{REACT}_T \frac{B\langle b,c \rangle \xrightarrow{b} B'\langle b,c \rangle}{B\langle b,c \rangle \mid A'\langle a,b \rangle \xrightarrow{\tau} B'\langle b,c \rangle \mid A\langle a,b \rangle}
\]
We can infer:

\[
\begin{align*}
    A(a,b) &= a.A'(a,b) & B(b,c) &= b.B'(b,c) \\
    A'(a,b) &= \overline{b}.A(a,b) & B'(b,c) &= \overline{c}.B(b,c)
\end{align*}
\]
Structural Congruence vs. Transition

If $P \xrightarrow{\alpha} P'$ and $P \equiv Q$, then there exists $Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \equiv Q'$.

The proof is by induction on the depth of the inference of $P \xrightarrow{\alpha} P'$. That is, the full proof must treat all possible cases for the final step of the inference of $P \xrightarrow{\alpha} P'$. As a simplification it is enough to show the result in the special case that the congruence $P \equiv Q$ is due to a single application of a structural congruence rule. The general case follows just by iterating the special case.
What is the relationship between the reaction relation $\rightarrow$ and the labeled transition relation $\xrightarrow{\alpha}$?

The transition rules mimic the reaction rules (and do more). In fact, the REACT rule can be mimicked by the transition rules $\text{SUM}_T$ and $\text{REACT}_T$. 
Tau Transition

\[ P \quad \Rightarrow \quad P' \quad \Rightarrow \quad P \xrightarrow{\tau} \equiv P' \]

Note, \( P \xrightarrow{\tau} \equiv P' \) is an instance of relational composition; it means that, for some \( P'' \), \( P \xrightarrow{\tau} P'' \) and \( P'' \equiv P' \).

The proof is by induction of the inference of \( P \xrightarrow{} P' \).
Unrestricted Transition

Consider a transition $P \xrightarrow{\alpha} P'$ with $\alpha \neq \tau$.

Intuitively, this transition has to arise from some summation $\alpha.Q + M$ inside $P$, with $\alpha$ unrestricted.

Let $P \xrightarrow{\alpha} P'$. Then $P$ and $P'$ can be expressed, up to structural congruence, in the form

$$P \equiv \text{new } \tilde{z}((\alpha.Q + M) | R)$$
$$P' \equiv \text{new } \tilde{z}(Q | R)$$

where $a$ is not restricted by $\text{new } \tilde{z}$.

The proof is by induction on the structure of inference of $P \xrightarrow{\alpha} P'$. 
Reaction and Tau Transition

\[ \text{P} \xrightarrow{\tau} \equiv \text{P}' \iff \text{P} \rightarrow \text{P}' \]
Properties of Transitions

- Given $P$, there are only finitely many transitions $P \xrightarrow{\alpha} P'$. 

- If $P \xrightarrow{\alpha} P'$ then $\text{fn}(P', \alpha) \subseteq \text{fn}(P)$. 

- If $P \xrightarrow{\alpha} P'$ and $\sigma$ is any substitution then $\sigma P \xrightarrow{\sigma \alpha} \sigma P'$. 
Structural Congruence vs. Bisimulation

- Structural congruence is a strong bisimulation over concurrent processes.

- If \( P \equiv Q \) then \( P \sim Q \).

\[ S = \{ (P,Q) \mid P \equiv Q \} \]

Both statements follow from the proposition:
- If \( P \xrightarrow{\alpha} P' \) and \( P \equiv Q \), then there exists \( Q' \) such that \( Q \xrightarrow{\alpha} Q' \) and \( P' \equiv Q' \).
A n-ary semaphore $S^{(n)}(p,v)$ is a process used to ensure that no more than $n$ instances of some activity run concurrently. An activity is started by acquiring a lock, denoted by the action $p$, and terminated by releasing the lock, denoted by the action $v$. 

\[ S^{(n)} \]
Unary and Binary Semaphores

We can define a unary and binary semaphore as

\[ S^{(1)} = p \cdot S_1^{(1)} \]
\[ S_1^{(1)} = v \cdot S^{(1)} \]
\[ S^{(2)} = p \cdot S_1^{(2)} \]
\[ S_1^{(2)} = v \cdot S^{(2)} + p \cdot S_2^{(2)} \]
\[ S_2^{(2)} = v \cdot S_1^{(2)} \]

where the subscript \( k \) represents how many instances of the activity are running concurrently.
Unary vs. Binary Semaphores

Our expectation of the semantics of the defined semaphores is that a binary semaphore should behave like two unary semaphores running concurrently, that is, $S^{(1)} | S^{(1)}$.

Intuitively, each unary semaphore represents a single unit of some resource and an n-ary semaphore is simply a combination of n units of some resource. For example, in the case of a binary semaphore, we have $n = 2$ and we require:

$$S^{(1)} | S^{(1)} \sim S^{(2)}$$
Bisimulation Check

$S^{(1)} \ | \ S^{(1)} \sim S^{(2)}$

Proof:

We have to verify that the relation $R$, as defined below, is a strong bisimulation.

$$R = \{(S^{(1)}|S^{(1)},S^{(2)}), (S^{(1)}|S^{(1)},S_{1}^{(2)}), (S^{(1)}|S_{1}^{(1)},S_{1}^{(2)}), (S_{1}^{(1)}|S_{1}^{(1)},S_{2}^{(2)})\}$$
Strong Simulation up-to

- The relation $R$ contains two pairs

  $$(S_1(1)|S(1),S_1(2)), (S(1)|S_1(1),S_1(2))$$

  which involve states that are the same up to structural congruence
  $(S_1(1)|S(1) \equiv S(1)|S_1(1))$.

- A binary relation $S$ over $P$ is a strong simulation up-to $\equiv$ if, whenever $PSQ$,

  if $P \xrightarrow{\alpha} P'$ then there exists $Q'$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \equiv S \equiv Q'$.

  S is a strong bisimulation up-to $\equiv$ if its converse has also this property.
$\equiv S \equiv$ is a composite relation. $P \equiv S \equiv Q$ means that for some $P'$ and $Q'$ we have $P \equiv P'$, $P' \overrightarrow{S} Q'$, and $Q \equiv Q'$.

In order for $S$ to be a strong simulation up-to $\equiv$ we must be able to complete the following diagram, given the top row and the left transition:
Consider two processes $P$ and $Q$. To show that $P \sim Q$, it is enough to establish that the pair $(P, Q)$ belongs to some strong bisimulation up-to $\equiv$.

If $S$ is a strong bisimulation up-to $\equiv$ and $PSQ$ then $P \sim Q$.

Proof:
Let $P \equiv S \equiv Q$ and $P \xrightarrow{\alpha} P'$. We have to find a $Q'$ that completes the following diagram:

\[
\begin{array}{c}
 P \equiv S \equiv Q \\
 \downarrow \alpha \\
 P' \\
\end{array} \\
\Rightarrow \quad \\
\begin{array}{c}
 P \equiv S \equiv Q \\
 \downarrow \alpha \\
 P' \equiv S \equiv Q' \\
\end{array}
\]
Proof

For some \( P_1 \) and \( Q_1 \) we have \( P \equiv P_1, P_1 S Q_1 \), and \( Q_1 \equiv Q \). Now, since \( \equiv \) is a strong bisimulation over concurrent processes, \( P \equiv Q \Rightarrow P \sim Q \), and knowing that \( S \) is a strong bisimulation up-to \( \equiv \), we can complete the following diagrams:
Semaphores Revisited

\[ S^{(1)} \mid S^{(1)} \sim S^{(2)} \]

\[ R' = \{(S^{(1)}|S^{(1)},S^{(2)}), (S_{1}^{(1)}|S^{(1)},S_{1}^{(2)}),(S_{1}^{(1)}|S_{1}^{(1)},S_{2}^{(2)})\} \]

\( R' \) is a bisimulation up-to \( \equiv \). By \( P \equiv S \equiv Q \Rightarrow P \sim Q \) we conclude \( S^{(1)} \mid S^{(1)} \sim S^{(2)} \).

Using the up-to technique we can construct simpler (or smaller) relations that are bisimulations.
Strong Equivalence vs. Strong Congruence

Consider the following example

\[ a.0 \mid b.0 \sim a.b.0 + b.a.0 \]

Both processes exhibit the same interactive behavior.

In fact, when one interact with a process as a black box one cannot tell its structure; a parallel composition is behaviorally indistinguishable from a sum.
Parallel Composition vs. Sum

For all processes $P \in \mathcal{P}$, $P \sim \Sigma \{ \beta.Q \mid P \xrightarrow{\beta} Q \}$.

Proof:

Let $S = \{ \beta.Q \mid P \xrightarrow{\beta} Q \}$, so that the right-hand side is $\Sigma S$. We have to show that the transitions of $P$ and $\Sigma S$ are actually identical.

Suppose $P \xrightarrow{\alpha} P'$, then by definition $\alpha.P' \in S$, and hence we have $\Sigma S \xrightarrow{\alpha} P'$ by SUMT.

In the other direction suppose $\Sigma S \xrightarrow{\alpha} P'$, then it must have been inferred by SUMT, and hence $P \xrightarrow{\alpha} P'$ by definition of $S$. 
Transitions of Compositions

A transition of a multiple composition occurs either due to one of its components, or due to a reaction between two sub-components.

For all $n \geq 0$ and processes $P_1, \ldots, P_n$:

$$P_1 | \ldots | P_n \sim \sum\{\alpha.(P_1 | \ldots | P_i' | \ldots | P_n) \mid 1 \leq i \leq n, P_i \xrightarrow{\alpha} P_i'\}$$
$$+ \sum\{\tau.(P_1 | \ldots | P_i' | \ldots | P_j' | \ldots | P_n) \mid 1 \leq i \leq j \leq n, P_i \xrightarrow{\alpha} P_i', P_j \xrightarrow{\bar{a}} P_j'\}$$

Proof: By induction on $n$. 
Transition of Standard Forms

Every process can be expressed in standard form.

\[
\text{new } \tilde{a} (M_1 | ... | M_n)
\]

For all \( n \geq 0 \) and processes \( P_1, ..., P_n \):

\[
\text{new } \tilde{a} (P_1 | ... | P_n)
\]

\[
\sum \{ \text{\( \alpha \).new } \tilde{a} (P_1 | ... | P'_i | ... | P_j) | 1 \leq i \leq n, P_i \xrightarrow{\alpha} P_i' \text{ and } \alpha, \bar{\alpha} \notin \tilde{a} \} + \sum \{ \text{\( \tau \).new } \tilde{a} (P_1 | ... | P'_i | ... | P'_j | ... | P_i) | 1 \leq i \leq j \leq n, P_i \xrightarrow{a} P_i' \text{ and } P_j \xrightarrow{\bar{a}} P_j' \}\]

Proof:

The left-hand side takes the form \text{new } a_1 ... \text{new } a_k(P_1 | ... | P_n), and the result is shown by induction on \( k \).
Sequential Composition

The parallel composition $P \parallel Q$ allows concurrent activity of $P$ and $Q$. However, sometimes one wishes to define sequential composition $P;Q$ to mean ‘when $P$ finishes, $Q$ starts’.

Sequential composition can be modeled in our process language, if we adopt the convention that each process performs a special action $\text{done}$ as its last action before evolving into an inactive agent. Using this approach, sequential composition can be defined as follows:

$$P;Q = \textbf{new} \text{start}(\{\text{start/done}\}P \parallel \text{start}.Q)$$

where the name $\text{start}$ does not occur free in $P$ or $Q$. 
Congruence

- Congruence means that we can ‘substitute equals for equals’.

- When a strong equivalence is a congruence, then if $P \sim Q$ then, in any system we can build with our process constructions, we can replace $P$ by $Q$ without altering the behavior of the system.

A strong equivalence is a process congruence if $P \sim Q$ implies

- $\alpha.P + M \sim \alpha.Q + N$
- $\text{new } a P \sim \text{new } a Q$
- $P | R \sim Q | R$
- $R | P \sim R | Q$

Proof:

For each case we have to show that a corresponding relation $S$ is a strong bisimulation (e.g. $S = \{(P|R,Q|R) \mid P \sim Q \}$).