Concurrency Theory

Lecture 2: Linear Time vs. Branching Time

Dr. Stephan Mennicke

Institute for Theoretical Computer Science
Knowledge-Based Systems Group

April 8, 2025

TECHNISCHE UNIVERSITÄT DRESDEN

International Center for Computational Logic

Process (Equivalence) Relations

Definition 11 Any binary relation $\mathcal{R} \subseteq \mathsf{Pr} \times \mathsf{Pr}$ is called a *process relation*. $\mathcal{R}$ is a *process equivalence* if it is a process relation and an equivalence.

We have seen two instances of process equivalences.

Theorem 12 $\leftrightarrow$ and $\equiv_{\mathsf{tr}}$ are process equivalences.

Proof: in a few slides … ∎

Throughout the course, we will explore many more process equivalences, each time with a different set of requirements.

Isomorphic equivalence ($\leftrightarrow$) and trace equivalence ($\equiv_{\mathsf{tr}}$) form meaninful boundaries.

Trivial boundaries: $\mathcal{U} = \mathsf{Pr} \times \mathsf{Pr}$ (the *universal equivalence*) and $\emptyset$ (the *non-equivalence*).

A Proof of Theorem 12

Theorem 12 $\leftrightarrow$ and $\equiv_{\mathsf{tr}}$ are process equivalences.

Proof: For all processes $p, q, r \in \mathsf{Pr}$,

1. $p \leftrightarrow p$ by $\mathrm{id} : \mathsf{Pr} \to \mathsf{Pr}$ ($\mathrm{id}(q) = q$ for all $q \in \mathsf{Pr}$) being an isomorphism.
2. $p \leftrightarrow q$ implies $q \leftrightarrow p$ since the inverse f^{-1} of an isomorphism f is an isomorphism (cf. Lemma 7).
3. $p \leftrightarrow q$ and $q \leftrightarrow r$ implies $p \leftrightarrow r$ since isomorphisms f and g compose to an isomorphism $g \circ f$ (if unclear, let's make it another exercise 😀).

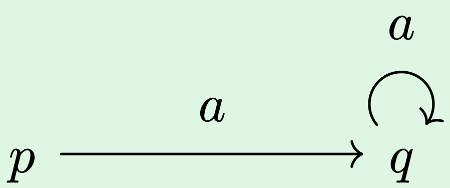
For all processes $p, q, r \in \mathsf{Pr}$,

1. $p \equiv_{\mathsf{tr}} p$ iff $\mathsf{traces}(p) = \mathsf{traces}(p)$ by reflexivity of $=$.
2. $p \equiv_{\mathsf{tr}} q$ iff $\mathsf{traces}(p) = \mathsf{traces}(q)$ iff $\mathsf{traces}(q) = \mathsf{traces}(p)$ iff $q \equiv_{\mathsf{tr}} p$ by symmetry of $=$.
3. $p \equiv_{\mathsf{tr}} q$ and $q \equiv_{\mathsf{tr}} r$ iff □□ iff $p \equiv_{\mathsf{tr}} r$ by transitivity of $=$.

∎

Reminder: $\leftrightarrow$ and $\equiv_{\text{tr}}$

Example. *Reconsider processes p and q and find that $p \equiv_{\text{tr}} q$*



We have $p \not\leftrightarrow q$ but $p \equiv_{\text{tr}} q$.

- this means, $\leftrightarrow \neq \equiv_{\text{tr}}$
- but does $\equiv_{\text{tr}} \subseteq \leftrightarrow$? ✘
- or $\leftrightarrow \subseteq \equiv_{\text{tr}}$? ✓

Process equivalence $\mathcal{E}_1$ ……… process equivalence $\mathcal{E}_2$

- *is finer (than)* if $\mathcal{E}_1 \subseteq \mathcal{E}_2$ *strictly* if if $\mathcal{E}_1 \subsetneq \mathcal{E}_2$
- *is coarser (than)* if $\mathcal{E}_1 \supseteq \mathcal{E}_1$ *strictly* if if $\mathcal{E}_1 \supsetneq \mathcal{E}_2$
- *is incomparable with* if **neither** finer **nor** coarser

Towards a Spectrum of Process Equivalences

Theorem 13

$$\emptyset \overset{(1)}{\subsetneq} \leftrightarrow \overset{(2)}{\subsetneq} \equiv_{\mathsf{tr}} \overset{(3)}{\subsetneq} \mathcal{U} = \mathsf{Pr} \times \mathsf{Pr}$$

Towards a Spectrum of Process Equivalences

Theorem 13

$$\emptyset \overset{(1)}{\subsetneq} \leftrightarrow \overset{(2)}{\subsetneq} \equiv_{\mathsf{tr}} \overset{(3)}{\subsetneq} \mathcal{U} = \mathsf{Pr} \times \mathsf{Pr}$$

Proof: Parts (1) and (3) are clear. Proper inclusions stem from the examples we have seen.

Regarding (2), let $p, q \in \mathsf{Pr}$ such that $p \leftrightarrow q$. Then there is an isomorphism f between the graphs $G(p)$ and $G(q)$, meaning

1. $f(p) = q$ (since p and q are the roots of their respective process graphs) and
2. $p_1 \xrightarrow{a} p_2$ ($p_1 \in \text{Reach}(p)$) if and only if $f(p_1) \xrightarrow{a} f(p_2)$ ($f(p_1) \in \text{Reach}(q)$)

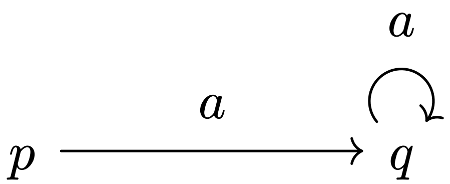
... to be continued ∎

Towards a Spectrum of Process Equivalences

Proof: For every trace $\sigma = a_1 a_2 \ldots a_n \in \mathsf{Act}^\star$,

$$
\begin{aligned}
\sigma \in \mathsf{traces}(p) &\text{ iff } \exists p_1, \ldots, p_n \in \mathsf{Pr}\ .p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} p_n && \text{(by definition)} \\
&\text{ iff } \exists p_1, \ldots, p_n \in \mathsf{Pr}\ .f(p) \xrightarrow{a_1} f(p_1) \xrightarrow{a_2} \cdots \xrightarrow{a_n} f(p_n) && (f \text{ is an isomorphism}) \\
&\text{ iff } \exists q_1, \ldots, q_n \in \mathsf{Pr}\ .q \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n && (\text{take } q_1 = f(p_1) \ldots q_n = f(p_n)) \\
&\text{ iff } \sigma \in \mathsf{traces}(q) && \text{(by definition)}
\end{aligned}
$$

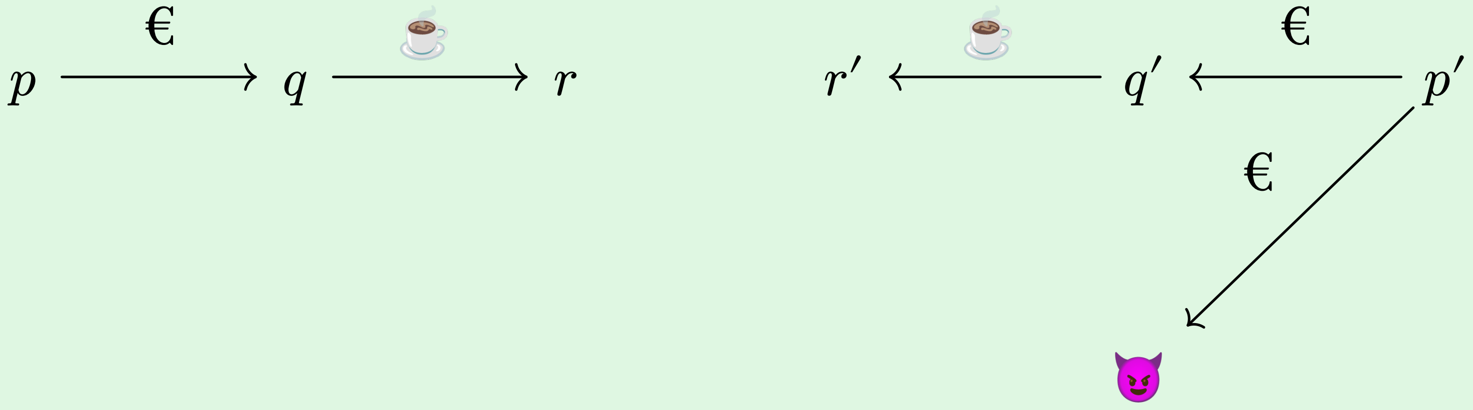
For $\leftrightarrow \neq \equiv_{\mathsf{tr}}$, reconsider p and q below, having $p \equiv_{\mathsf{tr}} q$ but $p \not\leftrightarrow q$.



$\blacksquare$

Trace Equivalence: End of Story?

Example.



$$\mathsf{traces}(p) = \{\varepsilon, €, €☕\} = \{\varepsilon, €, €, €☕\} = \mathsf{traces}(p')$$

There is one trace, namely €, that is a completed trace of p' but not of p.

In other words, trace equivalence (i.e., $\equiv_{\mathsf{tr}}$) is **not** sensitive to deadlocks.

The Completed Trace Semantics

Definition 14 A process $p \in \mathsf{Pr}$ is a *deadlock* if $p \stackrel{a}{\not\longrightarrow}$ for all $a \in \mathsf{Act}$.

The set of *completed traces* of a process $p \in \mathsf{Pr}$, denoted by $\mathsf{ctraces}(p)$ is the set of all traces $\sigma \in \mathsf{traces}(p)$ such that $p \stackrel{\sigma}{\longrightarrow} q$ and q is a deadlock.

Processes $p, q \in \mathsf{Pr}$ are *completed trace equivalent*, denoted by $p \equiv_{\mathsf{ctr}} q$, if $p \equiv_{\mathsf{tr}} q$ and $\mathsf{ctraces}(p) = \mathsf{ctraces}(q)$.

Theorem 15

$$\leftrightarrow \quad \stackrel{(1)}{\subsetneq} \quad \equiv_{\mathsf{ctr}} \quad \stackrel{(2)}{\subsetneq} \quad \equiv_{\mathsf{tr}}$$

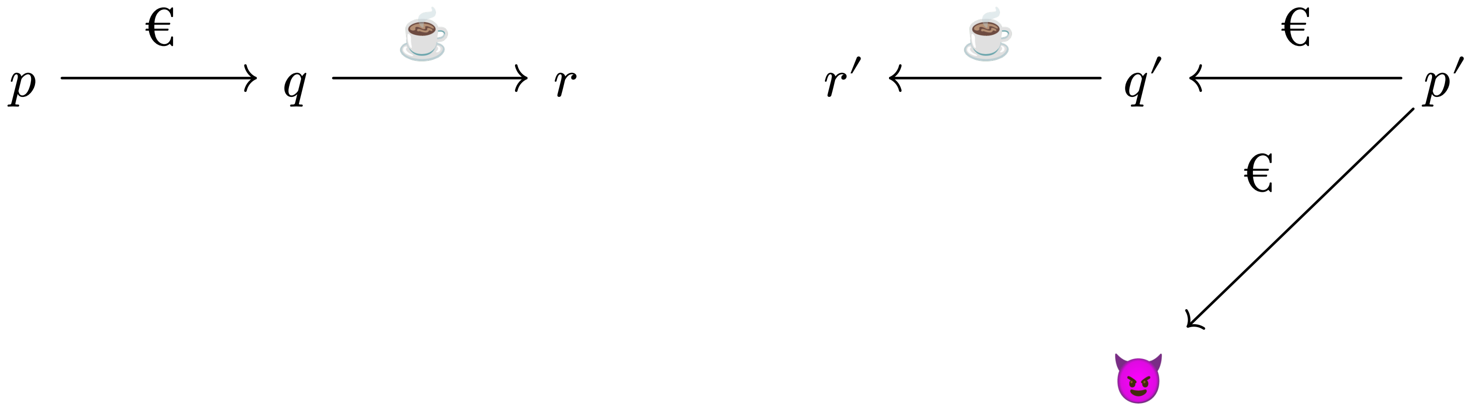
Proof of Theorem 15

Theorem 15

$$\leftrightarrow \quad \overset{(1)}{\subsetneq} \quad \equiv_{\mathsf{ctr}} \quad \overset{(2)}{\subsetneq} \quad \equiv_{\mathsf{tr}}$$

Regarding (2),

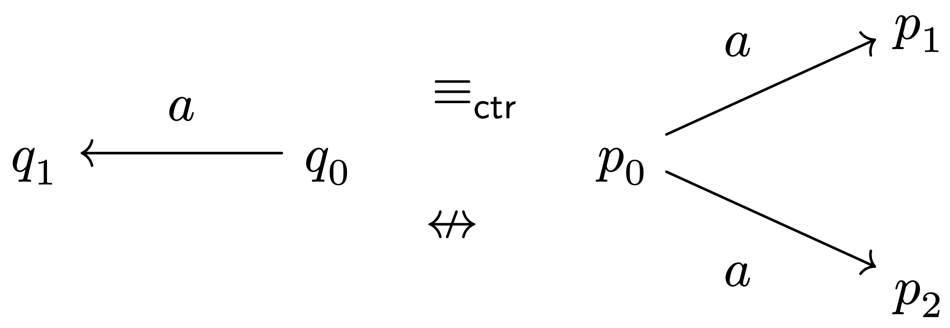
- observe that trace equivalence is part of the definition of $\equiv_{\mathsf{ctr}}$;
- in fact, $\mathsf{ctraces}(p) \subseteq \mathsf{traces}(p)$ for all processes $p \in \mathsf{Pr}$;
- furthermore, 😈 serves as a counterexample, proving $\equiv_{\mathsf{ctr}} \neq \equiv_{\mathsf{tr}}$.



Proof of Theorem 15

Towards (1),

- observe that a deadlock process $p \in \mathsf{Pr}$ can only be isomorphic to other deadlock processes;
- in fact, $p \leftrightarrow q$ for all processes $p, q \in \mathsf{Pr}$ that are deadlocks;
- hence, any completed trace of $p \in \mathsf{Pr}$ must be a a completed trace of $f(p)$ (by the same arguments as in proof of Theorem 13);
- also, $\leftrightarrow \neq \equiv_{\mathsf{ctr}}$ (e.g., p_0 and q_0 below).



∎

Completed Traces: End of Story?

Definition 14 A process $p \in \mathsf{Pr}$ is a *deadlock* if $p \stackrel{a}{\not\longrightarrow}$ for all $a \in \mathsf{Act}$.

The set of *completed traces* of a process $p \in \mathsf{Pr}$, denoted by $\mathsf{ctraces}(p)$ is the set of all traces $\sigma \in \mathsf{traces}(p)$ such that $p \xrightarrow{\sigma} q$ and q is a deadlock.

Processes $p, q \in \mathsf{Pr}$ are *completed trace equivalent*, denoted by $p \equiv_{\mathsf{ctr}} q$, if $p \equiv_{\mathsf{tr}} q$ and $\mathsf{ctraces}(p) = \mathsf{ctraces}(q)$.

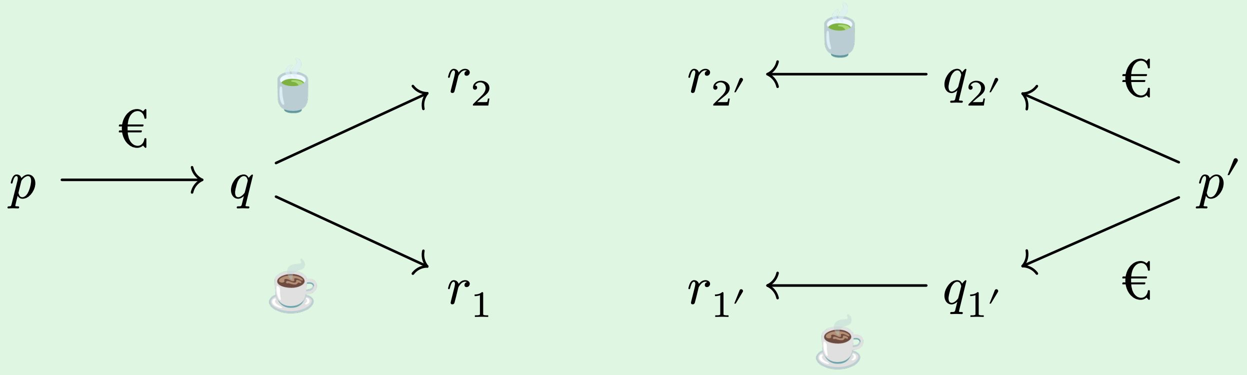
Theorem 15

$$\leftrightarrow \quad \overset{(1)}{\subsetneq} \quad \equiv_{\mathsf{ctr}} \quad \overset{(2)}{\subsetneq} \quad \equiv_{\mathsf{tr}}$$

$\equiv_{\mathsf{ctr}}$ preserves traces (2) and deadlocks (😈)

Completed Traces are Insensitive to Nondeterminism

Example.



What more do we need?

1. We are looking for the intimate connection between nondeterminism and interaction.
2. We are aiming at equivalences going beyond *linear-time* ($\equiv_{\mathsf{tr}}$ and $\equiv_{\mathsf{ctr}}$ are linear-time).

Recall

Definition 11 Any binary relation $\mathcal{R} \subseteq \mathsf{Pr} \times \mathsf{Pr}$ is called a *process relation*. $\mathcal{R}$ is a *process equivalence* if it is a process relation and an equivalence.

Theorem 15

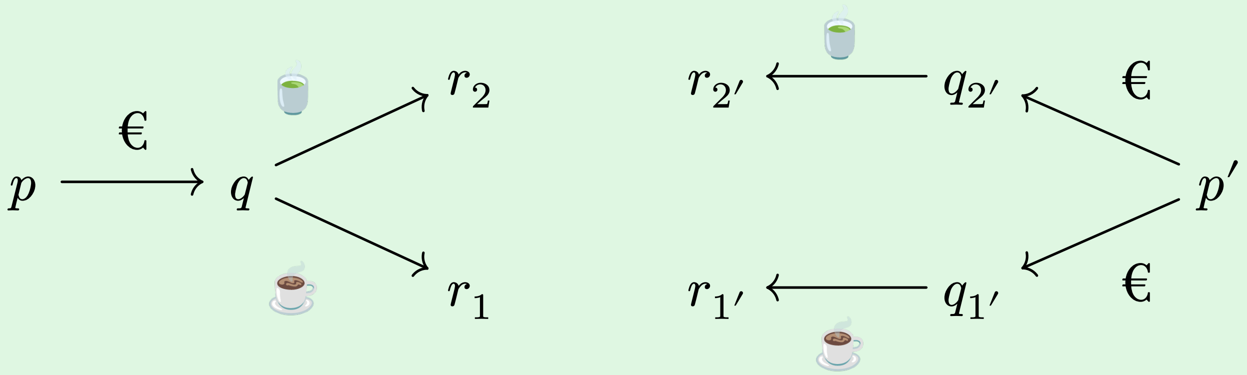
$$\leftrightarrow \quad \overset{(1)}{\subsetneq} \quad \equiv_{\mathsf{ctr}} \quad \overset{(2)}{\subsetneq} \quad \equiv_{\mathsf{tr}}$$

If, between two process equivalences $\mathcal{R}_1$ and $\mathcal{R}_2$, it holds that $\mathcal{R}_1 \subseteq \mathcal{R}_2$, we say that $\mathcal{R}_1$ *is finer than* $\mathcal{R}_2$, and $\mathcal{R}_2$ *is coarser than* $\mathcal{R}_1$.

The coarsest process equivalence of all is $\mathcal{U} \subseteq \mathsf{Pr} \times \mathsf{Pr}$.

Towards More Meaningful Equivalences

Example.



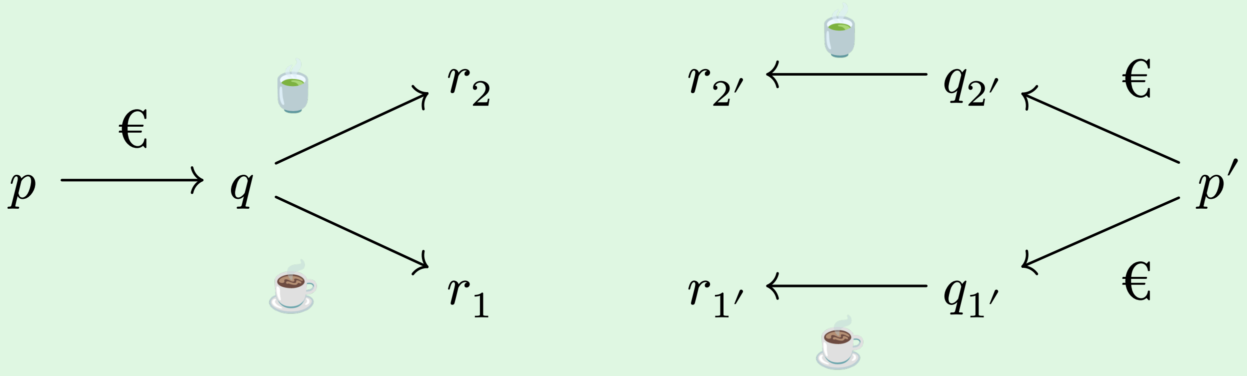
Maybe induction helps?

Suppose, $p \equiv p'$ ($\longleftarrow$ **claim**);

1. since $p \xrightarrow{€} q$, p' needs to have a *similar* step
2. $p' \xrightarrow{€} q_{1'}$ and $p' \xrightarrow{€} q_{2'}$
3. thus, the **claim** holds if $q \equiv q_{1'}$ or $q \equiv q_{2'}$
4. but as $q \xrightarrow{\text{☕}}$ and $q_{2'} \overset{\text{☕}}{\not\longrightarrow}$, $q \not\equiv q_{2'}$; similarly, $q \xrightarrow{\text{🍵}}$ but $q_{1'} \overset{\text{🍵}}{\not\longrightarrow}$, $q \not\equiv q_{1'}$

Induction Seems to Work

Example.



$p \not\equiv p'$ because $q \not\equiv q_{1'}$ and $q \not\equiv q_{2'}$.

Cooking up Equivalence $\equiv$

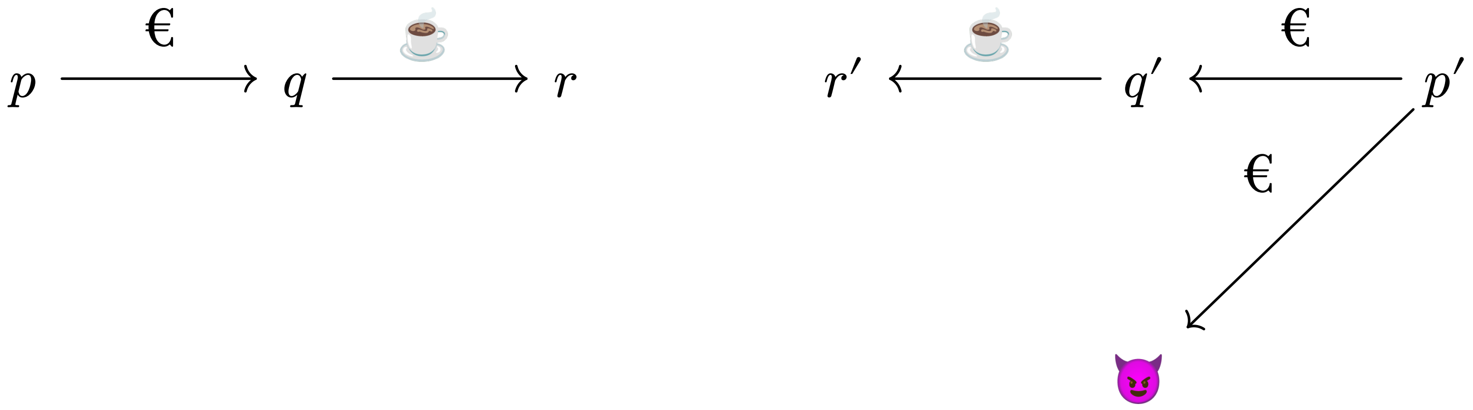
$p \equiv q$ if, for all $a \in \mathsf{Act}$,

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \equiv q'$;
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \equiv q'$.

Induction Seems to Work

$p \equiv q$ if, for all $a \in \mathsf{Act}$,

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \equiv q'$;
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \equiv q'$.



$p \not\equiv p'$ because $q \not\equiv$ 😈

Note, $r \equiv r' \equiv$ 😈

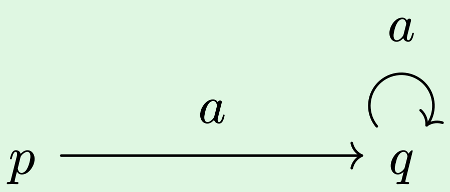
All deadlock processes are equivalent under $\equiv$.

Where Does Induction Fail?

$p \equiv q$ if, for all $a \in \mathsf{Act}$,

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \equiv q'$;
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \equiv q'$.

Example. *Reconsider processes p and q and find that $p \equiv_{\mathsf{tr}} q$*



To prove that $p \equiv q$, we have to show that $q \equiv q$ because

1. $p \xrightarrow{a} q$ *and there is a* q' *such that* $q \xrightarrow{a} q'$*, namely* $q' = q$*, for which* $q \equiv q' = q$*, and*
2. $q \xrightarrow{a} q$ *and there is a* p' *such that* $p \xrightarrow{a} p'$*, namely* $p' = q$*, for which* $p' = q \equiv q$*.*

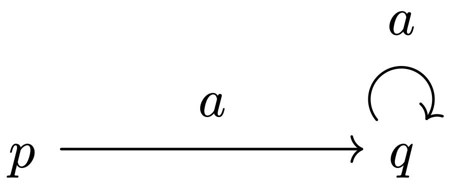
To prove that $q \equiv q$, we have to show that $q \equiv q$ … To prove that $q \equiv q$, we have to show that $q \equiv q$ … To prove that $q \equiv q$, we have to show that $q \equiv q$ … … ∎

Why Does Induction Fail?

$p \equiv q$ if, for all $a \in \mathsf{Act}$,

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \equiv q'$;
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \equiv q'$.

- Induction requires a **base case** start with **nothing**: $\mathcal{R}_0 = \{\}$
- By definition, in order to know that $p \equiv q$, we have to already know that $p' \equiv q'$
- In the example, to know/prove that $p \equiv q$, we have to already know that $q \equiv q$



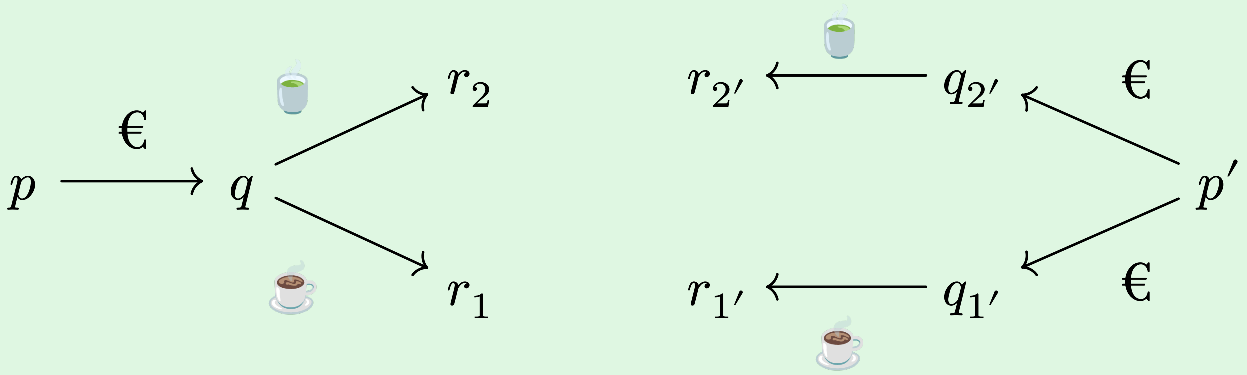
What went wrong?

What went well?

$p \equiv q$ if, for all $a \in \mathsf{Act}$,

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \equiv q'$;
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \equiv q'$.

Example.



An Inductive Approach to Process Equivalence in Reverse

> **Note**
>
> The coarsest process equivalence of all is $\mathcal{U} \subseteq \mathsf{Pr} \times \mathsf{Pr}$.

Compute $\simeq_0, \simeq_1, \ldots$ and define $\simeq_\omega := \bigcap_{i \geq 0} \simeq_i$

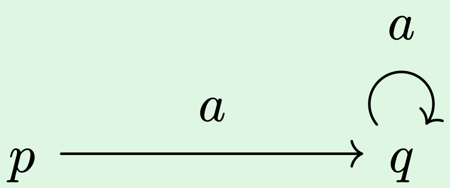
1. set $\simeq_0 = \mathcal{U}$
2. $p \simeq_{n+1} q$ for $n \geq 0$ if for all $a \in \mathsf{Act}$:
   a. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \simeq_n q'$;
   b. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \simeq_n q'$.

An Inductive Approach to Process Equivalence in Reverse

Compute $\simeq_0, \simeq_1, \ldots$ and define $\simeq_\omega := \bigcap_{i \geq 0} \simeq_i$

1. set $\simeq_0 = \mathcal{U}$
2. $p \simeq_{n+1} q$ for $n \geq 0$ if for all $a \in \mathsf{Act}$:
   a. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \simeq_n q'$;
   b. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \simeq_n q'$.

Example.

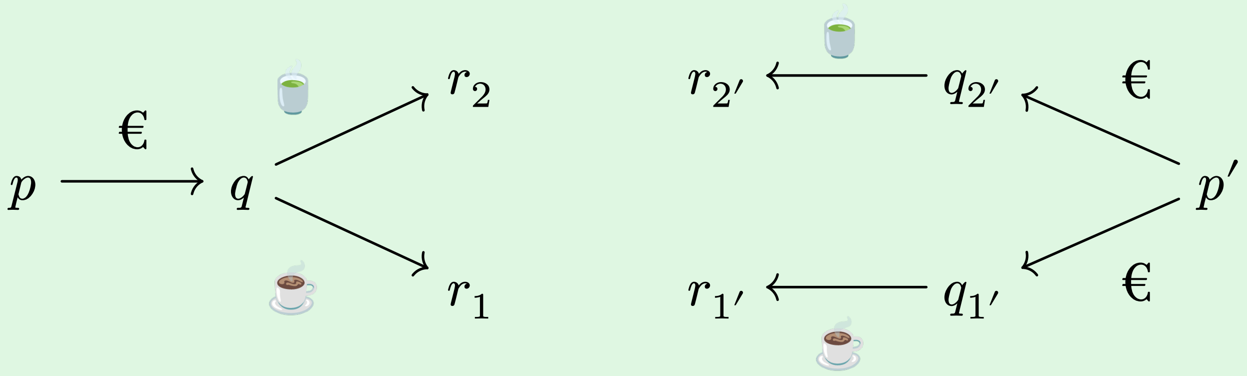


$$\simeq_0 = \{(p,p),(p,q),(q,p),(q,q)\}$$

$$\simeq_1 = \{(p,p),(p,q),(q,p),(q,q)\} = \simeq_0 = \simeq_\omega$$

An Inductive Approach to Process Equivalence in Reverse

Example.



$$\simeq_0 = \{(p,p), \cancel{(p,q)}, \cancel{(p,r_1)}, \cancel{(p,r_2)}, \ldots\}$$

$$\simeq_1 = \{(p,p), (p,p'), \ldots, \cancel{(q,q_{2'})}, \cancel{(q,q_{1'})}, \ldots, (r_1,r_{1'}), (r_1,r_{2'}), \ldots\}$$

$$\simeq_2 = \{(p,p), \cancel{(p,p')}, \cancel{(p',p)}, (p',p'), (q,q), (q_{1'},q_{1'}), (q_{2'},q_{2'}), \ldots\}$$

$$\simeq_3 = \{(p,p), (p',p'), (q,q), (q_{1'},q_{1'}), (q_{2'},q_{2'}), \ldots\} = \simeq_\omega$$

$$p \not\simeq_\omega p'$$

Rebooting Process Equivalence

A process relation $\mathcal{R} \subseteq \mathsf{Pr} \times \mathsf{Pr}$ is called a *(strong) bisimulation* if, for all $p, q \in \mathsf{Pr}$, $p \mathrel{\mathcal{R}} q$ implies

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \mathrel{\mathcal{R}} q'$, and
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \mathrel{\mathcal{R}} q'$

for all $a \in \mathsf{Act}$. We call p and q *bisimilar*, denoted $p \simeq q$, if there is a bisimulation $\mathcal{R}$ such that $p \mathrel{\mathcal{R}} q$. $\simeq$ is called *the bisimilarity*.

Rebooting Process Equivalence

Definition 16 (Bisimulation, Bisimilarity) A process relation $\mathcal{R} \subseteq \mathsf{Pr} \times \mathsf{Pr}$ is called a *(strong) bisimulation* if, for all $p, q \in \mathsf{Pr}$, $p \mathrel{\mathcal{R}} q$ implies

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \mathrel{\mathcal{R}} q'$, and
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \mathrel{\mathcal{R}} q'$

for all $a \in \mathsf{Act}$. We call p and q *bisimilar*, denoted $p \simeq q$, if there is a bisimulation $\mathcal{R}$ such that $p \mathrel{\mathcal{R}} q$. $\simeq$ is called *bisimilarity*.

Rebooting Process Equivalence

Definition 16 (Bisimulation, Bisimilarity) A process relation $\mathcal{R} \subseteq \mathsf{Pr} \times \mathsf{Pr}$ is called a *(strong) bisimulation* if, for all $p, q \in \mathsf{Pr}$, $p \mathrel{\mathcal{R}} q$ implies

1. for all p' with $p \xrightarrow{a} p'$, there is a q' with $q \xrightarrow{a} q'$ and $p' \mathrel{\mathcal{R}} q'$, and
2. for all q' with $q \xrightarrow{a} q'$, there is a p' with $p \xrightarrow{a} p'$ and $p' \mathrel{\mathcal{R}} q'$

for all $a \in \mathsf{Act}$. We call p and q *bisimilar*, denoted $p \simeq q$, if there is a bisimulation $\mathcal{R}$ such that $p \mathrel{\mathcal{R}} q$. $\simeq$ is called *bisimilarity*.

Consequences

1. bisimilarity $\simeq$ is the union of all bisimulations
2. showing that $p \simeq q$ holds reduces to finding a bisimulation $\mathcal{R}$ such that $p \mathrel{\mathcal{R}} q$
3. conversely, $p \not\simeq q$ can be shown by excluding the existence of any such bisimulation $\mathcal{R}$