Bisimulation Games

Game Theory Study Group

Joe Hurd

University of Cambridge

A Simple Process Calculus

We introduce a simple process calculus (consisting of operators taken from Milner's CCS) in which to write our examples and motivate bisimulation.

Fix a set A of actions containing the *internal* action τ.

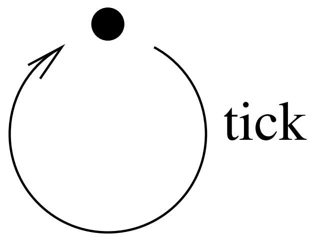
Write $\mathsf{E} \xrightarrow{a} \mathsf{F}$ if process E becomes process F after performing action $a \in A$.

A Simple Process Calculus

Prefix Axiom: $a.\mathsf{E} \xrightarrow{a} \mathsf{E}$

So after making the definition $\mathsf{Cl} = \mathit{tick}.\mathsf{Cl}$, we can apply the axiom to derive $\mathsf{Cl} \xrightarrow{\mathit{tick}} \mathsf{Cl}$, and the process Cl models an idealised clock.

This can be visualised in a labelled transition system:



A Simple Process Calculus

Choice Rule:

$$\frac{\exists i \in I.\ \mathsf{E}_i \xrightarrow{a} \mathsf{F}}{\left(\sum_{i \in I} \mathsf{E}_i\right) \xrightarrow{a} \mathsf{F}}$$

We often make use of the binary choice + operator and the nil process 0, both special cases.

As an example of this consider the family of processes

$$\mathsf{Cl}_i = \underbrace{\mathit{tick}.\mathit{tick}\ldots\mathit{tick}.}_{i \text{ times}}0$$

Now the process

$$\mathsf{Clock} = \sum_{i \in \mathbb{N}} \mathsf{Cl}_i$$

models a clock that will eventually break down.

A Simple Process Calculus

Let every action $a \in A$ have a co-action $\bar{a} \in A$, satisfying $\bar{\bar{a}} = a$.

Concurrent composition rules:

$$\frac{\mathsf{E} \xrightarrow{a} \mathsf{E}'}{\mathsf{E}|\mathsf{F} \xrightarrow{a} \mathsf{E}'|\mathsf{F}} \qquad \frac{\mathsf{F} \xrightarrow{a} \mathsf{F}'}{\mathsf{E}|\mathsf{F} \xrightarrow{a} \mathsf{E}|\mathsf{F}'}$$

$$\frac{\mathsf{E} \xrightarrow{a} \mathsf{E}' \quad \mathsf{F} \xrightarrow{\bar{a}} \mathsf{F}'}{\mathsf{E}|\mathsf{F} \xrightarrow{\tau} \mathsf{E}'|\mathsf{F}'} \quad a, \bar{a} \neq \tau$$

Abstraction rules:

$$\frac{\mathsf{E} \xrightarrow{a} \mathsf{F}}{\mathsf{E}\backslash J \xrightarrow{a} \mathsf{F}\backslash J} \quad a, \bar{a} \notin J$$

where $J \subset A$ is a subset of possible actions.

A Simple Process Calculus

An example process modelling a level crossing:

$$\begin{aligned}
\mathsf{Road} &= car.up.\overline{ccross}.\overline{down}.\mathsf{Road} \\
\mathsf{Rail} &= train.green.\overline{tcross}.\overline{red}.\mathsf{Rail} \\
\mathsf{Signal} &= \overline{green}.red.\mathsf{Signal} + \overline{up}.down.\mathsf{Signal} \\
\mathsf{Crossing} &= (\mathsf{Road}|\mathsf{Rail}|\mathsf{Signal})\backslash \\
&\quad \{green, red, up, down\}
\end{aligned}$$

The Simulation Game

The simulation game $\mathcal{S}(\mathsf{E}_0, \mathsf{F}_0)$ is played by Spoiler and Duplicator. A play of the game is a sequence of positions $(\mathsf{E}_0, \mathsf{F}_0), (\mathsf{E}_1, \mathsf{F}_1), \ldots$ determined by the following rules:

If $(\mathsf{E}_i, \mathsf{F}_i)$ is the current position, then Spoiler chooses a transition $\mathsf{E}_i \xrightarrow{a} \mathsf{E}_{i+1}$ and then Duplicator chooses a transition $\mathsf{F}_i \xrightarrow{a} \mathsf{F}_{i+1}$.

If Duplicator cannot match Spoiler's action then Spoiler has won. If Spoiler cannot make a move or the play goes on forever, then Duplicator has won. These are mutually exclusive.

Note: we could instead play these games with the observable transitions

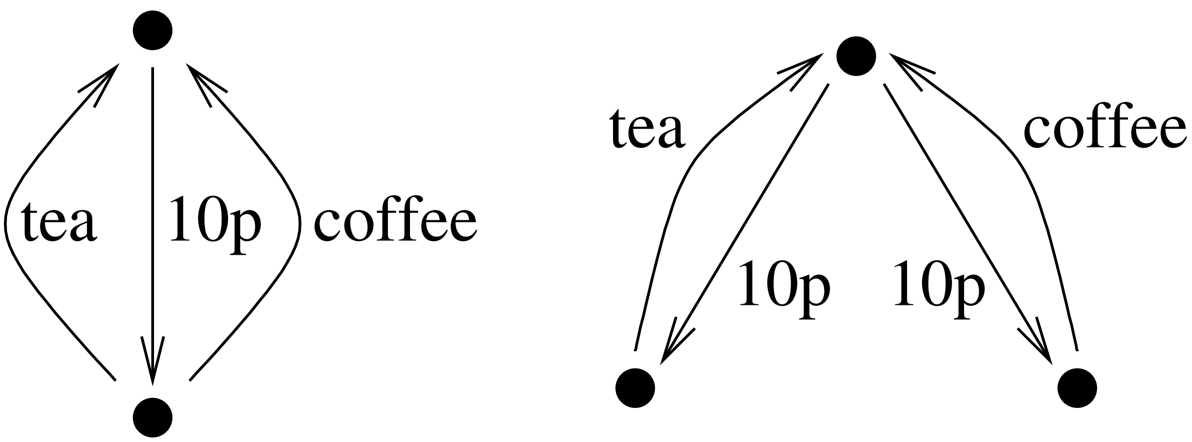
$$\mathsf{E} \overset{a}{\Longrightarrow} \mathsf{F} = \mathsf{E} \xrightarrow{\tau^*} \mathsf{E}' \xrightarrow{a} \mathsf{F}' \xrightarrow{\tau^*} \mathsf{F} \qquad a \neq \tau$$

The Simulation Game

As an example consider the following two vending machine processes:

$$\mathsf{V}_1 = 10p.(tea.\mathsf{V}_1 + coffee.\mathsf{V}_1)$$

$$\mathsf{V}_2 = 10p.tea.\mathsf{V}_2 + 10p.coffee.\mathsf{V}_2$$



The Simulation Game

A strategy for Spoiler is a (necessarily partial) function from a pair of processes (E, F) to a transition $\mathsf{E} \xrightarrow{a} \mathsf{E}'$.

A strategy for Duplicator is a (partial) function from a transition $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ and a process F to a transition $\mathsf{F} \xrightarrow{a} \mathsf{F}'$.

A strategy π is a winning strategy if π wins all possible plays.

Proposition: The simulation game defines a pre-order on processes, where $\mathsf{E} \preceq \mathsf{F}$ iff Duplicator has a winning strategy for the game $\mathcal{S}(\mathsf{E}, \mathsf{F})$.

Note: this idea of simulation is useful for refining a design interface A to a functional implementation B in a sequence of steps:

$$\mathsf{A} = \mathsf{E}_0 \preceq \mathsf{E}_1 \preceq \cdots \preceq E_n = \mathsf{B}$$

The Bisimulation Game

The bisimulation game $\mathcal{B}(\mathsf{E}_0, \mathsf{F}_0)$ is the same as the simulation game $\mathcal{S}(\mathsf{E}_0, \mathsf{F}_0)$, except that at each stage Spoiler can now choose to make a transition either from E_i or from F_i, and Duplicator must match it in the other process.

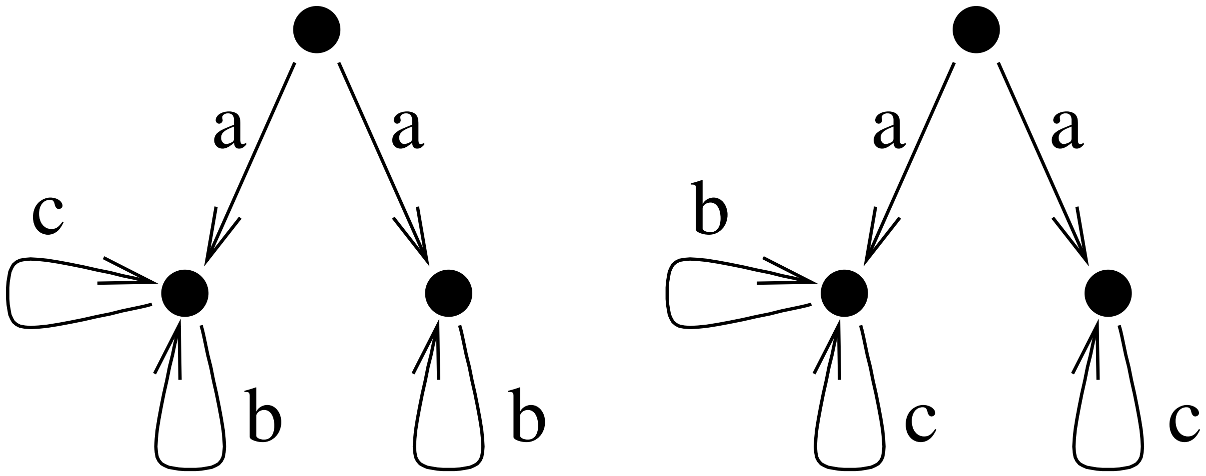
Proposition: If Spoiler can win $\mathcal{S}(\mathsf{E}, \mathsf{F})$ then he can win $\mathcal{B}(\mathsf{E}, \mathsf{F})$.

Equivalently: If Duplicator can win the bisimulation game $\mathcal{B}(\mathsf{E}, \mathsf{F})$ then she can win the simulation games $\mathcal{S}(\mathsf{E}, \mathsf{F})$ and $\mathcal{S}(\mathsf{F}, \mathsf{E})$.

But is the converse true?

The Bisimulation Game

Counter-example:



The Bisimulation Game

Say two processes E and F are game-equivalent if Duplicator has a winning strategy for $\mathcal{B}(\mathsf{E}, \mathsf{F})$.

A binary relation $\mathcal{R}$ between processes is a bisimulation if for all $(\mathsf{E}, \mathsf{F}) \in \mathcal{R}$ and $a \in A$:

1. if $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ then there exists an F' such that $\mathsf{F} \xrightarrow{a} \mathsf{F}'$ and $(\mathsf{E}', \mathsf{F}') \in \mathcal{R}$.
2. if $\mathsf{F} \xrightarrow{a} \mathsf{F}'$ then there exists an E' such that $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ and $(\mathsf{E}', \mathsf{F}') \in \mathcal{R}$.

Say two processes E and F bisimulate (written $\mathsf{E} \sim \mathsf{F}$) if there exists a bisimulation relation $\mathcal{R}$ containing (E, F).

Proposition: Two processes E and F are game-equivalent iff $\mathsf{E} \sim \mathsf{F}$.

Modal Logic

Let $\mathcal{M}_\infty$ be the following family of modal formulas:

$$\Phi ::= \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i \mid [K]\Phi \mid \langle K \rangle \Phi$$

where $K \subset A$, and I is an arbitrary index set.

The following inductive stipulation defines when a process E has a modal property Φ, written $\mathsf{E} \models \Phi$

$$\begin{aligned}
\mathsf{E} \models \bigwedge_{i \in I} \Phi_i &\equiv \forall i \in I.\ \mathsf{E} \models \Phi_i \\
\mathsf{E} \models \bigvee_{i \in I} \Phi_i &\equiv \exists i \in I.\ \mathsf{E} \models \Phi_i \\
\mathsf{E} \models [K]\Phi &\equiv \forall \mathsf{F}. \forall a \in K.\ \mathsf{E} \xrightarrow{a} \mathsf{F} \Rightarrow \mathsf{F} \models \Phi \\
\mathsf{E} \models \langle K \rangle \Phi &\equiv \exists \mathsf{F}. \exists a \in K.\ \mathsf{E} \xrightarrow{a} \mathsf{F} \wedge \mathsf{F} \models \Phi
\end{aligned}$$

Modal Logic

Write $\mathsf{E} \overset{\mathcal{M}_\infty}{\equiv} \mathsf{F}$ if $\forall\, \Phi \in \mathcal{M}_\infty.\ (\mathsf{E} \models \Phi$ iff $\mathsf{F} \models \Phi)$.

Theorem: $\mathsf{E} \sim \mathsf{F}$ iff $\mathsf{E} \overset{\mathcal{M}_\infty}{\equiv} \mathsf{F}$.

Proof: $\Rightarrow$: By induction on modal formulas: Suppose $\mathsf{E} \models [K]\Phi$. To show that $\mathsf{F} \models [K]\Phi$, we must show that $\mathsf{F}' \models \Phi$ for $\mathsf{F} \xrightarrow{a} \mathsf{F}'$ and $a \in K$. Since $\mathsf{E} \sim \mathsf{F}$, we can find $\mathsf{E} \xrightarrow{a} \mathsf{E}'$ with $\mathsf{E}' \sim \mathsf{F}'$. Now we are done by the induction hypothesis.

$\Leftarrow$: Show by contradiction that the relation $\{(\mathsf{E}, \mathsf{F}) : \mathsf{E} \overset{\mathcal{M}_\infty}{\equiv} \mathsf{F}\}$ is a bisimulation.

This is a modal characterization of bisimulation, due to Hennessey and Milner.

Modal Logic

Why the need for infinite conjunctions and disjunctions?

Let $\mathcal{M}$ denote the subset of $\mathcal{M}_\infty$ restricted to finite conjunctions and disjunctions.

Consider the simulation game $\mathcal{S}(\mathsf{Cl}, \mathsf{Clock})$. This is clearly a win for Spoiler, since Duplicator's clock is eventually going to break down. Therefore $\mathcal{B}(\mathsf{Cl}, \mathsf{Clock})$ is also a win for Spoiler, but $\mathsf{Cl} \overset{\mathcal{M}}{\equiv} \mathsf{Clock}$.

This occurs because there are an infinite number of transitions from Clock with the same action. On the set of processes where this is forbidden (called image finite processes), $\mathcal{M}$-equivalence is the same as bisimulation.

References

This talk was mostly taken from the notes of a course given by Colin Stirling entitled "Bisimulation, Model Checking and Other Games" at a Mathfit Instructional Course on Games and Computation. The notes can be found on his website.

The mutual simulation but no bisimulation example was taken from "Model Checking", E. M. Clarke, O. Grumberg and D. A. Peled, MIT Press 1999.