Defining Equivalence

Ehrenfeucht-Fraisse Games

1. Two structures (e.g., processes)

2. How alike are they?

3. Play games to distinguish them

Ingredients

1. Two players V (verifier) R (refuter)

   R wants to show structures are distinguishable

   V wants to show they are not

2. What is a move?

3. What is it to win?

Distinguishable Processes

A pair of processes E and F is distinguishable if one has a transition the other doesnt

$$E \xrightarrow{a} E' \quad \text{and} \quad \text{not}(F \xrightarrow{a}) \quad \text{or} \quad F \xrightarrow{a} F' \quad \text{and} \quad \text{not}(E \xrightarrow{a})$$

Alternatively, w.r.t. *observable* disitinguishability

$$E \stackrel{a}{\Longrightarrow} E' \quad \text{and} \quad \text{not}(F \stackrel{a}{\Longrightarrow}) \quad \text{or} \quad F \stackrel{a}{\Longrightarrow} F' \quad \text{and} \quad \text{not}(E \stackrel{a}{\Longrightarrow})$$

Bisimulation Game $\mathsf{G}(E_0, F_0)$

Play of $\mathsf{G}(E_0, F_0)$ is a finite/infinite sequence $(E_0, F_0) \ldots (E_i, F_i) \ldots$

If $(E_0, F_0) \ldots (E_j, F_j)$ then (E_{j+1}, F_{j+1}) is determined by move

- Player R chooses a transition $E_j \xrightarrow{a} E_{j+1}$, then player V chooses a transition with the same label $F_j \xrightarrow{a} F_{j+1}$
- Player R chooses a transition $F_j \xrightarrow{a} F_{j+1}$, then player V chooses a transition with the same label $E_j \xrightarrow{a} E_{j+1}$

Winning a play

R wins if reach a distinguishable pair

V wins otherwise (play is infinite or becomes stuck)

Examples

V wins every play of $\mathsf{G}(\mathtt{Cl}, \mathtt{Cl}')$

$$\begin{array}{lcl} \mathtt{Cl} & \stackrel{\text{def}}{=} & \mathtt{tick.Cl} \\ \mathtt{Cl}' & \stackrel{\text{def}}{=} & \mathtt{tick.tick.Cl}' \end{array}$$

V and R both win plays of $\mathsf{G}(\mathtt{Ven}_1, \mathtt{Ven}_2)$

$$\begin{array}{lcl} \mathtt{Ven}_1 & \stackrel{\text{def}}{=} & \mathtt{1p.1p.(tea.Ven}_1 + \mathtt{coffee.Ven}_1) \\ \mathtt{Ven}_2 & \stackrel{\text{def}}{=} & \mathtt{1p.(1p.tea.Ven}_2 + \mathtt{1p.coffee.Ven}_2) \\ \mathtt{Ven}_3 & \stackrel{\text{def}}{=} & \mathtt{1p.1p.tea.Ven}_3 + \mathtt{1p.1p.coffee.Ven}_3 \end{array}$$

However, V is able always to win $\mathsf{G}(\mathtt{Ven}_1, \mathtt{Ven}_2)$

Strategies

1. A strategy for a player: family of rules

   (a) Refuter: a rule "if the play so far is $(E_0, F_0) \dots (E_i, F_i)$, then choose transition t, where t is either $E_i \xrightarrow{a_i} E_{i+1}$ or $F_i \xrightarrow{a_i} F_{i+1}$."

   (b) Verifier: a rule "if the play so far is $(E_0, F_0) \dots (E_i, F_i)$, and player R has chosen transition t, then choose transition t'," where t' is a corresponding transition of the other process.

2. History-free strategy

   (a) For R: "at position (E, F) choose transition t", where t is either $E \xrightarrow{a} E'$ or $F \xrightarrow{a} F'$.

   (b) For V: "at position (E, F) when player R has chosen t choose t'", where t' is a corresponding transition of the other process

Determinacy

- A player uses a strategy in a play if all her moves obey the rules in it
- A strategy is winning if the player wins every play in which she uses it
- If a player has a winning strategy for a game, we say that the player "wins the game."
- Theorem (Determinacy) For every game G(E, F), one of the players has a history-free winning strategy

  Proof: Using ordinal induction on Force sets.

Simple Results

1. V wins G(E, E) using copy-cat strategy
2. If P wins G(E, F) then P wins G(F, E) using converse strategy
3. if P wins G(E, F) and G(F, G) then P wins G(E, G) using strategy composition
4. If P wins G(E, F) then P wins G$(C[E], C[F])$ for any CCS context $C[\ ]$ using ...

E and F are equivalent iff V wins G(E, F)

Game equivalence = Bisimulation equivalence

Bisimulations

A binary relation B between processes is a bisimulation provided that, whenever $(E, F) \in B$ and $a \in$ A,

- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in B$, and
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in B$

Two processes E and F are bisimulation equivalent (or bisimilar) if there is a bisimulation relation B such that $(E, F) \in B$

We write $E \sim F$ if E and F are bisimilar.

Examples

1. $a.(b.0 + c.0) \not\sim a.b.0 + a.c.0$

2. $\mathtt{Cnt} \sim \mathtt{Ct}'_0$

$$
\begin{array}{lcll}
\mathtt{Cnt} & \stackrel{\text{def}}{=} & \mathtt{up}.(\mathtt{Cnt} \mid \mathtt{down}.0) & \\
\mathtt{Ct}'_0 & \stackrel{\text{def}}{=} & \mathtt{up}.\mathtt{Ct}'_1 & \\
\mathtt{Ct}'_{i+1} & \stackrel{\text{def}}{=} & \mathtt{up}.\mathtt{Ct}'_{i+2} + \mathtt{down}.\mathtt{Ct}'_i & i \geq 0.
\end{array}
$$

The bisimulation

Let P_i be the following families of processes for $i \geq 0$ (when brackets are dropped between parallel components)

$$\begin{array}{lcl} P_0 & = & \{\mathtt{Cnt} \mid \mathsf{0}^j \ : \ j \geq 0\} \\ P_{i+1} & = & \{E \mid \mathsf{0}^j \mid \mathtt{down.0} \mid \mathsf{0}^k \ : \ E \in P_i \text{ and} \\ & & j \geq 0 \text{ and } k \geq 0\}, \end{array}$$

where $F \mid \mathsf{0}^0 = F$ and $F \mid \mathsf{0}^{i+1} = F \mid \mathsf{0}^i \mid \mathsf{0}$

$$B \ = \ \{(E, \mathtt{Ct}'_i) \ : \ i \geq 0 \text{ and } E \in P_i\}$$

is a bisimulation

Proof Technique: "Co-inductive"

To establish $E \sim F$

1. Present a candidate relation B with $(E, F) \in B$

2. Prove that indeed it obeys the hereditary conditions

Proof Technique Continued

Same sort of argument establishes that $\sim$ is a congruence

1. if $E \sim F$ then $G|E \sim G|F$

2. Proof: Assume that $E \sim F$, so there is a bisimulation B with $(E, F) \in B$.

3. Let C be the relation

$$\{(H|E', H|F') \,:\, (E', F') \in B\}$$

4. Show that C is a bisimulation ...

Observable bisimulations

Game equivalence = bisimulation equivalence

Neither, as presented, abstract from τ.

Observable games and observable bisimulation relations, appeal to the thicker transition relations $\stackrel{a}{\Longrightarrow}$, $\stackrel{\epsilon}{\Longrightarrow}$.

E and F are observably bisimilar, written as $E \approx F$, if there is an observable bisimulation B with $(E, F) \in B$

Protocol that may lose messages

$$
\begin{array}{lcl}
\texttt{Sender} & \stackrel{\text{def}}{=} & \texttt{in}(x).\overline{\texttt{sm}}(x).\texttt{Send1}(x) \\
\texttt{Send1}(x) & \stackrel{\text{def}}{=} & \texttt{ms}.\overline{\texttt{sm}}(x).\texttt{Send1}(x) + \texttt{ok}.\texttt{Sender} \\
\texttt{Medium} & \stackrel{\text{def}}{=} & \texttt{sm}(y).\texttt{Med1}(y) \\
\texttt{Med1}(y) & \stackrel{\text{def}}{=} & \overline{\texttt{mr}}(y).\texttt{Medium} + \tau.\overline{\texttt{ms}}.\texttt{Medium} \\
\texttt{Receiver} & \stackrel{\text{def}}{=} & \texttt{mr}(x).\overline{\texttt{out}}(x).\overline{\texttt{ok}}.\texttt{Receiver} \\
 & & \\
\texttt{Protocol} & \equiv & (\texttt{Sender} \mid \texttt{Medium} \mid \texttt{Receiver})\backslash\{\texttt{sm}, \texttt{ms}, \texttt{mr}, \texttt{ok}\}
\end{array}
$$

Example

$\texttt{Protocol} \approx \texttt{Cop}$. The following relation is an observable bisimulation

$$
\begin{array}{l}
\{(\texttt{Protocol}, \texttt{Cop})\} \cup \\
\{((\texttt{Send1}(m) \mid \texttt{Medium} \mid \overline{\texttt{ok}}.\texttt{Receiver}) \backslash J, \\
\qquad \texttt{Cop}) \ : \ m \in D\} \cup \\
\{((\overline{\texttt{sm}}(m).\texttt{Send1}(m) \mid \texttt{Medium} \mid \texttt{Receiver}) \backslash J, \\
\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \ : \ m \in D\} \cup \\
\{((\texttt{Send1}(m) \mid \texttt{Med1}(m) \mid \texttt{Receiver}) \backslash J, \\
\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \ : \ m \in D\} \cup \\
\{((\texttt{Send1}(m) \mid \texttt{Medium} \mid \overline{\texttt{out}}(m).\overline{\texttt{ok}}.\texttt{Receiver}) \backslash J, \\
\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \ : \ m \in D\} \cup \\
\{((\texttt{Send1}(m) \mid \overline{\texttt{ms}}.\texttt{Medium} \mid \texttt{Receiver}) \backslash J, \\
\qquad \overline{\texttt{out}}(m).\texttt{Cop}) \ : \ m \in D\}
\end{array}
$$

But

- Observable bisimulation equivalence is not a congruence because of initial preemptive power of τ
- $E \approx \tau.E$ but many cases where $E + F \not\approx \tau.E + F$
- $\approx^c$ is the largest subset of $\approx$ that is also a congruence.
- Observable equivalence is a congruence for all the operators of CCS, except sum

Defining $\approx^c$

$\approx^c$ can be described independently of process contexts in terms of transitions

$E \approx^c F$ *iff*

1. $E \approx F$ *and*
2. *if* $E \xrightarrow{\tau} E'$, *then* $F \xrightarrow{\tau} F_1 \stackrel{\varepsilon}{\Longrightarrow} F'$ *and* $E' \approx F'$ *for some* F_1 *and* F', *and*
3. *if* $F \xrightarrow{\tau} F'$, *then* $E \xrightarrow{\tau} E_1 \stackrel{\varepsilon}{\Longrightarrow} E'$ *and* $E' \approx F'$ *for some* E_1 *and* E'.

Conditional equational reasoning

A direct proof that two processes are bisimilar, or observably equivalent, is to exhibit the appropriate bisimulation relation that contains them

In the case that processes are finite state, this can be done automatically. There is a variety of tools that do this including the Edinburgh Concurrency Workbench

Alternatively, equivalence proofs can utilize conditional equational reasoning.

There is an assortment of algebraic, and semi-algebraic, theories of processes depending on the equivalence and the process combinators

It is essential that the equivalence be a congruence

Illustration

For $\approx^c$.

$$
\begin{array}{lcll}
a.\tau.x & = & a.x & \\
x + \tau.x & = & \tau.x & \\
(x + y)\backslash K & = & x\backslash K + y\backslash K & \\
(a.x)\backslash K & = & a.(x\backslash K) & \text{if } a \notin K \cup \overline{K} \\
(a.x)\backslash K & = & 0 & \text{if } a \in K \cup \overline{K} \\
x + 0 & = & x &
\end{array}
$$

Expansion law

$$x_i = \sum\{a_{ij}.x_{ij} \ : \ 1 \leq j \leq n_i\} \ \text{ for } i : 1 \leq i \leq m,$$
$$x_1 \mid \ldots \mid x_m \ =$$

$$\sum\{a_{ij}.y_{ij} \ : \ 1 \leq i \leq m \text{ and } 1 \leq j \leq n_i\} +$$
$$\sum\{\tau.y_{klij} \ : \ 1 \leq k < i \leq m \text{ and } a_{kl} = \overline{a}_{ij}\},$$

$$y_{ij} \ \equiv \ x_1 \mid \ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m$$

$$y_{klij} \ \equiv \ x_1 \mid \ldots \mid x_{k-1} \mid x_{kl} \mid x_{k+1} \mid$$
$$\ldots \mid x_{i-1} \mid x_{ij} \mid x_{i+1} \mid \ldots \mid x_m.$$

For instance

$$\begin{aligned} x_1 &= a.x_{11} + b.x_{12} + a.x_{13} \\ x_2 &= \overline{a}.x_{21} + c.x_{22}, \end{aligned}$$

$$\begin{aligned} x_1|x_2 \quad = \quad & a.(x_{11}|x_2) + b.(x_{12}|x_2) + a.(x_{13}|x_2) + \\ & \overline{a}.(x_1|x_{21}) + \\ & c.(x_1|x_{22}) + \tau.(x_{11}|x_{21}) + \tau.(x_{13}|x_{21}). \end{aligned}$$

Rules for recursion I

If E does not contain any occurrences of the parallel operator $|$, then P is "guarded" in E, provided that all occurrences of P in E are within the scope of a prefix $a.$ and a is an observable action (that is, not τ)

Assume that P is the only process constant in E. The guardedness condition guarantees that the equation $P = E$ has a unique solution up to $\approx^c$.

A solution to the equation $P = E$ is a process F such that $F \approx^c E\{F/P\}$. Uniqueness of solution is that, if both F and G are solutions, then $F \approx^c G$.

Rules for recursion II

- if $P \stackrel{\text{def}}{=} E$ then $P = E$
- if $P = E$ and P is guarded in E, and $Q = F$ and Q is guarded in F, and $E\{Q/P\} = F$, then $P = F$

Example $\mathtt{Cl} \stackrel{\text{def}}{=} \mathtt{tick.Cl}$, $\mathtt{Cl}' \stackrel{\text{def}}{=} \mathtt{tick.tick.Cl}'$.

$$
\begin{aligned}
&\mathtt{Cl} = \mathtt{tick.Cl} \\
&\mathtt{tick.Cl} = \mathtt{tick.tick.Cl} \\
&\mathtt{Cl} = \mathtt{tick.tick.Cl} \\
&\mathtt{Cl}' = \mathtt{tick.tick.Cl}' \\
&(\mathtt{tick.tick.Cl})\{\mathtt{Cl}'/\mathtt{Cl}\} = \mathtt{tick.tick.Cl}' \\
&\mathtt{Cl} = \mathtt{Cl}'
\end{aligned}
$$

Origins of bisimulation

1. Concurrency [Hennessy, Milner, Park]

   Language equivalence is not appropriate for "interacting automata" because it is not a congruence (w.r.t. natural operations)

2. Modal logic [van Benthem]

   Modal logic is a fragment of first-order logic (over transition graphs). Is there an independent characterisation of which fragment? Bisimulation invariance is key notion.

   $\Phi(x)$ is bisim invariant iff if $T \models \Phi[s]$ and $s \sim t$ then $T \models \Phi[t]$