Causal Process Algebra

Steve Reeves and David Streader

\{dstr,stever\}@cs.waikato.ac.nz.

Department of Computer Science
University of Waikato
Hamilton, New Zealand
Shadowy formal methods

- Formal methods illuminate (by hiding)
- The better the formal method the stronger the light and the darker the shadows
- Looking in the shadows is no criticism
Shadowy formal methods

- Formal methods illuminate (by hiding)
- The better the formal method the stronger the light and the darker the shadows
- Looking in the shadows is no criticism
- process algebras communication abstracts away:
  1. one action causing another
  2. one process choosing
Process Model

- One set (kind) of observable actions
- communication or synchronisation:
  1. is exclusive - must occur if it can.
  2. is private - unobserved

\[
\text{CCS } a;\!\!x \parallel \overline{b};\!\!\overline{x} \\
\text{CSP } a;\!\!x \parallel \alpha \overline{b};\!\!\overline{x}
\]
Process Model

- One set (kind) of observable actions
- communication or synchronisation:
  1. is exclusive - must occur if it can.
  2. is private - unobserved

\[
\begin{align*}
\text{CCS} & \quad a;x |\ | b;\bar{x} \\
\text{CSP} & \quad a;x, b;\bar{x} \\
\text{CCS VM} & \quad \parallel \text{Rob}_\alpha \text{ called restricted Composition by Milner} \\
\text{CSP} & \quad (\text{VM}_\alpha \text{Rob})/\alpha
\end{align*}
\]
Causal Process Algebra CPA has two sets of observable actions: the active or causal actions and the passive or reactive actions.

+ priority
Causal Process Algebra CPA has two sets of observable actions: the active or causal actions and the passive or reactive actions.

+ priority

Four advantages:
Causal Process Algebra

Causal Process Algebra CPA has two sets of observable actions: the active or causal actions and the passive or reactive actions.

+ priority

Four advantages:
1. extend refinement
2. determinism by construction (Milner)
3. small deterministic processes are implementable
4. hiding private communication does not introduce nondeterminism
Example

A world of Vending machines and Robots!
Example

- A world of Vending machines and Robots!

- Specification: The robot must select a button
  
  VM1 \[\parallel\] Rob will return one
  VM2 \[\parallel\] Rob will return two
  VM \[\parallel\] Rob will return one.
Example

A world of Vending machines and Robots!

 Specification: The robot must select a button

VM1 || Rob will return one
VM2 || Rob will return two
VM || Rob will return one.

Unsatisfiable in process algebra?
Processes are at the wrong level of detail?
Solutions

Without Priority Rob
Solutions

Without Priority Rob

(VM_α Rob) / α
Solutions

- Without Priority Rob

- \((\text{VM}_\alpha \parallel \text{Rob})/\alpha\)

- With Priority

\[ \begin{array}{c}
\text{Rob}^P \quad \bar{b}^1_2 \quad \bar{d}^2 \quad \tau^2 \\
\text{s} \quad \tau \\
\end{array} \]
One - Extending refinement

VM1 || R returns one

R1 \( \overset{\text{coin}}{s} \rightarrow \overset{\text{d1}}{\rightarrow} \overset{\text{one}}{\rightarrow} \overset{\text{b1}}{e} \)
One - Extending refinement

- VM1 || R returns one
- VM2 || R returns two
One - Extending refinement

- VM1 || R returns one
- VM2 || R returns two
- R1 ⊑ Rob
One - Extending refinement

- VM1 || R returns one
- VM2 || R returns two
- R1 ⊑ Rob
- VM || R returns one
  requires our process model be rebuilt.
Two - Preserving Determinism

If a deterministic robot interacts with a deterministic vending machine is the drink returned determined?
Two - Preserving Determinism

If a deterministic robot interacts with a deterministic vending machine is the drink returned determined?

We say yes Process algebra no!
Two - Preserving Determinism

If a deterministic robot interacts with a deterministic vending machine is the drink returned determined?

We say yes Process algebra no!

We are not the first to find this problematic!

Milner [?, p232] express the hope of finding “design rules, not too restrictive, which ensure that from determinate components we must arrive at a determinate system”
Two - Preserving Determinism

- If a deterministic robot interacts with a deterministic vending machine is the drink returned determined?
- We say yes Process algebra no!
- We are not the first to find this problematic!
- Milner [?, p232] express the hope of finding “design rules, not too restrictive, which ensure that from determinate components we must arrive at a determinate system”
- defines confluence [?, p237] “to strengthen determinacy in such a way that it will be preserved by restricted Composition”
Three - Implementable

We assume Nondeterministic behaviour not implementable on a deterministic finite state machine
Three - Implementable

- We assume Nondeterministic behaviour not implementable on a deterministic finite state machine
- Hence CSP/CCS $\mathcal{R}_0$ and $\mathcal{V}M$ not implementable!
Three - Implementable

- We assume Nondeterministic behaviour not implementable on a deterministic finite state machine
- Hence CSP/CCS $Rob$ and $VM$ not implementable!
- Occam does not implement CSP/CCS style parallel composition
- Occam is consistent with CSP failure refinement:
  $$\left( VM_{\alpha} \parallel Rob \right) / \alpha \sqsubseteq_F VM \parallel Occam \ Rob$$
- Not consistent with may and must testing refinement:
  $$\left( VM_{\alpha} \parallel Rob \right) / \alpha \not\sqsubseteq_{Test} VM \parallel Occam \ Rob$$
a set of names $Names$:

- passive actions $Act \overset{\text{def}}{=} \{ a \mid a \in Names \}$ and

- active actions $\overline{Act} \overset{\text{def}}{=} \{ \overline{a} \mid a \in Names \}$

- irreflexive priority relation $\triangleleft \subseteq Pri \times Pri$

- priority function $Act_{Pri} : Names \rightarrow Pri$

- function and relation lifted to actions or transitions.

PLTS

$$T_A \subseteq \{(n, (x, p), m) \mid n, m \in N_A \land (x, p) \in Act_{Pri} \land x \in Obs\} \cup \{(n, (\tau, p), m) \mid p \in Pri\}$$

$$n \stackrel{(x,p)}{\rightarrow} m$$
A is p-deterministic if all transitions leaving the same node have different names, whereas A is deterministic if transitions with the same priority and pre-node are passive actions with different names.
A is p-deterministic if all transitions leaving the same node have different names, whereas

A is deterministic if transitions with the same priority and pre-node are passive actions with different names.

A is deterministic iff \( \forall \bar{a} \in \text{Act}, \bar{b} \in \text{Act} \cup \{\tau\} \):

\[
\begin{align*}
x \xrightarrow{(\bar{a}, p)} z & \land x \xrightarrow{(\bar{b}, p)} y \Rightarrow z = y \land \bar{a} = \bar{b} \\
x \xrightarrow{(a, p)} z & \land x \xrightarrow{(a, p)} y \Rightarrow z = y
\end{align*}
\]
A is p-deterministic if all transitions leaving the same node have different names, whereas

A is deterministic if transitions with the same priority and pre-node are \textit{passive} actions with different names.

A is deterministic iff \( \forall \bar{a} \in \text{Act}, \bar{b} \in \text{Act} \cup \{\tau\} \)

\[ x \xrightarrow{\bar{a}, p} z \land x \xrightarrow{\bar{b}, p} y \Rightarrow z = y \land \bar{a} = \bar{b} \]

\[ x \xrightarrow{(a, p)} z \land x \xrightarrow{(a, p)} y \Rightarrow z = y \]

We define \( \preceq : \text{Act}_{\text{Pri}} \times \text{Act}_{\text{Pri}} \) by \( (x, p) \preceq (y, p') \) \( \overset{\text{def}}{=} \) \( p \preceq p' \).

Define \( \sqsubseteq_p \subseteq \text{Act}_{\text{Pri}} \times \text{Act}_{\text{Pri}} \) by \( \text{Act}_{\text{Pri}1} \sqsubseteq_p \text{Act}_{\text{Pri}2} \) \( \overset{\text{def}}{=} \) \( \preceq_1 \subseteq \preceq_2 \).

Lift \( \sqsubseteq_p \) to processes.

\( \sqsubseteq_{tp} \) is the transitive closure of \( \sqsubseteq_p \cup \sqsubseteq_t \)
Observational semantics

\[
\begin{array}{c}
\text{A} \\
S^{l}
\end{array}
\quad
\begin{array}{c}
\tau^{P} \\
\end{array}
\quad
\begin{array}{c}
\text{e}^{P} \\
\text{b}^{l}
\end{array}
\]
Observational semantics
Observational semantics

A

\[ s^l \]

\[ \tau^P \]

\[ e^P \]

\[ b^l \]

\[ S \]

Causal Process Algebra – p.12/14
Observational semantics (2)
Observational semantics (2)

Abs(A)

Abs(B)
Causal Process Algebra CPA has two sets of observable actions: the active or causal actions and the passive or reactive actions.

- priority

Disadvantage more complexe semantics

Four advantages:
1. extend refinement
2. determinism by construction (Milner)
3. small deterministic processes are implementable
4. hiding private communication does not introduce nondeterminism