A Non–Interleaving Semantics for MSC

Stefan Heymer
Institute for Telematics
Medical University at Lübeck

July 1, 1998

Overview

▷ Existing semantics

▷ Some Motivation on using a Translation to Process Algebra

▷ A Process Algebra P

▷ Operators for the Translation of MSCs to P

▷ Families of posets as a model

▷ A famlies of posets semantics for P

Existing Semantics for MSC

- ▷ Translation from MSC to Process Algebra
  (Mauw/Reniers)
- ▷ Signal/Next Event–Automata
  (Ladkin/Leue)
- ▷ Petri–Nets
  (Graubmann/Grabowski/Rudolph)
- ▷ Operational Semantics for MSC'96
  (Mauw/Reniers)
- ▷ Transition Systems with conditions as choice points
  (Rensink et al.)
- ▷ Pomsets
  (Katoen/Lambert)

YAPA?

- ▷ Why yet another translation to process algebra?
- ▷ Process Algebra as intermediate language
- ▷ Two–step construction:
  - Translation of MSC specification to process algebra
  - Giving a semantics to the process algebra
- ▷ Different semantics may be assigned to process algebra in second step:
  - Petri nets
  - Transition systems
  - Event structures
  - ...
- ▷ Here: Non–interleaving semantics for P

The Process Algebra

- ▷ Process Algebra P used in the construction:

$$p ::= \mathsf{nil} \mid a \cdot p \mid p_0 \oplus p_1 \mid p_0 \mp p_1 \mid p_0 \times p_1 \mid p \upharpoonright \Lambda \mid p\{\Xi\} \mid x \mid rec\ x.p$$

- ▷ Operators:

nil	Empty process
$a \cdot p$	Action prefixing
$p_0 \oplus p_1$	Choice
$p_0 \mp p_1$	Delayed choice
$p_0 \times p_1$	Parallel product
$p \upharpoonright \Lambda$	Restriction of behaviour
$p\{\Xi\}$	Relabelling of actions
$rec\ x.p$	Recursive behavior

- ▷ Synchronization of processes is defined by a combination of parallel product, restriction of behavior and relabelling

Problem 1: Delayed choice

- ▷ Z.120: Choice between MSCs in an alternative composition remains unresolved until behaviors differ
- ▷ Solution: Use delayed choice operator $\mp$ instead of usual choice operator $\oplus$
- ▷ Problem: Delayed choice operator cannot be defined in terms of other operators
- ▷ Semantics for the operators of the process algebra should be categorial constructions
- ▷ Solution of the problem:
  Introduce delayed choice as a special operator

Problem 2: Weak sequential composition

- ▷ Z.120: Sequential composition of MSCs by concatenating instance axes
- ▷ Instances in MSC may proceed even if the other instances have not reached the end of the MSC (weak sequential composition)
- ▷ Problem: Process algebra only has action prefixing for building up sequential behaviors
- ▷ Idea:
  - Introduce special *start* and *end* actions
  - Emulate weak sequential composition by parallel composition,...
  - ...forcing synchronization on *end* and *start* actions, thus "pasting" behaviors together, ...
  - ...and hiding the points of synchronization
- ▷ Define weak sequential composition as a shorthand notation:

$$p_1 \circ p_2 = ((p_1 \times p_2) \upharpoonright \Lambda_\circ)\{\Xi_\circ\}$$

Families of lposets

- ▷ A suitable class of models for the semantics of the process algebra
- ▷ Model used here: Families of labelled partial orders (Rensink 93)
- ▷ **Definition** A *family of labelled posets* is a non–empty, prefix–closed set of labelled partial orders $P = \langle E_P, \leq_P, \ell_P \rangle$, which are compatibly labelled, i. e. for a family of lposets $\mathcal{P}$ the condition

$$\forall P, Q \in \mathcal{P}.\ e \in E_P \cap E_Q \Rightarrow \ell_P(e) = \ell_Q(e)$$

  holds. The class of all families of labelled partial orders is denoted $\mathbf{f}_{\mathrm{LPO}}$.
- ▷ Special cases: labelled total orders (ltosets) and families of ltosets

Compositional semantics for the process algebra

▷ Define operations on families of lposets mirroring the intended effects of the operators of the process algebra

▷ Define a compositional semantics for the process algebra

▷ For families of lposets as a model:

$$
\begin{aligned}
[\![\mathsf{nil}]\!]_1 &= \{\varepsilon\} \\
[\![a \cdot p]\!]_1 &= {}_e a \cdot [\![p]\!]_1 \\
[\![p_1 \oplus p_2]\!]_1 &= [\![p_1]\!]_1 \oplus [\![p_2]\!]_1 \\
[\![p_1 \mp p_2]\!]_1 &= [\![p_1]\!]_1 \mp [\![p_2]\!]_1 \\
[\![p_1 \times p_2]\!]_1 &= [\![p_1]\!]_1 \times [\![p_2]\!]_1 \\
[\![p \upharpoonright \Lambda]\!]_1 &= [\![p]\!]_1 \upharpoonright \Lambda \\
[\![p\{\Xi\}]\!]_1 &= [\![p]\!]_1\{\Xi\}
\end{aligned}
$$

▷ Define a similar denotation $[\![\cdot]\!]_2$ for families of ltosets as a model

Relation to standardized semantics

▷ How do these semantics relate to the standardized one for MSC'92?

▷ Define the transition system for a family of lposets $\mathcal{P}$ as

$$Trans(\mathcal{P}) = (\mathcal{P}, \varepsilon, \rightarrow)$$

with

$$\rightarrow = \{(P, a, Q) \mid P, Q \in \mathcal{P} \wedge P \prec\!\!\cdot\, Q \wedge a \in \operatorname{ran} \ell_{Q \setminus P}\}$$

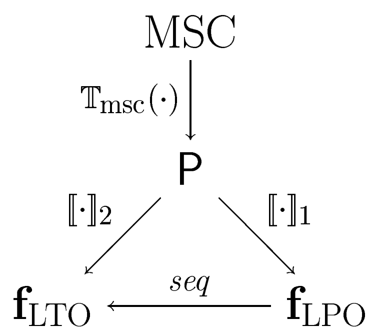
▷ **Theorem** For a BMSC M, the transition system $Trans([\![M\{\Xi\}]\!]_2)$ and the synchronization tree obtained from unfolding the process graph for $\mathcal{P}[\![M]\!]$ are bisimilar, with Ξ being the relabelling function

$$\Xi : a \mapsto \begin{cases} \text{undefined} & \text{if } a = \mathit{start}@i \text{ or } a = \mathit{end}@i \text{ for some } i \in \mathit{InstId} \\ a & \text{otherwise.} \end{cases}$$

▷ Define the sequentialization function

$$seq : \mathbf{f}_{\mathrm{LPO}} \to \mathbf{f}_{\mathrm{LTO}}$$
$$seq(\mathcal{P}) = \bigcup_{P \in \mathcal{P}} \{T \in \mathrm{LTO} \mid T \sqsubseteq P\}$$

▷ Observe the commutativity of the triangle in the following diagram:



Conclusions

▷ Two semantics for (a subset of) MSC'96

▷ Two–step construction

▷ Process algebra as "intermediate language"

▷ Possiblity to give several semantics to the process algebra, interleaving as well as non–interleaving

▷ Event–oriented model

▷ Possible extension: introduction of time annotations to events