A Real-Time Process Algebra with Open Intervals and Maximal Progress

Harald Fecher

Universität Mannheim, Institut für Informatik

Action Occurrences in Open Intervals

Examples:

- Alur and Dill's timed automaton
- some real time process algebras (e.g. Chen's or Daniels')

Reasons:

- user can formulate priority in a natural way
- dense time: to model dynamic behavior (refine time steps)

Maximal Progress Assumption

Definition: An action must occur at the instant at which all participants are ready.

$$E \xrightarrow{t} \;\Rightarrow E \stackrel{\tau}{\not\longrightarrow} \quad \wedge \, \forall \, t' < t : E \xrightarrow{t'} \stackrel{\tau}{\not\longrightarrow}$$

Example: Wang's timed CCS, Schneider's timed CSP

Reason: Otherwise the model is not forced to do anything.

Open Timed CCS

... is a real time process algebra based on CCS, with

- action occurrences in open intervals
- maximal progress assumption
- relative time
- instantaneous actions

Syntax:

$$\begin{array}{rcl} E & ::= & \mathrm{NIL} \mid \alpha^{\prec' \mathrm{e}'}_{\succ \mathrm{e}}.\mathrm{E} \mid \mathrm{E} + \mathrm{E} \mid \mathrm{E}|\mathrm{E} \mid \\ & & E\backslash a \mid E[f] \mid X \mid \mu X.E \end{array}$$

where $\succ \in \{>, \geq\}$, $\prec' \in \{<, \leq\}$, $e' \in R_0^+ \cup \{\infty\}$, $e \in R_0^+$ and $\alpha \in \mathrm{Act}$.

(Act will be defined more precisely later on)

Example

A plane (E_p) sends regularly an *ok*-signal ($\overline{ok}$-action).

The safety officer is allowed to send off ($send$) a rescue team (F).

A received *ok*-signal disables the sending for 5 minutes.

$$E_{officer} = \left(\mu X. \left(ok_{\geq 0}^{\leq \infty}.X + send_{\geq 5}^{\leq \infty}.F\right) | E_p\right) \setminus \{ok\}$$

The maximal progress assumption is needed, because otherwise the $send$-action could stay enabled even if an *ok*-signal was received.

Problem

Consider: $\tau^{<1}_{>0}$

1. $\tau^{<1}_{>0} \stackrel{d}{\not\longrightarrow}$ for all $d \in R^+$ because of the maximal progress assumption.

2. $\tau^{<1}_{>0} \stackrel{\tau}{\not\longrightarrow}$ because $\tau^{<1}_{>0}.E_1 + \tau^{<1}_{\geq 0}.E_2$ shall not be able to perform E_1.

<u>Solution:</u>

Extend the time domain R^+ by an additional *non-observable* time step (denoted by ι).

$\tau^{<1}_{>0} \xrightarrow{\iota} \tau^{<1}_{\geq 0}$

Resulting Problem

Consider: $a^{<1}_{>0}$

1. $a^{<1}_{>0} \xrightarrow{\iota}\xrightarrow{a}$ because $a^{<1}_{>0}|\bar{a}^{<1}_{>0} \xrightarrow{\iota}\xrightarrow{\tau}$.

2. $a^{<1}_{>0} \xrightarrow{\iota}\not\xrightarrow{a}$ since otherwise a can be observed at time point 0, which contradicts the specification of open intervals.

<u>Solution:</u> marked actions

$a^{<1}_{>0} \xrightarrow{\iota} \hat{a}^{<1}_{\geq 0}$

Marked actions are not observable and are only used for communication.

Extended Syntax

Act:

- Λ is the set of observable actions (a and $\bar{a}$).
- $\widehat{\Lambda} := \{\hat{c} \mid c \in \Lambda\}$ is the set of marked actions.
- $\mathrm{Act} := \Lambda \,\dot{\cup}\, \widehat{\Lambda} \,\dot{\cup}\, \{\tau\}$ is the set of all actions.

Relabelling functions (f): are those endofunctions on Act with

- $f^{-1}(\tau) = \{\tau\}$
- homomorphism for the complement $(^{-})$
- preserves marking $(\hat{\ })$ and unmarking

Operational Semantics

Transition systems with labels from $\mathrm{Act} \mathbin{\dot\cup} \mathrm{R}^+ \mathbin{\dot\cup} \{\iota\}$ are used.

For simplicity, only guarded expressions are allowed.

Nil:

$$\boxed{\dfrac{}{\mathrm{NIL} \xrightarrow{\mathrm{t}} \mathrm{NIL}}}$$

Action Prefix:

$$\boxed{\dfrac{0 \succ e \quad 0 \prec' e'}{\alpha^{\prec' e'}_{\succ\, e}.E \xrightarrow{\alpha} E}}$$

Operational Semantics (2)

Action Prefix (time):

$$\boxed{\frac{0 < e'}{\alpha_{\succ\, 0}^{\prec' e'}.E \xrightarrow{\iota} \hat{\alpha}_{\geq\, 0}^{\prec' e'}.E}}$$ (deadlocks are allowed)

where $\hat{}$ marks the action ($\hat{\tau} = \tau$)

$$\boxed{\frac{0 < e' \quad (e = 0 \Rightarrow \succ = \geq) \quad (\alpha = \tau \Rightarrow 0 < e)}{\alpha_{\succ\, e}^{\prec' e'}.E \xrightarrow{\iota} \alpha_{\succ\, e}^{\prec' e'}.E}}$$

$$\boxed{\frac{d \prec' e' \quad d \leq e}{\alpha_{\succ\, e}^{\prec' e'}.E \xrightarrow{d} \check{\alpha}_{\succ\, e-d}^{\prec' e'-d}.E}} \quad d \in R^+$$

where $\check{}$ unmarks the action

$$\boxed{\frac{d \prec' e' \quad \alpha \neq \tau \quad d > e}{\alpha_{\succ\, e}^{\prec' e'}.E \xrightarrow{d} \check{\alpha}_{\geq\, 0}^{\prec' e'-d}.E}} \quad d \in R^+$$

The action has to get unmarked, since it is expected that $a_{>0}^{\leq 3} \xrightarrow{\iota}\xrightarrow{1}\xrightarrow{a}$.

Comments

If we replaced the second action prefix time rule by the following rule

$$\frac{0 < e' \quad (e = 0 \Rightarrow \succ = \geq) \quad (\alpha = \tau \Rightarrow 0 < e)}{\alpha^{\prec' e'}_{\succ\, e}.E \stackrel{\iota}{\longrightarrow} \hat{\alpha}^{\prec' e'}_{\succ\, e}.E}$$

(marking α),

the process $\left(\tau^{\leq 1}_{>0}.b^{\leq 0}_{\geq 0}\right)\Big| a^{\leq 1}_{\geq 0}$ would not be able to perform the a-action after the b-action at the same time. (counterintuitive)

$$\left(\tau^{\leq 1}_{>0}.b^{\leq 0}_{\geq 0}\right)\Big| a^{\leq 1}_{\geq 0} \stackrel{\iota}{\longrightarrow}\stackrel{\tau}{\longrightarrow} b^{\leq 0}_{\geq 0}\Big|\hat{a}^{\leq 1}_{\geq 0} \stackrel{b}{\longrightarrow}\stackrel{a}{\not\longrightarrow}$$

Operational Semantics (3)

Parallel:

$$\boxed{\frac{E \xrightarrow{\alpha} E'}{E|F \xrightarrow{\alpha} E'|F}}$$ and symmetrically

$$\boxed{\frac{E \xrightarrow{\alpha_1} E' \quad F \xrightarrow{\alpha_2} F' \quad \check{\check{\alpha}}_1 = \check{\alpha}_2 \quad \alpha_i \neq \tau}{E|F \xrightarrow{\tau} E'|F'}}$$

$$\boxed{\frac{E \xrightarrow{t} E' \quad F \xrightarrow{t} F'}{E|F \xrightarrow{t} E'|F'} \; \Phi(t, E|F)}$$

where $\Phi(t, E')$ denotes that no internal action is possible for the process E' during the time interval $[0, t)$.

Upper Time Limit

$| \; |_{\mathcal{T}_\mathrm{c}}$: Guarded Expr. $\to ((R^+ \dot\cup \{\iota\}) \times \{<, \leq\}) \dot\cup \{0, \infty\}$

$$
\begin{aligned}
&|\mathrm{NIL}|_{\mathcal{T}_\mathrm{c}} = \infty \\
&\left|\alpha_{\succ e}^{\prec' e'}.E\right|_{\mathcal{T}_\mathrm{c}} = \begin{cases} 0 & \text{if } e' = 0 \\ \infty & \text{if } e' = \infty \\ (e', \prec') & \text{otherwise} \end{cases} \\
&|E + E'|_{\mathcal{T}_\mathrm{c}} = \max(|E|_{\mathcal{T}_\mathrm{c}}, |E'|_{\mathcal{T}_\mathrm{c}}) \\
&|E|E'|_{\mathcal{T}_\mathrm{c}} = \min(|E|_{\mathcal{T}_\mathrm{c}}, |E'|_{\mathcal{T}_\mathrm{c}}) \\
&|E \backslash a|_{\mathcal{T}_\mathrm{c}} = |E|_{\mathcal{T}_\mathrm{c}} \\
&|E[f]|_{\mathcal{T}_\mathrm{c}} = |E|_{\mathcal{T}_\mathrm{c}} \\
&|\mu X.E|_{\mathcal{T}_\mathrm{c}} = |E|_{\mathcal{T}_\mathrm{c}}
\end{aligned}
$$

Occurrences of the Internal Action

Sort : Guarded Expr. $\rightarrow ((\mathrm{R}_0^+ \dot\cup \{\iota\}) \rightarrow 2^{\Lambda \cup \{\tau\}})$

$$\mathrm{Sort}(\mathrm{NIL})_{(\mathrm{t})} = \emptyset$$

$$\mathrm{Sort}(\alpha^{\prec' \mathrm{e}'}_{\succ \mathrm{e}}.\mathrm{E})_{(\mathrm{t})} = \begin{cases} \{\check\alpha\} & \text{if } t \succ e \text{ and } t \prec' e' \\ \emptyset & \text{otherwise} \end{cases}$$

$$\mathrm{Sort}(\mathrm{E} + \mathrm{E}')_{(\mathrm{t})} = \mathrm{Sort}(\mathrm{E})_{(\mathrm{t})} \cup \mathrm{Sort}(\mathrm{E}')_{(\mathrm{t})}$$

$$\mathrm{Sort}(\mathrm{E}|\mathrm{E}')_{(\mathrm{t})} = \begin{cases} I & \text{if } (t, \leq) \leq |E|E'|_{\mathcal{T}_\mathrm{c}} \\ \emptyset & \text{otherwise} \end{cases}$$

where $\mathrm{I} = \mathrm{Sort}(\mathrm{E})_{(\mathrm{t})} \cup \mathrm{Sort}(\mathrm{E}')_{(\mathrm{t})}$

$$\cup \begin{cases} \{\tau\} & \text{if } \mathrm{Sort}(\mathrm{E})_{(\mathrm{t})} \cap \overline{\mathrm{Sort}(\mathrm{E}')_{(\mathrm{t})}} \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

$$\mathrm{Sort}(\mathrm{E}\backslash \mathrm{a})_{(\mathrm{t})} = \mathrm{Sort}(\mathrm{E})_{(\mathrm{t})} \backslash \{\mathrm{a}, \bar{\mathrm{a}}\}$$

$$\mathrm{Sort}(\mathrm{E}[\mathrm{f}])_{(\mathrm{t})} = \mathrm{f}(\mathrm{Sort}(\mathrm{E})_{(\mathrm{t})})$$

$$\mathrm{Sort}(\mu \mathrm{X}.\mathrm{E})_{(\mathrm{t})} = \mathrm{Sort}(\mathrm{E})_{(\mathrm{t})}$$

<u>Formula:</u> $\Phi : (R^+ \dot\cup \{\iota\}) \times$ Guarded Expr.

$$\Phi(t, E) = \forall t' \in (R_0^+ \dot\cup \{\iota\}) : \ t' < t \Rightarrow \tau \notin \mathrm{Sort}(\mathrm{E})_{(\mathrm{t}')}$$

Operational Semantics (4)

<u>Choice:</u>

$$\boxed{\frac{E \xrightarrow{\alpha} E'}{E + F \xrightarrow{\alpha} E'}}$$ and symmetrically

$$\boxed{\frac{E \xrightarrow{t} E'}{E + F \xrightarrow{t} E'} \ (|F|_{\mathcal{T}_c} < (t, \leq)) \wedge \Phi(t, F)}$$ and sym.

where $|F|_{\mathcal{T}_c}$ denotes the upper time limit to which F can pass (with respect to open intervals).

$$\boxed{\frac{E \xrightarrow{t} E' \quad F \xrightarrow{t} F'}{E + F \xrightarrow{t} E' + F'}}$$

Operational Semantics (5)

Restriction:

$$\boxed{\dfrac{E \stackrel{\alpha}{\longrightarrow} E' \quad \check{\alpha} \notin \{a, \bar{a}\}}{E \backslash a \stackrel{\alpha}{\longrightarrow} E' \backslash a}} \quad \boxed{\dfrac{E \stackrel{t}{\longrightarrow} E'}{E \backslash a \stackrel{t}{\longrightarrow} E' \backslash a}}$$

Relabelling:

$$\boxed{\dfrac{E \stackrel{\alpha}{\longrightarrow} E'}{E[f] \stackrel{f(\alpha)}{\longrightarrow} E'[f]}} \quad \boxed{\dfrac{E \stackrel{t}{\longrightarrow} E'}{E[f] \stackrel{t}{\longrightarrow} E'[f]}}$$

Recursion:

$$\boxed{\dfrac{E[\mu X.E/X] \stackrel{\alpha}{\longrightarrow} E'}{\mu X.E \stackrel{\alpha}{\longrightarrow} E'}} \quad \boxed{\dfrac{E[\mu X.E/X] \stackrel{t}{\longrightarrow} E'}{\mu X.E \stackrel{t}{\longrightarrow} E'}}$$

Properties of Open Timed CCS

- Time Determinism
$$\left(E \stackrel{t}{\longrightarrow} E' \wedge E \stackrel{t}{\longrightarrow} E''\right) \Rightarrow E' = E''$$

- Time Additivity $(d, d' \in R^+)$
$$\left(E \stackrel{d}{\longrightarrow} F \wedge F \stackrel{d'}{\longrightarrow} E'\right) \iff E \stackrel{d+d'}{\longrightarrow} E'$$

$$\left(E \stackrel{\iota}{\longrightarrow} F \wedge F \stackrel{d}{\longrightarrow} E'\right) \iff E \stackrel{d}{\longrightarrow} E'$$

$$\left(E \stackrel{\iota}{\longrightarrow} E'' \wedge E'' \stackrel{\iota}{\longrightarrow} E'\right) \Longrightarrow E \stackrel{\iota}{\longrightarrow} E'$$

- Deadlocks (e.g. $a^{\leq 0}_{\geq 1}$)

- Maximal Progress

Equivalences

- strong bisimulation; considers marked actions as adequate (context observation)
- neglect marked actions
- abstract away from ι-steps
- abstract away from τ-steps
- combinations of the last three points

Equivalence (2)

Consider the equivalence which abstracts away from ι-steps and neglects branches which contain a marked action:

<u>Bisimulation definition:</u> $(E, F) \in B$ then
if $E \xrightarrow{\gamma} E'$ and $\gamma \in \Lambda \dot{\cup} \{\tau, \iota\} \dot{\cup} R^+$ then
$\exists F' : F \overset{\gamma}{\longmapsto} F' \wedge (E', F') \in B$
and symmetrically

where
$\overset{\iota}{\longmapsto} = (\xrightarrow{\iota})^*$
$\overset{d}{\longmapsto} = \xrightarrow{d}$
$\overset{\alpha}{\longmapsto} = \overset{\iota}{\longmapsto} \xrightarrow{\alpha} \overset{\iota}{\longmapsto}$

Unfortunately this equivalence fails to be a congruence. For example, $\tau^{\leq 1}_{>0} + a^{\leq 1}_{>0}.b^{\leq 0}_{\geq 0}$ and $\tau^{\leq 1}_{>0}$ are equivalent, but in parallel with $\bar{a}^{\leq 1}_{>0}$ the former may perform a b-action.

Priority

Open Timed CCS allows a dynamic priority of actions at the same point in time by using $\tau^{\leq e+1}_{>e}$.

<u>Example:</u> Consider $a^{\leq 7}_{\geq 2}.E + b^{\leq 6}_{\geq 2}.F$, where a has priority 0 and b has priority -2.

Then

$$a^{\leq 7}_{\geq 2}.E + \tau^{\leq 3}_{>2}.\left(a^{\leq 5}_{\geq 0}.E + \tau^{\leq 1}_{>0}.\left(a^{\leq 5}_{\geq 0}.E + b^{\leq 4}_{\geq 0}.F\right)\right)$$

has the desired behavior by abstracting away from ι-steps and internal actions.

Further Work

- find a model without extending the syntax (marked actions)

- possibility to enable actions only for communication $(a^{<1}_{\geq 0}.\hat{b}^{\leq 0}_{\geq 0}.)$

- extend the algebra with time variables

  ? $\tau(x)^{\leq 1}_{>0}.a^{\leq 0-x}_{\geq 0-x} \xrightarrow{\iota}\xrightarrow{\tau}\xrightarrow{a}$?

- establish equational laws

- extend other process algebras with the presented ideas