Read Operators and their Expressiveness In Process Algebra

Speaker: **Maria Rita Di Berardini**[1] F. Corradini[1]
W. Vogler[2]

[1]Dip. di Matematica e Informatica, Università di Camerino

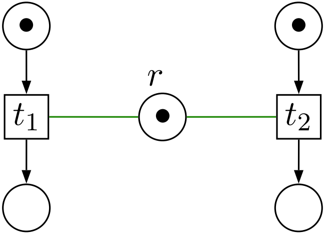
[2]Institut für Informatik, Universität Augsburg

Paco Meeting, Camerino 15 Settembre 2010

Non-blocking/non-consuming reading operations

Motivating idea: provide a way of modelling reading of resources without consuming them – multiple concurrent accesses to the some resource

Non-blocking reading is known from Petri nets in the form of **read arcs**



Non-blocking/non-consuming reading operations

Read arcs:

- have been used to model a variety of applications (transaction serialisability, concurrent constraint programming, cryptographic protocols, ...)
- add relevant expressivity (the MUTEX problem cannot be solved without read arcs)

Also note that non-blocking reading recall the notion of *persistence* that exists e.g. in several calculi for describing and analysing security protocols.

In this paper

We enhance PAFAS (a process algebra for asynchronous timed concurrent systems) with non-blocking reading

- read-action prefix operator (flexible, complex semantics) – PAFAS$_r$
- read-set prefix operator (simplier semantics, but with syntactic restrictions) – PAFAS$_s$

We also study the expressiveness of these read operators:

- added expressivity of read-action prefixes (w.r.t. fair behaviour)
- compare the expressivity of PAFAS$_s$ with those of PAFAS$_r$ and Petri nets with read-arcs
- it is still an open problem if PAFAS$_r$ is more expressive than PAFAS$_s$

A Process Algebra for Faster Asynchronous systems

Basic Assumption:

- Actions have an upper time bound (1 or 0) as a maximal delay
- Patient prefixes: $\alpha.P$ (delay 1), α is either a visible action or τ
- Urgent prefixes: $\underline{\alpha}.P$ (delay 0)
- Patient processes: $\underline{a}.P \stackrel{1}{\not\rightarrow}$ but $\underline{a}.P \parallel_a a.\mathsf{nil} \xrightarrow{1} \underline{a}.P \parallel_a \underline{a}.\mathsf{nil}$

PAFAS

$$P \;::=\; \mathsf{nil} \mid x \mid \alpha.P \mid P + P \mid P \parallel_A P \mid P[\Phi] \mid \mathsf{rec}\ x.P$$

$$Q \;::=\; P \mid \underline{\alpha}.P \mid Q + Q \mid Q \parallel_A Q \mid Q[\Phi] \mid \mathsf{rec}\ x.Q$$

Initial processes: $P, P', \ldots$ **General processes** $Q, Q', \ldots$

Transitional Semantics of PAFAS 1/2

Functional Behaviour: $Q \stackrel{\alpha}{\mapsto} Q'$ – Q evolves in Q' by performing α

$$\frac{\mu \in \{\alpha, \underline{\alpha}\}}{\mu.P \stackrel{\alpha}{\mapsto} P}$$

$$\frac{Q_1 \stackrel{\alpha}{\mapsto} Q'}{Q_1 + Q_2 \stackrel{\alpha}{\mapsto} Q'}$$

$$\frac{\alpha \notin A \quad Q_1 \stackrel{\alpha}{\mapsto} Q_1'}{Q_1 \|_A Q_2 \stackrel{\alpha}{\mapsto} Q_1' \|_A Q_2}$$

$$\frac{\alpha \in A \quad Q_1 \stackrel{\alpha}{\mapsto} Q_1' \quad Q_2 \stackrel{\alpha}{\mapsto} Q_2'}{Q_1 \|_A Q_2 \stackrel{\alpha}{\mapsto} Q_1' \|_A Q_2'}$$

other rules are as expected

Transitional Semantics of PAFAS 2/2

Temporal Behaviour: Q let pass 1 unit of time and becomes Q'

- $Q \xrightarrow{X}_r Q'$ conditional time step of duration 1
  $X \subseteq \mathbb{A}$ (visible actions) is a refusal set

$$\frac{}{\alpha.P \xrightarrow{X}_r \underline{\alpha}.P}$$

$$\frac{\alpha \notin X \cup \{\tau\}}{\underline{\alpha}.P \xrightarrow{X}_r \underline{\alpha}.P}$$

$$\frac{Q_i \xrightarrow{X}_r Q_i'}{Q_1 + Q_2 \xrightarrow{X}_r Q_1' + Q_2'}$$

$$\frac{Q_i \xrightarrow{X_i}_r Q_i' \quad X \subseteq (A \cap (X_1 \cup X_2)) \cup ((X_1 \cap X_2) \backslash A)}{Q_1 \parallel_A Q_2 \overset{\alpha}{\mapsto} Q_1' \parallel_A Q_2'}$$

other rules are as expected

Transitional Semantics of PAFAS 2/2

Temporal Behaviour: Q let pass 1 unit of time and becomes Q'

- $Q \xrightarrow{X}_r Q'$ conditional time step of duration 1
  $X \subseteq \mathbb{A}$ (visible actions) is a refusal set

$$\frac{}{\alpha.P \xrightarrow{X}_r \underline{\alpha}.P}$$

$$\frac{\alpha \notin X \cup \{\tau\}}{\underline{\alpha}.P \xrightarrow{X}_r \underline{\alpha}.P}$$

$$\frac{Q_i \xrightarrow{X}_r Q_i'}{Q_1 + Q_2 \xrightarrow{X}_r Q_1' + Q_2'}$$

$$\frac{Q_i \xrightarrow{X_i}_r Q_i' \quad X \subseteq (A \cap (X_1 \cup X_2)) \cup ((X_1 \cap X_2) \backslash A)}{Q_1 \|_A Q_2 \overset{\alpha}{\mapsto} Q_1' \|_A Q_2'}$$

Whenever $Q \xrightarrow{\mathbb{A}}_r Q'$, we write $Q \xrightarrow{1} Q'$ and call it a **1-step**

Adding non-blocking reading I (PAFAS$_r$)

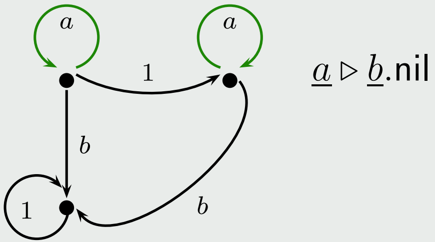
PAFAS +
Read-action prefix operator

$\mu \triangleright Q$ where $\mu \in \{\alpha, \underline{\alpha}\}$

behave as Q but can also been read with α (**read action**)

since being read does not change the state, α can be performed repeatedly

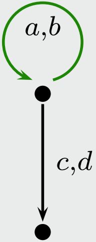
$a \triangleright b.\text{nil}$



The operational semantics of $\mu \triangleright Q$ needs two type of transition relations

Nested reading behaviours

$P = a \triangleright (b \triangleright c + d)$



Ordinary action transitions: $Q \overset{\alpha}{\mapsto} Q'$

Read action transitions: $Q \overset{\alpha}{\rightsquigarrow} Q'$
(due to nested read actions)

Basically:

- $P = a \triangleright (b \triangleright c + d) \overset{a}{\rightsquigarrow} P$
- $P' = b \triangleright c + d \overset{b}{\rightsquigarrow} P'$ and $P = a \triangleright P' \overset{b}{\rightsquigarrow} P$ performing a read action does not change the state, also in a choice context
- $P' = b \triangleright c + d \overset{c}{\mapsto}$ nil and $P = a \triangleright P' \overset{c}{\mapsto}$ nil (similarly for the ordinary d)

$$\overset{\alpha}{\longrightarrow} \; = \; \overset{\alpha}{\longmapsto} \; \cup \; \overset{\alpha}{\rightsquigarrow}$$

Transitional Semantics of $\mu \triangleright Q$

Functional behaviour

$$\frac{\mu \in \{\alpha, \underline{\alpha}\}}{\mu \triangleright Q \overset{\alpha}{\leadsto} \mu \triangleright Q}$$

$$\frac{Q \overset{\alpha}{\leadsto} Q'}{\mu \triangleright Q \overset{\alpha}{\leadsto} \mu \triangleright Q'}$$

$$\frac{Q \overset{\alpha}{\mapsto} Q'}{\mu \triangleright Q \overset{\alpha}{\mapsto} \mu \triangleright Q'}$$

Temporal behaviour

$$\frac{Q \xrightarrow{X}_r Q'}{\alpha \triangleright Q \xrightarrow{X}_r \underline{\alpha} \triangleright Q'}$$

$$\frac{Q \xrightarrow{X}_r Q' \quad \alpha \notin X \cup \{\tau\}}{\underline{\alpha} \triangleright Q \xrightarrow{X}_r \underline{\alpha} \triangleright Q'}$$

Adding non-blocking reading II (PAFAS$_s$)

PAFAS +
Read-set prefix operator

$$\{\mu_1, ..., \mu_n\} \triangleright Q$$

$\{\mu_1, ..., \mu_n\}$ contains all read actions currently enabled

We try to avoid nested reading

$\{a, b\} \triangleright c$	$a \triangleright (b \triangleright c)$
$\{a, b\} \triangleright (c + d)$	$a \triangleright (b \triangleright c + d)$

Functional behaviour: $\overset{\alpha}{\mapsto}$ ($= \xrightarrow{\alpha}$)

$$\frac{\mu_i \in \{\alpha, \underline{\alpha}\}}{\{\mu_1, ..., \mu_n\} \triangleright Q \overset{\alpha}{\mapsto} Q'}$$

$$\frac{Q \overset{\alpha}{\mapsto} Q'}{\{\mu_1, ..., \mu_n\} \triangleright Q \overset{\alpha}{\mapsto} Q'}$$

Proper terms = read-proper + rec-proper

Not all PAFAS$_s$ terms have a reasonable semantics:

- $P_1 = \{a\} \triangleright (\{b\} \triangleright c) \overset{b}{\mapsto} \{b\} \triangleright c$ instead of $\{a\} \triangleright (\{b\} \triangleright c)$
- $P_2 = (\{b\} \triangleright c) + a \overset{b}{\mapsto} \{b\} \triangleright c$ instead of $\{a\} \triangleright (\{b\} \triangleright c)$

We only consider **read-proper** terms:

1. for all subterms $\{\mu_1, \cdots, \mu_n\} \triangleright Q_1$, Q_1 is read-guarded
2. all subterm $Q_1 + Q_2$ are read-guarded

where **read-guarded** = read-set prefix operators are in the scope of some action-prefix: $a.\{b\} \triangleright c$ but not $\{b\} \triangleright c$ or $\{b\} \triangleright c + a$

Proper terms = read-proper + rec-proper

Not all PAFAS$_s$ terms have a reasonable semantics:

- $P_1 = \{a\} \triangleright (\{b\} \triangleright c) \overset{b}{\mapsto} \{b\} \triangleright c$ instead of $\{a\} \triangleright (\{b\} \triangleright c)$
- $P_2 = (\{b\} \triangleright c) + a \overset{b}{\mapsto} \{b\} \triangleright c$ instead of $\{a\} \triangleright (\{b\} \triangleright c)$

We only consider **read-proper** terms:

1. for all subterms $\{\mu_1, \cdots, \mu_n\} \triangleright Q_1$, Q_1 is read-guarded
2. all subterm $Q_1 + Q_2$ are read-guarded

where **read-guarded** = read-set prefix operators are in the scope of some action-prefix: $a.\{b\} \triangleright c$ but not $\{b\} \triangleright c$ or $\{b\} \triangleright c + a$

Proper terms = read-proper + rec-proper

$P_3 = \text{rec}\, x.\{a\} \triangleright b.(c + x)$ is read-proper, but $P_3 \xmapsto{b} c + P_3$ that is not read-proper because of P_3

P_3 is not **rec-proper**:

1. $\{a\} \triangleright b.(c + x)$ is not read-guarded, and
2. x is not guarded in $c + x$

Proper terms = read-proper + rec-proper

$P_3 = \text{rec } x.\{a\} \triangleright b.(c + x)$ is read-proper, but $P_3 \stackrel{b}{\mapsto} c + P_3$ that is not read-proper because of P_3

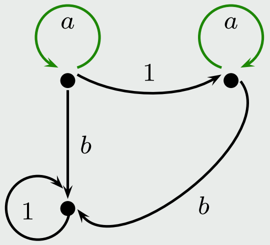
P_3 is not **rec-proper**:

1. $\{a\} \triangleright b.(c + x)$ is not read-guarded, and
2. x is not guarded in $c + x$

Expressivity of read prefixes

(Weak) Fairness of Actions: an action must be performed whenever countinuosly enabled in a run
Fair Traces: sequence of visible actions that occur in a transition sequence with infinitely many 1-step

$a \triangleright b$.nil



$$\mathsf{FairL}(a \triangleright b) = \{a^i b \mid i \geq 0\}$$

Theorem

If P is a finite state process with **no read-prefixes** and $sort(P) = \{a, b\}$ then $\mathsf{FairL}(P) \neq \{a^i b \mid i \geq 0\}$

Mapping PAFAS$_s$ into PAFAS$_r$

$[\![\_]\!]_s$: proper PAFAS$_s$ terms $\longrightarrow$ terms in PAFAS$_r$

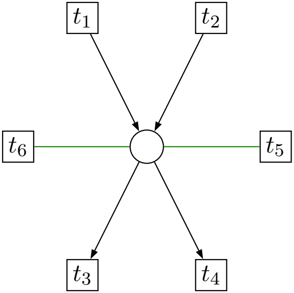
$[\![Q]\!]_s$ is Q where each subterm $\{\mu_1, \ldots, \mu_n\} \triangleright Q_1$ is replaced with $\mu_1 \triangleright \ldots \triangleright \mu_n \triangleright [\![Q_1]\!]_r$

$$[\![\{a\} \triangleright (\tau.\{b, c\} \triangleright d \parallel_\emptyset e)]\!]_s = a \triangleright (\tau.b \triangleright c \triangleright d \parallel_\emptyset e)$$

Theorem

For all proper Q in PAFAS$_s$, Q and $[\![Q]\!]_r$ are **(timed) bisimilar**

PAFAS$_s$ and Petri nets with read-arcs



$$P_0 = t_1.P_1 + t_2.P_1$$
$$P_1 = \{t_5, t_6\} \triangleright (t_3.P_0 + t_4.P_0)$$

Mapping PAFAS$_r$ into PAFAS$_s$

- We first exhibit a subset of PAFAS$_r$ terms that have an easy translation in PAFASs (terms in *Read Normal Form*, RNF)
- We also discuss how terms that are not in RNF can be normalised and the problems of such a normalisation

Terms in RNF

The set of terms in RNF is the image of the mapping function $[\![\_]\!]_r$

We only consider (sub)terms in PAFAS$_r$

1. $\mu_1 \triangleright \cdots \mu_n \triangleright Q_1$ where Q_1 is read-guarded
2. $Q_1 + Q_2$ read-guarded
3. rec $x.Q$ where Q is rec-proper (as above)

where **read-guarded** = all read-action prefix operators are in the scope of some action-prefix: $a.b \triangleright c$ but not $b \triangleright c$ or $b \triangleright c + a$

Translating terms in RNF

$[\![\_]\!]_s$: terms in RNF $\rightarrow$ proper PAFAS$_s$-terms

$[\![Q]\!]_s$ is Q where each subterm $\mu_1 \triangleright \ldots \triangleright \mu_n \triangleright Q_1$ is replaced with $\{\nu_1, \cdots, \nu_k\} \triangleright [\![Q_1]\!]_s$ (here $\{\nu_1, \cdots, \nu_k\}$ is a "proper" read-set)

$$[\![a \triangleright b \triangleright a \triangleright Q]\!]_s = \{a, b\} \triangleright [\![Q]\!]_s$$

$$[\![a \triangleright b \triangleright \underline{a} \triangleright Q]\!]_s = \{\underline{a}, b\} \triangleright [\![Q]\!]_s$$

Theorem

For each Q in RNF, Q and $[\![Q]\!]_s$ are **(timed) bisimilar**

Terms not in RNF

Basic Idea

Use laws to rewrite a term not in RNF into a bisimilar one in RNF
Ex: $(a \triangleright b) + c \sim a \triangleright (b + c)$

L1	$\mu \triangleright (\nu \triangleright Q) \sim \nu \triangleright (\mu \triangleright Q)$
L2	$\alpha \triangleright (\mu \triangleright Q) \sim \mu \triangleright Q$, $\underline{\alpha} \triangleright (\mu \triangleright Q) \sim \underline{\alpha} \triangleright Q$ provided that $\mu \in \{\alpha, \underline{\alpha}\}$
L3	$(\mu \triangleright Q) + R \sim \mu \triangleright (Q + R)$
L4	$a \triangleright (Q_1 \parallel_A Q_2) \sim ((a \triangleright Q_1) \parallel_{A \cup \{a\}} (a \triangleright Q_2))$, $\underline{a} \triangleright (Q_1 \parallel_A Q_2) \sim ((\underline{a} \triangleright Q_1) \parallel_{A \cup \{a\}} (\underline{a} \triangleright Q_2))$ prov. that $a \notin sort(Q)$
L5	$(\alpha \triangleright Q)[\Phi] \sim \Phi(\alpha) \triangleright (Q[\Phi])$, $(\underline{\alpha} \triangleright Q)[\Phi] \sim \underline{\Phi(\alpha)} \triangleright (Q[\Phi])$
L6	$(Q[\Phi])[\Psi] \sim Q[\Psi \circ \Phi]$
L7	rec $x.Q \sim Q\{$rec $x.Q/x\}$

Terms not in RNF

The idea of the translation into RNF is to perform rewriting by induction on the term size:

- action-prefix, parallel composition and relabeling preserve RNF
- read-prefixes $\mu \triangleright Q$ can be dealt with distributing μ among Q's components

  Ex: $a \triangleright (b \triangleright c \,\|_{\emptyset}\, d) \sim a \triangleright b \triangleright c \,\|_a\, a \triangleright d$
- choice and recursion still pose an unsolved problem

Normalising choices

How to normalise $Q + (R_1 \parallel_A R_2)$?

(1) if Q is deterministic (Ex: $Q = a.b + c.d$) we can use the law

$$Q + (R_1 \parallel_A R_2) \sim (Q + R_1) \parallel_{A \cup sort(Q)} (Q + R_2)$$

But if $Q = a.b + a.d$ (non-deterministic):

- $(Q + R_1) \parallel_{\{a,b,c,d\}} (Q + R_2) \overset{a}{\mapsto} b \parallel_{\{a,b,c,d\}} d$ while
- $Q + (R_1 \parallel_A R_2) \overset{a}{\mapsto} b$ or in c

Normalising choices

How to normalise $Q + (R_1 \parallel_A R_2)$?

(2) replace the secon copy of Q with the "top-part" of Q

The top-part of $Q = a.b + a.d$ is a and

$$Q + (R_1 \parallel_A R_2) \sim (Q + R_1) \parallel_{A \cup \{a\}} (a + R_2)$$

The top-part of $Q = a \triangleright b.c$ is $a \triangleright b$ and

$$Q + (R_1 \parallel_A R_2) \sim (Q + R_1) \parallel_{A \cup \{a,b\}} (a \triangleright b + R_2)$$

We need a suitable expansion law

But, what is the top-part of $Q = a \parallel_\emptyset b$? (note that $Q \not\approx a.b + b.a$)

- $Q \xrightarrow{1} \underline{a} \parallel_\emptyset \underline{b} \xmapsto{a} \mathsf{nil} \parallel_\emptyset \underline{b}$,
- $a.b + b.a \xrightarrow{1} \underline{a}.b + \underline{b}.a \xmapsto{a} b$

To solve this problem, a proper expansion law is needed

But even for the standard PAFAS this law is unknown

Conclusion and Future Work

We have studied two different way to enhance PAFAS with non-blocking reading actions

We also study the expressiveness of these read operators:

- it is still an open problem if PAFAS$_r$ is more expressive than PAFAS$_s$

In the future:

- we will try to complete this translation; this is related to finding an expansion law for generic PAFAS$_r$ (and PAFAS) terms
- This expansion law should also provide us with an axiomatisation for the full PAFAS language
- We plan to use read prefixes for modelling systems and comparing their efficiency or proving them correct under the progress assumption

Thank you for your attention