Reactive Processes

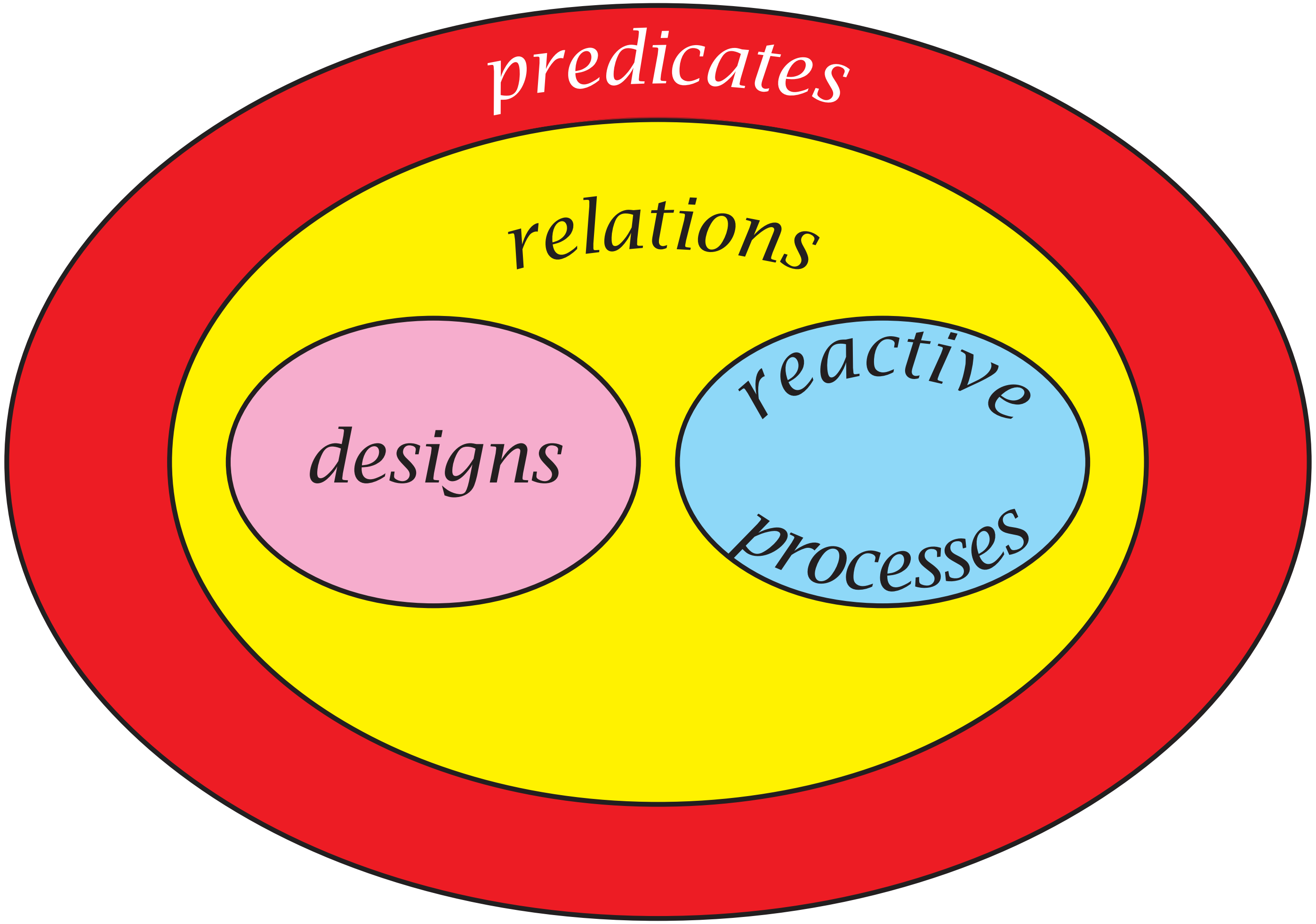
Jim Woodcock
University of York
May 2005

0. Introduction

1. *Reactive processes*
2. *Healthiness conditions*
2. *Exercises*

1. Reactive processes

- no longer just initial-final relation
- before-after, including intermediate states
- new observations for reactive processes: tr, ref, and $wait$
- healthiness conditions for reactive processes


predicates
relations
designs
reactive
processes

Observations

$okay \wedge \neg\, wait$	*started in a stable state*
$okay \wedge wait$	*not started, but in a stable state*
$\neg\, okay$	*not started, but in an unstable state*
$okay' \wedge \neg\, wait'$	*terminated*
$okay' \wedge wait'$	*in an intermediate state*
$\neg\, okay'$	*in an unstable state*

2. Healthiness conditions for reactive processes

Never undo

R1 $P \ = \ P \wedge tr \leq tr'$

Ignore history

R2 $P(tr, tr') \ = \ P(\langle\rangle, tr' - tr)$

Wait to start, preserving divergence

R3 $P \ = \ (I\!I_{rea} \lhd wait \rhd P)$

Another abbreviation

$P_b \mathrel{\widehat{=}} P[b/wait]$

recall that, for ***H1-H2*** predicates

$P = \neg P^f \vdash P^t$

a similar theorem exists for certain reactive processes involving

$\neg P^f_f \vdash P^t_f$

$L1 \quad (\mathbf{H1}(P))^c_b = \mathbf{H1}(P^c_b)$ ***H1****-wait-okay′*

Reactive $I\!I$

$$
\begin{aligned}
I\!I_{rea} \;\hat{=}\; & \neg\, okay \wedge tr \le tr' \\
& \vee \\
& okay' \wedge tr' = tr \wedge wait' = wait \wedge ref' = ref
\end{aligned}
$$

L2 $\quad I\!I_{rea} = \neg\, okay \wedge tr \le tr' \;\vee\; I\!I_{rel}$ $\qquad$ $I\!I_{rea}$-$I\!I_{rel}$

L3 $\quad I\!I_{rea} = I\!I_{rel} \lhd okay \rhd tr \le tr'$ $\qquad$ $I\!I_{rea}$-$I\!I_{rel}$-conditional

L4 $\quad okay \wedge I\!I_{rea} = okay \wedge I\!I_{rel}$ $\qquad$ $okay$-$I\!I_{rea}$-$I\!I_{rel}$

L5 $\quad okay \wedge I\!I_{rea} \,;\, P = okay \wedge P$ $\qquad$ $okay$-$I\!I_{rea}$-sequence-unit

L6 $\quad I\!I_{rea} = \mathbf{H2}(I\!I_{rea})$ $\qquad$ $I\!I_{rea}$-**H2**

L7 $\quad \mathbf{H1}(I\!I_{rea}) = \mathbf{H1}(I\!I_{rel})$ $\qquad$ **H1**-$I\!I_{rea}$-$I\!I_{rel}$

Proof of L6:

$$\boxed{\mathbf{H2}(I\!I_{rea})} \qquad \text{[J-splitting]}$$

$$= I\!I_{rea}{}^{f} \vee (I\!I_{rea}{}^{t} \wedge okay') \qquad [I\!I_{rea}]$$

$$= (\neg\, okay \wedge tr \leq tr')^{f} \qquad \text{[propositional calculus]}$$

$$\vee$$

$$(\neg\, okay \wedge tr \leq tr' \;\vee\; I\!I_{rel})^{t} \wedge okay'$$

$$= (\neg\, okay \wedge tr \leq tr') \vee (\neg\, okay \wedge tr \leq tr' \wedge okay') \qquad \text{[absorption]}$$

$$\vee$$

$$I\!I_{rel}{}^{t} \wedge okay'$$

$$= \neg\, okay \wedge tr \leq tr' \;\vee\; I\!I_{rel}{}^{t} \wedge okay' \qquad \text{[Leibniz]}$$

$$= \neg\, okay \wedge tr \leq tr' \;\vee\; okay' \wedge I\!I_{rel} \qquad [I\!I_{rea}]$$

$$= \boxed{I\!I_{rea}}$$

Healthiness condition ***R1***

never undo

$$P \;=\; P \wedge tr \leq tr'$$

$$\mathbf{R1}(P) \;=\; P \wedge tr \leq tr'$$

reactive processes can't change history

R1

P is ***R1*** healthy iff

$$P = P \wedge tr \leq tr'$$

$$\boldsymbol{R1}(P) = P \wedge tr \leq tr'$$

Laws for **R1**

L1	$\mathbf{R1} \circ \mathbf{R1} = \mathbf{R1}$	**R1**-idempotent
L2	$\mathbf{R1}(P \wedge Q) = \mathbf{R1}(P) \wedge \mathbf{R1}(Q)$	**R1**-$\wedge$
L3	$\mathbf{R1}(P \vee Q) = \mathbf{R1}(P) \vee \mathbf{R1}(Q)$	**R1**-$\vee$
L4	$\mathbf{R1}(P) \wedge Q = \mathbf{R1}(P \wedge Q)$	**R1**-extend-$\wedge$
L5	$\mathbf{R1}(P \triangleleft b \triangleright Q) = \mathbf{R1}(P) \triangleleft b \triangleright \mathbf{R1}(Q)$	**R1**-conditional
L6	$\mathbf{R1}(\neg\, \mathbf{R1}(P)) = \mathbf{R1}(\neg\, P)$	**R1**-negate-**R1**
L7	$(\mathbf{R1}(P))_b^c = \mathbf{R1}(P_b^c)$	**R1**-$wait$-$okay'$
L8	$\mathit{II}_{rel} = \mathbf{R1}(\mathit{II}_{rel})$	II_{rel}-**R1**
L9	$\mathit{II}_{rea} = \mathbf{R1}(\mathit{II}_{rea})$	II_{rea}-**R1**

*Proof that $I\!I_{rea}$ is **R1**-healthy*

$$
\begin{array}{ll}
\boxed{\mathbf{R1}(I\!I_{rea})} & [I\!I_{rea}\text{-conditional }] \\
= \mathbf{R1}(I\!I_{rel} \lhd okay \rhd tr \le tr') & [\mathbf{R1}\text{ conditional}] \\
= \mathbf{R1}(I\!I_{rel}) \lhd okay \rhd \mathbf{R1}(\mathbf{true}) & [\text{propositional calculus, }\mathbf{R1}] \\
= \mathbf{R1}(I\!I_{rel}) \lhd okay \rhd \mathbf{R1} \circ \mathbf{R1}(\mathbf{true}) & [\mathbf{R1}\text{ idempotent}] \\
= \mathbf{R1}(I\!I_{rel}) \lhd okay \rhd \mathbf{R1}(\mathbf{true}) & [\mathbf{R1}\text{, propositional calculus}] \\
= \mathbf{R1}(I\!I_{rel}) \lhd okay \rhd tr \le tr' & [I\!I_{rel}\text{-}\mathbf{R1}] \\
= I\!I_{rel} \lhd okay \rhd tr \le tr' & [I\!I_{rea}\text{ conditional}] \\
= \boxed{I\!I_{rea}} &
\end{array}
$$

Closure properties

Providing P and Q are ***R1*** healthy,

L10	$\mathbf{R1}(P \wedge Q) = P \wedge Q$	***R1***-$\wedge$-*closure*
L11	$\mathbf{R1}(P \vee Q) = P \vee Q$	***R1***-$\vee$-*closure*
L12	$\mathbf{R1}(P \lhd b \rhd Q) = P \lhd b \rhd Q$	***R1***-*conditional-closure*
L13	$\mathbf{R1}(P \,;\, Q) = P \,;\, Q$	***R1***-*sequence-closure*

Proof of sequence closure

$[\ (P \wedge x \preceq x' \ ;\ Q \wedge x \preceq x') \Rightarrow x \preceq x'\ ]$ *[sequence-transitive-relation]*

Proof of *L13*:

$\boxed{P\ ;\ Q}$ *[assumption: P and Q both **R1**]*

$= \mathbf{R1}(P)\ ;\ \mathbf{R1}(Q)$ *[**R1**, twice]*

$= P \wedge tr \leq tr'\ ;\ Q \wedge tr \leq tr'$ *[sequence-transitive-relation]*

$= (P \wedge tr \leq tr'\ ;\ Q \wedge tr \leq tr') \wedge tr \leq tr'$ *[**R1**, three times]*

$= \mathbf{R1}(\mathbf{R1}(P)\ ;\ \mathbf{R1}(Q))$ *[assumption: P and Q both **R1**]*

$= \boxed{\mathbf{R1}(P\ ;\ Q)}$

*A connection between **R1** and **H1***

$L14 \quad I\!I_{rea} = \mathbf{R1} \circ \mathbf{H1}(I\!I_{rea})$ $\qquad I\!I_{rea}$-**R1**-**H1**

Proof of *L14*:

$$\boxed{\mathbf{R1} \circ \mathbf{H1}(I\!I_{rea})} \qquad [\mathbf{H1}\text{-}I\!I_{rea}\text{-}I\!I_{rel}]$$

$$= \mathbf{R1} \circ \mathbf{H1}(I\!I_{rel}) \qquad [\mathbf{H1}]$$

$$= \mathbf{R1}(okay \Rightarrow I\!I_{rel}) \qquad [\text{propositional calculus}]$$

$$= \mathbf{R1}(\neg\, okay \vee I\!I_{rel}) \qquad [\mathbf{R1}\text{-}\vee\text{-distribution}]$$

$$= \mathbf{R1}(\neg\, okay) \vee \mathbf{R1}(I\!I_{rel}) \qquad [I\!I_{rel}\text{-}\mathbf{R1}]$$

$$= \mathbf{R1}(\neg\, okay) \vee I\!I_{rel} \qquad [\mathbf{R1}]$$

$$= \neg\, okay \wedge tr \le tr' \;\vee\; I\!I_{rel} \qquad [I\!I_{rea}]$$

$$= \boxed{I\!I_{rea}}$$

R1-H2 commutativity

L15 $\mathbf{H2} \circ \mathbf{R1} = \mathbf{R1} \circ \mathbf{H2}$ **R1**-**H2**-*commutativity*

Proof of Law *L15*:

$\quad \mathbf{H2} \circ \mathbf{R1}(P)$ [**R1**]

$= \mathbf{H2}(P \wedge tr \leq tr')$ [**H2**-$\wedge$-*non-okay*]

$= \mathbf{H2}(P) \wedge tr \leq tr'$ [**R1**]

$= \mathbf{R1} \circ \mathbf{H2}(P)$ $\square$

R2

ignore history

$$\boxed{\mathbf{R2a}(P(tr, tr')) \mathrel{\widehat{=}} \sqcap s \bullet P(s, s \frown (tr' - tr))}$$

$$\boxed{\mathbf{R2b}(P(tr, tr')) \mathrel{\widehat{=}} P(\langle\rangle, tr' - tr)}$$

$$\begin{aligned} & \mathbf{R2a}(tr = \langle a \rangle) \\ =\; & \sqcap s \bullet s = \langle a \rangle \\ =\; & \mathbf{true} \sqcap \mathbf{false} \\ =\; & \mathbf{true} \end{aligned}$$

$$\begin{aligned} & \mathbf{R2b}(tr = \langle a \rangle) \\ =\; & (tr = \langle a \rangle)[\langle\rangle, tr' - tr / tr, tr'] \\ =\; & (\langle\rangle = \langle a \rangle) \\ =\; & \mathbf{false} \end{aligned}$$

*Every **R2b** relation is **R2a***

$L1 \quad \mathbf{R2b} = \mathbf{R2a} \circ \mathbf{R2b}$ **R2b-R2a**

Proof of $L1$:

$$
\begin{array}{ll}
\boxed{\mathbf{R2a} \circ \mathbf{R2b}(P(tr, tr'))} & [\mathbf{R2b}] \\
= \mathbf{R2a}(P(\langle\,\rangle, tr' - tr)) & [\mathbf{R2a}] \\
= \sqcap s \bullet P(\langle\,\rangle, tr' - tr)(s, s \frown (tr' - tr)) & [\text{substitution}] \\
= \sqcap s \bullet P(\langle\,\rangle, s \frown (tr' - tr) - s) & [\text{property of } -] \\
= \sqcap s \bullet P(\langle\,\rangle, tr' - tr) & [\text{property of } \sqcap] \\
= P(\langle\,\rangle, tr' - tr) & [\mathbf{R2b}] \\
= \boxed{\mathbf{R2b}(P)} &
\end{array}
$$

*Every **R2a** relation is **R2b***

$L2 \quad \boldsymbol{R2a} = \boldsymbol{R2b} \circ \boldsymbol{R2a}$ ***R2a-R2b***

Proof of *L2*:

$$\boxed{ckeyR2b \circ \boldsymbol{R2a}(P(tr, tr'))} \quad [\boldsymbol{R2a}]$$

$$= \boldsymbol{R2b}(\sqcap s \bullet P(s, s \frown (tr' - tr))) \quad [\boldsymbol{R2b}]$$

$$= (\sqcap s \bullet P(s, s \frown (tr' - tr)))(\langle\,\rangle, tr' - tr) \quad [substitution]$$

$$= \sqcap s \bullet P(s, s \frown (tr' - tr) - \langle\,\rangle) \quad [property\ of\ -]$$

$$= \sqcap s \bullet P(s, s \frown (tr' - tr)) \quad [\boldsymbol{R2a}]$$

$$= \boxed{\boldsymbol{R2a}(P)}$$

R2

$\mathbf{R2} \mathrel{\widehat{=}} \mathbf{R2b}$

L3 $\mathbf{R2} \circ \mathbf{R2} = \mathbf{R2}$ ***R2**-idempotent*

Providing P and Q are ***R2*** healthy, and that $tr, tr' \notin \alpha b$

L4 $\mathbf{R2}(P \wedge Q) = P \wedge Q$ ***R2**-$\wedge$-closure*

L5 $\mathbf{R2}(P \vee Q) = P \vee Q$ ***R2**-$\vee$-closure*

L6 $\mathbf{R2}(P \lhd tr' = tr \rhd Q) = P \lhd tr' = tr \rhd Q$ ***R2**-cond-closure-1*

L7 $\mathbf{R2}(P \lhd b \rhd Q) = P \lhd b \rhd Q$ ***R2**-cond-closure-2*

L8 $\mathbf{R2}(P \mathbin{;} Q) = P \mathbin{;} Q$ ***R2**-sequence-closure*

*Lemma: **R2** sequence*

For a fresh variable u,

$$\mathbf{R2}(P\, ;\, Q) = (\, P(\langle\rangle, tr')\, ;\, Q(tr, tr' - u)\,)[tr/u]$$

Proof

$$\boxed{\mathbf{R2}(P\, ;\, Q)} \qquad \textit{[R2]}$$

$$= (P\, ;\, Q)(\langle\rangle, tr' - tr) \qquad \textit{[substitution]}$$

$$= (P\, ;\, Q)(\langle\rangle, tr' - u)[tr/u] \qquad \textit{[sequence substitution]}$$

$$= \boxed{(\, P(\langle\rangle, tr')\, ;\, Q(tr, tr' - u)\,)[tr/u]}$$

Proof of L8:

$$\boxed{\mathbf{R2}(P\ ;\ Q)}$$ *[**R2** sequence]*

$= (\,P(\langle\rangle, tr')\ ;\ Q(tr, tr' - u)\,)[tr/u]$ *[assumption: Q is **R2**]*

$= (\,P(\langle\rangle, tr')\ ;\ Q(\langle\rangle, tr' - tr)(tr, tr' - u)\,)[tr/u]$ *[substitution]*

$= (\,P(\langle\rangle, tr')\ ;\ Q(\langle\rangle, tr' - (u \frown tr))\,)[tr/u]$ *[leading assign]*

$= (\,P(\langle\rangle, tr')\ ;\ tr := u \frown tr\ ;\ Q(\langle\rangle, tr' - tr)\,)[tr/u]$ *[following assign]*

$= (\,P(\langle\rangle, tr' - u)\ ;\ Q(\langle\rangle, tr' - tr)\,)[tr/u]$ *[substitution]*

$= P(\langle\rangle, tr' - tr)\ ;\ Q(\langle\rangle, tr' - tr)$ *[assumption: P and Q are **R2**]*

$= \boxed{P\ ;\ Q}$

More laws of ***R2***

L9	$\mathbf{R2}(P\,;\mathbf{R2}(Q)) = \mathbf{R2}(P)\,;\mathbf{R2}(Q)$	**R2** *composition*
L10	$(\mathbf{R2}(P))_b^c = \mathbf{R2}(P_b^c)$	**R2**-*wait-okay'*
L11	$J = \mathbf{R2}(J)$	*J* **R2**
L12	$\mathbf{H1} \circ \mathbf{R2} = \mathbf{R2} \circ \mathbf{H1}$	*commutativity*-**R2**-**H1**
L13	$\mathbf{H2} \circ \mathbf{R2} = \mathbf{R2} \circ \mathbf{H2}$	*commutativity*-**R2**-**H2**
L14	$\mathbf{R1} \circ \mathbf{R2} = \mathbf{R2} \circ \mathbf{R1}$	*commutativity*-**R2**-**R1**

Proof of L13:

$\mathbf{R2} \circ \mathbf{H2}(P)$	[**H2**]
$= \mathbf{R2}(P \, ; J)$	[J **R2**]
$= \mathbf{R2}(P \, ; \mathbf{R2}(J))$	[**R2** composition]
$= \mathbf{R2}(P) \, ; \mathbf{R2}(J)$	[J **R2**]
$= \mathbf{R2}(P) \, ; J$	[**H2**]
$= \mathbf{H2} \circ \mathbf{R2}(P)$	

Proof of L14:

$\textbf{R2} \circ \textbf{R1}(P(tr, tr'))$	[**R1**, **R2**]
$= (P \wedge tr \leq tr')(\langle\rangle, tr' - tr)$	[substitution]
$= P(\langle\rangle, tr' - tr) \wedge \langle\rangle \leq tr' - tr$	[$\leq$ and $-$]
$= P(\langle\rangle, tr' - tr) \wedge tr \leq tr'$	[**R1**, **R2**]
$= \textbf{R1} \circ \textbf{R2}(P(tr, tr'))$	

R3

wait to start, preserving divergence

$P \;=\; okay' \land wait' \land tr' = tr$

$Q \;=\; okay' \land \neg\, wait' \land tr' = tr \frown \langle a \rangle$

$\boxed{P \,;\, Q}$ *[sequence]*

$= \exists\, okay_0, wait_0, tr_0, ref_0 \bullet$ *[definition]*

$\quad P[\,okay_0, wait_0, tr_0, ref_0 / okay', wait', tr', ref'\,] \land$

$\quad Q[\,okay_0, wait_0, tr_0, ref_0 / okay, wait, tr, ref\,]$

$= \exists\, okay_0, wait_0, tr_0, ref_0 \bullet$ *[one point]*

$\quad okay_0 \land wait_0 \land tr_0 = tr \land okay' \land \neg\, wait' \land tr' = tr_0 \frown \langle a \rangle$

$= \exists\, ref_0 \bullet okay' \land \neg\, wait' \land tr' = tr \frown \langle a \rangle$ *[predicate calculus]*

$= okay' \land \neg\, wait' \land tr' = tr \frown \langle a \rangle$ *[definition]*

$$= \boxed{Q}$$

Healthiness

$$\boxed{P = (I\!I_{rea} \lhd wait \rhd P)}$$

$$\mathbf{R3}(P) \;\widehat{=}\; (\, I\!I_{rea} \lhd wait \rhd P\,)$$

L1	$(\mathbf{R3}(P))_t = (I\!I_{rea})_t$	***R3**-wait-true*
L2	$(\mathbf{R3}(P))_f = P_f$	***R3**-not-wait-false*
L3	$(\mathbf{R3}(P))^c = ((I\!I_{rea})^c \lhd wait \rhd P^c)$	***R3**-okay′*

Closure properties

Provided P and Q are ***R3***

L4 $\quad \mathbf{R3}(P \land Q) = P \land Q$ *closure-∧-**R3***

L5 $\quad \mathbf{R3}(P \lor Q) = P \lor Q$ *closure-∨-**R3***

L6 $\quad \mathbf{R3}(P \lhd \_ \rhd Q) = P \lhd \_ \rhd Q$ *closure-_ ◁ _ ▷ _-**R3***

Provided P is ***R3***, and Q is ***R1*** and ***R3***

L7 $\quad \mathbf{R3}(P \,;\, Q) = P \,;\, Q$ *closure-;-**R3***

Idempotence and commutativity

L8	$\mathbf{R3} \circ \mathbf{R3} = \mathbf{R3}$	***R3**-idempotent*
L9	$\mathbf{H2} \circ \mathbf{R3} = \mathbf{R3} \circ \mathbf{H2}$	*commutativity-**R3**-**H2***
L10	$\mathbf{R1} \circ \mathbf{R3} = \mathbf{R3} \circ \mathbf{R1}$	*commutativity-**R3**-**R1***
L11	$\mathbf{R2} \circ \mathbf{R3} = \mathbf{R3} \circ \mathbf{R2}$	*commutativity-**R3**-**R2***

R3-H1-non-commutativity

Why don't ***R3*** and ***H1*** commute?

$$\boxed{\mathbf{H1} \circ \mathbf{R3}(P)} \qquad [\mathbf{H1}, \mathbf{R3}]$$

$$= okay \Rightarrow (I\!I_{rea} \lhd wait \rhd P) \qquad [\text{conditional}]$$

$$= (okay \Rightarrow I\!I_{rea}) \lhd wait \rhd (okay \Rightarrow P) \qquad [\mathbf{H1}]$$

$$= \mathbf{H1}(I\!I_{rea}) \lhd wait \rhd \mathbf{H1}(P) \qquad [I\!I_{rea} \text{ is not } \mathbf{H1}]$$

$$\neq I\!I_{rea} \lhd wait \rhd \mathbf{H1}(P) \qquad [\mathbf{R3}]$$

$$= \boxed{\mathbf{R3} \circ \mathbf{H1}(P)}$$

Consequences

L12 $\mathbf{H1} \circ \mathbf{R3} = \mathbf{H1} \circ \mathbf{R3} \circ \mathbf{H1}$ ***R3-H1*** *sub-commutativity*

L13 $\mathbf{R3} \circ \mathbf{R1} \circ \mathbf{H1} = \mathbf{R1} \circ \mathbf{H1} \circ \mathbf{R3}$ ***R3-H1-R1*** *sub-commutativity*

R

$$\boldsymbol{R \;\widehat{=}\; R1 \circ R2 \circ R3}$$

For reactive P

$L1 \quad (tr \leq tr') \,;\, P = tr \leq tr'$ *reactive-left-zero*

$L2 \quad I\!I_{rea} \,;\, P \;=\; P \lhd okay \rhd tr \leq tr'$ *restricted identity*

Proof of L1

first note that

$$\boxed{\mathbf{R3}(P)}$$

$$= I\!I_{rea} \lhd wait \rhd P \qquad [I\!I_{rea}]$$

$$= (I\!I_{rel} \lhd okay \rhd tr \le tr') \lhd wait \rhd P \qquad [\text{propositional calculus}]$$

$$= \boxed{(\neg\, okay \land wait \land tr \le tr') \lor (okay \land wait \land I\!I_{rel}) \lor (\neg\, wait \land P)}$$

Proof of L1

$$
\begin{array}{ll}
\boxed{tr \le tr';\ P} & \text{[assumption: } P \text{ is } \mathbf{R3}\text{]} \\
= tr \le tr';\ \mathbf{R3}(P) & \text{[}\mathbf{R3}(P) \text{ expansion]} \\
= tr \le tr';\ \neg\, okay \wedge wait \wedge tr \le tr' & \text{[right one point, } \mathit{I\!I}_{rel}\text{]} \\
\quad \vee\ tr \le tr';\ okay \wedge wait \wedge \mathit{I\!I}_{rel} & \\
\quad \vee\ tr \le tr';\ \neg\, wait \wedge P & \\
= (tr \le tr';\ tr \le tr') \vee (tr \le tr') \vee (tr \le tr';\ P) & \text{[sequence]} \\
= (tr \le tr') \vee (tr \le tr';\ \wedge P) & \text{[assumption: } P \text{ is } \mathbf{R1}\text{]} \\
= (tr \le tr') \vee (tr \le tr';\ \wedge P \wedge tr \le tr') & \text{[sequence transitivity]} \\
= tr \le tr' \vee ((tr \le tr';\ \wedge P \wedge tr \le tr') \wedge tr \le tr') & \text{[absorption]} \\
= \boxed{tr \le tr'} &
\end{array}
$$

Laws for $\boldsymbol{R}$

L3	$(\boldsymbol{R}(P))_f = \boldsymbol{R1} \circ \boldsymbol{R2}(P_f)$	$\boldsymbol{R}$*-wait-false*
L4	$(\boldsymbol{R}(P))_t = (I\!I_{rea})_t$	$\boldsymbol{R}$*-wait-true*
L5	$(\boldsymbol{R}(P))^c = ((I\!I_{rea})^c \lhd wait \rhd \boldsymbol{R1} \circ \boldsymbol{R2}(P^c))$	$\boldsymbol{R}$*-okay'*

Provided P and Q are $\boldsymbol{R}$-healthy

L6	$\boldsymbol{R}(P \land Q) = P \land Q$	*closure-*$\land$*-*$\boldsymbol{R}$
L7	$\boldsymbol{R}(P \lor Q) = P \lor Q$	*closure-*$\lor$*-*$\boldsymbol{R}$
L8	$\boldsymbol{R}(P \lhd tr' = tr \rhd Q) = P \lhd tr' = tr \rhd Q$	*closure-cond-*$\boldsymbol{R}$
L9	$\boldsymbol{R}(P\, ;\, Q) = P\, ;\, Q$	*closure-;-*$\boldsymbol{R}$

Summary

a ***reactive process*** is a relation satisfying the following

- alphabet contains $okay, wait, tr, ref, okay', wait', tr', ref'$
- alphabet contains $\mathcal{A}$ set of events ('process alphabet')
- fixed-point of monotonic idempotent $\boldsymbol{R} \mathrel{\widehat{=}} \boldsymbol{R1} \circ \boldsymbol{R2} \circ \boldsymbol{R3}$

3. Exercises

Specify the following reactive processes as predicates

1. *the deadlocked reactive process*

   the process that never diverges, never does anything, and never terminates

2. *show that the deadlocked reactive process is a left-zero for sequential composition of reactive processes*

3. *communicating reactive process*

   a reactive process that performs an "a" event

4. *imperative reactive process*

   a for-loop that performs an "a" event 10 times

Exercise 1: DEAD — the deadlocked reactive process

the process that never diverges, never does anything, and never terminates

$$okay' \wedge (tr' = tr) \wedge wait'$$

is it reactive?

- *never undoes past events:* ***R1***
- *ignores history:* ***R2***
- *doesn't wait to start, preserving divergence*
  - *refusal set left unconstrained*
  - *recovers from divergence*

make it ***R3****-healthy*

$$DEAD \mathrel{\widehat{=}} \mathbf{R3}(okay' \wedge (tr' = tr) \wedge wait')$$

Exercise 2: DEAD is a left zero for sequential composition

*suppose P is **R3***

$$\boxed{DEAD \,;\, P} \qquad [DEAD \text{ is } \mathbf{R3}]$$

$$= \mathbf{R3}(DEAD \,;\, P) \qquad [\mathbf{R3}]$$

$$= I\!I_{rea} \lhd wait \rhd (DEAD \,;\, P) \qquad [DEAD]$$

$$= I\!I_{rea} \lhd wait \rhd (DEAD \land okay' \land wait' \,;\, P) \qquad [\text{assumption: } P\ \mathbf{R3}]$$

$$= I\!I_{rea} \lhd wait \rhd (DEAD \land okay' \land wait' \,;\, \mathbf{R3}(P)) \qquad [\text{sequence, } \mathbf{R3}]$$

$$= I\!I_{rea} \lhd wait \rhd (DEAD \land okay' \land wait' \,;\, I\!I_{rea}) \qquad [I\!I_{rea}]$$

$$= I\!I_{rea} \lhd wait \rhd (DEAD \land okay' \land wait') \qquad [DEAD]$$

$$= I\!I_{rea} \lhd wait \rhd DEAD \qquad [\mathbf{R3}]$$

$$= \mathbf{R3}(DEAD) \qquad [DEAD \text{ is } \mathbf{R3}]$$

$$= \boxed{DEAD}$$

Exercise 3: communication

a reactive process that performs an "a" event

1. *either the trace stays constant or it is extended by "a"*
2. *if the trace is constant, then the process is waiting for interaction with the environment*
3. *if the trace is constant, then it must not be refusing "a"*
4. *if the trace is extended, then the process has terminated*
5. *it doesn't diverge*

for $a \in \mathcal{A}$

$$okay' \wedge (\, (tr' = tr \wedge a \notin ref') \lhd wait' \rhd (tr' = tr \frown \langle a \rangle) \,)$$

Doing an "a" event

$$okay' \land ((tr' = tr \land a \notin ref') \lhd wait' \rhd (tr' = tr \frown \langle a \rangle))$$

is it a reactive process?

- *is it **R1** healthy?*
- *is it **R2** healthy?*
- *is it **R3** healthy?*

$$perform(a) \;\widehat{=}\; \mathbf{R3}(okay' \land ((tr' = tr \land a \notin ref') \lhd wait' \rhd (tr' = tr \frown \langle a \rangle)))$$

Exercise 4: imperative reactive process

a for-loop that performs an "a" event 10 times

$\textbf{\textit{for}}\ i := 1\ \textbf{\textit{to}}\ 10\ \textbf{\textit{do}}\ perform(a)$

$\textbf{\textit{var}}\ i := 0;$

$\quad (i < 10) * (\, perform(a)_{+i};\ i := i + 1\,)$

$\textbf{\textit{end}}\ i$

is this process reactive?

- *is it **R1** healthy?*
- *is it **R2** healthy?*
- *is it **R3** healthy?*

Assignments

the program uses two design *assignments*

remember:* *no design is reactive

$$\textbf{\textit{var}}\ i;\quad \boxed{\textbf{\textit{R3}}(i := 0)}\quad (i < 10) * (\,\mathit{perform}(a)_{+i};\ \boxed{\textbf{\textit{R3}}(i := i + 1)}\,)\quad \textbf{\textit{end}}\ i$$

what else needs to be done?

Assignments

the program uses two *design* *assignments*

remember: ***no design is reactive***

$$
\begin{array}{l}
\textbf{\textit{var}}\ i;\\
\quad \textbf{\textit{R3}}(i := 0)\\
\quad (i < 10) * (\ \boxed{\textbf{\textit{R3}}(perform(a)_{+i})}\ ;\ \textbf{\textit{R3}}(i := i + 1)\ )\\
\textbf{\textit{end}}\ i
\end{array}
$$

can we simplfy this?