

CSP Processes

Ana Cavalcanti
Jim Woodcock
University of York
ICTAC School 2006

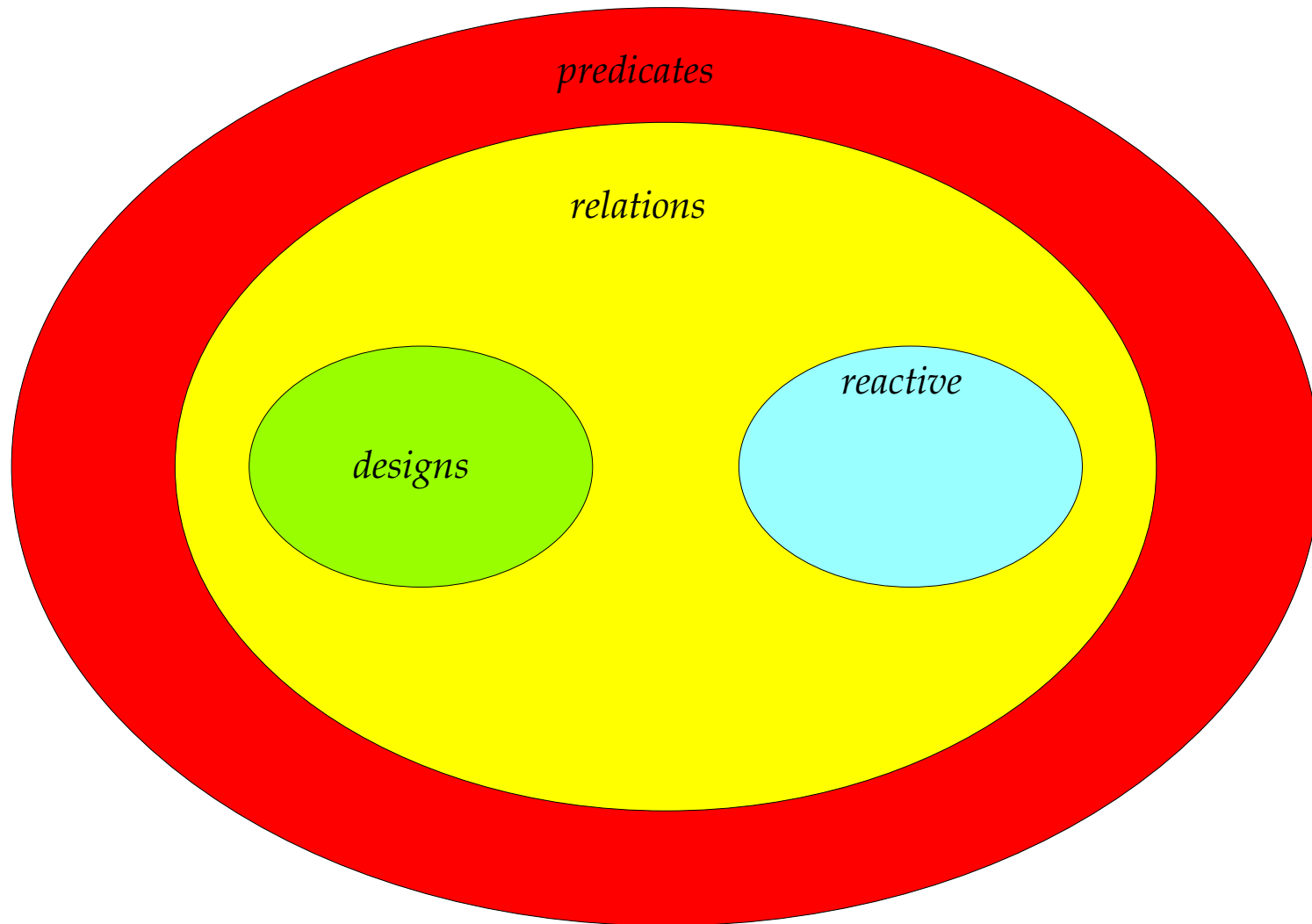
Agenda

- Lecture 1: Alphabetised relational calculus
- Lecture 2: Designs
- Lecture 3: Reactive processes
- Lecture 4: CSP processes

Agenda

- fundamental healthiness conditions
- extra healthiness conditions
- CSP processes as reactive designs
- failures divergences

Recap



Why do we need another theory?

Is the following a reactive process?

$$\mathbb{I}_{rea} \triangleleft wait \triangleright \neg okay \wedge tr' = tr \hat{\ } \langle a \rangle$$

Are we happy with that?

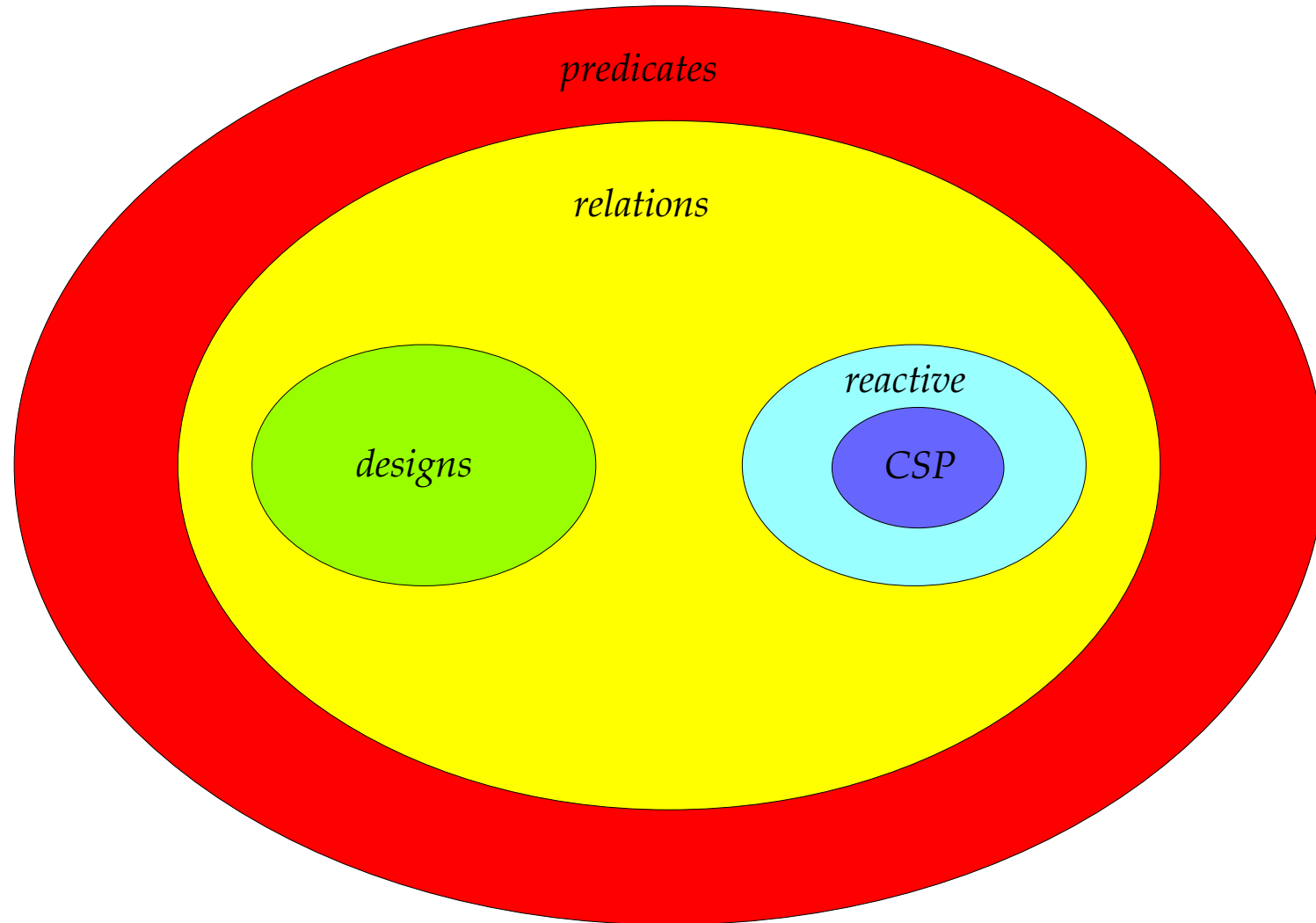
What about

$$\mathbb{I}_{rea} \triangleleft wait \triangleright \neg okay' \quad ?$$

CSP process

A reactive process that satisfies two other healthiness conditions: *CSP1* and *CSP2*.

CSP process



CSP1

$$P = P \vee (\neg \text{okay} \wedge \text{tr} \leq \text{tr}')$$

Idempotent

$$\mathbf{CSP1}(P) = P \vee \neg \text{okay} \wedge \text{tr} \leq \text{tr}'$$

What does a *CSP1* process look like?

$$\boxed{P}$$

$$[P \text{ is reactive and } \mathbf{CSP1}]$$

$$= \mathbf{CSP1} \circ \mathbf{R1} \circ \mathbf{R2} \circ \mathbf{R3}(P)$$

$$= \mathbf{CSP1} \circ \mathbf{R1} \circ \mathbf{R3}(\mathbf{R2}(P))$$

$$= \mathbf{CSP1} \circ \mathbf{R3}(\mathbf{R2}(P) \wedge tr \leq tr')$$

$$= \mathbf{CSP1}(\mathbf{I}_{rea} \triangleleft wait \triangleright \mathbf{R2}(P) \wedge tr \leq tr')$$

$$= \mathbf{CSP1}(\mathbf{I}_{rea} \triangleleft wait \triangleright \mathbf{R2}(P) \wedge tr \leq tr')$$

$$= (\mathbf{I}_{rea} \triangleleft wait \triangleright \mathbf{R2}(P) \wedge tr \leq tr') \vee \neg okay \wedge tr \leq tr'$$

What does a *CSP1* process look like?

$$\begin{aligned}
 & (\mathbf{I}_{rea} \triangleleft wait \triangleright \mathbf{R2}(P) \wedge tr \leq tr') \vee \neg okay \wedge tr \leq tr' \\
 = & (\mathbf{I}_{rel} \triangleleft wait \triangleright \mathbf{R2}(P) \wedge tr \leq tr') \vee \neg okay \wedge tr \leq tr' \\
 = & \neg okay \wedge tr \leq tr' \\
 & \vee \\
 & wait \wedge \mathbf{I}_{rel} \\
 & \vee \\
 & \neg wait \wedge \mathbf{R2}(P) \wedge tr \leq tr'
 \end{aligned}$$

What does a *CSP1* process look like?

$$= \neg \textit{okay} \wedge \textit{tr} \leq \textit{tr}'$$

$$\vee$$

$$\textit{okay} \wedge \textit{wait} \wedge \mathbf{I}_{rel}$$

$$\vee$$

$$\textit{okay} \wedge \neg \textit{wait} \wedge \mathbf{R2}(P) \wedge \textit{tr} \leq \textit{tr}'$$

$$= \neg \textit{okay} \wedge \textit{tr} \leq \textit{tr}'$$

$$\vee$$

$$\textit{okay} \wedge \textit{wait} \wedge \mathbf{I}_{rel}$$

$$\vee$$

$$\textit{okay} \wedge \neg \textit{wait} \wedge \mathbf{R1} \circ \mathbf{R2}(P)$$

Commutativity – *CSP1*

commutativity-*CSP1-R1*

$$CSP1 \circ R1 = R1 \circ CSP1$$

commutativity-*CSP1-R2*

$$CSP1 \circ R2 = R2 \circ CSP1$$

commutativity-*CSP1-R3*

$$CSP1 \circ R3 = R3 \circ CSP1$$

CSP1-R1-H1

$$CSP1(P) = R1 \circ H1(P) \quad \text{provided } P \text{ is } R1 \text{ healthy}$$

Closure – *CSP1*

closure- \wedge -*CSP1*

$CSP1(P \wedge Q) = P \wedge Q$ provided P and Q are *CSP1* healthy

closure- \vee -*CSP1*

$CSP1(P \vee Q) = P \vee Q$ provided P and Q are *CSP1* healthy

closure- \triangleleft $_$ \triangleright $_$ -*CSP1*

$CSP1(P \triangleleft _ \triangleright Q) = P \triangleleft _ \triangleright Q$ provided P and Q are *CSP1* healthy

closure- $;$ -*CSP1*

$CSP1(P ; Q) = P ; Q$ provided P and Q are *CSP1* healthy

CSP2 – H2

$$P = P ; J$$

$$J \hat{=} (okay \Rightarrow okay') \wedge wait' = wait \wedge tr' = tr \wedge ref' = ref$$

Idempotent

$$CSP2(P) = P ; J$$

Commutativity – *CSP2*

commutativity-*CSP2-H1*

$$CSP2 \circ H1 = H1 \circ CSP2$$

commutativity-*CSP2-R1*

$$CSP2 \circ R1 = R1 \circ CSP2$$

commutativity-*CSP2-R2*

$$CSP2 \circ R2 = R2 \circ CSP2$$

commutativity-*CSP1-R3*

$$CSP2 \circ R3 = R3 \circ CSP2$$

commutativity-*CSP2-CSP1*

$$CSP1 \circ CSP2 = CSP2 \circ CSP1$$

What does a *CSP1-CSP2* process look like?

$$\boxed{P}$$

$$= \mathit{CSP2} \circ \mathit{CSP1} \circ \mathit{R1} \circ \mathit{R2} \circ \mathit{R3}(P)$$

$$= \mathit{CSP1} \circ \mathit{R1} \circ \mathit{R2} \circ \mathit{R3} \circ \mathit{CSP2}(P)$$

$$= \dots$$

$$= \neg \mathit{okay} \wedge \mathit{tr} \leq \mathit{tr}'$$

$$\vee$$

$$\mathit{okay} \wedge \mathit{wait} \wedge \mathbf{I}_{rel}$$

$$\vee$$

$$\mathit{okay} \wedge \neg \mathit{wait} \wedge \mathit{R1} \circ \mathit{R2} \circ \mathit{CSP2}(P)$$

What does a *CSP1-CSP2* process look like?

$$= \neg \textit{okay} \wedge \textit{tr} \leq \textit{tr}'$$

$$\vee$$

$$\textit{okay} \wedge \textit{wait} \wedge \mathbf{I}_{rel}$$

$$\vee$$

$$\textit{okay} \wedge \neg \textit{wait} \wedge \mathbf{CSP2} \circ \mathbf{R1} \circ \mathbf{R2}(P)$$

$$= \neg \textit{okay} \wedge \textit{tr} \leq \textit{tr}'$$

$$\vee$$

$$\textit{okay} \wedge \textit{wait} \wedge \mathbf{I}_{rel}$$

$$\vee$$

$$\textit{okay} \wedge \neg \textit{wait} \wedge \mathbf{H2} \circ \mathbf{R1} \circ \mathbf{R2}(P)$$

What does a *CSP1-CSP2* process look like?

$$\begin{aligned} &= \neg \textit{okay} \wedge \textit{tr} \leq \textit{tr}' \\ &\vee \\ &\textit{okay} \wedge \textit{wait} \wedge \mathbf{I}_{rel} \\ &\vee \\ &\textit{okay} \wedge \neg \textit{wait} \wedge \mathbf{H1} \circ \mathbf{H2} \circ \mathbf{R1} \circ \mathbf{R2}(P) \end{aligned}$$

Closure – *CSP2*

closure- \vee -*CSP2*

$CSP2(P \vee Q) = P \vee Q$ provided P and Q are *CSP2* healthy

closure- \triangleleft $_ \triangleright$ $_$ -*CSP2*

$CSP2(P \triangleleft _ \triangleright Q) = P \triangleleft _ \triangleright Q$ provided P and Q are *CSP2* healthy

closure- $;$ -*CSP2*

$CSP2(P ; Q) = P ; Q$ provided Q is *CSP2* healthy

Question: what about \wedge ?

Designs

reactive-design-*CSP1*

$$\mathbf{R}(P \vdash Q) = \mathbf{CSP1}(\mathbf{R}(P \vdash Q))$$

reactive-design-*CSP2*

$$\mathbf{R}(P \vdash Q) = \mathbf{CSP2}(\mathbf{R}(P \vdash Q))$$

For every CSP process P ,

$$P = \mathbf{R}(\neg P_f^f \vdash P_f^t)$$

For every CSP process P ,

$$P = \mathbf{R}(\neg P_f^f \vdash P_f^t)$$

Proof

$$P$$

$$= \mathbf{R} \circ \mathbf{CSP1} \circ \mathbf{R} \circ \mathbf{CSP2}(P)$$

$$= \mathbf{R} \circ \mathbf{R1} \circ \mathbf{H1} \circ \mathbf{R} \circ \mathbf{CSP2}(P)$$

$$= \mathbf{R} \circ \mathbf{R1} \circ \mathbf{H1} \circ \mathbf{R} \circ \mathbf{H2}(P)$$

$$= \mathbf{R} \circ \mathbf{R1} \circ \mathbf{H1} \circ \mathbf{H2} \circ \mathbf{R}(P)$$

$$= \mathbf{R} \circ \mathbf{R1} \circ \mathbf{H1} \circ \mathbf{H2}(P)$$

$$= \mathbf{R} \circ \mathbf{H1} \circ \mathbf{H2}(P)$$

$$= \mathbf{R}(\neg \mathbf{H1} \circ \mathbf{H2}(P)^f \vdash \mathbf{H1} \circ \mathbf{H2}(P)^t)$$

$$\begin{aligned}
&= \mathbf{R}(\neg \mathbf{H1}(P)^f \vdash \mathbf{H1}(P)^t) \\
&= \mathbf{R}(\neg (okay \Rightarrow P)^f \vdash (okay \Rightarrow P)^t) \\
&= \mathbf{R}((okay \wedge \neg (okay \Rightarrow P^f)) \Rightarrow ((okay \Rightarrow P^t) \wedge okay')) \\
&= \mathbf{R}(\neg okay \vee P^f \vee \neg okay \wedge okay' \vee P^t \wedge okay') \\
&= \mathbf{R}(\neg okay \vee P^f \vee P^t \wedge okay') \\
&= \mathbf{R}(\neg P^f \vdash P^t) \\
&= \mathbf{R1} \circ \mathbf{R2}(\mathbf{I}_{rea} \triangleleft wait \triangleright (\neg P^f \vdash P^t)) \\
&= \mathbf{R1} \circ \mathbf{R2}(\mathbf{I}_{rea} \triangleleft wait \triangleright (\neg P^f \vdash P^t) \wedge \neg wait) \\
&= \mathbf{R1} \circ \mathbf{R2}(\mathbf{I}_{rea} \triangleleft wait \triangleright (\neg P_f^f \vdash P_f^t)) \\
&= \boxed{\mathbf{R}(\neg P_f^f \vdash P_f^t)}
\end{aligned}$$

CSP processes as reactive designs

- express all processes as reactive designs
- pre and postcondition precisely specify required process
- embeds assertional reasoning within theory of CSP
- parallel program development = sequential program development

$$STOP = \mathbf{R}(true \vdash tr' = tr \wedge wait')$$

CSP processes as reactive designs

- express all processes as reactive designs
- pre and postcondition precisely specify required process
- embeds assertional reasoning within theory of CSP
- parallel program development = sequential program development

$$STOP = \mathbf{R}(true \vdash tr' = tr \wedge wait')$$

$$SKIP = \mathbf{R}(true \vdash tr' = tr \wedge \neg wait')$$

CSP processes as reactive designs

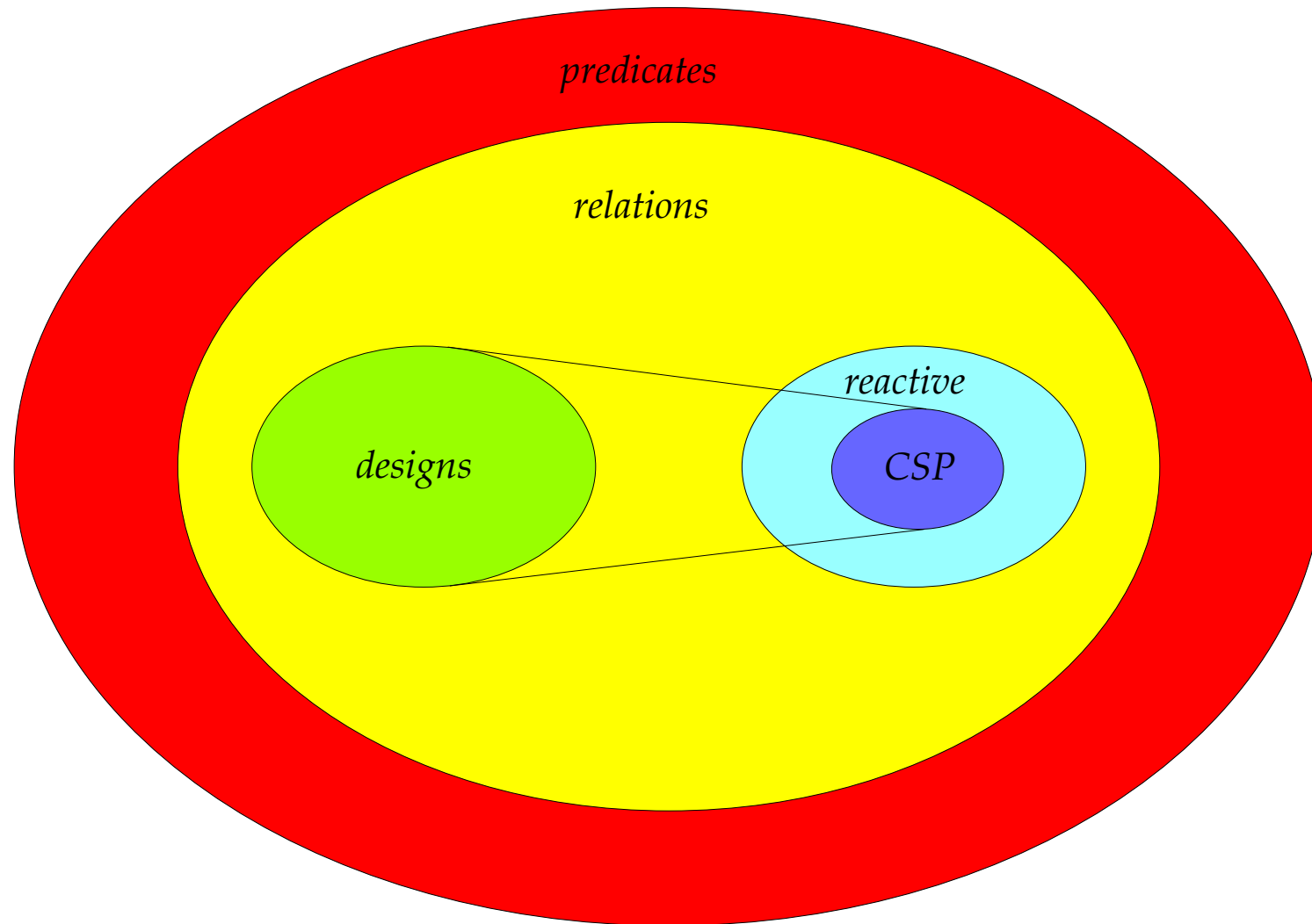
- express all processes as reactive designs
- pre and postcondition precisely specify required process
- embeds assertional reasoning within theory of CSP
- parallel program development = sequential program development

$$STOP = \mathbf{R}(true \vdash tr' = tr \wedge wait')$$

$$SKIP = \mathbf{R}(true \vdash tr' = tr \wedge \neg wait')$$

$$CHAOS = \mathbf{R}(false \vdash true)$$

CSP processes



Are *CSP1* and *CSP2* enough?

$$\boxed{SKIP ; P = P} \quad ?$$

$$\boxed{P ; SKIP = P} \quad ?$$

CSP3

$$P = \mathbf{SKIP} ; P$$

What does this mean?

$$\mathbf{SKIP} ; P$$

$$= \mathbf{R}(true \vdash tr' = tr \wedge \neg wait') ; P$$

$$= \mathbf{R}((true \vdash tr' = tr \wedge \neg wait') ; P)$$

$$= \mathbf{R}(\exists ref \bullet P)$$

$$= \mathbf{I}_{rea} \triangleleft wait \triangleright \exists ref \bullet P$$

Meaning of *CSP3*

P is *CSP3* if and only if $\neg wait \Rightarrow (P = \exists ref \bullet P)$

So, we cannot write processes like

$$\mathbf{R}(a \in ref \vdash \neg wait' \wedge tr' = tr \hat{\ } \langle b \rangle)$$

CSP₄

$$P = P ; \text{SKIP}$$

What does this mean?

$$P ; \text{SKIP}$$

$$\begin{aligned} &= (\exists \text{ref}' \bullet P) \wedge \text{okay}' \wedge \neg \text{wait}' \\ &\vee \\ &P \wedge \text{okay}' \wedge \text{wait}' \\ &\vee \\ &(P \wedge \neg \text{okay}') ; \text{tr} \leq \text{tr}' \end{aligned}$$

Meaning of *CSP*₄

So, we cannot write processes like

$$\mathbf{R}(true \vdash \neg wait' \wedge \{a, b\} \subseteq ref')$$

CSP5

$$P = P \parallel SKIP$$

$$P = \mathbf{R3}(P ; (true \vdash (ref' \subseteq ref))_{+\{tr,wait\}})$$

Question: what does it mean?

When do we stop thinking about healthiness conditions?

It all depends on what we want to do...

Are the models isomorphic?

- Not really...
- UTP top

$P \sqsubseteq R(\mathit{true} \vdash \mathit{false})$, for every CSP process P .

- Failures and divergences
 - No top
 - Every non-divergent deterministic process is maximal

Challenges

- Further healthiness conditions
- Reactive designs for all the constructors
- Time, resources, mobility, probability, ...
- Mechanisation
- Refinement laws

Assessment exercises

- Exercise 2, page 225
- Exercise 5, page 228
- Exercise 8, page 233
- Exercise 12, page 248
- Exercise 17, page 261