Interfaces as games, programs as strategies

Markus Michelbrink

Department of Computer Science
University of Wales, Swansea

TYPES 2004
Paris, December 2004

Interfaces

Due to Peter Hancock and Anton Setzer

An (state dependent) Interface is given by

- $\mathsf{S} : \mathsf{Set}$
  a set of states S,
- $\mathsf{C} : \mathsf{S} \to \mathsf{Set}$
  a set of commands $\mathsf{C}\ s$ for every state $s : \mathsf{S}$,
- $\mathsf{R} : (s : \mathsf{S}) \to \mathsf{C}\ s \to \mathsf{Set}$
  a set of responses $\mathsf{R}\ s\ c$ for every command $c : \mathsf{C}\ s$ and state $s : \mathsf{S}$,
- $\mathsf{nxt} : (s : \mathsf{S}) \to (c : \mathsf{C}\ s) \to \mathsf{R}\ s\ c \to \mathsf{S}$
  a function nxt giving a state $\mathsf{nxt}\ s\ c\ r$ for every response $r : \mathsf{R}\ s\ c$, every command $c : \mathsf{C}\ s$ and every state $s : \mathsf{S}$.

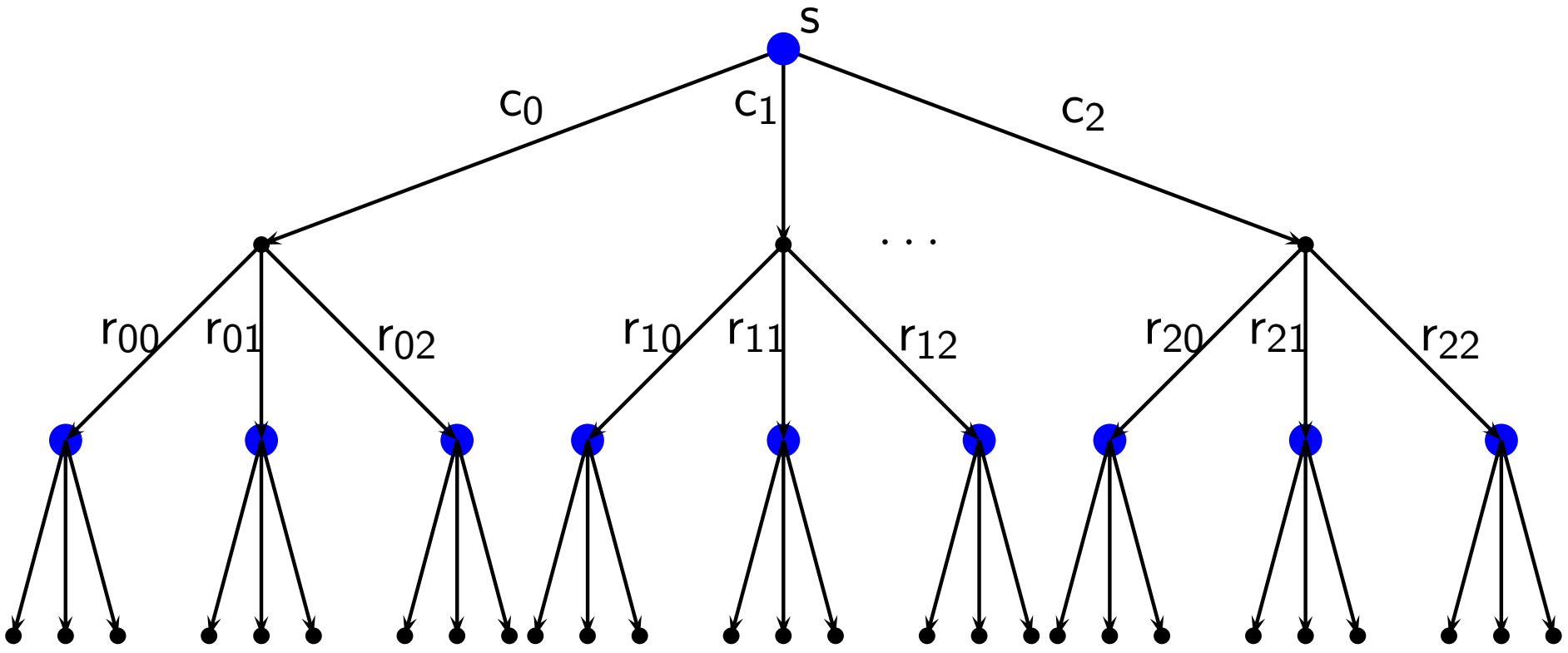
Programs

Due to Peter Hancock and Anton Setzer

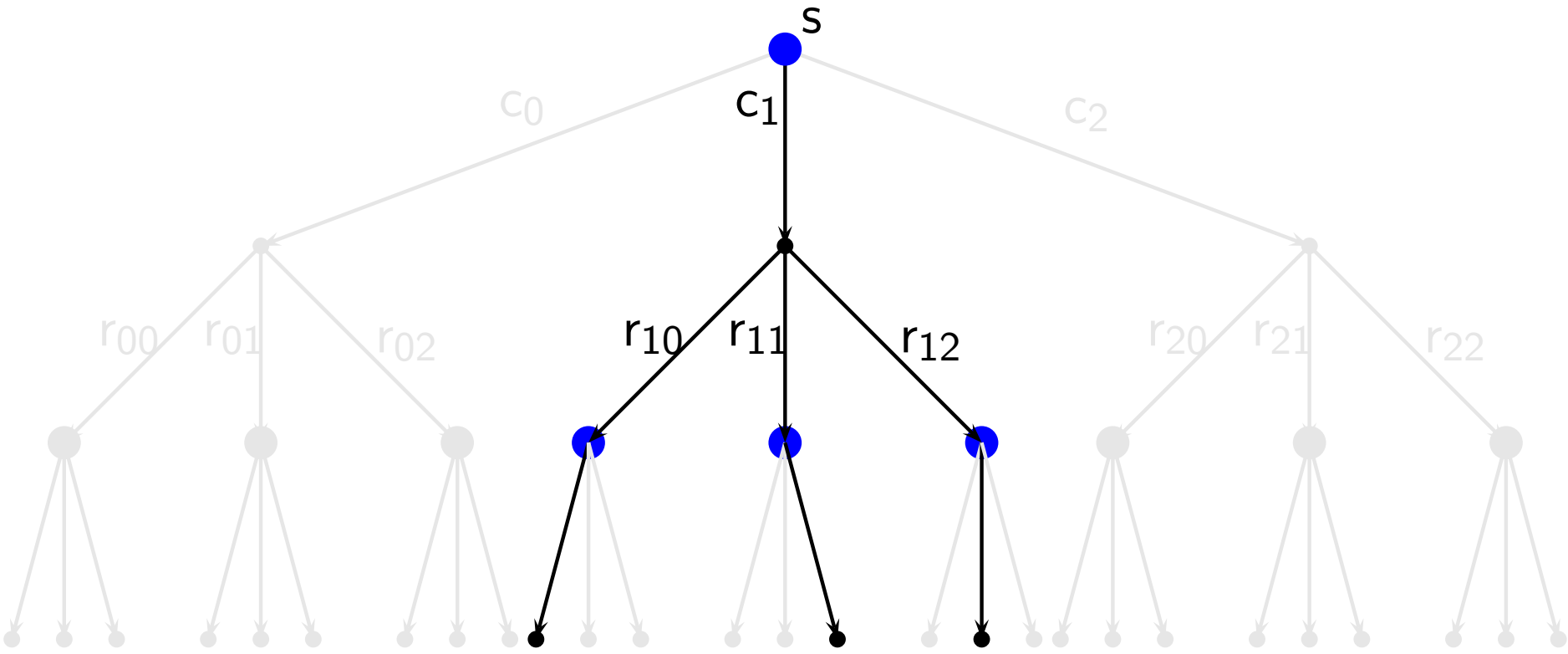
An interactive Program for such an Interface is given by

- IO : S $\longrightarrow$ Set
  a set IO s (of programs) for every state s : S,
- a function onestep
  giving for every program p : IO s and every state s : S a command c : C s and a function f : $(r$: R s $c)$ $\longrightarrow$ IO $(\mathsf{nxt}\ s\ c\ r)$ which gives for every response r to c a new program.
- an element p : IO s for some state s : S.

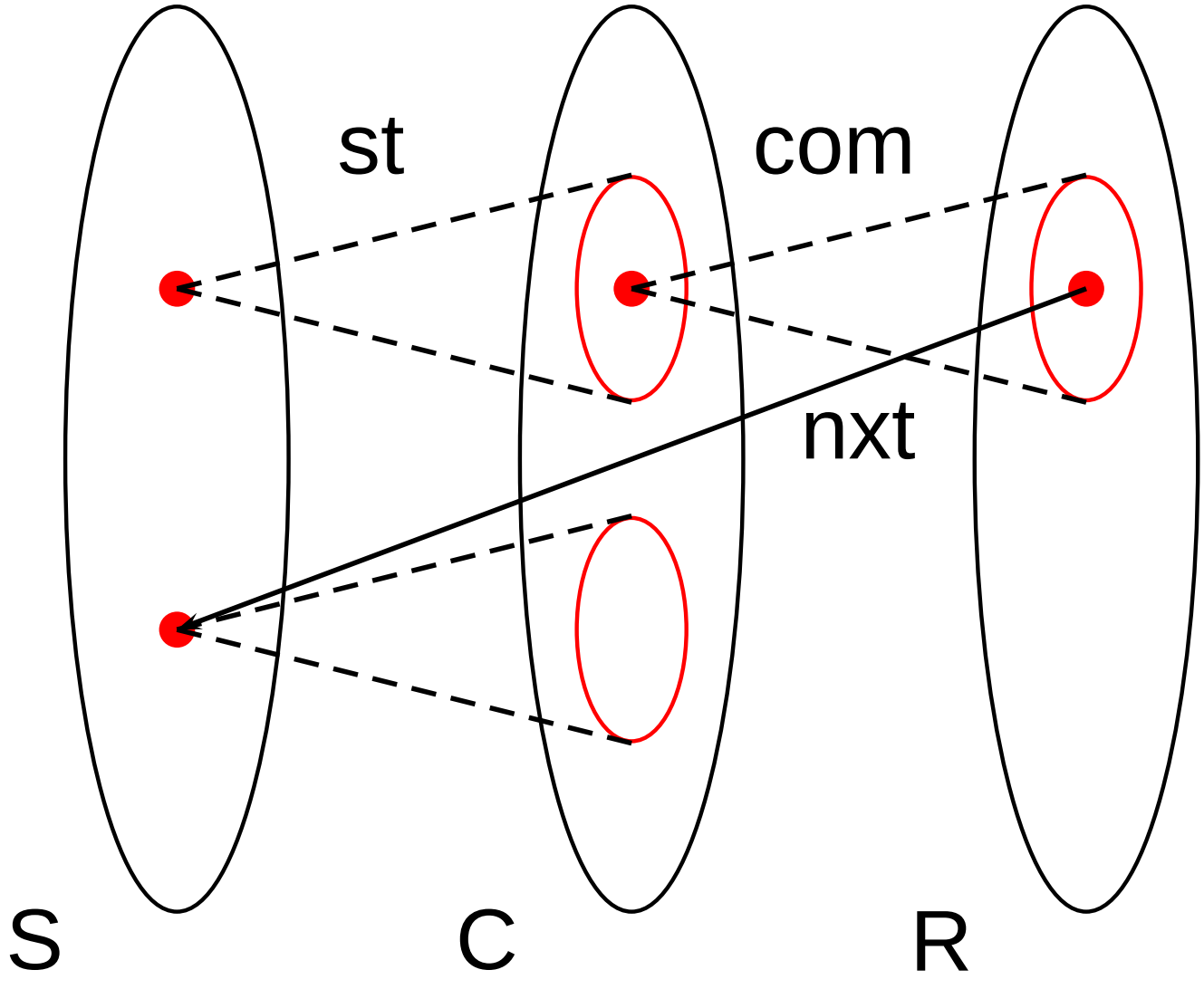
Interfaces



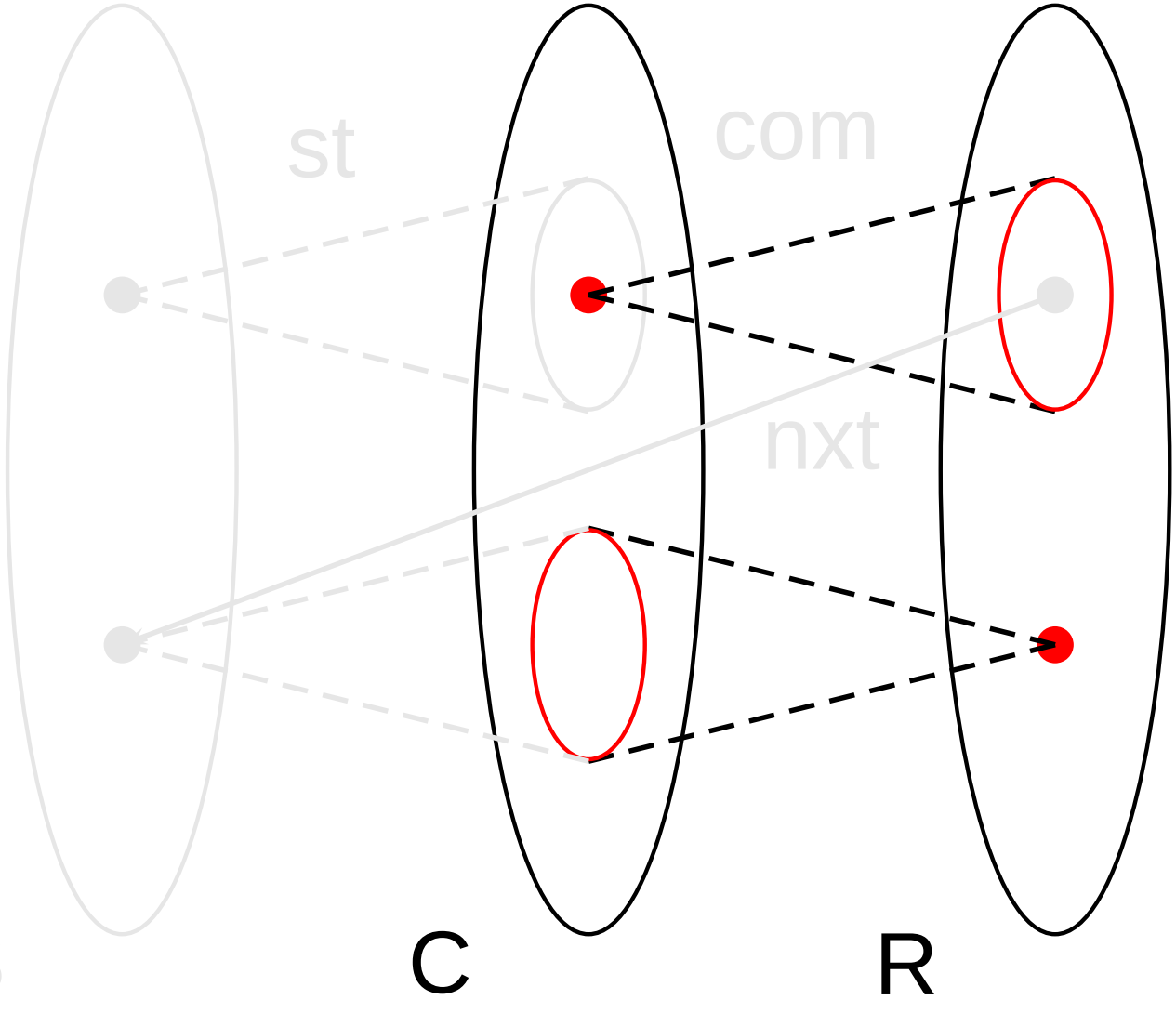
Programs



Motivation



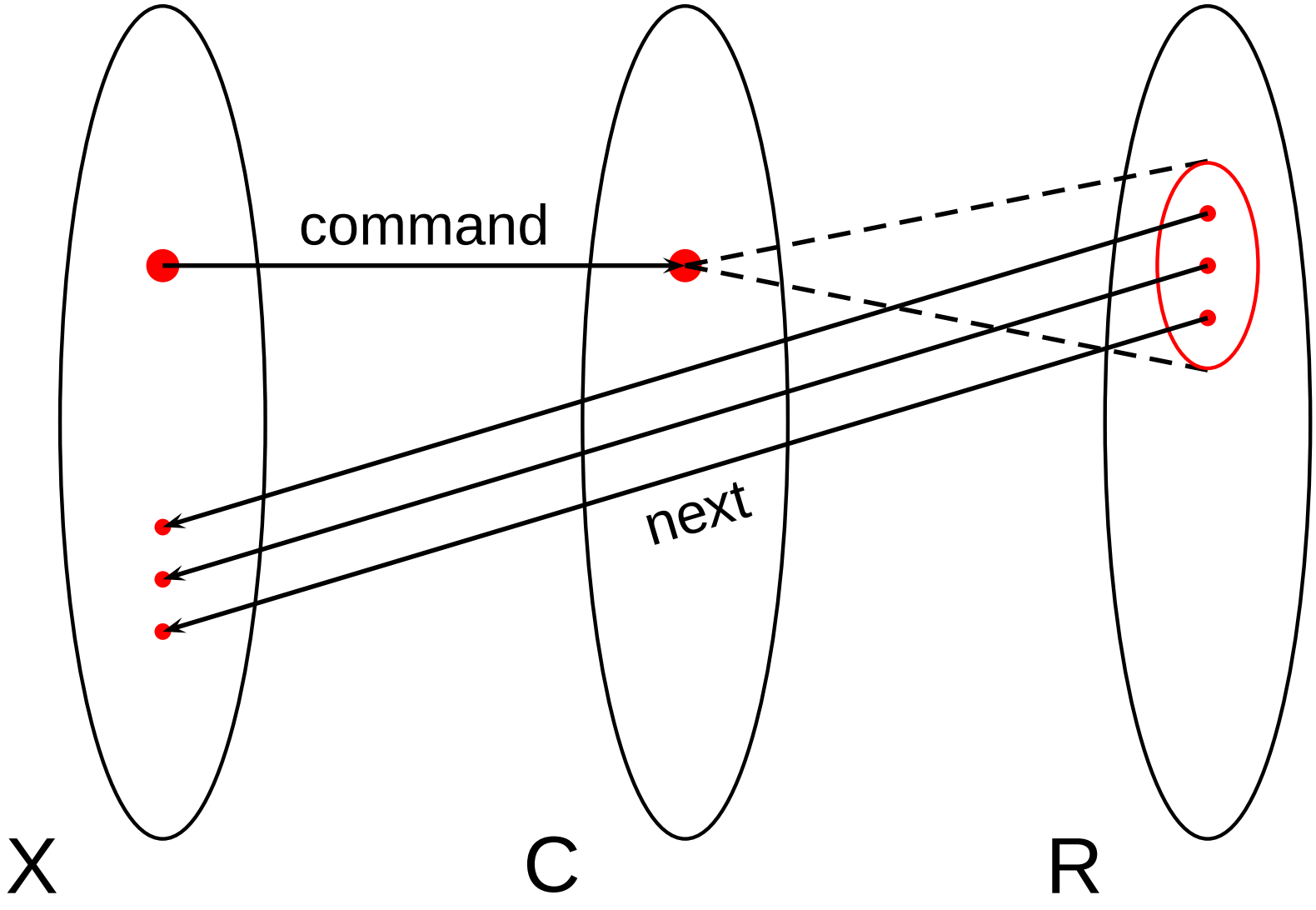
Motivation



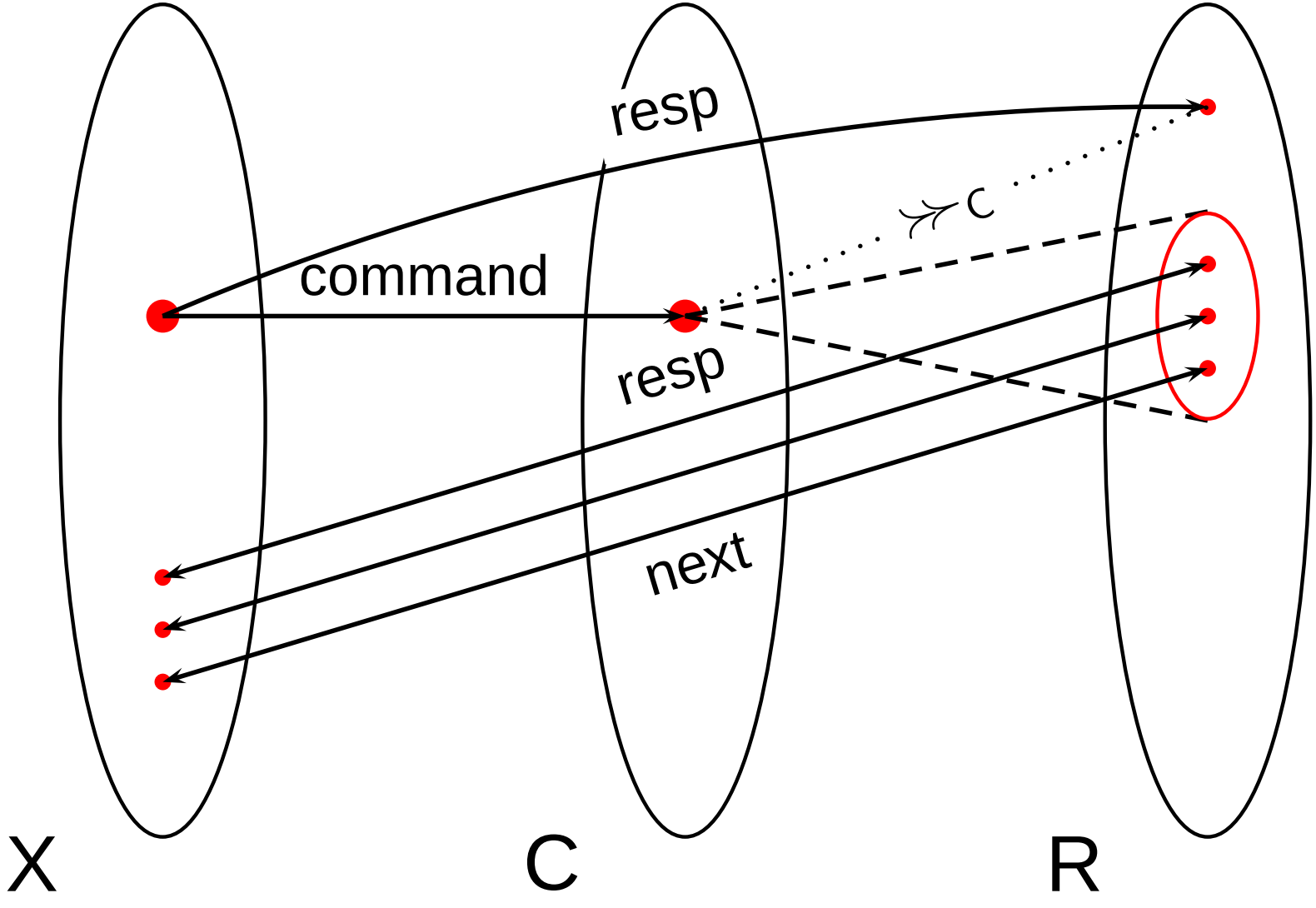
Interface/Game

- $C, R : \mathsf{Set}$
  a set of commands (players moves) and a set of responses (opponents moves),
- $\prec\!\!\prec_C : R \to C \to \mathsf{Set}$
  a relation between R and C,
- $\prec\!\!\prec_R : C \to R \to \mathsf{Set}$
  a relation between C and R.

Programs/Strategies



Soundness



Programs/Strategies

A program/strategy is given by

- X : Set
  a set X
- a function onestep
  giving for every $x : \mathsf{X}$ a command $c : \mathsf{C}$ and a function
  $f : (r : \mathsf{R}) \to c \prec\!\!\prec_{\mathsf{R}} r \to X$.

We write command x for c and next x for f.

A sound strategy is a strategy $(\mathsf{X}, \mathsf{onestep})$ and a function $\mathsf{resp} : \mathsf{X} \to \mathsf{R}$ such that for $x : \mathsf{X}$:

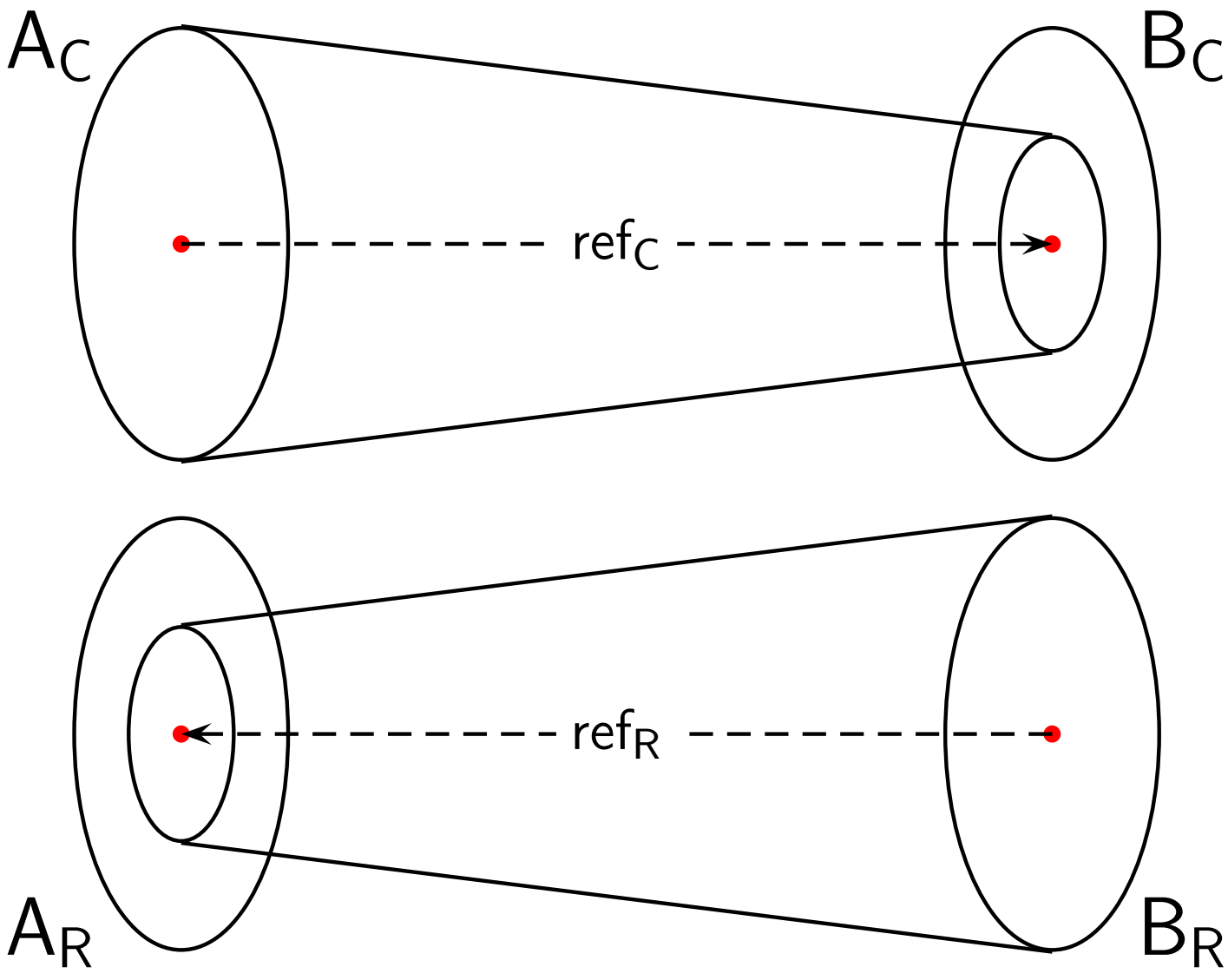
- $\mathsf{resp}\ x \prec\!\!\prec_{\mathsf{C}} \mathsf{command}\ x$
- $\mathsf{next}\ x\ r\ \mathsf{fitr} = r \quad$ for $\quad \mathsf{fitr} : \mathsf{command}\ x \prec\!\!\prec_{\mathsf{R}} r$.

Refinement

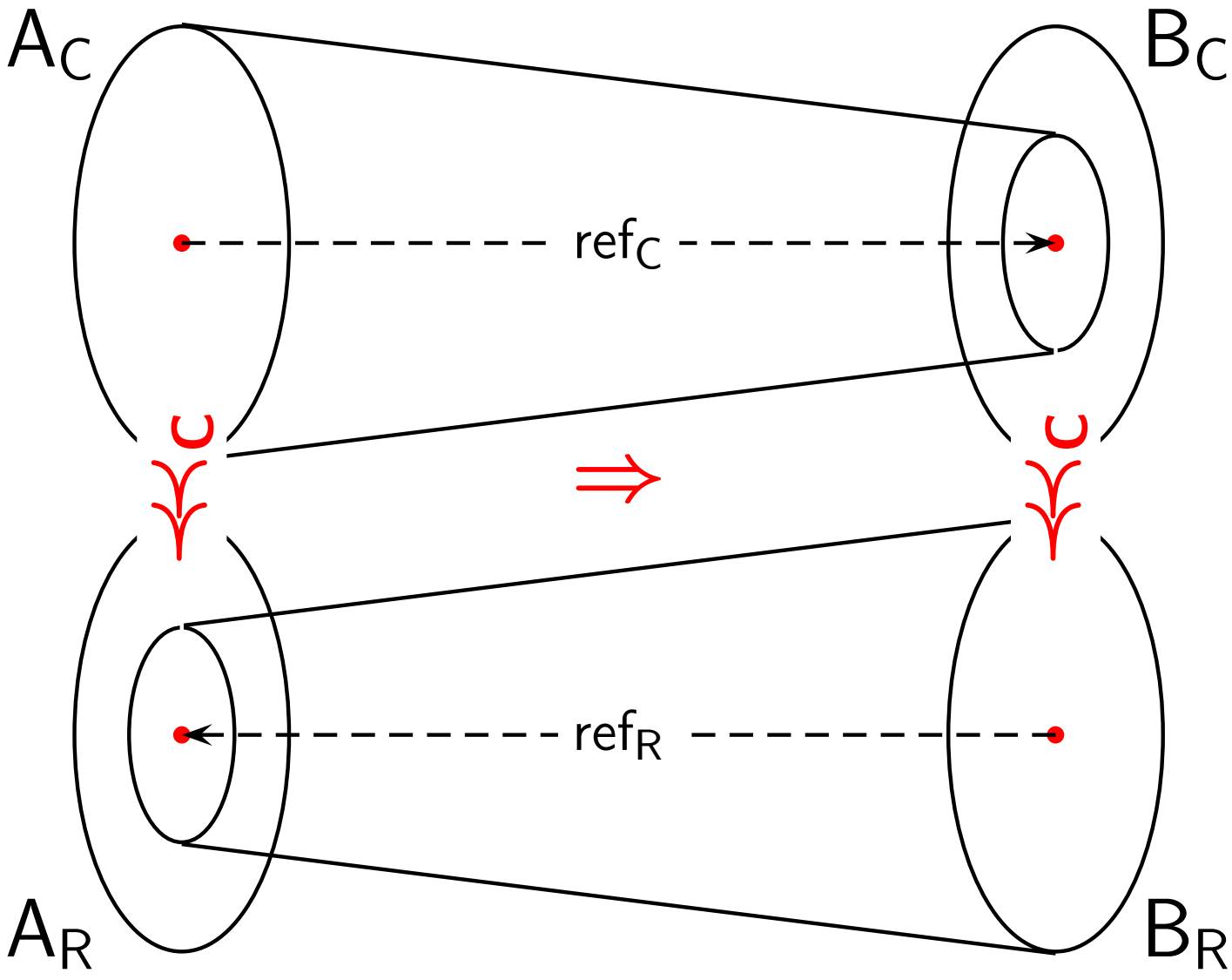
Idea: One side can do more,
the other side can do less

Let opponent have more freedom ...

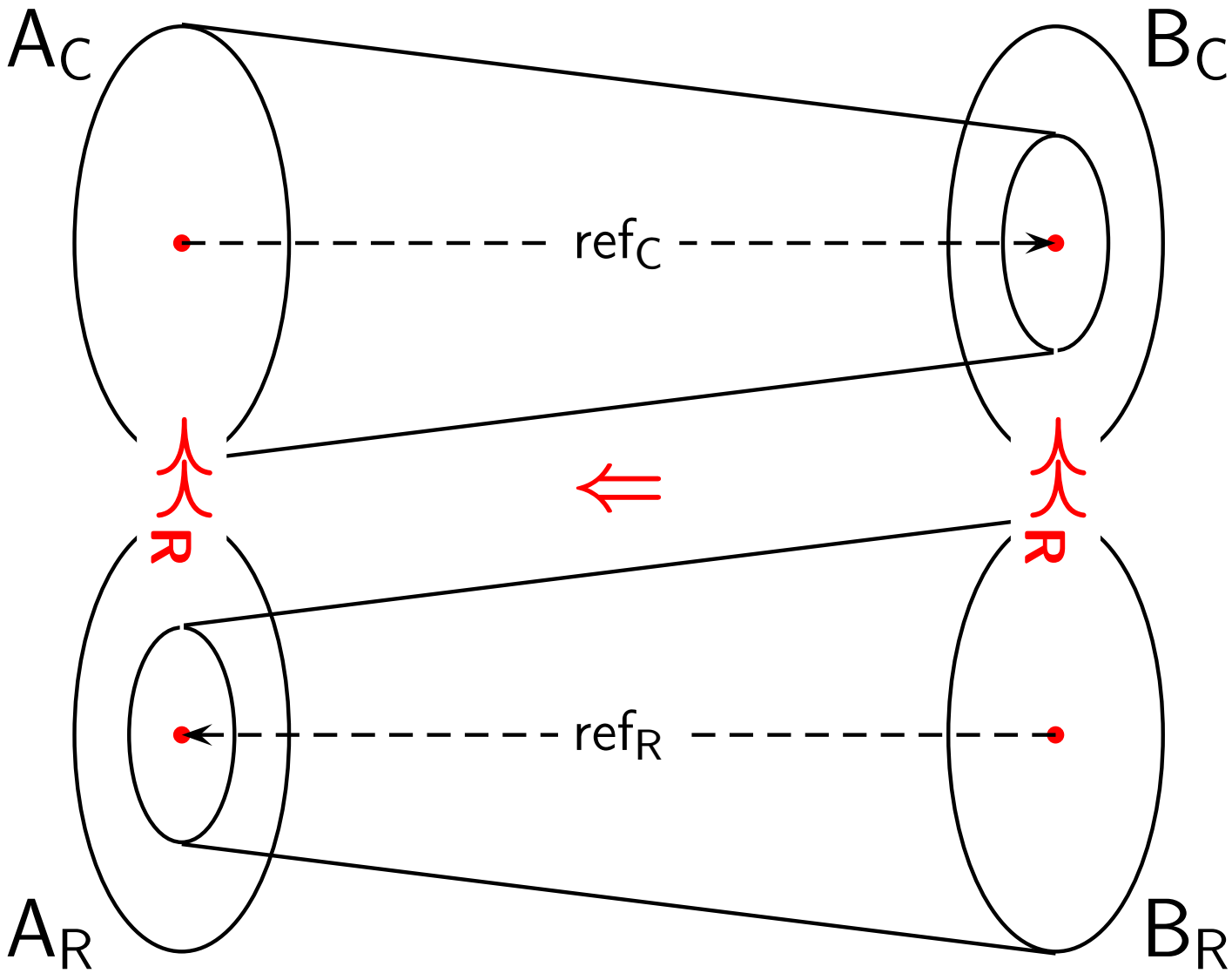
Refinement



Refinement



Refinement



Refinement

Given two interfaces $\mathsf{A} = (\mathsf{C_A}, \mathsf{R_A}, \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A}}, \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A}})$ and $\mathsf{B} = (\mathsf{C_B}, \mathsf{R_B}, \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{B}}, \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{B}})$.
A is a refinement of B is the type which elements consist of:

- a function
$$\mathsf{ref_C} : \mathsf{C_A} \to \mathsf{C_B},$$
- a function
$$\mathsf{ref_R} : \mathsf{R_B} \to \mathsf{R_A},$$
- and proofs for
$$\mathsf{ref_R}\ r_b \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A}}\ c_a \quad \Rightarrow \quad r_b \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{B}}\ \mathsf{ref_C}\ c_a$$
and
$$\mathsf{ref_C}\ c_a \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{B}}\ r_b \quad \Rightarrow \quad c_a \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A}}\ \mathsf{ref_R}\ r_b$$
for $r_b : \mathsf{R_B}$, $c_a : \mathsf{C_A}$.

Refinement

If A is a Refinement of B and $(\mathsf{X_A}, \mathsf{command_A}, \mathsf{next_A}, \mathsf{resp_A})$ a sound strategy on A then is $(\mathsf{X_B}, \mathsf{command_B}, \mathsf{next_B}, \mathsf{resp_B})$ a sound strategy on B where

- elements of $\mathsf{X_B}$ are triples

$$x : \mathsf{X}, r_b : \mathsf{R_B}, \mathsf{id} : \mathsf{resp_A}\ x = \mathsf{ref_R}\ r_b$$

- $$\mathsf{resp_B}\ x\ r_b\ \mathsf{id} = r_b$$

- $$\mathsf{command_B}\ x\ r_b\ \mathsf{id} = \mathsf{ref_C}(\mathsf{command_A}\ x)$$

- $$\mathsf{next_B}\ x\ r_b\ \mathsf{id}\ r'_b\ \mathsf{rfit_B} = \mathsf{next_B}\ x\ (\mathsf{ref_R}\ r'_b)\ \mathsf{rfit_A}$$

  and $\mathsf{rfit_A}$ obtained from $\mathsf{rfit_B}$ by the refinement.

Refinement

- Refinement relation on interfaces/games is reflexiv and transitiv.
- Infimum A of $\mathsf{A_i}$, $i : \mathsf{I}$ is given by

$$\mathsf{A_R} = (i : \mathsf{I}) \to \mathsf{A_{iR}}$$

$$\mathsf{A_C} = \sum(\mathsf{I}, \lambda i : \mathsf{I}.\mathsf{A_{iC}})$$

$$f \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A}} (i, c) \;:\Leftrightarrow\; f\ i \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A_i}} c$$

$$(i, c) \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A}} f \;:\Leftrightarrow\; c \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A_i}} f\ i$$

- Supremum A of $\mathsf{A_i}$, $i : \mathsf{I}$ is given by

$$\mathsf{A_R} = \sum(\mathsf{I}, \lambda i : \mathsf{I}.\mathsf{A_{iR}})$$

$$\mathsf{A_C} = (i : \mathsf{I}) \to \mathsf{A_{iC}}$$

$$(i, r) \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A}} f \;:\Leftrightarrow\; r \prec\!\!\prec_{\mathsf{C}}{}^{\mathsf{A_i}} f\ i$$

$$f \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A}} (i, r) \;:\Leftrightarrow\; f\ i \prec\!\!\prec_{\mathsf{R}}{}^{\mathsf{A_i}} r$$

Cogame, Tensor

Cogame: Player becomes opponent,
opponent becomes player

Tensor: As in usual game semantics:
opponent chooses where to play

Cogame, Tensor, Lolipop

- The cogame $\mathsf{A}^\perp$ of a game A is given by

$$\mathsf{C}_{\mathsf{A}}^{\perp} = \mathsf{R}_{\mathsf{A}}, \mathsf{R}_{\mathsf{A}}^{\perp} = \mathsf{C}_{\mathsf{A}}, \prec\!\!\prec_{\mathsf{C}}^{\mathsf{A}^\perp} = \prec\!\!\prec_{\mathsf{R}}^{\mathsf{A}}, \prec\!\!\prec_{\mathsf{R}}^{\mathsf{A}^\perp} = \prec\!\!\prec_{\mathsf{C}}^{\mathsf{A}} .$$

- The tensor $\mathsf{A} \otimes \mathsf{B}$ is given by

$$(\mathsf{A} \otimes \mathsf{B})_{\mathsf{C}} = \mathsf{A}_{\mathsf{C}} \times \mathsf{B}_{\mathsf{C}}$$

$$(\mathsf{A} \otimes \mathsf{B})_{\mathsf{R}} = (\mathsf{A}_{\mathsf{R}} \times \mathsf{B}_{\mathsf{C}}) + (\mathsf{A}_{\mathsf{C}} \times \mathsf{B}_{\mathsf{R}})$$

$$\mathsf{inl}(r_a, c_b) \prec\!\!\prec_{\mathsf{C}}^{\mathsf{A}\otimes\mathsf{B}} (c_a, c'_b) \;:\Leftrightarrow\; r_a \prec\!\!\prec_{\mathsf{C}}^{\mathsf{A}} c_a \quad \& \quad c_b = c'_b$$

$$\mathsf{inr}(c_a, r_b) \prec\!\!\prec_{\mathsf{C}}^{\mathsf{A}\otimes\mathsf{B}} (c'_a, c_b) \;:\Leftrightarrow\; r_b \prec\!\!\prec_{\mathsf{C}}^{\mathsf{B}} c_b \quad \& \quad c_a = c'_a$$

$$(c_a, c'_b) \prec\!\!\prec_{\mathsf{R}}^{\mathsf{A}\otimes\mathsf{B}} \mathsf{inl}(r_a, c_b) \;:\Leftrightarrow\; c_a \prec\!\!\prec_{\mathsf{R}}^{\mathsf{A}} r_a \quad \& \quad c_b = c'_b$$

$$(c'_a, c_b) \prec\!\!\prec_{\mathsf{R}}^{\mathsf{A}\otimes\mathsf{B}} \mathsf{inr}(c_a, r_b) \;:\Leftrightarrow\; c_b \prec\!\!\prec_{\mathsf{R}}^{\mathsf{B}} r_b \quad \& \quad c_a = c'_a$$

- Lolipop $\mathsf{A} \multimap \mathsf{B} := (\mathsf{A} \otimes \mathsf{B}^\perp)^\perp$.

A Copy-Cat Strategy

Strategy on

$$A \multimap A$$

Idea: Take command c, response r

and proof for $c \prec\!\!\prec r$ or $r \prec\!\!\prec c$.

Play the diagonal (r, r) or (c, c).

A Copy-Cat Strategy

$(\mathsf{CopyCat_A}, \mathsf{command_A}, \mathsf{next_A}, \mathsf{resp_A})$ given by

$$\begin{aligned}\mathsf{CopyCat_A} \quad = \quad & ((r_a : \mathsf{A_R}) \times (c_a : \mathsf{A_C}) \times (\mathsf{fitc} : r_a \prec\!\!\prec^{\mathsf{A}}_{\mathsf{C}} c_a)) + \\ & ((c_a : \mathsf{A_C}) \times (r_a : \mathsf{A_R}) \times (\mathsf{fitr} : c_a \prec\!\!\prec^{\mathsf{A}}_{\mathsf{R}} r_a))\end{aligned}$$

$$\mathsf{resp_A}\ \mathsf{inl}(r_a\ c_a\ \mathsf{fitc}) := (r_a, c_a)$$

$$\mathsf{resp_A}\ \mathsf{inr}(r_a\ c_a\ \mathsf{fitr}) := (r_a, c_a)$$

$$\mathsf{command_A}\ \mathsf{inl}(r_a\ c_a\ \mathsf{fitc}) := \mathsf{inr}(c_a, c_a)$$

$$\mathsf{command_A}\ \mathsf{inr}(r_a\ c_a\ \mathsf{fitr}) := \mathsf{inl}(r_a, r_a)$$

$$\mathsf{next_A}\ \mathsf{inl}(r_a\ c_a\ \mathsf{fitc})\ (r'_a, c_a)\ \mathsf{fitr} := \mathsf{inl}(r'_a, c_a, \mathsf{fitr})$$

$$\mathsf{next_A}\ \mathsf{inr}(r_a\ c_a\ \mathsf{fitr})\ (r_a, c'_a)\ \mathsf{fitc} := \mathsf{inr}(r_a, c'_a, \mathsf{fitc})$$

is a sound strategy on $\mathsf{A} \multimap \mathsf{A}$.

Composition

$$\sigma : A \multimap B \qquad \tau : B \multimap C$$

$$\sigma ; \tau : A \multimap C$$

Idea: Play strategy σ against strategy τ in B.

Problem: Infinite play in B.

Usual solution: Winning strategies.

Solution here: Fairness!

Some restrictions on games

For A : Set let

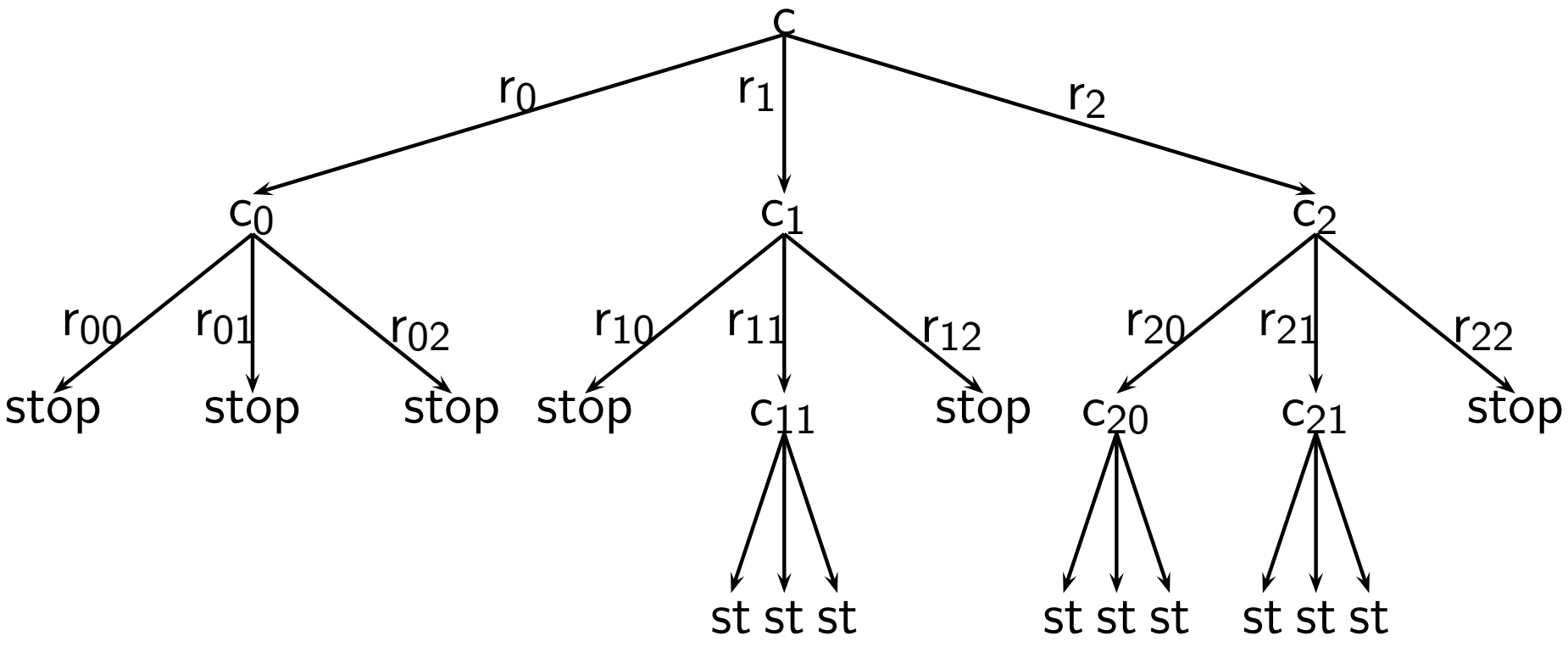
$$\mathsf{Collapse}\ A := (a, b : A) \to \mathsf{Id}\ A\ a\ b$$

$$\mathsf{UIP}\ A := (a, b : A) \to \mathsf{Collapse}\ (\mathsf{Id}\ A\ a\ b)$$

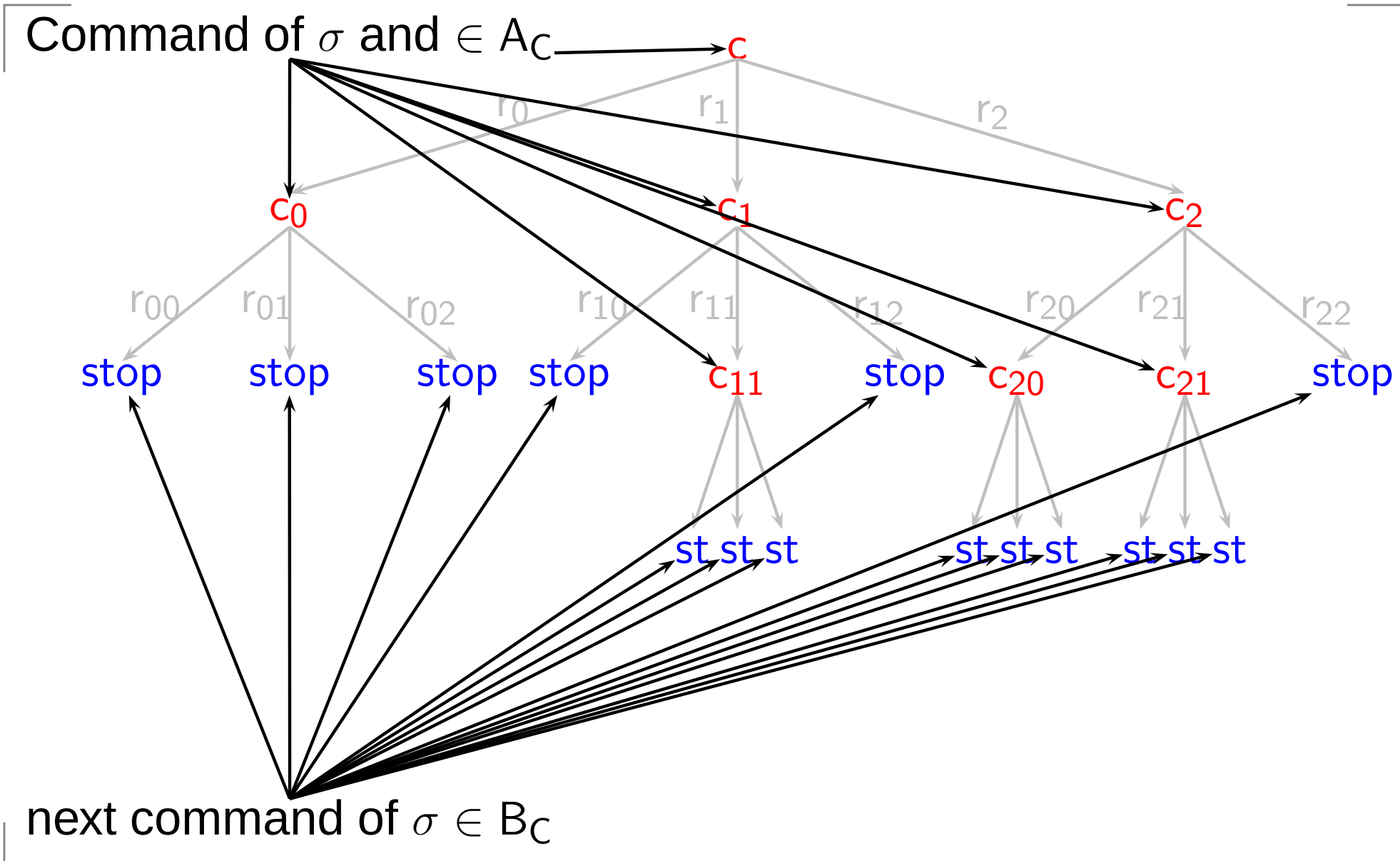
From now on for all games A

- $\mathsf{Collapse}\ r \prec\!\!\prec^{\mathsf{A}}_{\mathsf{C}} c$ and $\mathsf{Collapse}\ c \prec\!\!\prec^{\mathsf{A}}_{\mathsf{R}} r$
- $\mathsf{UIP}\ \mathsf{A}_{\mathsf{C}}$ and $\mathsf{UIP}\ \mathsf{A}_{\mathsf{R}}$.
- Everything must have a predecessor (reasonable game).

Beginning


c
r0
r1
r2
c0
c1
c2
r00
r01
r02
r10
r11
r12
r20
r21
r22
stop
stop
stop
stop
c11
stop
c20
c21
stop
st st st
st st st
st st st

Maxvalid A-Beginning $\sigma : A \multimap B$



Fair Strategy

For games A, B a fair strategy on $A \multimap B$ is given by

- a strategy $\sigma = (X_\sigma, \mathsf{command}_\sigma, \mathsf{next}_\sigma)$
- $$\mathsf{maxABeg}_\sigma, \mathsf{maxBBeg}_\sigma : X \to (\mathsf{Beginning}\ A \multimap B)$$
- with
  $$\mathsf{maxvalidABeginning}\ A\ B\ \sigma\ (\mathsf{maxABeg}_\sigma\ x)\ x$$
  and
  $$\mathsf{maxvalidBBeginning}\ A\ B\ \sigma\ (\mathsf{maxBBeg}_\sigma\ x)\ x$$

Link

For games $\mathsf{A}, \mathsf{B}, \mathsf{C}$, $c_{ab} : (\mathsf{A} \multimap \mathsf{B})_{\mathsf{C}}$, $c_{bc} : (\mathsf{B} \multimap \mathsf{C})_{\mathsf{C}}$

$$\mathsf{Link}\ \mathsf{A}\ \mathsf{B}\ \mathsf{C}\ c_{ab}\ c_{bc}$$

is given by

- $\mathsf{Link}\ \mathsf{A}\ \mathsf{B}\ \mathsf{C}\ \mathsf{inl}(r_a, r_b)\ \mathsf{inl}(r'_b, r_c) = \mathsf{Id}\ \mathsf{B}_{\mathsf{R}}\ r_b\ r'_b$
- $\mathsf{Link}\ \mathsf{A}\ \mathsf{B}\ \mathsf{C}\ \mathsf{inl}(r_a, r_b)\ \mathsf{inr}(c_b, c_c) = \mathsf{False}$
- $\mathsf{Link}\ \mathsf{A}\ \mathsf{B}\ \mathsf{C}\ \mathsf{inr}(c_a, c_b)\ \mathsf{inl}(r_b, r_c) = (c_b \prec\!\!\prec^{\mathsf{B}}_{\mathsf{R}}\ r_b) + (r_b \prec\!\!\prec^{\mathsf{B}}_{\mathsf{C}}\ c_b)$
- $\mathsf{Link}\ \mathsf{A}\ \mathsf{B}\ \mathsf{C}\ \mathsf{inr}(c_a, c_b)\ \mathsf{inr}(c'_b, c_c) = \mathsf{Id}\ \mathsf{B}_{\mathsf{C}}\ c_b\ c'_b$

Composition

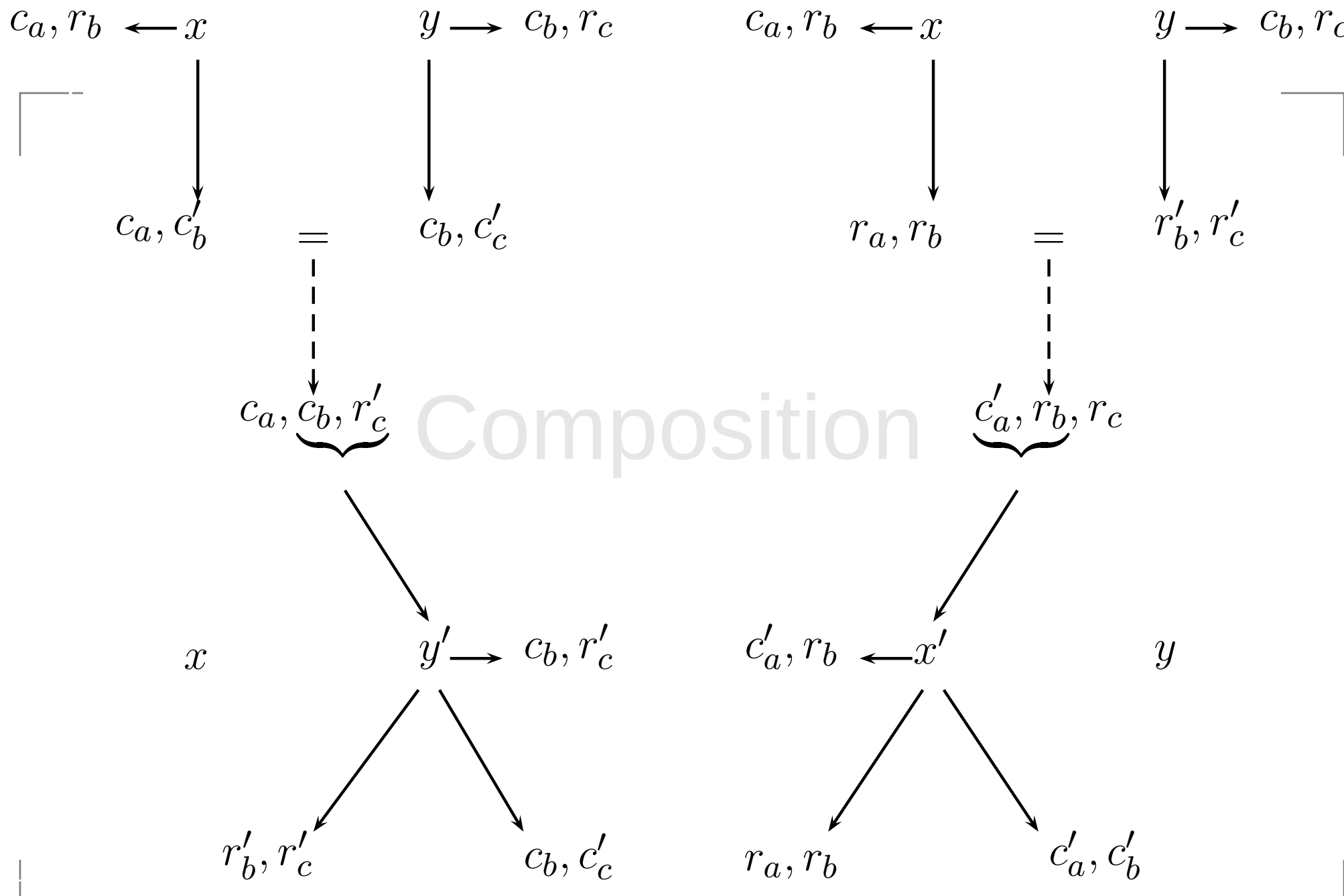
For fair strategies $\sigma = (\mathsf{X}_\sigma, \mathsf{command}_\sigma, \mathsf{next}_\sigma) : \mathsf{A} \multimap \mathsf{B}$ and $\tau = (\mathsf{X}_\tau, \mathsf{command}_\tau, \mathsf{next}_\tau) : \mathsf{B} \multimap \mathsf{C}$ let $X_{\sigma;\tau}$ be the set which elements are triples

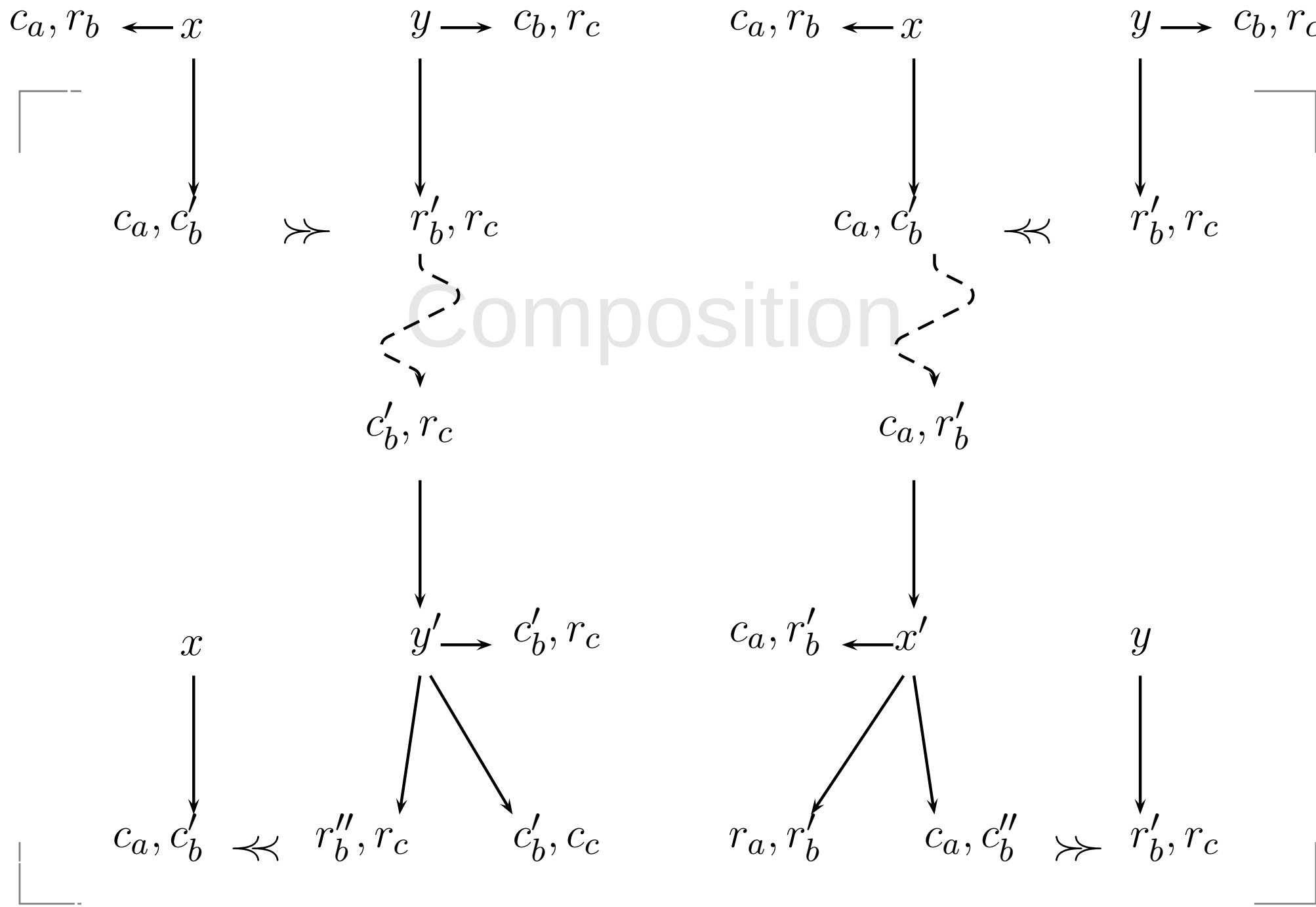
- $x : X_\sigma$
- $y : X_\tau$
- $\mathsf{link} : \mathsf{Link\ A\ B\ C}\ (\mathsf{command}_\sigma\ x)\ (\mathsf{command}_\tau\ y)$

and

$$\mathsf{resp}_{\sigma;\tau}\ (x, y, \mathsf{link}) = (c_a, r_c)$$

where $\mathsf{resp}_\sigma\ x = (c_a, r_b)$ and $\mathsf{resp}_\tau\ y = (c_b, r_c)$.





Bisimulation

Let $\sigma = (\mathsf{X}_\sigma, \mathsf{command}_\sigma, \mathsf{next}_\sigma)$ and $\tau = (\mathsf{X}_\tau, \mathsf{command}_\tau, \mathsf{next}_\tau)$ strategies on A. $x : X_\sigma$, $y : X_\tau$, n : Nat are n-bisimular ($x \sim_n y$) iff $n = 0$ or

$$\mathsf{command}_\sigma\ x = \mathsf{command}_\tau\ y$$

and for r :: A_R and rfit : $\mathsf{command}_\sigma\ x \prec\!\!\prec_\mathsf{R} r$

$$\mathsf{next}_\sigma\ x\ r\ \mathsf{rfit} \sim_{n-1} \mathsf{next}_\tau\ y\ r\ \mathsf{rfit}'$$

where rfit' is obtained by transfering rfit.

$x : X_\sigma$, $y : X_\tau$ are bisimular iff they are n-bisimular f.a. n : Nat.

Behavioural equivalence

Strategies $\sigma = (\mathsf{X}_\sigma, \mathsf{command}_\sigma, \mathsf{next}_\sigma)$ and $\tau = (\mathsf{X}_\tau, \mathsf{command}_\tau, \mathsf{next}_\tau)$ on A are behavioural equivalent ($x \approx y$) iff there are $f : X_\sigma \to X_\tau$ and $g : X_\tau \to X_\sigma$ such that

$$f\ x \sim x$$

and

$$g\ y \sim y$$

f.a. $x : X_\sigma$, $y : X_\tau$.

The Category of reasonable games with fair strategies

For a fair and sound strategy $\sigma : \mathsf{A} \multimap \mathsf{B}$ is

$$\mathsf{id}_\mathsf{A}; \sigma \approx \sigma \approx \sigma; \mathsf{id}_\mathsf{B}$$

where $\mathsf{id}_\mathsf{A} = \mathsf{copyCat}_\mathsf{A}$, $\mathsf{id}_\mathsf{B} = \mathsf{copyCat}_\mathsf{B}$ and A, B reasonable.

For fair and sound strategies
$\sigma : \mathsf{A} \multimap \mathsf{B}, \tau : \mathsf{B} \multimap \mathsf{C}, \mu : \mathsf{C} \multimap \mathsf{D}$:

$$(\sigma; \tau); \mu \approx \sigma; (\tau; \mu)$$

where $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$ reasonable.

Conclusion and future work

- Linear category.
- Investigate relationship to work of Hancock/Hyvernat.
- Improve notion of refinement. Investigate relation to refinement calculus.
- Formal topology.
- ...

Maxvalid Beginnings

For A game let

$$\text{Beginning A} := \text{data} \quad \text{stop} \mid \text{fork}(c : \text{A}_\text{C})(f : (r : \text{A}_\text{R}) \to c \prec\!\!\prec_\text{R} r \to \text{Beginning A})$$

For games A, B, $\sigma = (\text{X}, \text{command}, \text{next})$ strategy on A $\multimap$ B, x : X and beg : Beginning A $\multimap$ B let

$$\begin{aligned}
\text{maxvalidABeginning A B } \sigma \text{ stop } x &= \text{“sigma plays in B”} \\
\text{maxvalidABeginning A B } \sigma \text{ (fork } c\ f) \ x &= \text{“sigma plays in A and} \\
&\quad \text{command } x = c \text{ and} \\
&\quad \text{maxvalidABeginning A B } \sigma \ (f\ r \text{ rfit}) \ (\text{next } x\ r \text{ rfit}) \\
&\quad \text{for } r : (\text{A} \multimap \text{B})_\text{R} \text{ and rfit} : c \prec\!\!\prec_\text{R}^{\text{A}\multimap\text{B}} r\text{.”}
\end{aligned}$$

maxvalidBBeginning A B σ beg x analog.

Reasonable game

A is reasonable is the type which elements consists of

- a function $\mathsf{reas}_\mathsf{C} : \mathsf{A}_\mathsf{C} \to \mathsf{A}_\mathsf{R}$
- a function $\mathsf{reas}_\mathsf{R} : \mathsf{A}_\mathsf{R} \to \mathsf{A}_\mathsf{C}$

such that

$$\mathsf{reas}_\mathsf{C}\ c \prec\!\!\prec_\mathsf{C} c \text{ for } c : \mathsf{A}_\mathsf{C}$$

$$\mathsf{reas}_\mathsf{R}\ r \prec\!\!\prec_\mathsf{R} r \text{ for } r : \mathsf{A}_\mathsf{R}$$

"Everything is needed"