Data Refinement

refinement (-nm-) n. Refining or being refined; fineness of feeling or taste, polished manners etc.; subtle or ingenious manifestation of, piece of elaborate arrangement, (all the refinements of reasoning, torture; a countermine was a refinement beyond their skill); instance of improvement (up)on; piece of subtle reasoning, fine distinction.

Example

Refinement may involve the tightening of a contract, reducing the range of allowable behaviours:

European directive

$\sqsubseteq$

Act of Parliament

$\sqsubseteq$

Regulations

Example

Part of the contract may be rewritten to describe the same objective in different terms:

We would like to raise £1,000,000. We would like to do this in the United States of America. At the time of writing, the exchange rate is £1.00 = $1.50. How many dollars should we attempt to raise?

Abstract data types

An abstract data type combines a description of state with descriptions of operations upon that state.

$$\mathcal{A} == (AState, AInit, \langle AOp1, AOp2, \ldots \rangle)$$

We may use abstract data types to specify the behaviour of system components.

An abstract data type provides a view of system behaviour at some level of abstraction.

Example

An abstract data type may be used to specify a class:

```
class Example
  State state
  Example()
  MethodX()
  MethodY()
```

$AState$

$AInit$

$\langle AOp1, AOp2 \rangle$

Questions

- why might we want different views of the same system?
- what makes two different views consistent?
- what makes two consistent views different?

Refinement relation

Data type $\mathcal{B}$ is a refinement of $\mathcal{A}$,

$$\mathcal{A} \sqsubseteq \mathcal{B}$$

if and only if every interaction with $\mathcal{B}$ could have been an interaction with $\mathcal{A}$.

If this is the case, then $\mathcal{B}$ will be a worthy replacement for $\mathcal{A}$, whatever the context.

Proper refinement

It is quite possible that $\mathcal{A} \sqsubseteq \mathcal{B}$, but $\mathcal{B} \not\sqsubseteq \mathcal{A}$.

This will be the case when description $\mathcal{B}$ has resolved some of the uncertainty present in $\mathcal{A}$: the effect of at least one of the operations is now more precisely defined.

Refinement ordering

If $\mathcal{B}$ is a refinement of $\mathcal{A}$, and $\mathcal{C}$ is a refinement of $\mathcal{B}$, then $\mathcal{C}$ is also a refinement of $\mathcal{A}$.

$$\frac{A \sqsubseteq B \quad B \sqsubseteq C}{A \sqsubseteq C}$$

Iterative development of a system, from abstract specification to concrete implementation, is a form of refinement.

Example

$ResourceManager_A$

$free : \mathbb{F}\,\mathbb{N}$

$Allocate_A$

$\Delta ResourceManager$

$r! : \mathbb{N}$

$r! \in free \;\wedge\; free' = free \setminus \{r!\}$

$ResourceManager_B$

$free_B : \mathbb{F}\,\mathbb{N}$

$Allocate_B$

$\Delta ResourceManager_B$

$r! : \mathbb{N}$

$r! = \min free_B \;\wedge\; free'_B = free_B \setminus \{r!\}$

$ResourceManager_C$

$free_C : \mathrm{seq}\, \mathbb{N}$

$\forall\, i, j : \mathrm{dom}\, free_C \mid i < j \bullet free_C\, i < free_C\, j$

$Allocate_C$

$\Delta ResourceManager_C$

$r! : \mathbb{N}$

$r! = head\, free_C \;\wedge\; free'_C = tail\, free_C$

If the initialisation and release operations are defined consistently,

$$(ResourceManager_A, Init_A, \langle Allocate_A, Release_A \rangle)$$

$$\sqsubseteq$$

$$(ResourceManager_B, Init_B, \langle Allocate_B, Release_B \rangle)$$

$$\sqsubseteq$$

$$(ResourceManager_C, Init_C, \langle Allocate_C, Release_C \rangle)$$

Refinement to code

The refinement of state components and operations can take us to executable code.

- we replace the components of the state—sets, relations, sequences—with data structures from our chosen implementation language.
- we resolve any nondeterminism present in the description of operations.

Example

$ResourceManager_D$

$free_D : \text{array of } Bit$

Data refinement and relations

To derive a formal definition of refinement, and hence a set of useful proof rules, we consider the semantics of operations.

An operation may be seen as a relation between states: one state is related to another if the system could start in the first and end up in the second.

Our definition of refinement for data types will depend upon a similar definition of refinement for relations.

Total relations

Refinement of total relations is easily defined. If R and S are total relations, then

$$R \sqsubseteq S \Leftrightarrow S \subseteq R$$

Wherever R relates the same element x to two distinct elements y_1 and y_2, S may omit either $x \mapsto y_1$ or $x \mapsto y_2$.

Totalisation

To define refinement for partial relations, we consider their totalised forms:

$$\dot{\rho} = \rho \cup \{ x : X^{\perp}; \ y : Y^{\perp} \mid x \notin \operatorname{dom} \rho \bullet x \mapsto y \}$$

where

$$X^{\perp} = X \cup \{ \perp \}$$

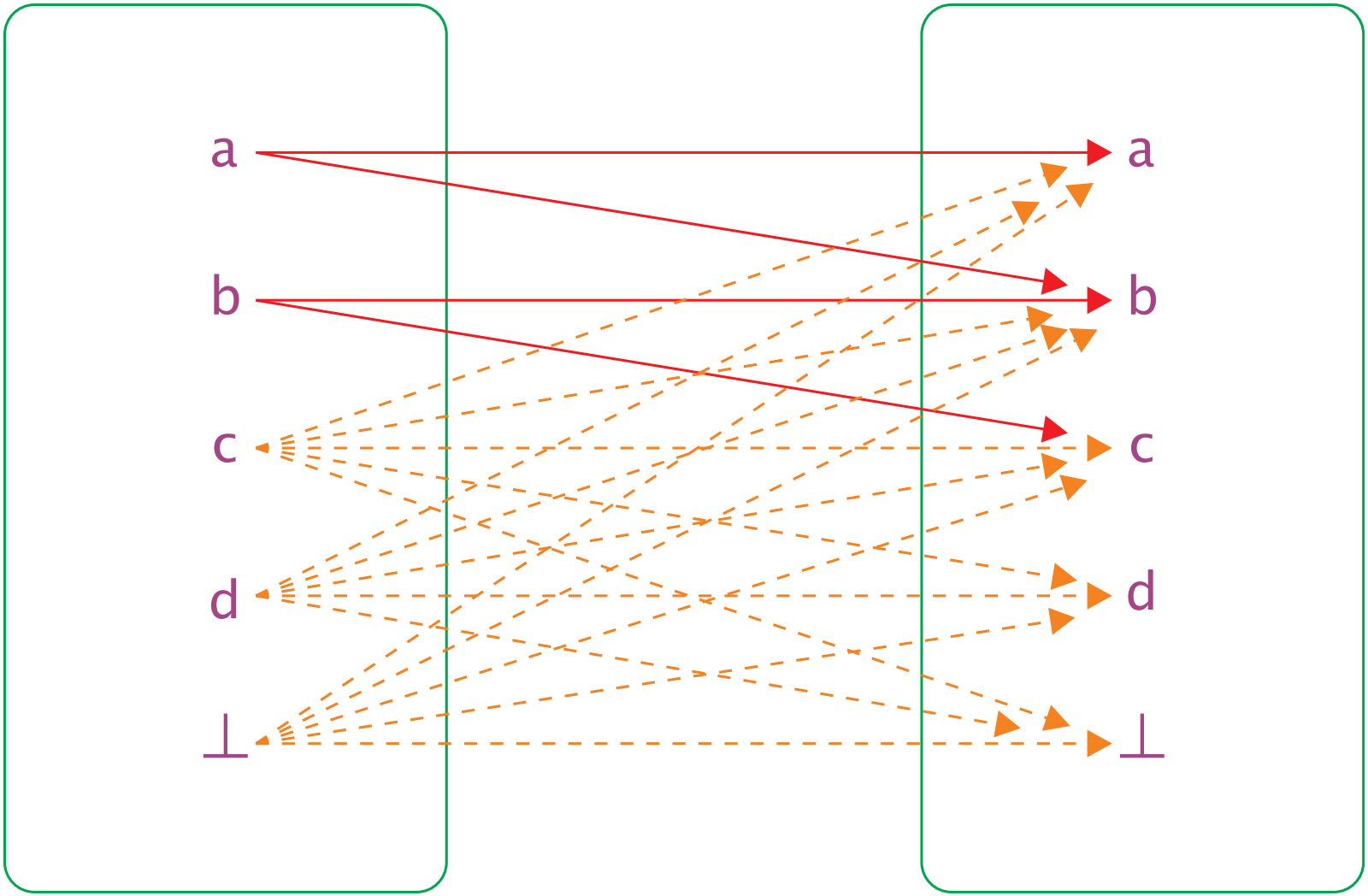
$$Y^{\perp} = Y \cup \{ \perp \}$$

Example

$$L ::= a \mid b \mid c \mid d$$

$$\rho == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c\}$$

$$\dot{\rho} == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c,$$
$$c \mapsto \bot, c \mapsto a, c \mapsto b, c \mapsto c, c \mapsto d,$$
$$d \mapsto \bot, d \mapsto a, d \mapsto b, d \mapsto c, d \mapsto d,$$
$$\bot \mapsto \bot, \bot \mapsto a, \bot \mapsto b, \bot \mapsto c, \bot \mapsto d\}$$


a
b
c
d
⊥
a
b
c
d
⊥

Why augment the source and target?

$$\kappa_0 \;==\; \{ z : \mathbb{Z} \bullet z \mapsto 0 \}$$

$$\dot{\varnothing} \mathbin{⨾} \dot{\kappa}_0 \;=\; \ldots$$

With

$$= (\varnothing \cup (\overline{\operatorname{dom} \varnothing}^{\perp} \times \mathbb{Z}^{\perp})) \fatsemi (\kappa_0 \cup (\overline{\operatorname{dom} \kappa_0}^{\perp} \times \mathbb{Z}^{\perp}))$$

[dot]

$$= (\overline{\varnothing}^{\perp} \times \mathbb{Z}^{\perp}) \fatsemi (\kappa_0 \cup (\overline{\mathbb{Z}}^{\perp} \times \mathbb{Z}^{\perp}))$$

[properties of $\cup$ and dom]

$$= (\mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp}) \fatsemi (\kappa_0 \cup (\varnothing^{\perp} \times \mathbb{Z}^{\perp}))$$

[properties of $\overline{}$]

$$= ((\mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp}) \fatsemi \kappa_0) \cup ((\mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp}) \fatsemi (\{\perp\} \times \mathbb{Z}^{\perp}))$$

[property of $\times$]

$$= \kappa_0 \cup (\mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp})$$ [properties of $\times$]

$$= \mathbb{Z}^{\perp} \times \mathbb{Z}^{\perp}$$ [property of $\cup$]

Without

$$= (\varnothing \cup (\overline{\operatorname{dom} \varnothing} \times \mathbb{Z}))\ ⨾\ (\kappa_0 \cup (\overline{\operatorname{dom} \kappa_0} \times \mathbb{Z}))$$

[dot without $\perp$]

$$= (\overline{\varnothing} \times \mathbb{Z})\ ⨾\ (\kappa_0 \cup (\overline{\mathbb{Z}} \times \mathbb{Z}))$$

[properties of $\cup$ and dom]

$$= (\mathbb{Z} \times \mathbb{Z})\ ⨾\ (\kappa_0 \cup (\varnothing \times \mathbb{Z}))$$ [properties of $\overline{}$]

$$= (\mathbb{Z} \times \mathbb{Z})\ ⨾\ (\kappa_0 \cup \varnothing)$$ [property of $\times$]

$$= (\mathbb{Z} \times \mathbb{Z})\ ⨾\ \kappa_0$$ [property of $\cup$]

$$= \kappa_0$$ [property of ⨾]

Refinement

$$\rho == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c\}$$
$$\sigma == \{a \mapsto a, b \mapsto b, b \mapsto c, c \mapsto c\}$$

$$\mathrm{dom}\,\sigma \;=\; \{a, b, c\} \;\supseteq\; \{a, b\} \;=\; \mathrm{dom}\,\rho$$

$$(\mathrm{dom}\,\rho \lhd \sigma)$$
$$= \{a \mapsto a, b \mapsto b, b \mapsto c, \}$$
$$\subseteq \rho$$

Using totalisation

$$
\begin{aligned}
\dot{\sigma} = \{ & a \mapsto a, b \mapsto b, b \mapsto c, c \mapsto c, \\
& d \mapsto \bot, d \mapsto a, d \mapsto b, d \mapsto c, d \mapsto d, \\
& \bot \mapsto \bot, \bot \mapsto a, \bot \mapsto b, \bot \mapsto c, \bot \mapsto d \} \\
\subseteq \dot{\rho}
\end{aligned}
$$

Not a refinement

$$\rho == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c\}$$

$$\tau == \{a \mapsto a, c \mapsto c\}$$

$$(\operatorname{dom} \rho) = \{a, b\} \not\subseteq \{a, c\} = (\operatorname{dom} \tau)$$

Corruption

$$\sim : Bit \rightarrow Bit$$

$$\sim 0 = 1$$

$$\sim 1 = 0$$

Never two in a row

$$\_corruptsto\_ : \mathrm{seq}\, Bit \leftrightarrow \mathrm{seq}\, Bit$$

$$\begin{aligned}
&\forall\, bs, bs' : \mathrm{seq}\, Bit \bullet \\
&\quad bs\ corruptsto\ bs' \Leftrightarrow \\
&\qquad \#bs' \leq \#bs \wedge \\
&\qquad \forall\, i : 1 \mathinner{..} \#bs' - 1 \bullet \\
&\qquad\quad bs\, i \neq bs'\, i \Rightarrow bs(i+1) = bs'(i+1)
\end{aligned}$$

$$\langle 1, 1, 0, 1, 1, 1, 0, 0 \rangle\ corruptsto\ \langle 0, 1, 0, 0, 1 \rangle$$

Odd ones

$$
\begin{array}{l}
\_changesto\_ : \mathrm{seq}\, Bit \leftrightarrow \mathrm{seq}\, Bit \\
\hline
\forall\, bs, bs' : \mathrm{seq}\, Bit \bullet \\
\quad bs\ changesto\ bs' \Leftrightarrow \\
\quad\quad \#bs' \leq \#bs \wedge \\
\quad\quad \forall\, i : 1 \mathinner{..} (\#bs' - 1) \bullet \\
\quad\quad\quad i \in \{ n : \mathbb{N}_1 \bullet 2 * n \} \Rightarrow bs\, i = bs'\, i \wedge \\
\quad\quad\quad i \in \{ n : \mathbb{N} \bullet 2 * n + 1 \} \Rightarrow bs\, i \neq bs'\, i
\end{array}
$$

$$\langle 1, 1, 0, 1, 1, 1, 0, 0 \rangle\ changesto\ \langle 0, 1, 1, 1, 0 \rangle$$

Programs

A program is a sequence of operations upon a data type.

$$\mathcal{D} = (D, di, df, \{do_1, do_2\})$$

$$di \fatsemi do_1 \fatsemi do_2 \fatsemi df$$

Parameters

$$P(\mathcal{X}) \;=\; xi \mathrel{\mathrel{;}} xo_1 \mathrel{\mathrel{;}} xo_2 \mathrel{\mathrel{;}} xf$$

Totalisation

$$\dot{\mathcal{X}} = (X^{\perp}, \dot{xi}, \dot{xf}, \{ i : I \bullet \dot{xo}_i \})$$

Refinement

$$P(\dot{C}) \subseteq P(\dot{\mathcal{A}})$$

$$\dot{ci} \mathbin{⨾} \dot{co}_{s_1} \mathbin{⨾} \dot{co}_{s_2} \mathbin{⨾} \cdots \mathbin{⨾} \dot{co}_{s_n} \mathbin{⨾} \dot{cf}$$
$$\subseteq$$
$$\dot{ai} \mathbin{⨾} \dot{ao}_{s_1} \mathbin{⨾} \dot{ao}_{s_2} \mathbin{⨾} \cdots \mathbin{⨾} \dot{ao}_{s_n} \mathbin{⨾} \dot{af}$$

Example

$$A == \mathrm{seq}\, Bit \times Action \times \mathrm{seq}\, Bit$$

$$C == \mathrm{seq}\, Bit \times Action \times \mathrm{seq}\, Bit$$

$$Action ::= yes \mid no$$

Initialisation

$$
\begin{array}{l}
\_ai\_ : \mathrm{seq}\, Bit \leftrightarrow A \\
\_ci\_ : \mathrm{seq}\, Bit \leftrightarrow C \\
\hline
\forall\, bs : \mathrm{seq}\, Bit;\ a : A;\ c : C \bullet \\
\qquad bs\ ai\ a \Leftrightarrow a = (bs, no, \langle\rangle) \\
\qquad bs\ ci\ c \Leftrightarrow c = (bs, no, \langle\rangle)
\end{array}
$$

Finalisation

$$
\begin{array}{l}
\_af\_ : A \leftrightarrow \mathrm{seq}\, Bit \\
\_cf\_ : C \leftrightarrow \mathrm{seq}\, Bit \\
\hline
\forall\, bs : \mathrm{seq}\, Bit;\ a : A;\ c : C \bullet \\
\qquad a\ af\ bs \Leftrightarrow bs = a.3 \\
\qquad c\ cf\ bs \Leftrightarrow bs = c.3
\end{array}
$$

$$\_ao\_ : A \leftrightarrow A$$

$$\forall\, a, a' : A \bullet$$
$$\quad a \; ao \; a' \Leftrightarrow$$
$$\quad\quad a'.1 = tail\; a.1$$
$$\quad\quad a.2 = yes \Rightarrow$$
$$\quad\quad\quad a'.3 = a.3 \frown \langle head\; a.1 \rangle \wedge a'.2 = no$$
$$\quad\quad a.2 = no \Rightarrow$$
$$\quad\quad\quad a'.3 = a.3 \frown \langle \sim head\; a.1 \rangle \wedge a'.2 = yes$$
$$\quad\quad\quad \vee$$
$$\quad\quad\quad a'.3 = a.3 \frown \langle head\; a.1 \rangle \wedge a'.2 = no$$

$$\_co\_ : C \leftrightarrow C$$

$$\forall\, c, c' : C \bullet$$

$$c \; co \; c' \Leftrightarrow$$

$$c'.1 = tail\, c.1$$

$$c.2 = yes \Rightarrow$$

$$c'.3 = c.3 \frown \langle head\, c.1 \rangle \wedge c'.2 = no$$

$$c.2 = no \Rightarrow$$

$$c'.3 = c.3 \frown \langle \sim head\, c.1 \rangle \wedge c'.2 = yes$$

$$ci \circ cf \subseteq ai \circ af$$

$$ci \circ co \circ cf \subseteq ai \circ ao \circ af$$

$$ci \circ co \circ co \circ cf \subseteq ai \circ ao \circ ao \circ af$$

$$\vdots$$

Simulation?

- is ci a subset of $ai \fatsemi \rho$?
- is $\rho \fatsemi cf$ a subset of af?
- is $\rho \fatsemi co_i$ a subset of $ao_i \fatsemi \rho$, for each index i?

Lifting

$$\mathring{\rho} \in X^{\perp} \leftrightarrow Y^{\perp}$$

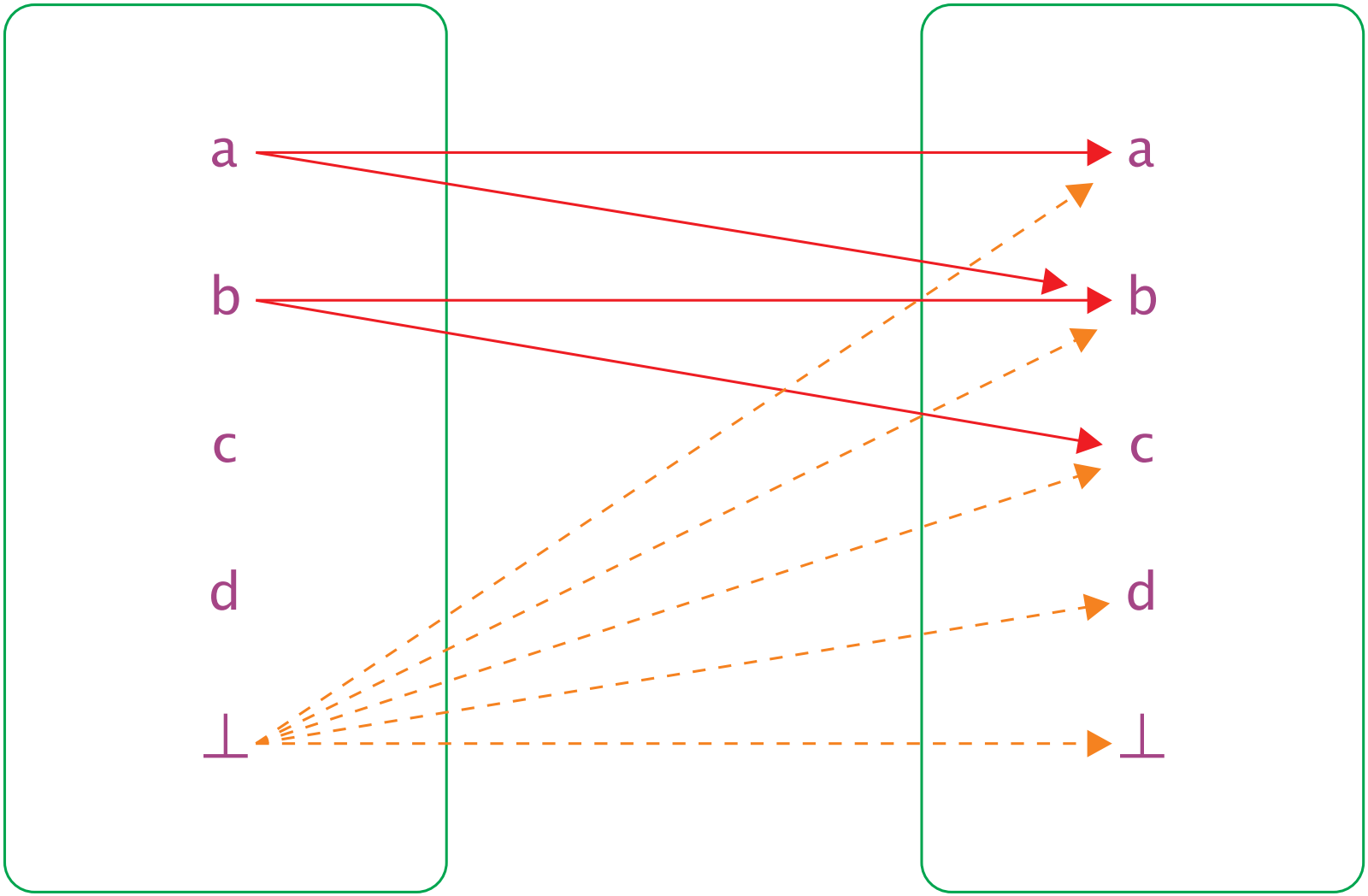
$$\mathring{\rho} = \rho \cup (\{\perp\} \times Y^{\perp})$$

Example

$$L ::= a \mid b \mid c \mid d$$

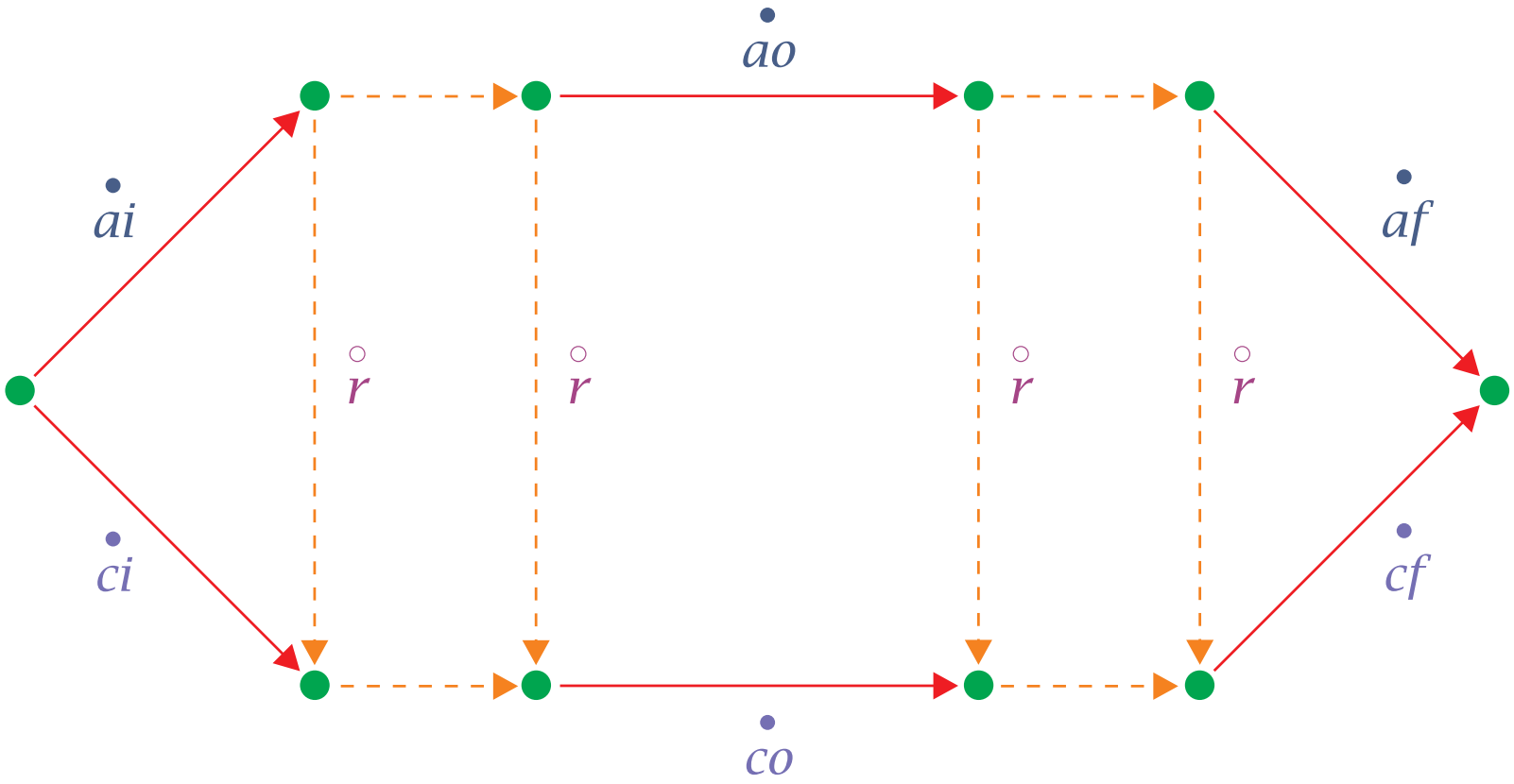
$$\rho == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c\}$$

$$\mathring{\rho} == \{a \mapsto a, a \mapsto b, b \mapsto b, b \mapsto c,$$
$$\bot \mapsto \bot, \bot \mapsto a, \bot \mapsto b, \bot \mapsto c, \bot \mapsto d\}$$


a
b
c
d
⊥
a
b
c
d
⊥

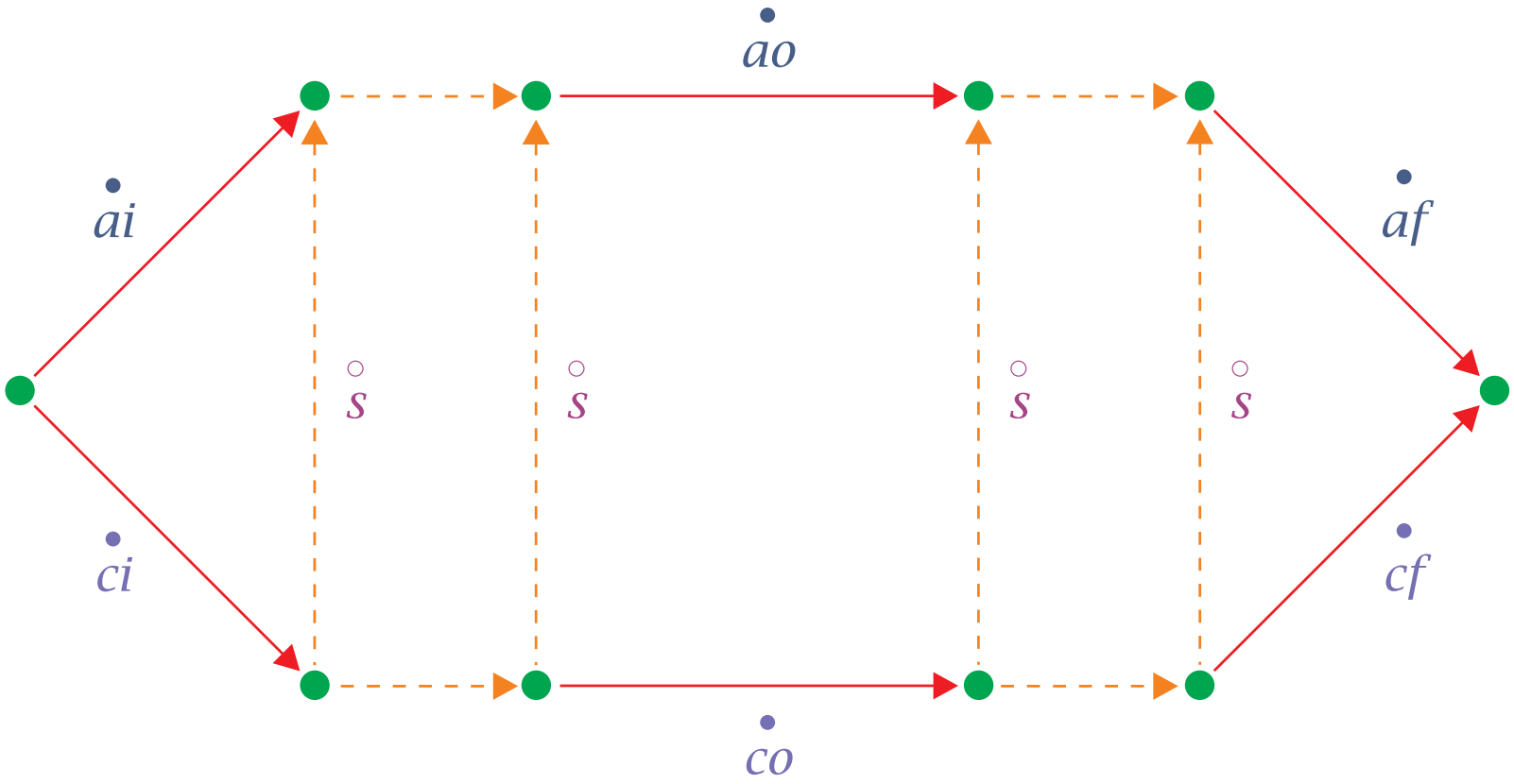
Forwards simulation

- $\overset{\bullet}{ci} \subseteq \overset{\bullet}{ai} \fatsemi \overset{\circ}{r}$
- $\overset{\circ}{r} \fatsemi \overset{\bullet}{cf} \subseteq \overset{\bullet}{af}$
- $\overset{\circ}{r} \fatsemi \overset{\bullet}{co}_i \subseteq \overset{\bullet}{ao}_i \fatsemi \overset{\circ}{r}$ for each index i


ao
ai
af
r
r
r
r
ci
cf
co

Backwards simulation

- $\dot{ci} \fatsemi \mathring{s} \subseteq \dot{ai}$
- $\dot{cf} \subseteq \mathring{s} \fatsemi \dot{af}$
- $\dot{co}_i \fatsemi \mathring{s} \subseteq \mathring{s} \fatsemi \dot{ao}_i$ for each index i


ao
ai
af
s
s
s
s
ci
cf
co

Relaxing

$$\dot{ci} \subseteq \dot{ai} \fatsemi \mathring{r}$$

$$\Leftrightarrow \mathring{ci} \subseteq \mathring{ai} \fatsemi \mathring{r} \qquad [ai \text{ and } ci \text{ are both total}]$$

$$\Leftrightarrow ci \subseteq \mathring{ai} \fatsemi \mathring{r} \ \wedge \ \{\bot\} \times C^{\bot} \subseteq \mathring{ai} \fatsemi \mathring{r}$$

$$[\text{property of subset}]$$

$$\Leftrightarrow ci \subseteq ai \fatsemi r \ \wedge \ \{\bot\} \times C^{\bot} \subseteq \mathring{ai} \fatsemi \mathring{r} \qquad [\bot \notin \operatorname{dom} ci]$$

$$\Leftrightarrow ci \subseteq ai \fatsemi r \ \wedge \ \{\bot\} \times C^{\bot} \subseteq \mathring{ai} \fatsemi (r \cup \{\bot\} \times C^{\bot})$$

$$[\text{lifting}]$$

$$\Leftrightarrow ci \subseteq ai \fatsemi r \qquad [\bot \in \operatorname{ran} \mathring{ai}]$$

Spot dot elimination

$$\rho \subseteq \dot{\sigma} \fatsemi \mathring{\tau}$$

$$\Leftrightarrow \rho \subseteq (\sigma \cup (\overline{\mathrm{dom}\,\sigma}^{\perp} \times Y^{\perp})) \fatsemi \mathring{\tau} \qquad \text{[totalisation]}$$

$$\Leftrightarrow \rho \subseteq (\sigma \fatsemi \mathring{\tau}) \cup ((\overline{\mathrm{dom}\,\sigma}^{\perp} \times Y^{\perp}) \fatsemi \mathring{\tau}) \qquad \text{[distribution]}$$

$$\Leftrightarrow \rho \subseteq (\sigma \fatsemi \tau) \cup ((\overline{\mathrm{dom}\,\sigma}^{\perp} \times Y^{\perp}) \fatsemi \mathring{\tau}) \qquad [\perp \notin \mathrm{ran}\,\sigma]$$

$$\Leftrightarrow \rho \subseteq (\sigma \fatsemi \tau) \cup ((\overline{\mathrm{dom}\,\sigma}^{\perp} \times Y^{\perp}) \fatsemi (\tau \cup \{\perp \times Z^{\perp})) \qquad \text{[lifting]}$$

$$\Leftrightarrow \rho \subseteq (\sigma \fatsemi \tau) \cup (\overline{\mathrm{dom}\,\sigma}^{\perp} \times Z^{\perp}) \qquad [\text{property of } \fatsemi]$$

$$\Leftrightarrow (\mathrm{dom}\,\sigma) \lhd \rho \subseteq \sigma \fatsemi \tau \qquad \text{[property of relations]}$$

Correctness

$$\mathring{r} \fatsemi \dot{co} \;\subseteq\; \dot{ao} \fatsemi \mathring{r}$$

$$\Leftrightarrow \operatorname{dom} ao \lhd (\mathring{r} \fatsemi \dot{co}) \;\subseteq\; ao \fatsemi r \quad \text{[spot dot elimination]}$$

$$\Leftrightarrow (\operatorname{dom} ao \lhd \mathring{r}) \fatsemi \dot{co} \;\subseteq\; ao \fatsemi r \quad \text{[property of } \lhd \text{ and } \fatsemi\text{]}$$

$$\Leftrightarrow (\operatorname{dom} ao \lhd r) \fatsemi \dot{co} \;\subseteq\; ao \fatsemi r \quad [\bot \notin \operatorname{dom} ao]$$

$$\Leftrightarrow (\operatorname{dom} ao \lhd r) \fatsemi (co \cup \overline{\operatorname{dom} co}^{\bot} \times C^{\bot}) \;\subseteq\; ao \fatsemi r$$

[totalisation]

$$\Leftrightarrow (\operatorname{dom} ao \lhd r) \fatsemi co \;\subseteq\; ao \fatsemi r \quad \text{[property of } \subseteq\text{]}$$

$$\wedge$$

$$(\operatorname{dom} ao \lhd r) \fatsemi (\overline{\operatorname{dom} co}^{\bot} \times C^{\bot}) \;\subseteq\; ao \fatsemi r$$

Relaxed rules for forwards simulation

$$ci \subseteq ai \fatsemi r$$

$$r \fatsemi cf \subseteq af$$

$$(\mathrm{dom}\, ao) \lhd r \fatsemi co \subseteq ao \fatsemi r$$

$$\mathrm{ran}((\mathrm{dom}\, ao) \lhd r) \subseteq \mathrm{dom}\, co$$

Relaxed rules for backwards simulation

$$ci \mathbin{⨾} s \subseteq ai$$

$$cf \subseteq s \mathbin{⨾} af$$

$$\mathrm{dom}(s \rhd (\mathrm{dom}\, ao)) \lhd co \mathbin{⨾} s \subseteq s \mathbin{⨾} ao$$

$$\overline{\mathrm{dom}\, co} \subseteq \mathrm{dom}(s \rhd (\mathrm{dom}\, ao))$$

Summary

- refinement
- abstract data types
- refinement of total relations
- refinement of partial relations
- simulation