10. Mathematical Models for Refinement and Decomposition

Jean-Raymond Abrial (ETH)

July 2007

Purpose of this Lecture

- To justify the Proof Obligation Rules we used in this course

- To study invariant preservation rules

- To define a notion of trace

- To study simple refinement in terms of traces

- To study two sufficient conditions for refinement

- To study discrete system decomposition

The Proof Obligations were introduced in the following chapters:

PO Rules	Chapters
INI_INV	II. 4.16
FIS	IV. 5.4
INV	II. 4.4
DLF	II. 4.20
INI_INV_REF	IV. 7.5
FIS_REF	

PO Rules	Chapters
GRD_REF	II. 5.5
INV_REF	IV. 7.2
DLF_REF	II. 5.15
WFD_REF1	II. 5.13
WFD_REF2	II. 5.13

General Approach

- Start from a sketchy discrete transition system

- Build set-theoretic models of:
  - Variables
  - Invariants
  - Events

- Link the sketchy transition system to the math models

1. Invariant Preservation

Invariant Preservation: Sketchy Model

variables: v

inv: $I(v)$

init
$v :| K(v')$

event$_i$
when
$G_i(v)$
then
$v :| R_i(v, v')$
end

Invariant Preservation: Mathematical Model

- let S be the set on which the variables v are moving with $I(v)$

- Let L be the initializing set as defined by the init event

- Let ae_i be the binary relation associated with event event$_i$

- Let ae be the binary relation associated with all transitions

- Properties:

$$\boxed{L \subseteq S}$$

$$\boxed{L \neq \varnothing}$$

$$\boxed{ae_i \in S \leftrightarrow S}$$

$$\boxed{ae = ae_1 \cup \ldots \cup ae_n}$$

$\boldsymbol{S}$	$\{\, \boldsymbol{v} \mid \boldsymbol{I(v)} \,\}$
$\boldsymbol{L}$	$\{\, \boldsymbol{v} \mid \boldsymbol{K(v)} \,\}$
$\boldsymbol{ae_i}$	$\{\, \boldsymbol{v} \mapsto \boldsymbol{v'} \mid \boldsymbol{I(v)} \;\wedge\; \boldsymbol{G_i(v)} \;\wedge\; \boldsymbol{R_i(v, v')} \,\}$
$\mathbf{dom}\,(\boldsymbol{ae_i})$	$\{\, \boldsymbol{v} \mid \boldsymbol{I(v)} \;\wedge\; \boldsymbol{G_i(v)} \,\}$

Invariant Preservation: the Rules

$\boldsymbol{S}$	$\{\, \boldsymbol{v} \mid \boldsymbol{I}(\boldsymbol{v}) \,\}$
$\boldsymbol{L}$	$\{\, \boldsymbol{v} \mid \boldsymbol{K}(\boldsymbol{v}) \,\}$

$\boldsymbol{L} \neq \varnothing \quad \leadsto$

$\vdash \; \exists \boldsymbol{v} \cdot \boldsymbol{K}(\boldsymbol{v})$	INI_FIS

$\boldsymbol{L} \subseteq \boldsymbol{S} \quad \leadsto$

$\boldsymbol{K}(\boldsymbol{v}) \; \vdash \; \boldsymbol{I}(\boldsymbol{v})$	INI_INV

S	$\{\, v \mid I(v) \,\}$
ae_i	$\{\, v \mapsto v' \mid I(v) \;\wedge\; G_i(v) \;\wedge\; R_i(v, v') \,\}$
$\mathrm{dom}\,(ae_i)$	$\{\, v \mid I(v) \;\wedge\; G_i(v) \,\}$

$$\mathrm{dom}\,(ae_i) \;=\; \{\, v \mid I(v) \;\wedge\; G_i(v) \;\wedge\; \exists v' \cdot R_i(v, v') \,\}$$

$\leadsto$

$I(v)$ $G_i(v)$ $\vdash$ $\exists v' \cdot R_i(v, v')$	FIS

S	$\{\, v \mid I(v) \,\}$
ae_i	$\{\, v \mapsto v' \mid I(v) \;\wedge\; G_i(v) \;\wedge\; R_i(v, v') \,\}$

$$ae_i \;\in\; S \leftrightarrow S \quad \rightsquigarrow$$

$I(v)$ $G_i(v)$ $R_i(v, v')$ $\vdash$ $I(v')$	INV

S	$\{\, v \mid I(v) \,\}$
$\mathrm{dom}\,(ae_i)$	$\{\, v \mid I(v) \;\wedge\; G_i(v) \,\}$
ae	$ae_1 \;\cup\; \ldots \;\cup\; ae_n$

- Deadlock freedom rule

$$S \;\subseteq\; \mathrm{dom}(ae) \quad \leadsto \quad \begin{array}{l} \;\; I(v) \\ \vdash \\ \;\; G_1(v) \vee \ldots \vee G_n(v) \end{array} \quad \text{DLF}$$

2.Traces

Trace: Definition

- We want to observe the behavior of discrete systems

- A trace is a record of the history of the observed transitions

- This allows us to define refinement: comparisons of behaviors.

- We first present an example

- Then we shall generalize the example

- Finally, we give a mathematical definition of traces

Trace: Example 1

- The action/weak-reaction design pattern

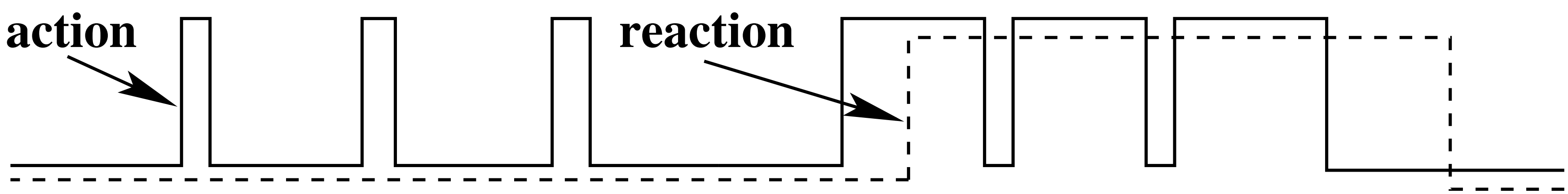
```
init
    a := 0
    r := 0
```

```
a_on
  when
    a = 0
  then
    a := 1
  end
```

```
a_off
  when
    a = 1
  then
    a := 0
  end
```

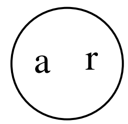
```
r_on
  when
    a = 1
    r = 0
  then
    r := 1
  end
```

```
r_off
  when
    a = 0
    r = 1
  then
    r := 0
  end
```

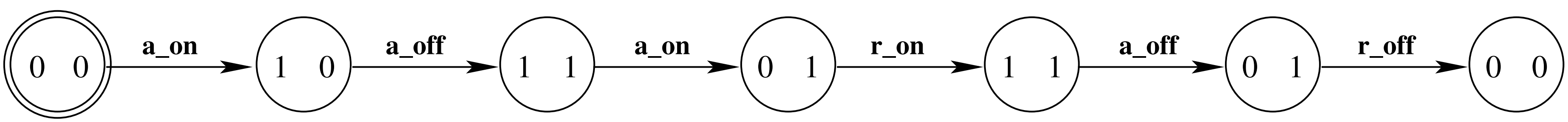


Trace

- Let's represent the state as follows:



- Here is what can be observed after 6 transitions:



- It is one among many others observations

- Such a succession of states is called a trace.

Characterizing a Trace

- It is a finite non-empty sequence.

- Its first element is a member of the initial set of states.

- Successive elements in it are related by a before-after predicate.

- All non-empty prefixes of a trace are traces as well

Mathematical Representation

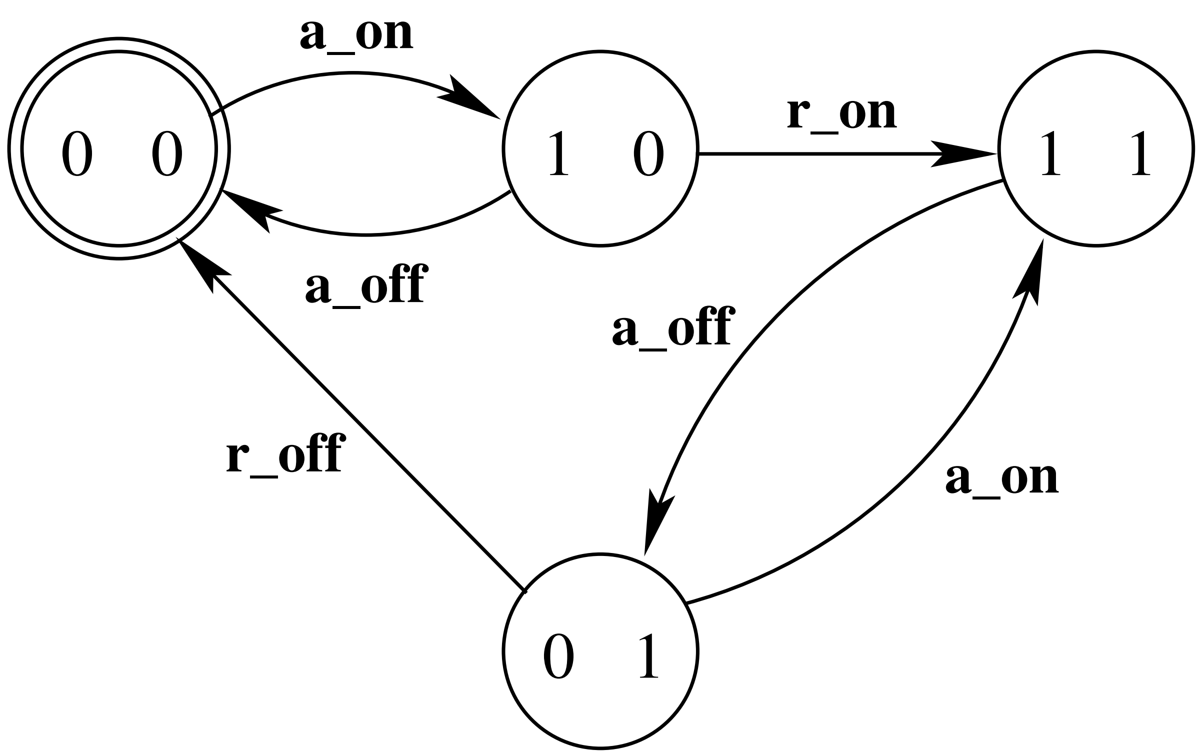
init $a := 0$ $r := 0$	a_on **when** $a = 0$ **then** $a := 1$ **end**	a_off **when** $a = 1$ **then** $a := 0$ **end**	r_on **when** $a = 1$ $r = 0$ **then** $r := 1$ **end**	r_off **when** $a = 0$ $r = 1$ **then** $r := 0$ **end**

$$L = \{0 \mapsto 0\}$$

$$\begin{array}{lcl}
\textbf{a_on_rel} & = & \{(0 \mapsto 0) \mapsto (1 \mapsto 0), (0 \mapsto 1) \mapsto (1 \mapsto 1)\} \\
\textbf{a_off_rel} & = & \{(1 \mapsto 0) \mapsto (0 \mapsto 0), (1 \mapsto 1) \mapsto (0 \mapsto 1)\} \\
\textbf{r_on_rel} & = & \{(1 \mapsto 0) \mapsto (1 \mapsto 1)\} \\
\textbf{r_off_rel} & = & \{(0 \mapsto 1) \mapsto (0 \mapsto 0)\}
\end{array}$$

$$\begin{array}{lcl}
ae & = & \{(0 \mapsto 0) \mapsto (1 \mapsto 0), \\
 & & \;(0 \mapsto 1) \mapsto (1 \mapsto 1), \\
 & & \;(1 \mapsto 0) \mapsto (0 \mapsto 0), \\
 & & \;(1 \mapsto 1) \mapsto (0 \mapsto 1), \\
 & & \;(1 \mapsto 0) \mapsto (1 \mapsto 1), \\
 & & \;(0 \mapsto 1) \mapsto (0 \mapsto 0)\ \}
\end{array}$$

Graph of Evolution



- A trace is a path in this graph

- A path starts at the initializing set

Set of Traces

- The set of traces T associated with:
  - an initial set L
  - a transition relation ae

$$T \;\in\; \mathbb{P}(S) \times (S \leftrightarrow S) \;\rightarrow \mathbb{P}(\mathbb{N}_1 \times (\mathbb{N}_1 \nrightarrow S))$$

- Definition:

$$\begin{array}{l} \quad n \mapsto t \;\in\; T(L \mapsto ae) \\ \Leftrightarrow \\ \left(\begin{array}{l} n \in \mathbb{N}_1 \\ t \in 1 \,..\, n \rightarrow S \\ t(1) \in L \\ \forall\, i \cdot (\, i \in 1 \,..\, n-1 \;\Rightarrow\; t(i) \mapsto t(i+1) \;\in\; ae\,) \end{array}\right) \end{array}$$

3. Simple Refinement

- We introduce another example and generalize it

- We compare the traces of two transition systems

- Trace inclusion is considered

- Adding more constraints

- A notion of external variables and set

Trace: Example 2

- The action/strong-reaction design pattern

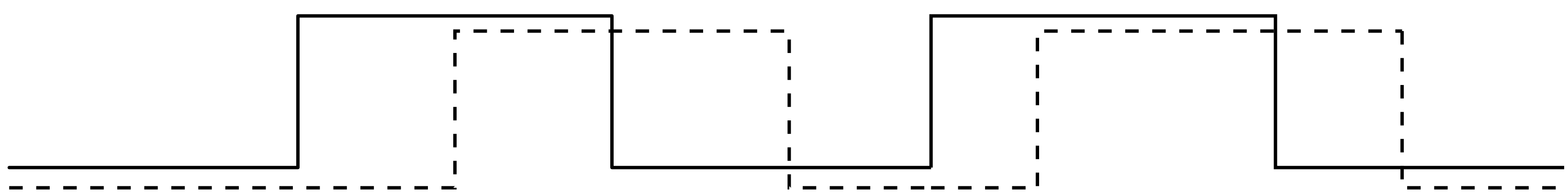
init
$a := 0$
$r := 0$

a_on
when
$a = 0$
$\underline{r = 0}$
then
$a := 1$
end

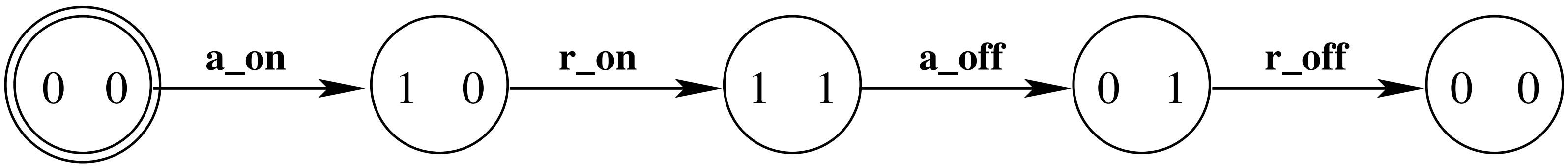
a_off
when
$a = 1$
$\underline{r = 1}$
then
$a := 0$
end

r_on
when
$a = 1$
$r = 0$
then
$r := 1$
end

r_off
when
$a = 0$
$r = 1$
then
$r := 0$
end

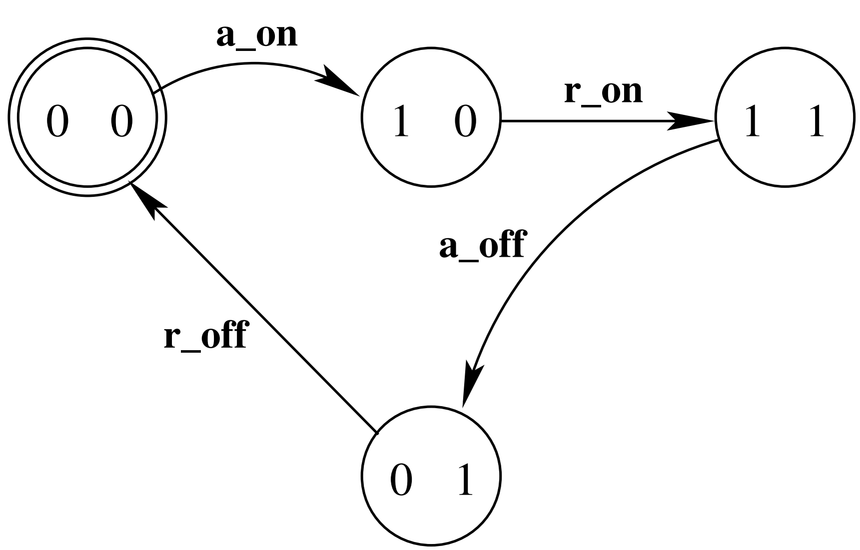


Mathematical Representations

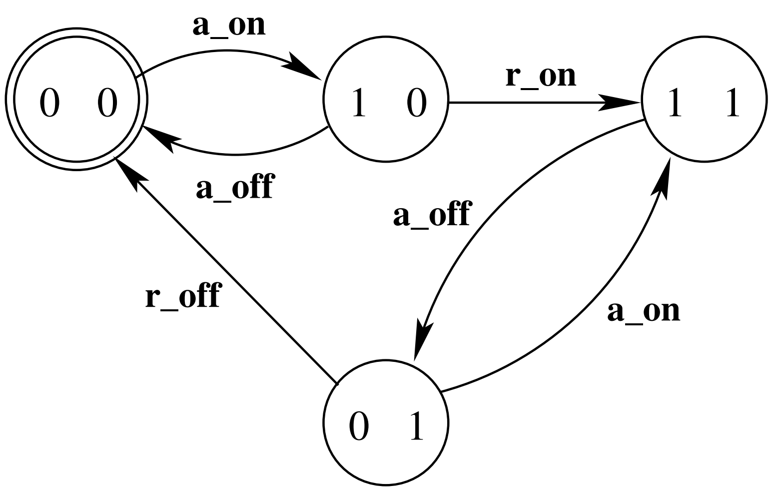


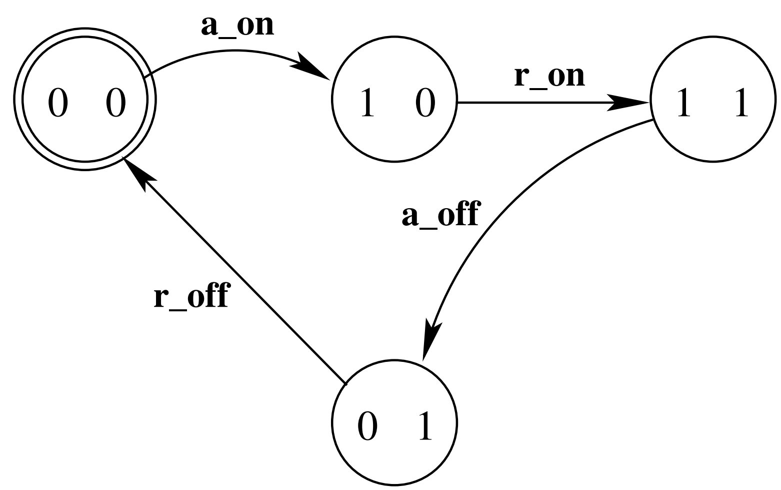
$$M = \{0 \mapsto 0\}$$

$$\begin{aligned} re = \{ \; & (0 \mapsto 0) \mapsto (1 \mapsto 0), \\ & (1 \mapsto 0) \mapsto (1 \mapsto 1), \\ & (1 \mapsto 1) \mapsto (0 \mapsto 1), \\ & (0 \mapsto 1) \mapsto (0 \mapsto 0) \; \} \end{aligned}$$



Comparing the Examples





- Traces of Example 2 are also traces of Example 1

- But trace inclusion is too strong to define refinement

- Trace continuation must be identical in abstraction and refinement

- Inclusion of concrete initializing set in abstract one

Simple Refinement: Formal Definition

$$L \subseteq S \qquad\qquad M \subseteq S$$

$$ae \in S \leftrightarrow S \qquad\qquad re \in S \leftrightarrow S$$

$$\boxed{\begin{array}{l} M \subseteq L \\ M \neq \varnothing \\ re \subseteq ae \\ \mathrm{dom}(ae) \subseteq \mathrm{dom}(re) \end{array}} \qquad \text{(I)}$$

$$
\begin{aligned}
L &= \{0 \mapsto 0\} \\
ae &= \{\ (0 \mapsto 0) \mapsto (1 \mapsto 0), \\
&\quad\ \ (0 \mapsto 1) \mapsto (1 \mapsto 1), \\
&\quad\ \ (1 \mapsto 0) \mapsto (0 \mapsto 0), \\
&\quad\ \ (1 \mapsto 1) \mapsto (0 \mapsto 1), \\
&\quad\ \ (1 \mapsto 0) \mapsto (1 \mapsto 1), \\
&\quad\ \ (0 \mapsto 1) \mapsto (0 \mapsto 0)\ \}
\end{aligned}
$$

$$
\begin{aligned}
M &= \{0 \mapsto 0\} \\
re &= \{\ (0 \mapsto 0) \mapsto (1 \mapsto 0), \\
&\quad\ \ (1 \mapsto 1) \mapsto (0 \mapsto 1), \\
&\quad\ \ (1 \mapsto 0) \mapsto (1 \mapsto 1), \\
&\quad\ \ (0 \mapsto 1) \mapsto (0 \mapsto 0)\ \}
\end{aligned}
$$

- Example 2 refines example 1

$$ae = ae_1 \cup \ldots \cup ae_n \qquad re = re_1 \cup \ldots \cup re_n$$

- Considering event containments

$$re_1 \subseteq ae_1 \ \wedge \ \ldots \ \wedge \ re_n \subseteq ae_n$$

- And possibly domain containments

$$\mathrm{dom}(ae_1) \subseteq \mathrm{dom}(re_1) \ \wedge \ \ldots \ \wedge \ \mathrm{dom}(ae_n) \subseteq \mathrm{dom}(re_n)$$

$$\begin{aligned}
&\boldsymbol{M} \subseteq \boldsymbol{L} \\
&\boldsymbol{M} \neq \varnothing \\
&\boldsymbol{re}_1 \subseteq \boldsymbol{ae}_1 \\
&\cdots \\
&\boldsymbol{re}_n \subseteq \boldsymbol{ae}_n \\
&\mathbf{dom}(\boldsymbol{ae}) \subseteq \mathbf{dom}(\boldsymbol{re})
\end{aligned} \qquad \text{(II)}$$

External and Internal Variables

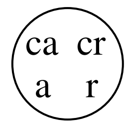
- We considered what can be observed from two system states

- What can be observed is just a convention

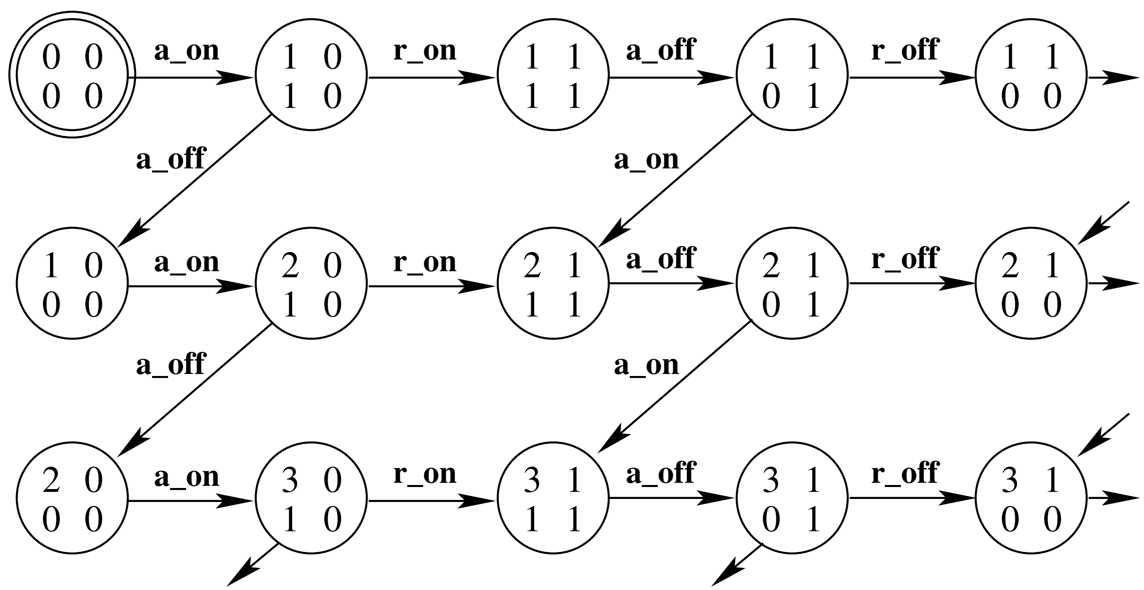
- It is possible that the real state is larger than what can be observed

- Variables ca and cr are considered internal

- Representation of the complete state



- The set of traces is now infinite



- Let f be the projection of the set of states S on the external set E

$$f \in S \rightarrow E$$

- We are also projecting the relation associated with each event

$$\begin{tikzcd}
& S \arrow[r, "ae_i"] \arrow[dl, "f"'] & S \arrow[dr, "f"] & \\
E & & & E \\
& S \arrow[r, "re_i"'] \arrow[ul, "f"] & S \arrow[ur, "f"'] &
\end{tikzcd}$$

$$
\begin{aligned}
&\boldsymbol{M} \subseteq \boldsymbol{L} \\
&\boldsymbol{M} \neq \varnothing \\
&\boldsymbol{f}^{-1} ; \boldsymbol{re}_1 ; \boldsymbol{f} \subseteq \boldsymbol{f}^{-1} ; \boldsymbol{ae}_1 ; \boldsymbol{f} \\
&\ldots \\
&\boldsymbol{f}^{-1} ; \boldsymbol{re}_n ; \boldsymbol{f} \subseteq \boldsymbol{f}^{-1} ; \boldsymbol{ae}_n ; \boldsymbol{f} \\
&\mathbf{dom}(\boldsymbol{ae}) \subseteq \mathbf{dom}(\boldsymbol{re})
\end{aligned}
\qquad \text{(III)}
$$

4. General Refinement

- We now suppose that when refining, the external set is changed

- We have thus an abstract external set E and a concrete one F

- But we want to be able to reconstruct E from F

- No loss of information

- We introduce a total function h mapping E to F

$$f \in S \rightarrow E \qquad g \in T \rightarrow F$$

$$h \in F \rightarrow E$$

$$ae_i \in S \leftrightarrow S \qquad re_i \in T \leftrightarrow T$$

$$\begin{tikzcd}
 & S \arrow[r, "ae_i"] \arrow[dl, "f"'] & S \arrow[dr, "f"] & \\
E & & & E \\
F \arrow[u, "h"] & & & F \arrow[u, "h"'] \\
 & T \arrow[ul, "g"] \arrow[r, "re_i"'] & T \arrow[ur, "g"'] &
\end{tikzcd}$$

- We compare $g^{-1} \,;\, re_i \,;\, g$ to $h \,;\, f^{-1} \,;\, ae_i \,;\, f \,;\, h^{-1}$

$$g[M] \subseteq h^{-1}[f[L]]$$

$$M \neq \varnothing$$

$$g^{-1} ; re_1 ; g \subseteq h ; f^{-1} ; ae_1 ; f ; h^{-1}$$

$$\ldots \qquad \text{(IV)}$$

$$g^{-1} ; re_n ; g \subseteq h ; f^{-1} ; ae_n ; f ; h^{-1}$$

$$h^{-1}[f[\mathrm{dom}(ae)]] \subseteq g[\mathrm{dom}(re)]$$

5. Sufficient Conditions for General Refinement

- We introduce a relation r linking the concrete and abstract states

- We introduce a relationship between r and h

- We introduce two possible additional conditions

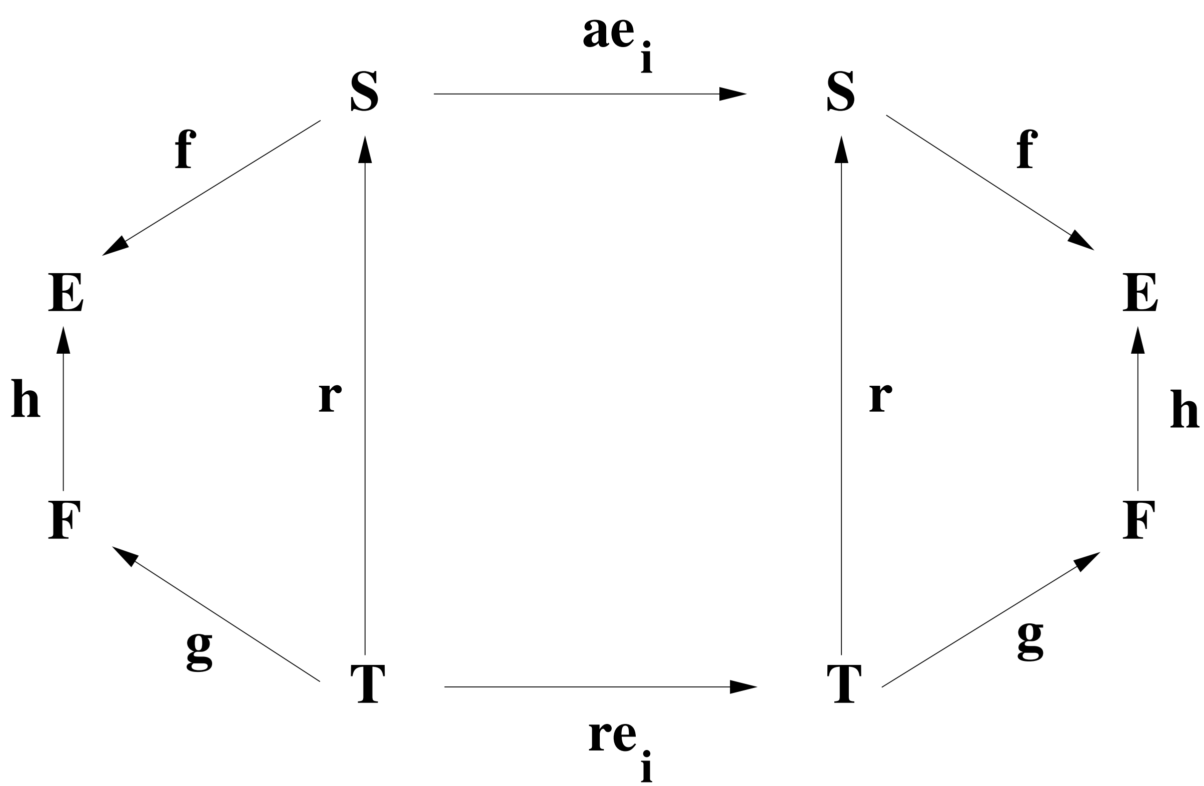
- This defines forward and backward simulations

- We reconstruct our refinement refinement Proof Obligations

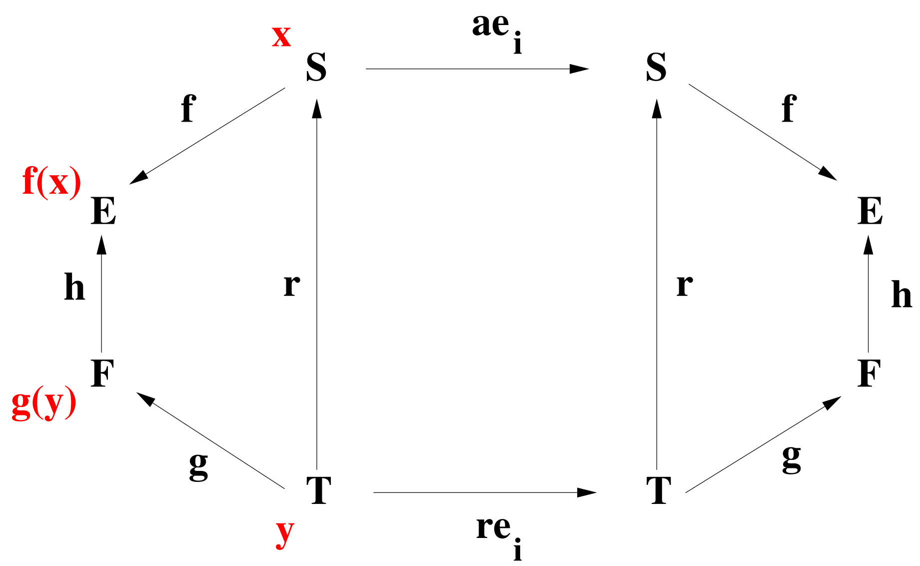
Sufficient Condition:Forward Simulation

- The relation r is total

$$\boxed{r \;\in\; T \twoheadleftarrow\!\!\!\rightarrow S}$$



First Condition: Relationship between r and h



$$\forall x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

The previous condition can be simplified to the following one:

$r^{-1}\,;\,g \;\subseteq\; f\,;\,h^{-1}$	C1

Proof

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

Proof

$$\forall x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

$\Leftrightarrow$

$$\forall x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,)$$

Proof

$$\forall x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

$$\Leftrightarrow$$

$$\forall x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall x, y, z \cdot (\, z = g(y) \;\wedge\; y \mapsto x \in r \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

Proof

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

$\Leftrightarrow$

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,)$$

$\Leftrightarrow$

$$\forall\, x, y, z \cdot (\, z = g(y) \;\wedge\; y \mapsto x \in r \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$\Leftrightarrow$

$$\forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$$
\begin{array}{ll}
& \forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,) \\
\Leftrightarrow & \\
& \forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,) \\
\Leftrightarrow & \\
& \forall\, x, y, z \cdot (\, z = g(y) \;\wedge\; y \mapsto x \in r \;\Rightarrow\; z \mapsto f(x) \in h\,) \\
\Leftrightarrow & \\
& \forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \;\Rightarrow\; z \mapsto f(x) \in h\,) \\
\Leftrightarrow & \\
& \forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \Rightarrow \\
& \qquad\qquad\qquad\qquad \exists u \cdot (u = f(x) \;\wedge\; z \mapsto u \in h\,)\,)
\end{array}
$$

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

$$\Leftrightarrow$$

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, y, z \cdot (\, z = g(y) \;\wedge\; y \mapsto x \in r \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \Rightarrow \exists u \cdot (u = f(x) \;\wedge\; z \mapsto u \in h\,)\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (x \mapsto y \in r^{-1} \;\wedge\; y \mapsto z \in g\,) \Rightarrow \exists u \cdot (x \mapsto u \in f \;\wedge\; u \mapsto z \in h^{-1}\,)\,)$$

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; f(x) = h(g(y))\,)$$

$$\Leftrightarrow$$

$$\forall\, x, y \cdot (\, y \mapsto x \in r \;\Rightarrow\; g(y) \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, y, z \cdot (\, z = g(y) \;\wedge\; y \mapsto x \in r \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \;\Rightarrow\; z \mapsto f(x) \in h\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (z = g(y) \;\wedge\; y \mapsto x \in r\,) \Rightarrow \exists u \cdot (u = f(x) \;\wedge\; z \mapsto u \in h\,)\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, \exists y \cdot (x \mapsto y \in r^{-1} \;\wedge\; y \mapsto z \in g\,) \Rightarrow \exists u \cdot (x \mapsto u \in f \;\wedge\; u \mapsto z \in h^{-1}\,)\,)$$

$$\Leftrightarrow$$

$$\forall\, x, z \cdot (\, x \mapsto z \in (r^{-1} \,;\, g) \;\Rightarrow\; x \mapsto z \in (f \,;\, h^{-1})\,)$$

Proof

$$\forall x,y\cdot(\,y\mapsto x\in r\;\Rightarrow\;f(x)=h(g(y))\,)$$

$$\Leftrightarrow$$

$$\forall x,y\cdot(\,y\mapsto x\in r\;\Rightarrow\;g(y)\mapsto f(x)\in h\,)$$

$$\Leftrightarrow$$

$$\forall x,y,z\cdot(\,z=g(y)\;\wedge\;y\mapsto x\in r\;\Rightarrow\;z\mapsto f(x)\in h\,)$$

$$\Leftrightarrow$$

$$\forall x,z\cdot(\,\exists y\cdot(z=g(y)\;\wedge\;y\mapsto x\in r\,)\;\Rightarrow\;z\mapsto f(x)\in h\,)$$

$$\Leftrightarrow$$

$$\forall x,z\cdot(\,\exists y\cdot(z=g(y)\;\wedge\;y\mapsto x\in r\,)\Rightarrow \exists u\cdot(u=f(x)\;\wedge\;z\mapsto u\in h\,)\,)$$

$$\Leftrightarrow$$

$$\forall x,z\cdot(\,\exists y\cdot(x\mapsto y\in r^{-1}\;\wedge\;y\mapsto z\in g\,)\Rightarrow \exists u\cdot(x\mapsto u\in f\;\wedge\;u\mapsto z\in h^{-1}\,)\,)$$

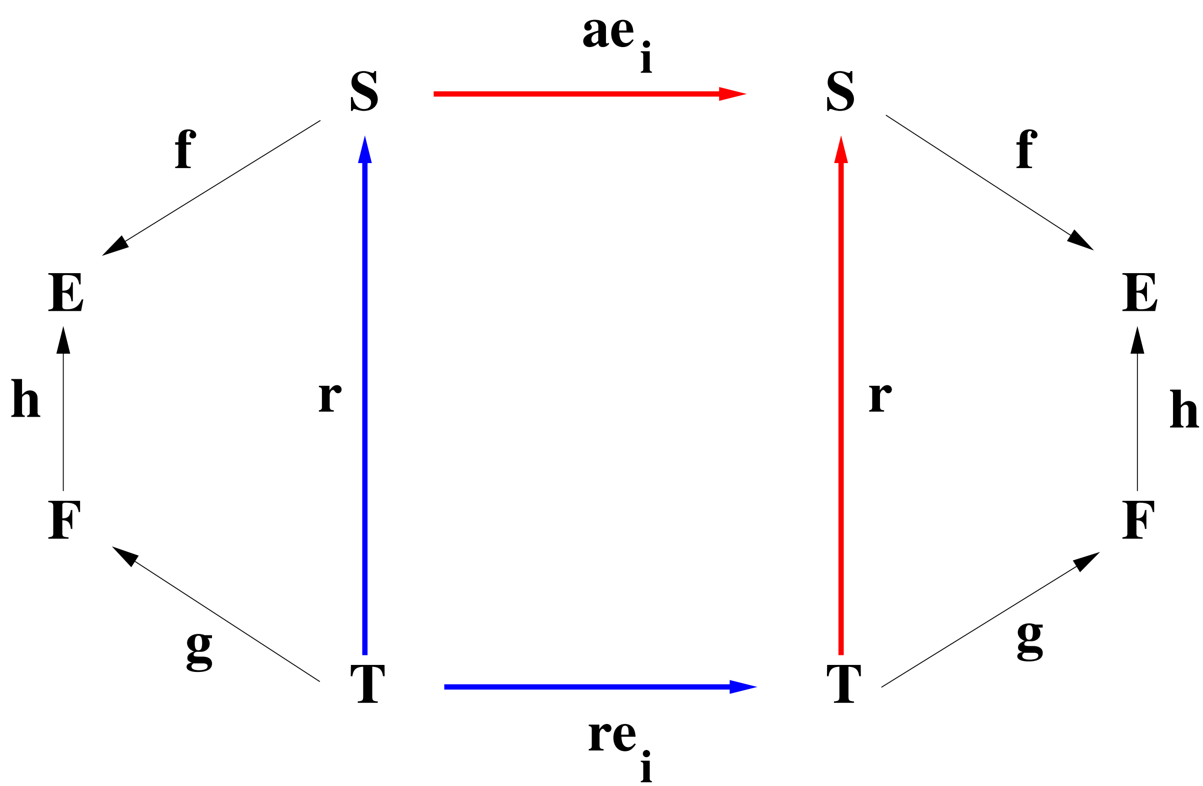
$$\Leftrightarrow$$

$$\forall x,z\cdot(\,x\mapsto z\in(r^{-1}\,;g)\;\Rightarrow\;x\mapsto z\in(f\,;h^{-1})\,)$$

$$\Leftrightarrow$$

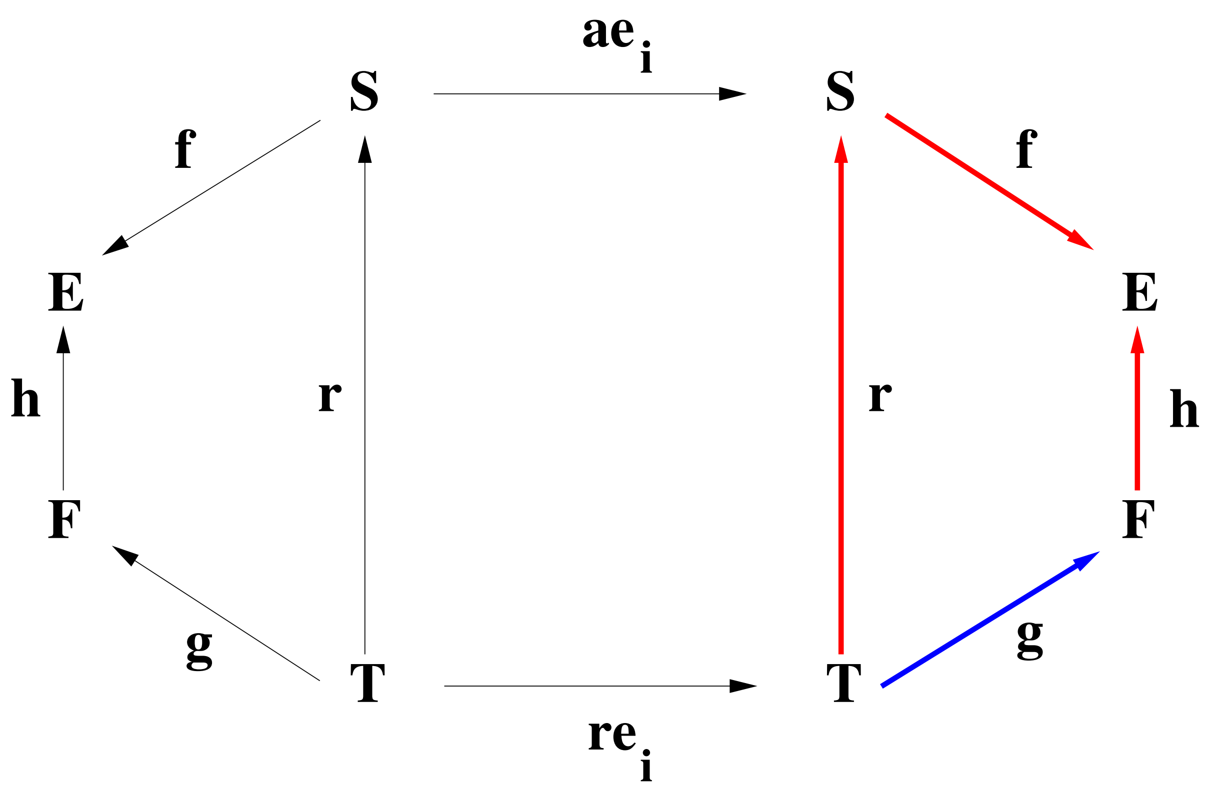
$$r^{-1}\,;g\;\;\subseteq\;\;f\,;h^{-1}$$

More Conditions (1)



$r^{-1}\,;re_i \quad \subseteq \quad ae_i\,;r^{-1}$	C2

More Conditions (2)



$g^{-1} \subseteq h\,;\,f^{-1}\,;\,r^{-1}$	C3

- C3 can be deduced from C1

	$r^{-1} ; g \subseteq f ; h^{-1}$	C1
$\Rightarrow$		Set Theory
	$r ; r^{-1} ; g \subseteq r ; f ; h^{-1}$	
$\Rightarrow$		**id** $(T) \subseteq r ; r^{-1}$ since $r \in T \leftrightarrow S$
	$g \subseteq r ; f ; h^{-1}$	
$\Leftrightarrow$		Set Theory
	$g^{-1} \subseteq h ; f^{-1} ; r^{-1}$	C3

$r^{-1}\,;g \quad\subseteq\quad f\,;h^{-1}$	C1
$r^{-1}\,;re_i \quad\subseteq\quad ae_i\,;r^{-1}$	C2
$g^{-1} \quad\subseteq\quad h\,;f^{-1}\,;r^{-1}$	C3

$$
\begin{array}{lll}
 & \underline{g^{-1}}\,;re_i\,;g & \\
\subseteq & & \text{C3} \\
 & h\,;f^{-1}\,;\underline{r^{-1}\,;re_i}\,;g & \\
\subseteq & & \text{C2} \\
 & h\,;f^{-1}\,;ae_i\,;\underline{r^{-1}\,;g} & \\
\subseteq & & \text{C1} \\
 & h\,;f^{-1}\,;ae_i\,;f\,;h^{-1} &
\end{array}
$$

Summary of Forward Simulation

$$\boxed{\boldsymbol{r^{-1} \,;\, re_i \;\;\subseteq\;\; ae_i \,;\, r^{-1}}}$$

$$\boxed{\boldsymbol{M \;\subseteq\; r^{-1}[L]}}$$

variables: w

inv1: $J(v, w)$

init
$w :| N(w')$

event$_i$
when
$H_i(w)$
then
$w :| S_i(w, w')$
end

- The previous concrete model is supposed to refine the following abstract one:

variables: v

inv0: $I(v)$

init
$v :| K(v')$

event$_i$
when
$G_i(v)$
then
$v :| R_i(v, v')$

$\boldsymbol{S}$	$\{\, v \mid I(v) \,\}$
$\boldsymbol{T}$	$\{\, w \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,)\,\}$
$\boldsymbol{L}$	$\{\, v \mid K(v) \,\}$
$\boldsymbol{M}$	$\{\, w \mid N(w) \,\}$
ae_i	$\{\, v \mapsto v' \mid I(v) \;\wedge\; G_i(v) \;\wedge\; R_i(v,v') \,\}$
re_i	$\{\, w \mapsto w' \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w) \;\wedge\; S_i(w,w') \,\}$
r	$\{\, w \mapsto v \mid I(v) \;\wedge\; J(v,w) \,\}$
$\mathrm{dom}\,(ae_i)$	$\{\, v \mid I(v) \;\wedge\; G_i(v) \,\}$
$\mathrm{dom}\,(re_i)$	$\{\, w \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w) \,\}$

$\boldsymbol{L}$	$\{\, \boldsymbol{v} \mid \boldsymbol{K}(\boldsymbol{v}) \,\}$
$\boldsymbol{M}$	$\{\, \boldsymbol{w} \mid \boldsymbol{N}(\boldsymbol{w}) \,\}$
$\boldsymbol{r}$	$\{\, \boldsymbol{w} \mapsto \boldsymbol{v} \mid \boldsymbol{I}(\boldsymbol{v}) \;\land\; \boldsymbol{J}(\boldsymbol{v}, \boldsymbol{w}) \,\}$

$\boldsymbol{M} \;\subseteq\; \boldsymbol{r}^{-1}[\boldsymbol{L}] \quad \rightsquigarrow$

$\boldsymbol{N}(\boldsymbol{w})$ $\vdash$ $\exists \boldsymbol{v} \cdot (\; \boldsymbol{K}(\boldsymbol{v}) \;\land\; \boldsymbol{J}(\boldsymbol{v}, \boldsymbol{w}) \;)$	INI_INV_REF

re_i	$\{\, w \mapsto w' \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w) \;\wedge\; S_i(w,w')\,\}$
$\mathrm{dom}\,(re_i)$	$\{\, w \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w)\,\}$

Here is the domain of re_i

$$\{\, w \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w) \;\wedge\; \exists w' \cdot S_i(w,w')\,\}$$

$\leadsto$

$I(v)$ $J(v,w)$ $H_i(w)$ $\vdash$ $\exists\, w' \cdot S_i(w,w')$	**FIS_REF**

General Refinement: the Rules (3)

ae_i	$\{\, v \mapsto v' \mid I(v) \;\wedge\; G_i(v) \;\wedge\; R_i(v,v') \,\}$
re_i	$\{\, w \mapsto w' \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w) \,) \;\wedge\; H_i(w) \;\wedge\; S_i(w,w') \,\}$
r	$\{\, w \mapsto v \mid I(v) \;\wedge\; J(v,w) \,\}$

$$r^{-1} \,;\, re_i \;\subseteq\; ae_i \,;\, r^{-1}$$

$I(v)$ $J(v,w)$ $H_i(w)$ $\vdash$ $G_i(v)$	**GRD_REF**

$I(v)$ $J(v,w)$ $H_i(w)$ $S_i(w,w')$ $\vdash$ $\exists\, v' \cdot \left(\begin{array}{l} R_i(v,v') \\ J(v',w') \end{array} \right)$	**INV_REF**

General Refinement: the Rules (4)

r	$\{\, w \mapsto v \mid I(v) \;\wedge\; J(v,w) \,\}$
$\mathrm{dom}\,(ae_i)$	$\{\, v \mid I(v) \;\wedge\; G_i(v) \,\}$
$\mathrm{dom}\,(re_i)$	$\{\, w \mid \exists v \cdot (\, I(v) \;\wedge\; J(v,w)\,) \;\wedge\; H_i(w) \,\}$

$$r^{-1}[\mathrm{dom}(ae)] \;\subseteq\; \mathrm{dom}(re)$$

$\leadsto$

$I(v)$ $J(v,w)$ $G_1(v) \;\vee\; \ldots \;\vee\; G_n(v)$ $\vdash$ $H_1(w) \;\vee\; \ldots \;\vee\; H_n(w)$	**DLF_REF**

$r^{-1} ; g \subseteq f ; h^{-1}$	C1
$r^{-1} ; re_i \subseteq ae_i ; r^{-1}$	C2
$g^{-1} \subseteq h ; f^{-1} ; r^{-1}$	C3

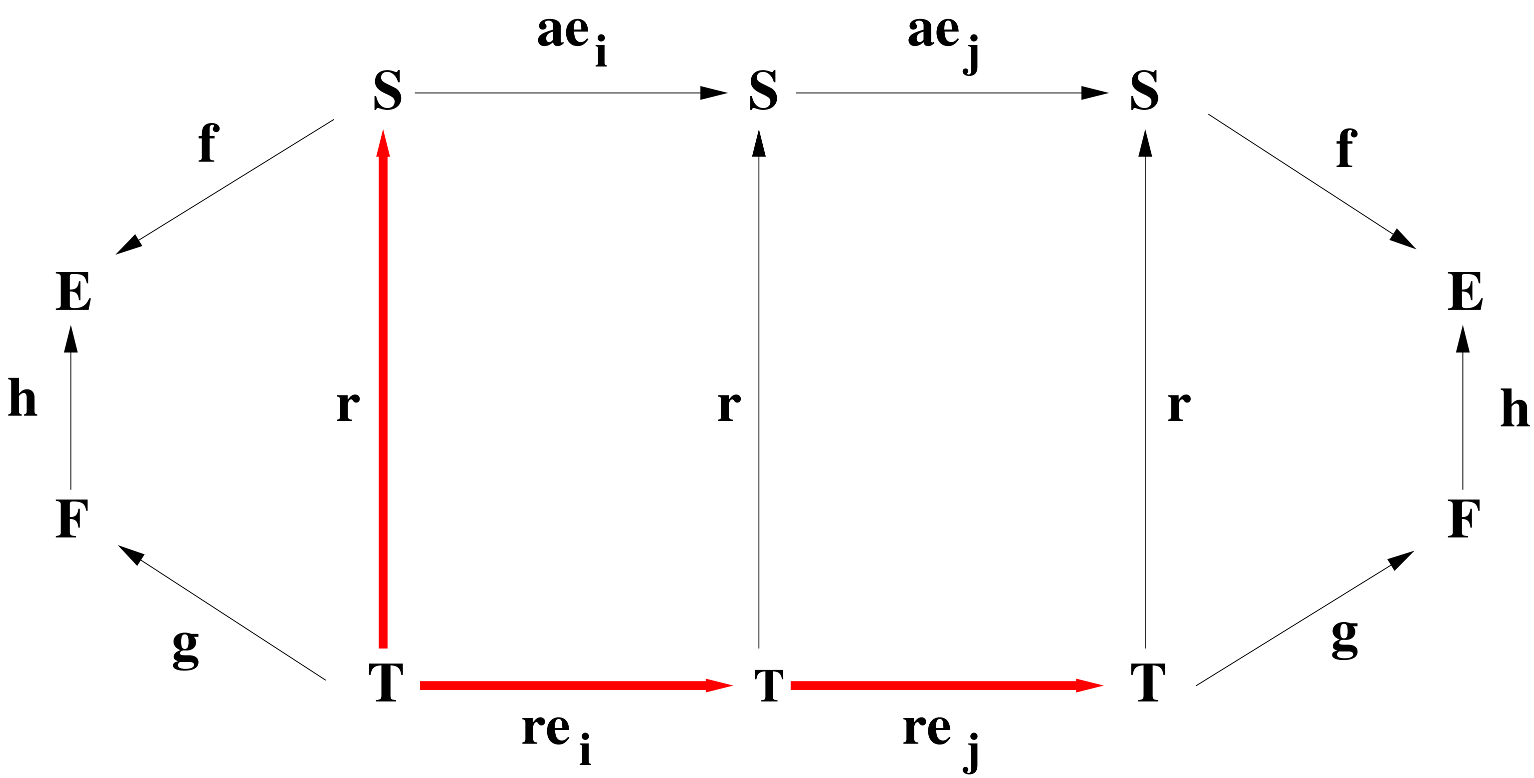
$r^{-1} ; g \subseteq f ; h^{-1}$	C1
$r^{-1} ; re_i^{-1} \subseteq ae_i^{-1} ; r^{-1}$	C2’
$g^{-1} \subseteq h ; f^{-1} ; r^{-1}$	C3

- Conditions C1, C2’, and C3 are identical to the following:

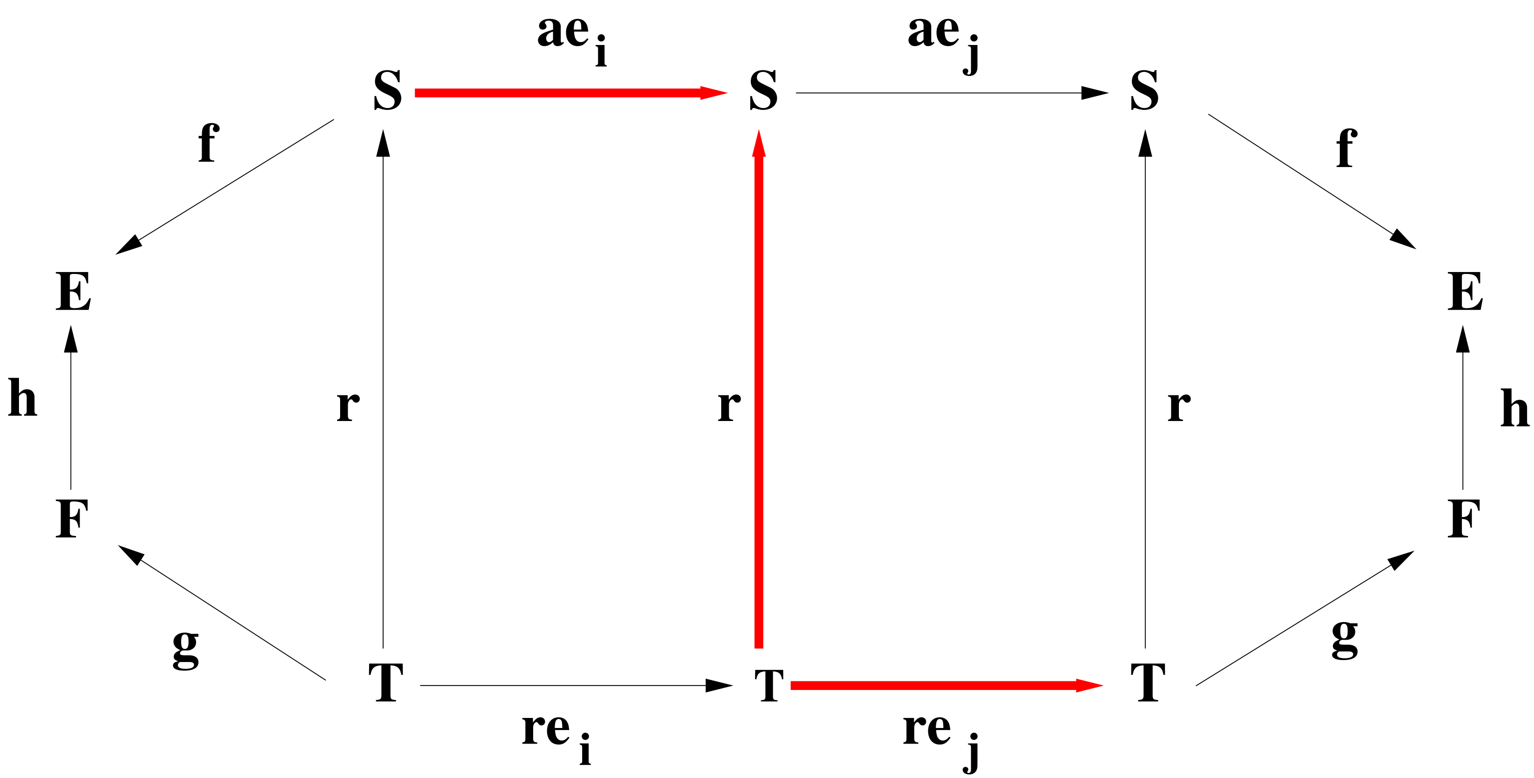
$g^{-1} ; r \subseteq h ; f^{-1}$	D1
$re_i ; r \subseteq r ; ae_i$	D2’
$g \subseteq r ; f ; h^{-1}$	D3

$g^{-1} ; r \subseteq h ; f^{-1}$	D1
$re_i ; r \subseteq r ; ae_i$	D2'
$g \subseteq r ; f ; h^{-1}$	D3

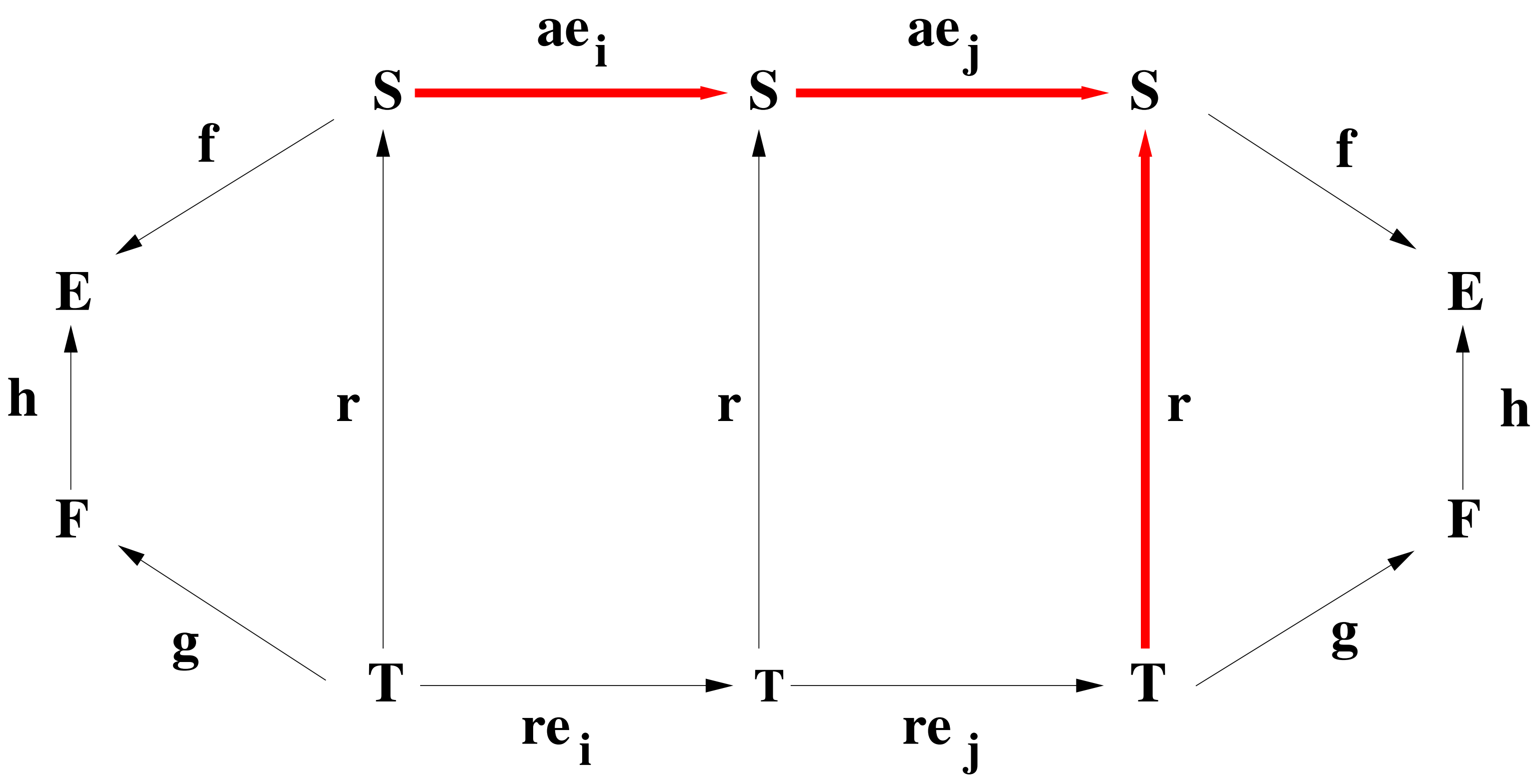
$$
\begin{array}{lll}
 & g^{-1} ; re_i ; \underline{g} & \\
\subseteq & & \text{D3} \\
 & g^{-1} ; \underline{re_i ; r} ; f ; h^{-1} & \\
\subseteq & & \text{D2'} \\
 & \underline{g^{-1} ; r} ; ae_i ; f ; h^{-1} & \\
\subseteq & & \text{D1} \\
 & h ; f^{-1} ; ae_i ; f ; h^{-1} &
\end{array}
$$



$$r^{-1}\,;re_i \;\subseteq\; ae_i\,;r^{-1}$$



$$r^{-1}\,;re_j \;\subseteq\; ae_j\,;r^{-1}$$



$$r^{-1} \,;\, re_i \,;\, re_j \quad \subseteq \quad ae_i \,;\, ae_j \,;\, r^{-1}$$

- An abstract event ae_i is split into 2 events re_{i1} and re_{i2}.

- One simply proves that these events both refine ae_i.

Merging

- It is also possible to merge two abstract events ae_i and ae_j

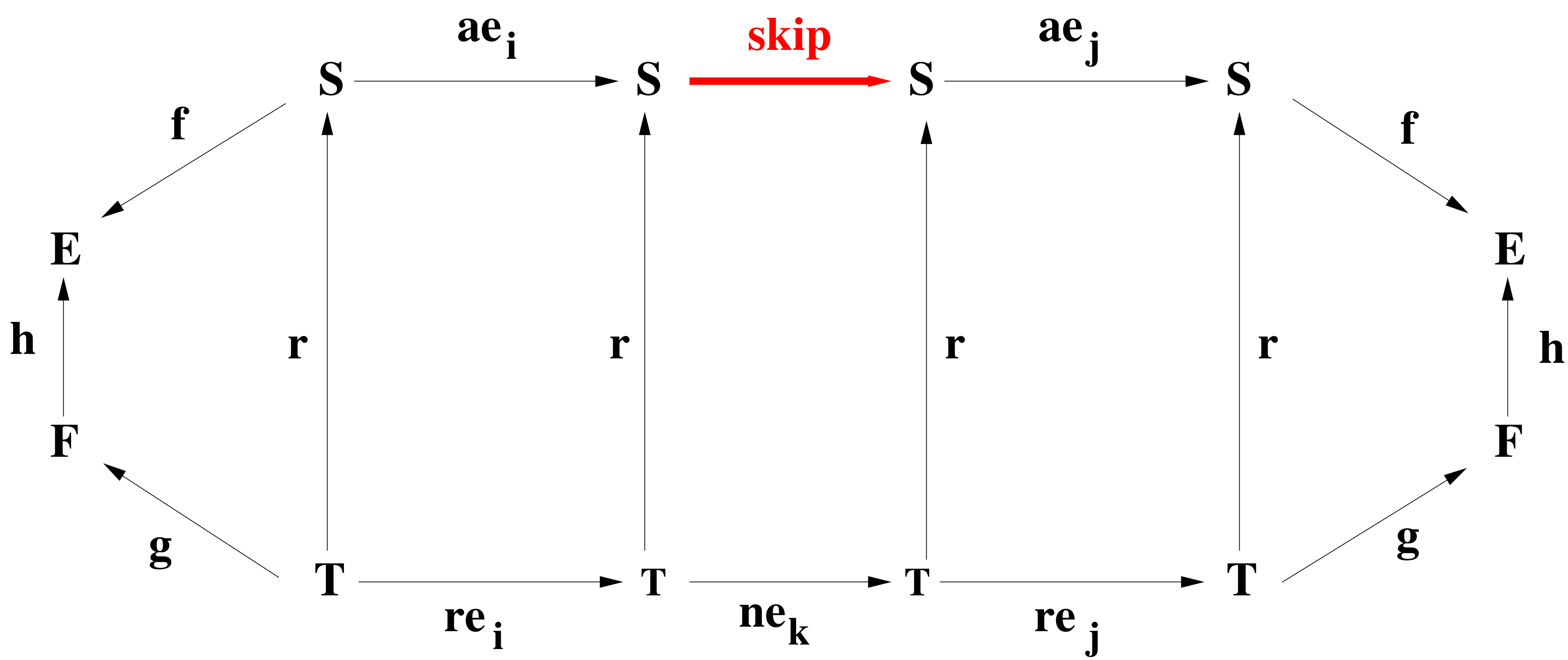
- We form a single refined event re_{ij}.

- One has simply to prove that re_{ij} refines $ae_i \cup ae_j$.

- This is very interesting when events ae_i and ae_j have the following shape:

event$_i$ **when** P Q **then** S **end**		event$_j$ **when** $\neg P$ Q **then** S **end**	$\leadsto$	event$_{ij}$ **when** Q **then** S **end**

- A new event refines skip



$$r^{-1}\,;ne_k \;\subseteq\; r^{-1}$$

new_event$_k$
when
$N_k(w)$
then
$w :| T_k(w, w')$
end

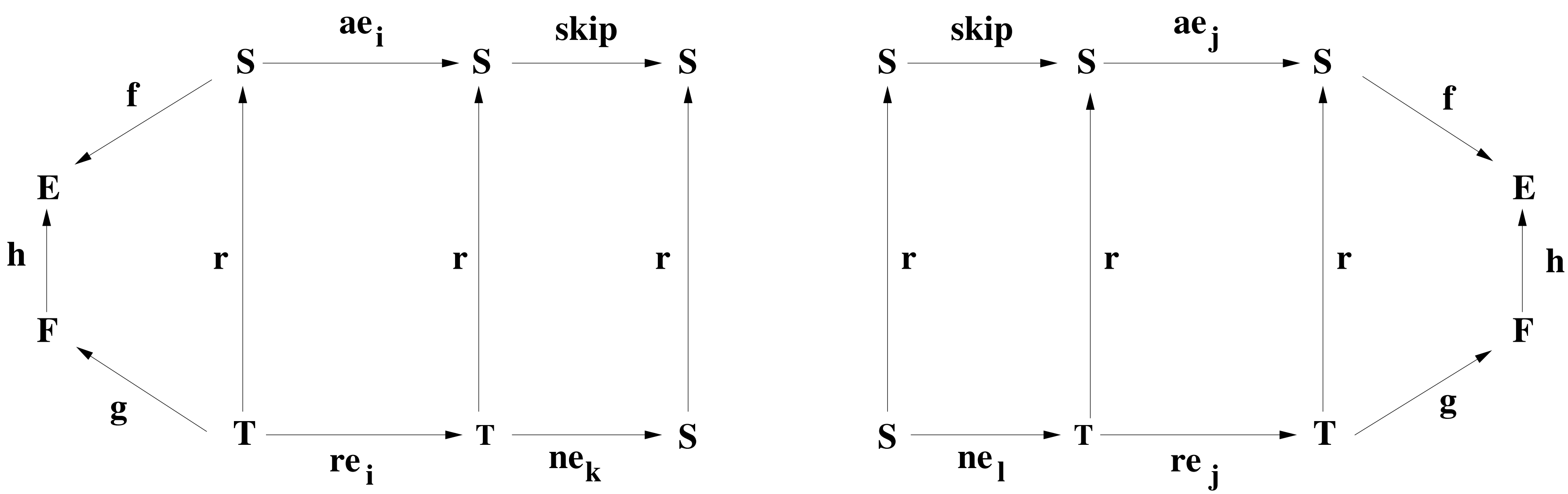
Invariant preservation requires an adaptation of rule INV_REF:

$I(v)$ $J(v, w)$ $H_i(w)$ $S_i(w, w')$ $\vdash$ $\exists\, v' \cdot \begin{pmatrix} R_i(v, v') \\ J(v', w') \end{pmatrix}$	**INV_REF**	$\leadsto$	$I(v)$ $J(v, w)$ $N_k(w)$ $T_k(w, w')$ $\vdash$ $J(v, w')$	**INV_REF**

- Modification of the Deadlock freedom rule

$I(v)$ $J(v,w)$ $G_1(v) \;\vee\; \ldots \;\vee\; G_n(v)$ $\vdash$ $H_1(w) \;\vee\; \ldots \;\vee\; H_n(w) \;\vee\; N_1(w) \;\vee\; \ldots \;\vee\; N_m(w)$	**DLF_REF**

Introducing New Events: No Divergence



- Exhibiting a decreasing variant

$I(v)$ $J(v,w)$ $N_k(w)$ $\vdash$ $V(w) \; \in \; \mathbb{N}$	WFD_REF1

$I(v)$ $J(v,w)$ $N_k(w)$ $T_k(w,w')$ $\vdash$ $V(w') < V(w)$	WFD_REF2

5. Decomposition

Decomposition: Basic Model

$$\boldsymbol{s} \in \boldsymbol{S} \leftrightarrow \boldsymbol{S}$$

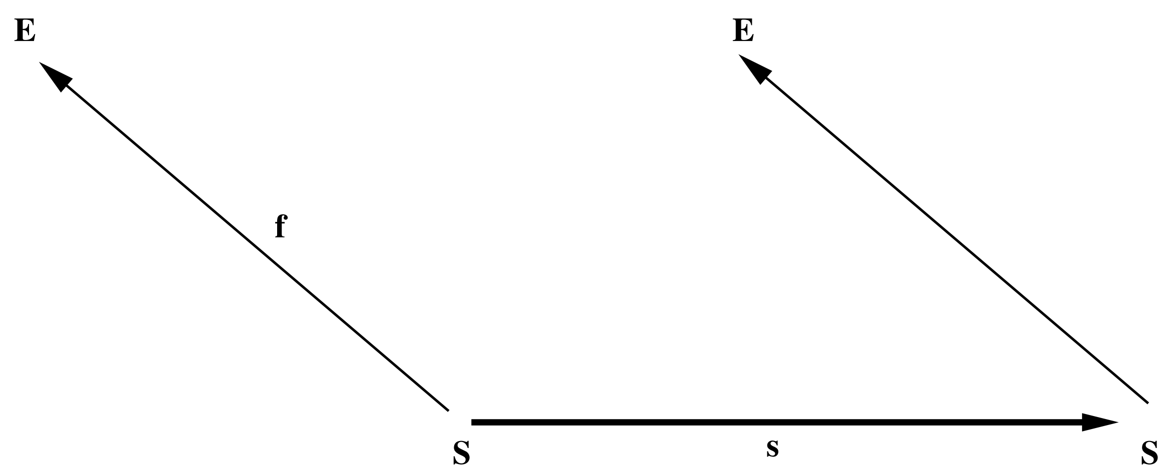
This can be represented by the following simple diagram:

$$\mathbf{S} \xrightarrow{\mathbf{s}} \mathbf{S}$$

The state has got some external variables belonging to the set $\boldsymbol{E}$. The projection of the state from $\boldsymbol{S}$ to this external set $\boldsymbol{E}$ is defined by means of the following function $\boldsymbol{f}$:

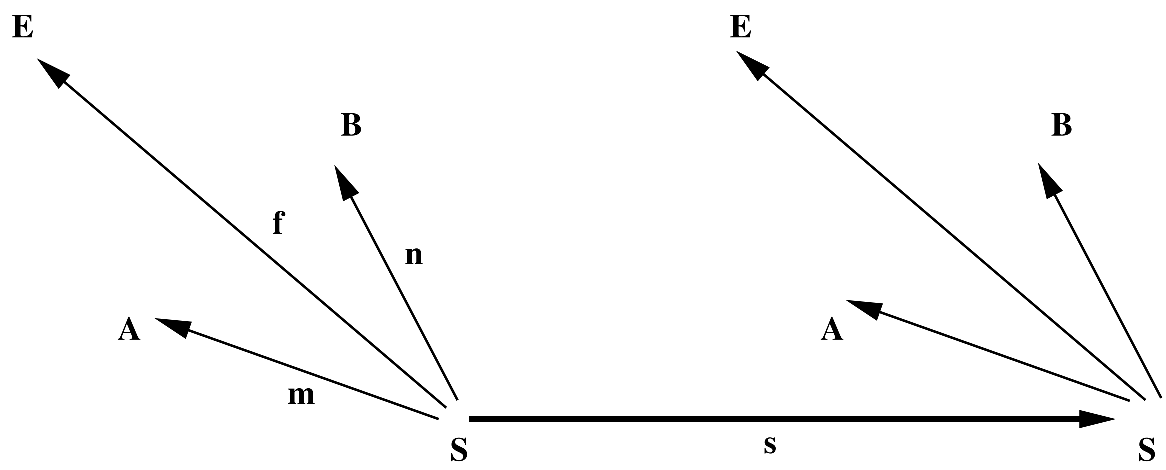
$$\boldsymbol{f} \;\in\; \boldsymbol{S} \longrightarrow \boldsymbol{E}$$

This can be represented by extending our previous diagram as follows:



$$m \in S \rightarrow A \qquad n \in S \rightarrow B$$

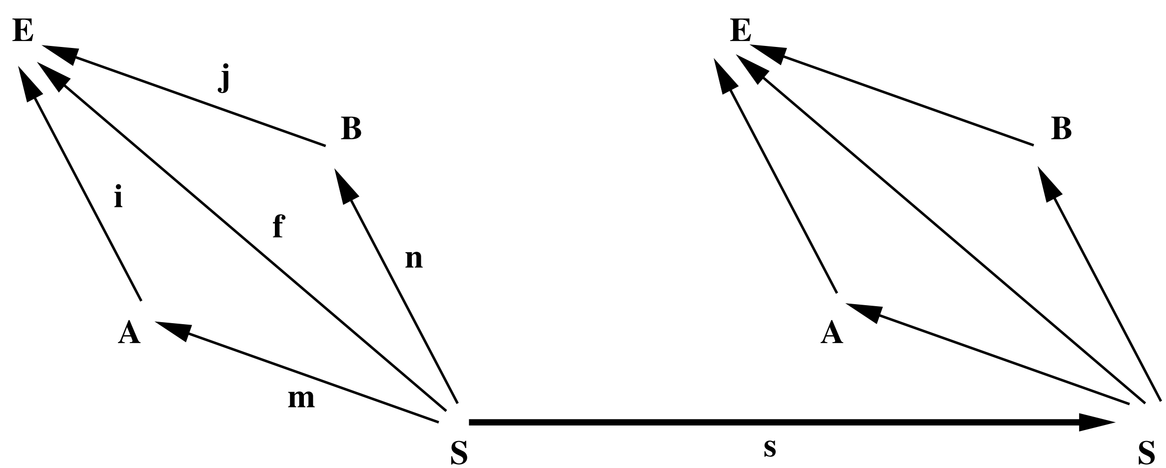
This can be represented by extending our previous diagram as follows:



$$i \in A \rightarrow E \qquad j \in B \rightarrow E$$

These two functions are related to the projection function f as follows:

$$f = m\,;i \qquad f = n\,;j$$



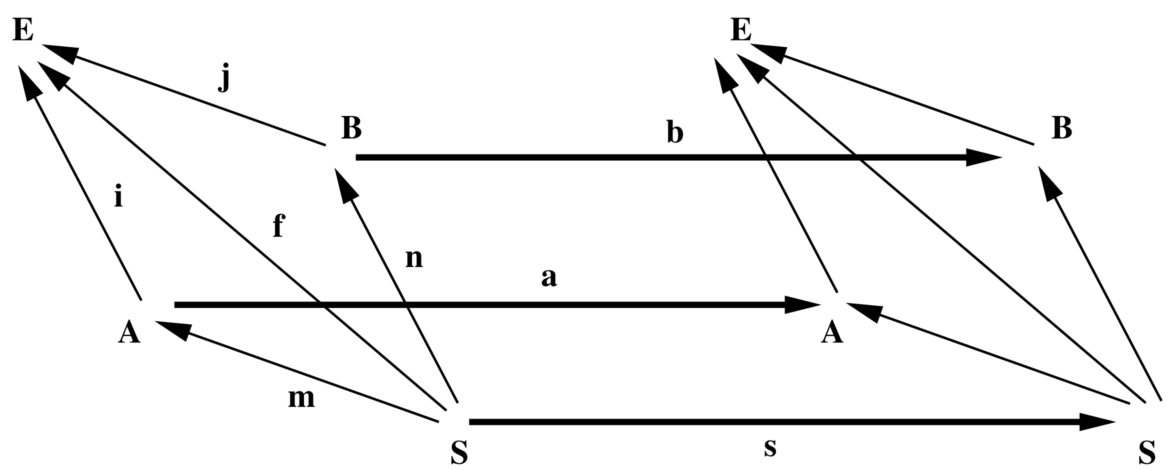
Likewise, event $\boldsymbol{s}$ is *decomposed* into two events $\boldsymbol{a}$ and $\boldsymbol{b}$. Formally:

$$\boldsymbol{a} \;\in\; \boldsymbol{A} \leftrightarrow \boldsymbol{A} \qquad \boldsymbol{b} \;\in\; \boldsymbol{B} \leftrightarrow \boldsymbol{B}$$

These events are the projections of event $\boldsymbol{s}$ on the sets $\boldsymbol{A}$ and $\boldsymbol{B}$. Formally:

$$\boldsymbol{a} \;=\; \boldsymbol{m}^{-1}\,;\boldsymbol{s}\,;\boldsymbol{m} \qquad \boldsymbol{b} \;=\; \boldsymbol{n}^{-1}\,;\boldsymbol{s}\,;\boldsymbol{n}$$

This can be represented by extending our previous diagram as follows:



$$\boldsymbol{f^{-1} \,;\, s \,;\, f \;\subseteq\; i^{-1} \,;\, a \,;\, i} \qquad \boldsymbol{f^{-1} \,;\, s \,;\, f \;\subseteq\; j^{-1} \,;\, b \,;\, j}$$

Here are the proofs of these statements:

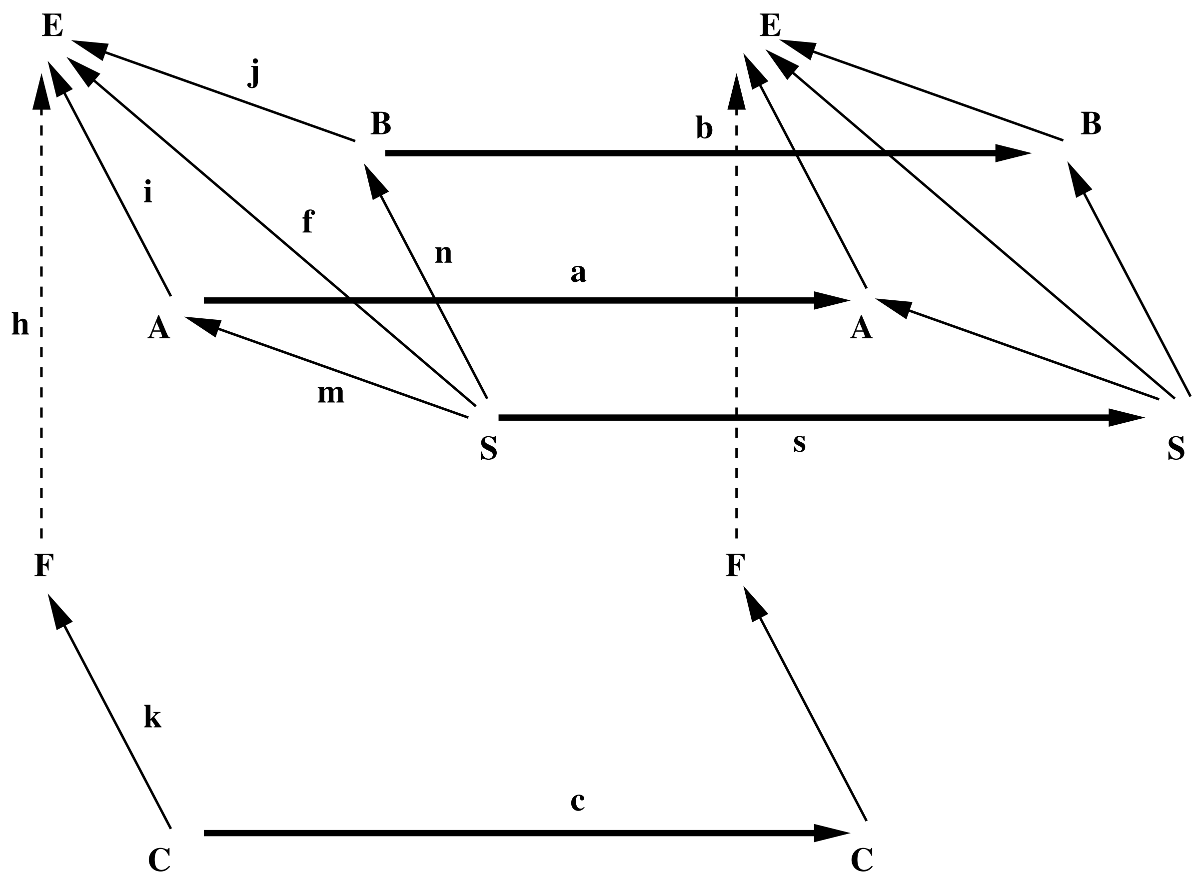
$$
\begin{aligned}
& f^{-1} \, ; s \, ; f \\
= & \\
& i^{-1} \, ; m^{-1} \, ; s \, ; m \, ; i \\
= & \\
& i^{-1} \, ; a \, ; i
\end{aligned}
\qquad\qquad
\begin{aligned}
& f^{-1} \, ; s \, ; f \\
= & \\
& j^{-1} \, ; n^{-1} \, ; s \, ; n \, ; j \\
= & \\
& j^{-1} \, ; b \, ; j
\end{aligned}
$$

$$\begin{aligned} \boldsymbol{k} &\in \boldsymbol{C} \longrightarrow \boldsymbol{F} \\ \boldsymbol{h} &\in \boldsymbol{F} \longrightarrow \boldsymbol{E} \end{aligned}$$

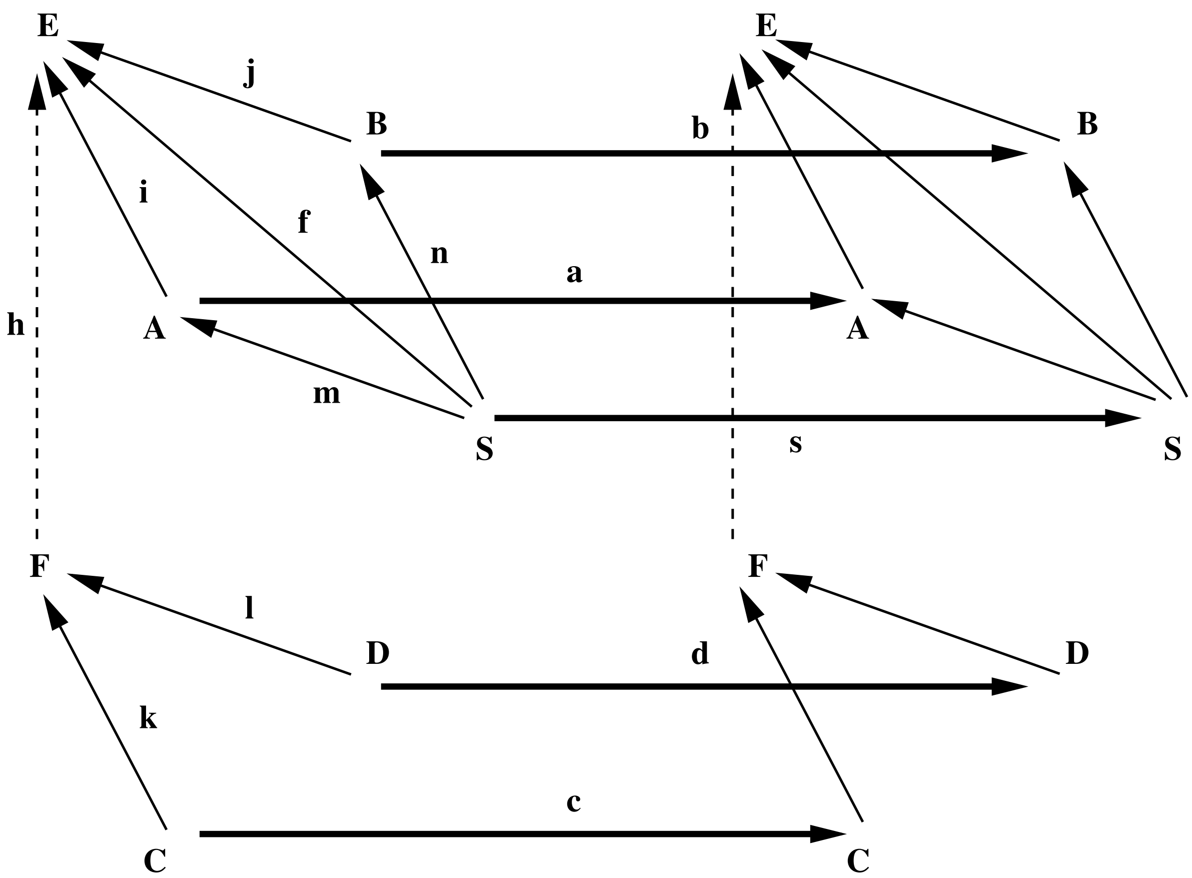
The refinement of $\boldsymbol{a}$ to $\boldsymbol{c}$ is formally expressed by applying rule REF yielding:

$$\boldsymbol{k}^{-1} \,;\, \boldsymbol{c} \,;\, \boldsymbol{k} \;\subseteq\; \boldsymbol{h} \,;\, \boldsymbol{i}^{-1} \,;\, \boldsymbol{a} \,;\, \boldsymbol{i} \,;\, \boldsymbol{h}^{-1}$$

Refining the Sub-systems



Refining the Sub-systems

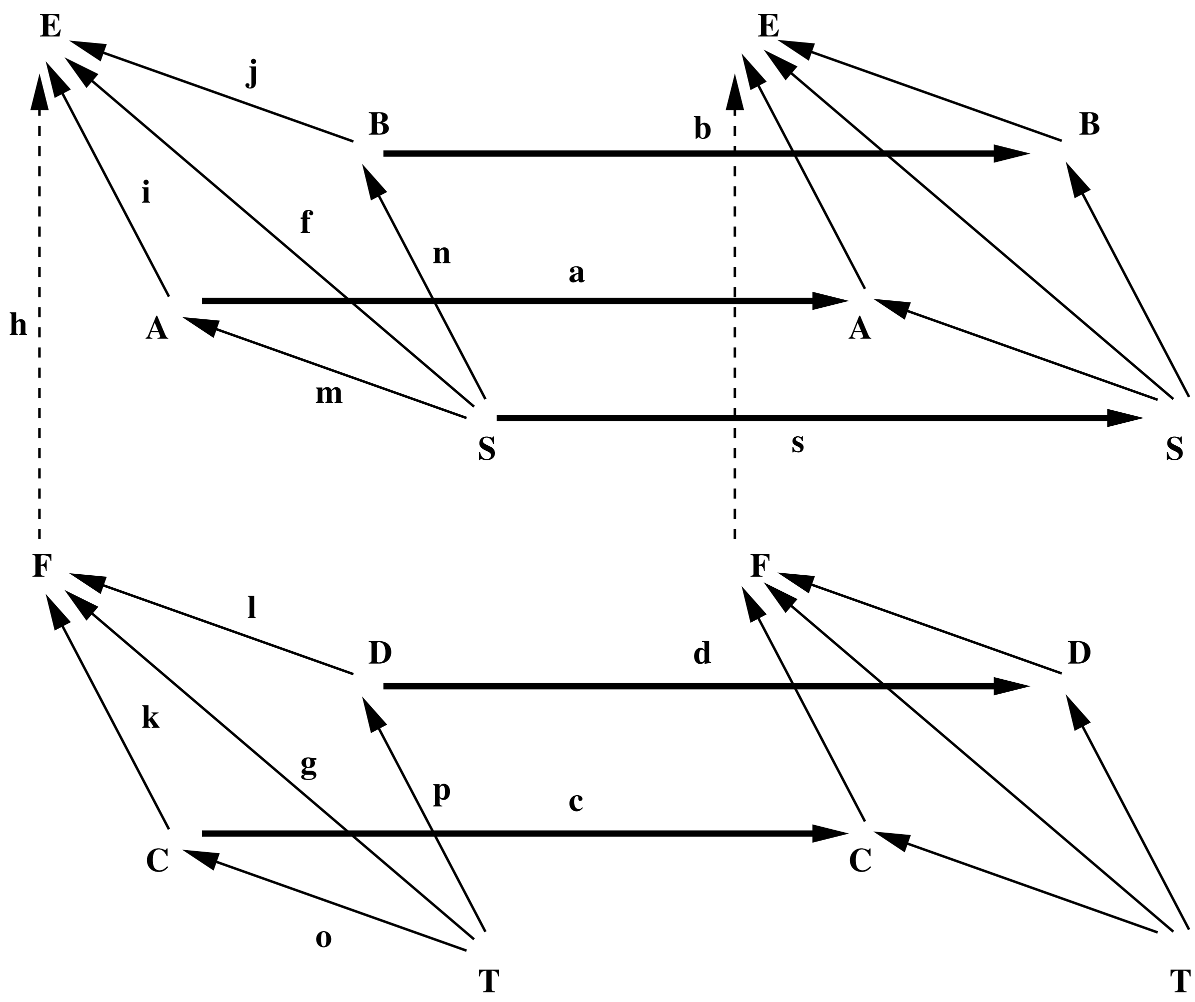


$$\begin{aligned} o &\in T \rightarrow C \\ p &\in T \rightarrow D \\ g &\in T \rightarrow F \end{aligned}$$

The fact that the external set F corresponds to the same projection is expressed by the following property:

$$\begin{aligned} g &= o \, ; k \\ g &= p \, ; l \end{aligned}$$

Re-composing



$$g^{-1} ; t ; g \quad \subseteq \quad h ; f^{-1} ; s ; f ; h^{-1}$$

that is

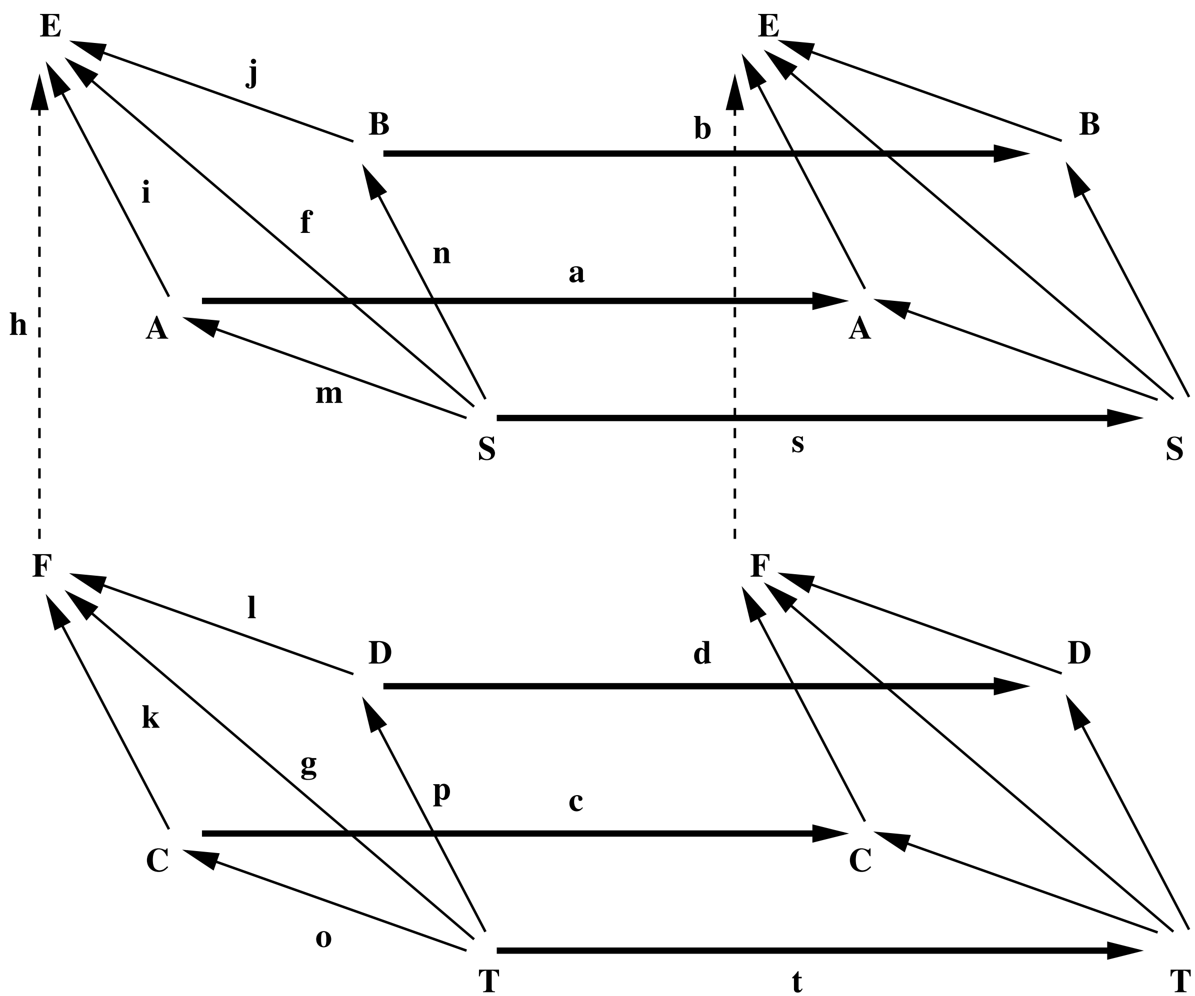
$$g^{-1} ; ((o ; c ; o^{-1}) \;\cap\; (p ; d ; p^{-1})) ; g \quad \subseteq \quad h ; f^{-1} ; s ; f ; h^{-1}$$

For proving this, it is sufficient to prove the following statements:

$$g^{-1} ; o ; c ; o^{-1} ; g \quad \subseteq \quad h ; f^{-1} ; s ; f ; h^{-1}$$

$$g^{-1} ; p ; d ; p^{-1} ; g \quad \subseteq \quad h ; f^{-1} ; s ; f ; h^{-1}$$

$$
\begin{aligned}
& g^{-1} ; o ; c ; o^{-1} ; g \\
= \\
& k^{-1} ; o^{-1} ; o ; c ; o^{-1} ; o ; k \\
\subseteq \\
& k^{-1} ; c ; k \\
\subseteq \\
& h ; i^{-1} ; a ; i ; h^{-1} \\
= \\
& h ; i^{-1} ; m^{-1} ; s ; m ; i ; h^{-1} \\
= \\
& h ; f^{-1} ; s ; f ; h^{-1}
\end{aligned}
$$


E
j
B
b
E
B
i
f
n
a
h
A
A
m
S
s
S
F
l
F
D
d
D
k
g
p
c
C
C
o
T
t
T