

3. File Transfer Protocol

Jean-Raymond Abrial

July 2007

- To introduce another example: **the file transfer protocol**
- To present a number of **additional mathematical conventions**
- To slightly enlarge the usage of the **Proof Obligation Rules**
- Example studied in many places, in particular in the following book
- L. Lamport *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers* Addison-Wesley 1999

- A file is to be transferred from a **Sender** to a **Receiver**
- On the Sender's side the file is called f
- On the Receiver's side the file is called g
- At the beginning of the protocol, g is supposed to be empty
- At the end of the protocol, g should be equal to f

The protocol ensures the copy of a file from one site to another one

FUN-1

The file is supposed to be made of a sequence of items

FUN-2

The file is send piece by piece between the two sites

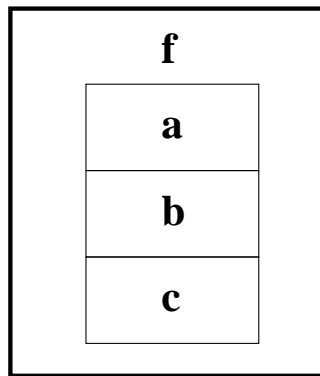
FUN-3

- Our approach at modeling is one of an **external observer**
- The observer “sees” the state space first **from very far away**
- He then approaches the future system and sees **more details**
- As he approaches he also sees **more things happening**

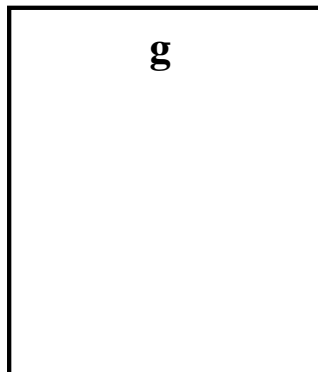
- **Initial model**: The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement**: The file is transmitted gradually (FUN3)
- **Second refinement**: The two agents are separated
- **Third refinement**: Towards an implementation
- **Decomposition**

INITIAL SITUATION

SENDER

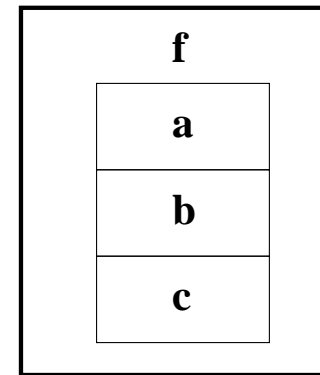


RECEIVER

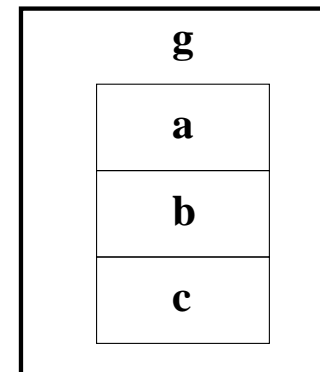


FINAL SITUATION

SENDER



RECEIVER



	f
1	a
	b
n	c

carrier sets: D

constants: n, f

prp0_1: $n \in \mathbb{N}$

prp0_2: $0 < n$

prp0_3: $f \in 1 .. n \rightarrow D$

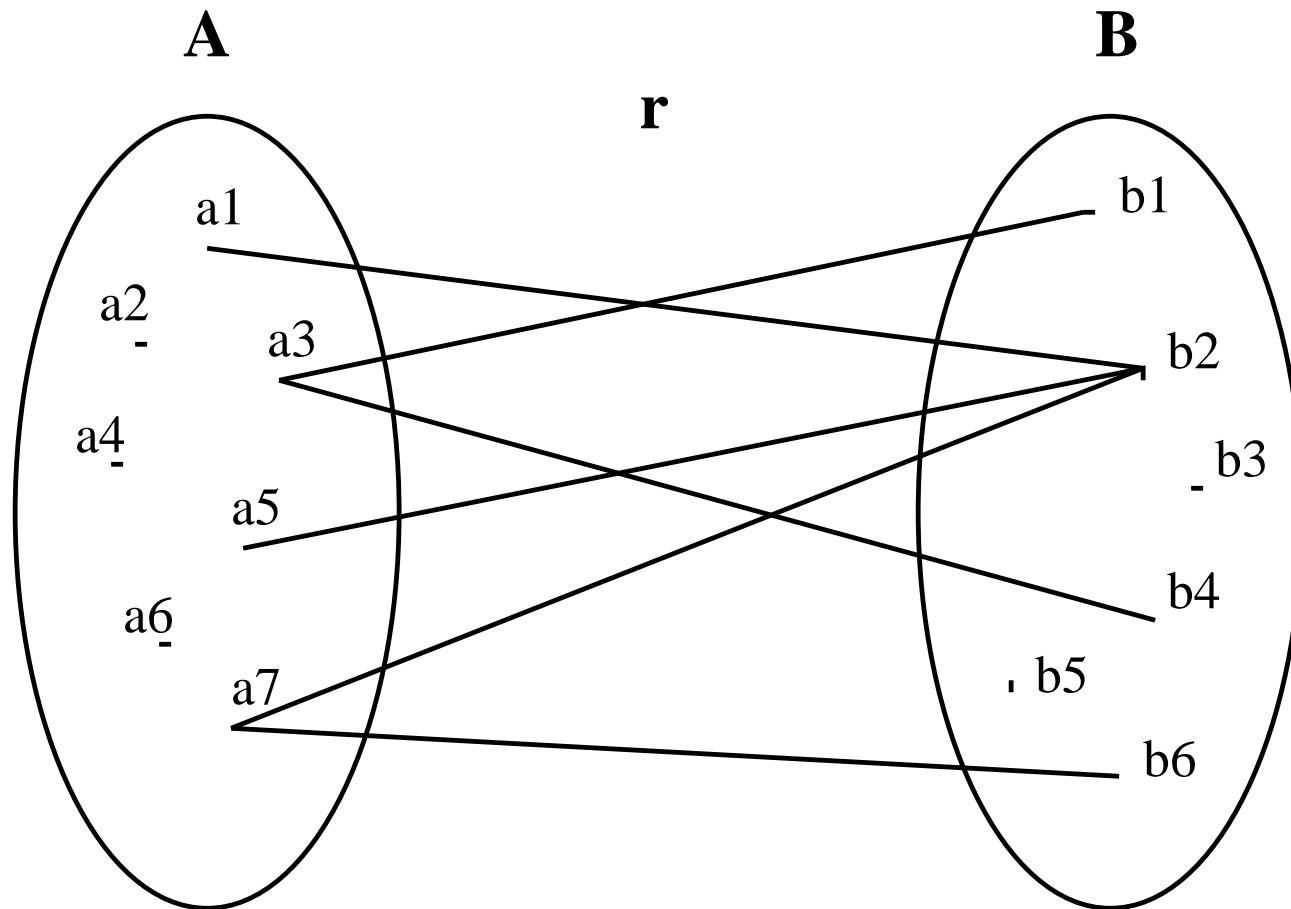
variables: g

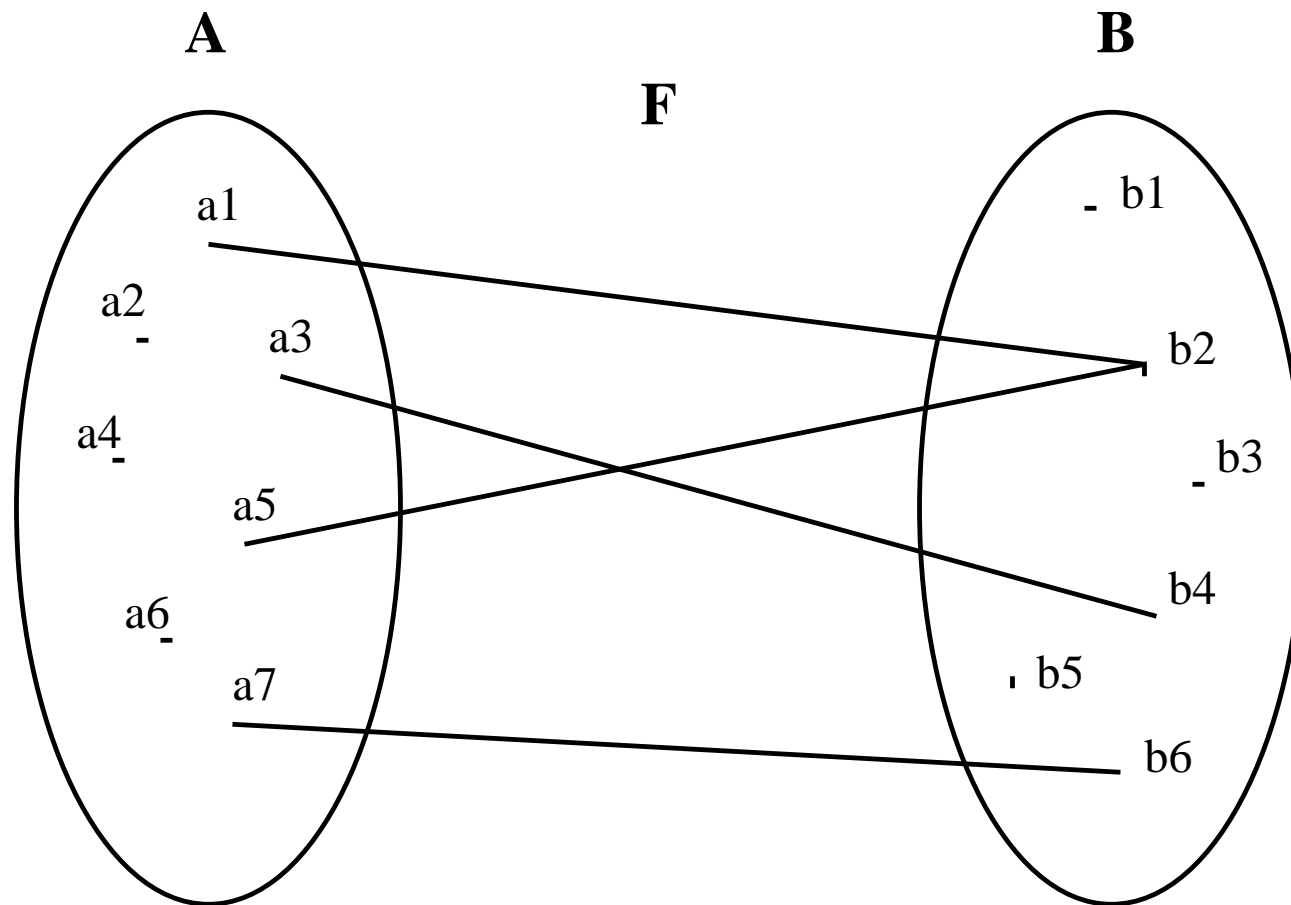
inv0_1: $g \in 1 .. n \leftrightarrow D$

- The **carrier set D** makes this development **generic**

$x \in S$	set membership operator
\mathbb{N}	set of natural numbers: $\{0, 1, 2, 3, \dots\}$
$a .. b$	interval from a to b : $\{a, a + 1, \dots, b\}$ (empty when $b < a$)
$a \mapsto b$	pair constructing operator
$S \times T$	Cartesian product operator
$S \subseteq T$	set inclusion operator
$\mathbb{P}(S)$	power set operator

$S \leftrightarrow T$	set of binary relations from S to T
$S \rightarrow T$	set of total functions from S to T
$S \twoheadrightarrow T$	set of partial functions from S to T
$\text{dom}(r)$	domain of a relation r
$\text{ran}(r)$	range of a relation r

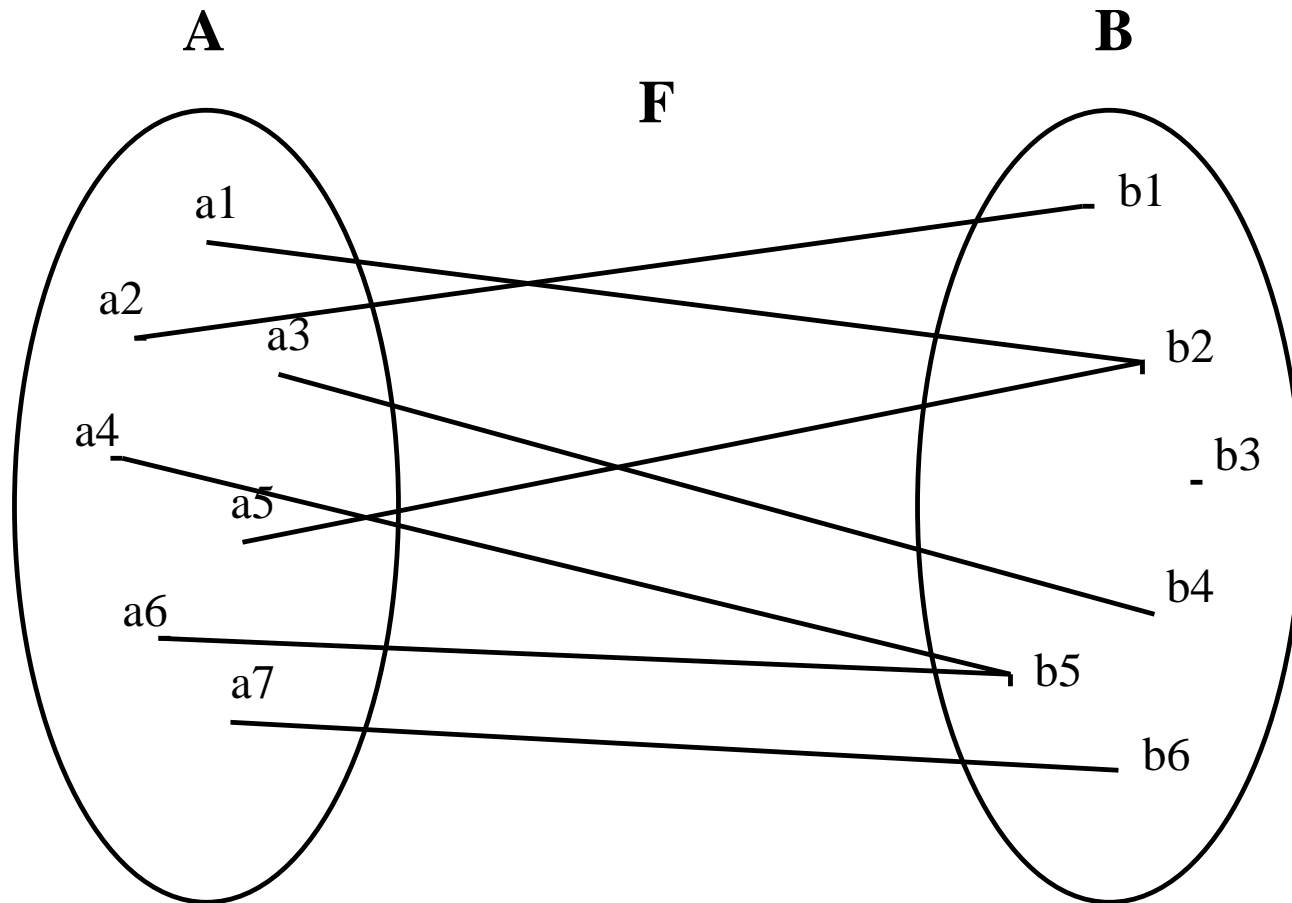




$$F = \{a1 \mapsto b2, a3 \mapsto b4, a5 \mapsto b2, a7 \mapsto b6\}$$

$$\text{dom}(F) = \{a1, a3, a5, a7\}$$

$$\text{ran}(F) = \{b2, b4, b6\}$$



$$\text{dom}(F) = A$$

final
 $g := f$

- The **guard** is true
- The **action** is a **simple assignment**

- There exists a special **initializing event**

init
 $g := \emptyset$

- We can add the following invariant:

inv0_2: $g = \emptyset \vee g = f$

carrier sets: D

constants: n, f

prp0_1: $n \in \mathbb{N}$

prp0_2: $0 < n$

prp0_3: $f \in 1 .. n \rightarrow D$

variables: g

inv0_1: $g \in 1 .. n \leftrightarrow D$

inv0_2: $g = \emptyset \vee g = f$

init

$g := \emptyset$

final

$g := f$

- Event **init establishes** invariants **inv0_1** and **inv0_2** (Rule INI_INV)
- Event **final preserves** invariants **inv0_1** and **inv0_2** (Rule INV)

inv0_1: $g \in 1 .. n \leftrightarrow D$

inv0_2: $g = \emptyset \vee g = f$

init

$g := \emptyset$

final

$g := f$

- For the init event in the initial model

Properties \vdash Modified Invariant	INI_INV
----------------------------------------------	---------

- Applying Rule INI_INV to invariant **inv0_1**

init

$g := \emptyset$

inv0_1: $g \in 1 .. n \leftrightarrow D$

prp0_1

prp0_2

prp0_3

⊢

modified **inv0_1**

$n \in \mathbb{N}$

$0 < n$

$f \in 1 .. n \rightarrow D$

⊢

$\emptyset \in 1 .. n \leftrightarrow D$

inv0_1 / INI_INV

- Applying Rule INI_INV to invariant **inv0_2**

init
 $g := \emptyset$

inv0_2: $g = \emptyset \vee g = f$

prp0_1
prp0_2
prp0_3
 \vdash
 modified **inv0_2**

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1 .. n \rightarrow D$
 \vdash
 $\emptyset = \emptyset \vee \emptyset = f$

inv0_2 / INI_INV

- For other events in the initial model

Properties Invariants Guards of the event \vdash Modified Invariant	INV
-----------------------------------------------------------------------------------	-----

- Applying Rule INV

final
g := f

prp0_1
prp0_2
prp0_3
inv0_1
inv0_2
⊢
modified **inv0_1**

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1..n \rightarrow D$
g $\in 1..n \leftrightarrow D$
 $g = \emptyset \vee g = f$
⊢
f $\in 1..n \leftrightarrow D$

final / **inv0_1** / INV

- Applying Rule INV

final
g := *f*

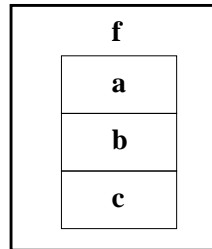
prp0_1
prp0_2
prp0_3
inv0_1
inv0_2
⊢
modified **inv0_2**

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1..n \rightarrow D$
 $g \in 1..n \leftrightarrow D$
g = \emptyset \vee *g* = *f*
⊢
f = \emptyset \vee *f* = *f*

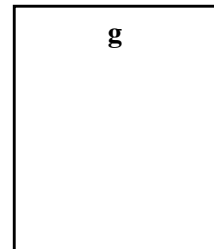
final / **inv0_2** / INV

INITIAL SITUATION

SENDER

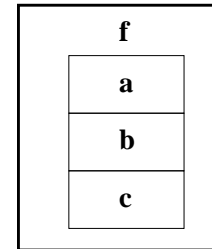


RECEIVER

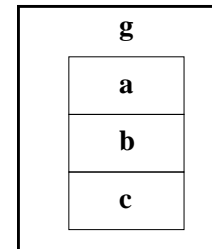


FINAL SITUATION

SENDER



RECEIVER

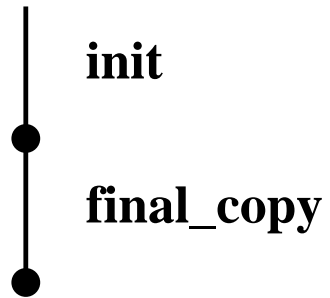


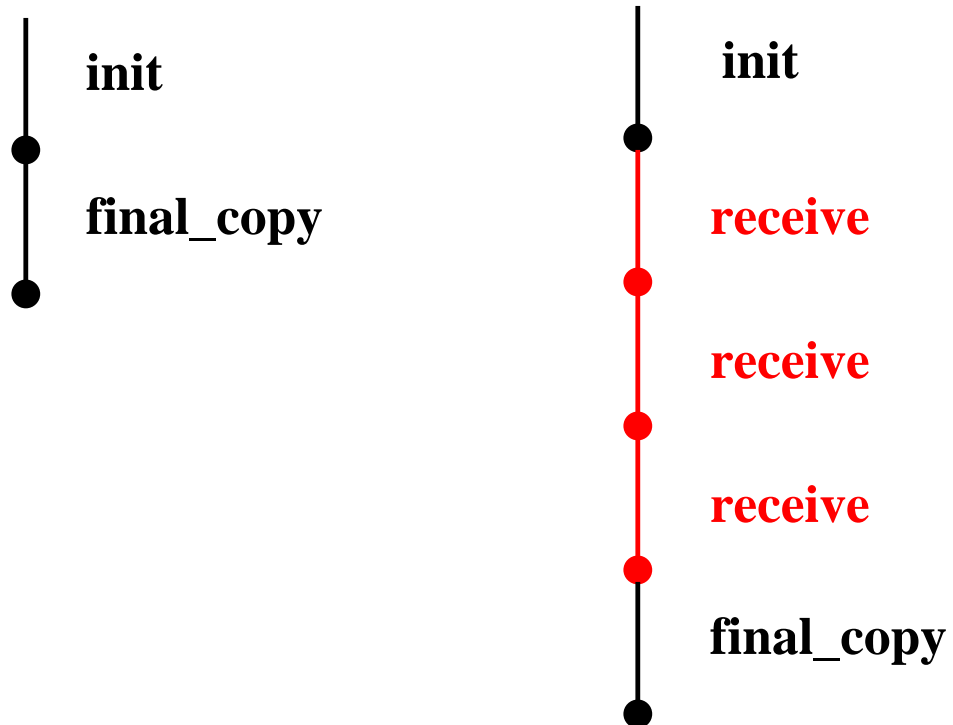
inv0_1: $g \in 1 .. n \leftrightarrow D$

inv0_2: $g = \emptyset \vee g = f$

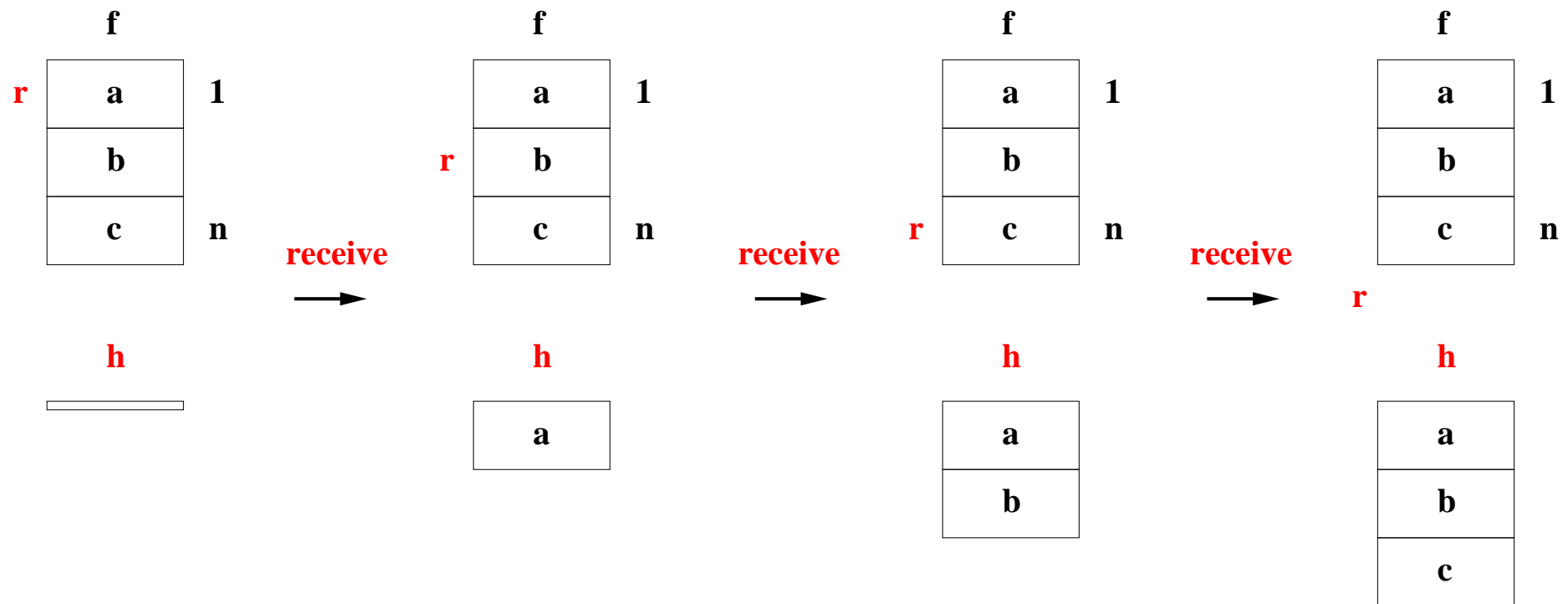
- **Initial model:** The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement:** The file is transmitted gradually (FUN3)
- **Second refinement:** The two agents are separated
- **Third refinement:** Towards an implementation
- **Decomposition**

- The observer **comes closer** to the future system
- So far he was just seeing **the beginning** and **the end**
- Now the observer will see **some intermediate moves**
- He sees the file being **gradually transfered** from Sender to Receiver
- But he still has a **partial view**





A new event is introduced: **receive**



- The new variable r lies within the interval $1 .. n + 1$
- The new variable h is equal to f restricted to its $r - 1$ first values

- Introducing additional variables h and r
- Variable g disappears

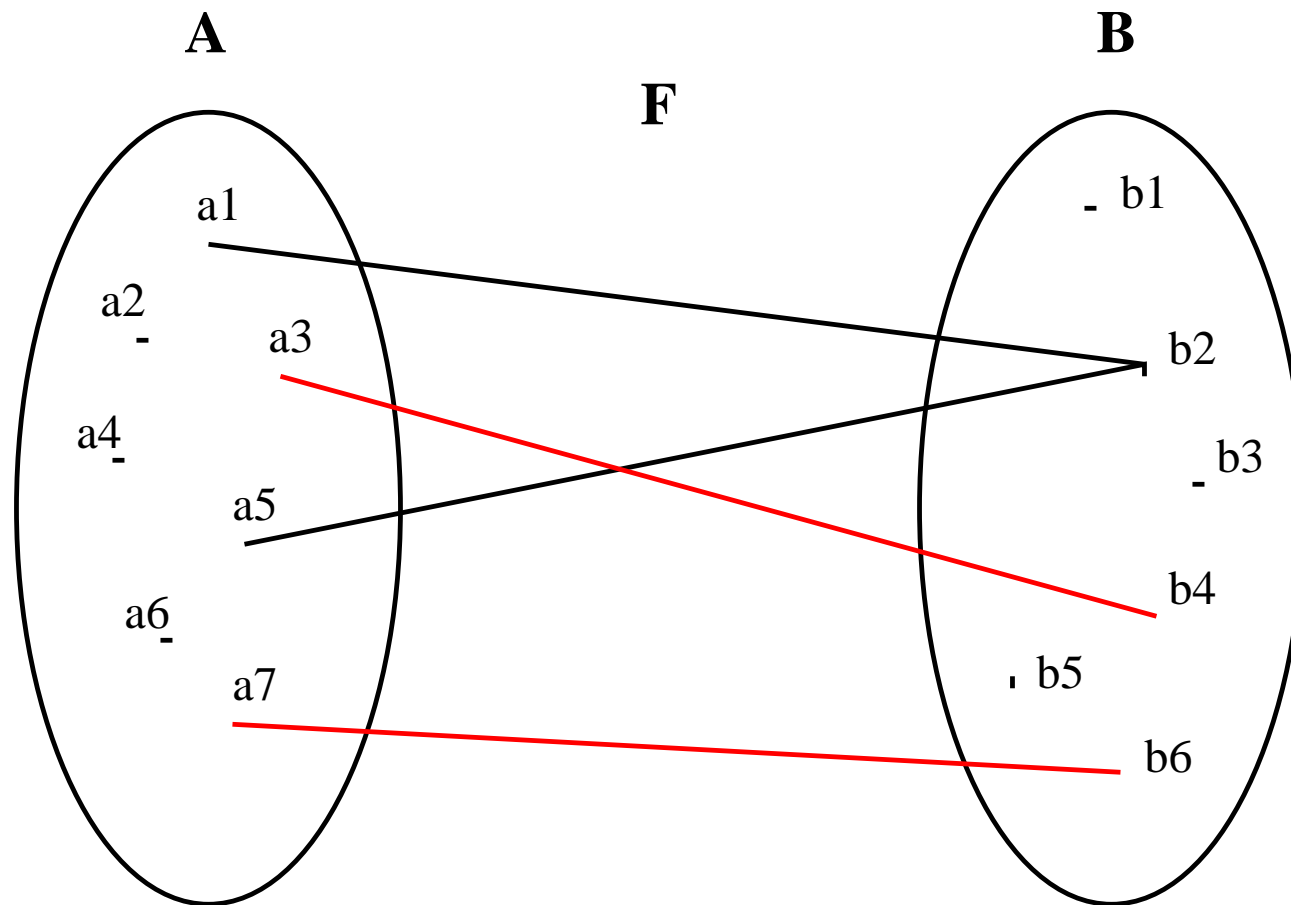
variables: h, r

inv1_1: $r \in 1 .. n + 1$

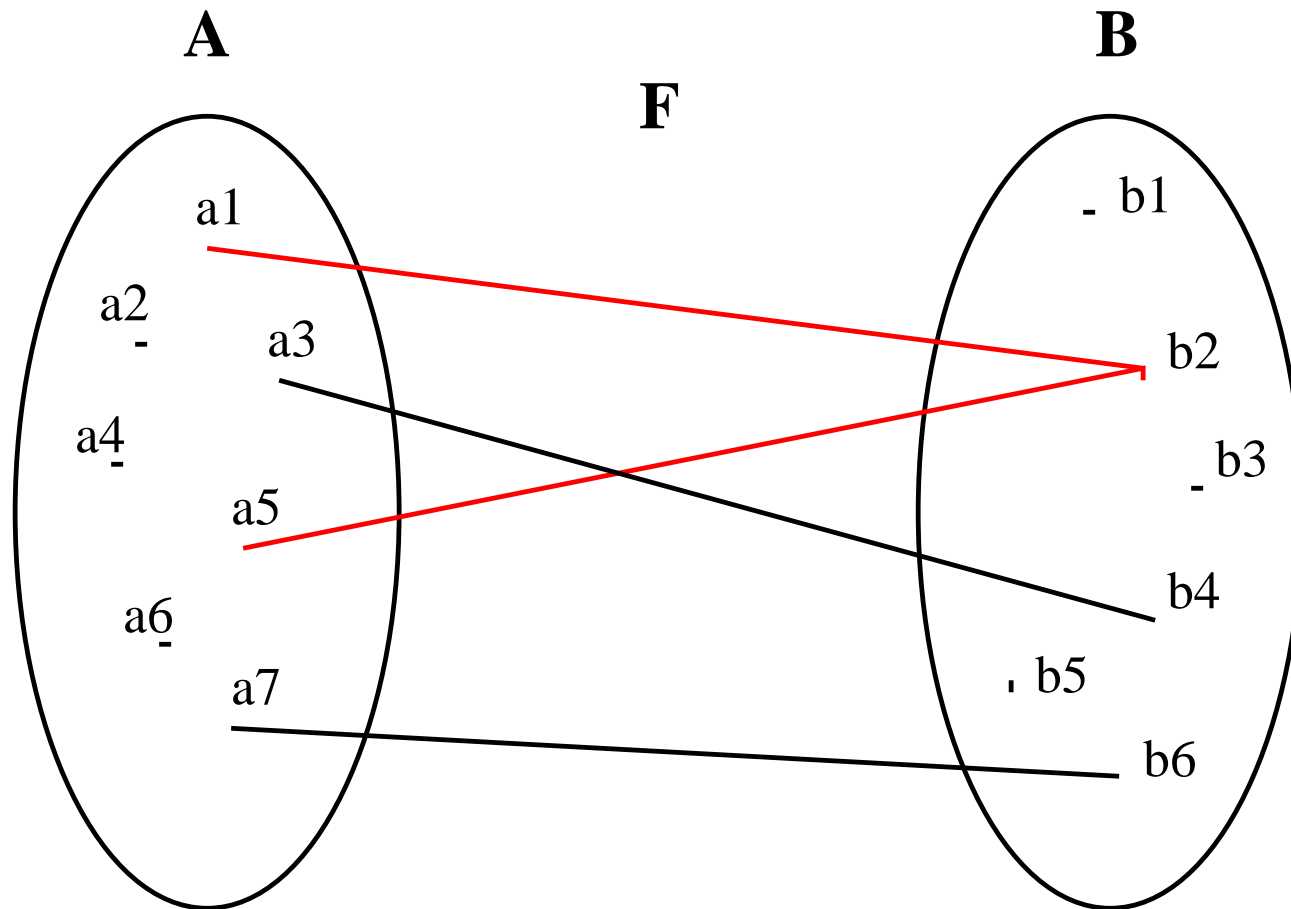
inv1_2: $h = (1 .. r - 1) \triangleleft f$

- h is defined to be the domain restriction of f to $1 .. r - 1$

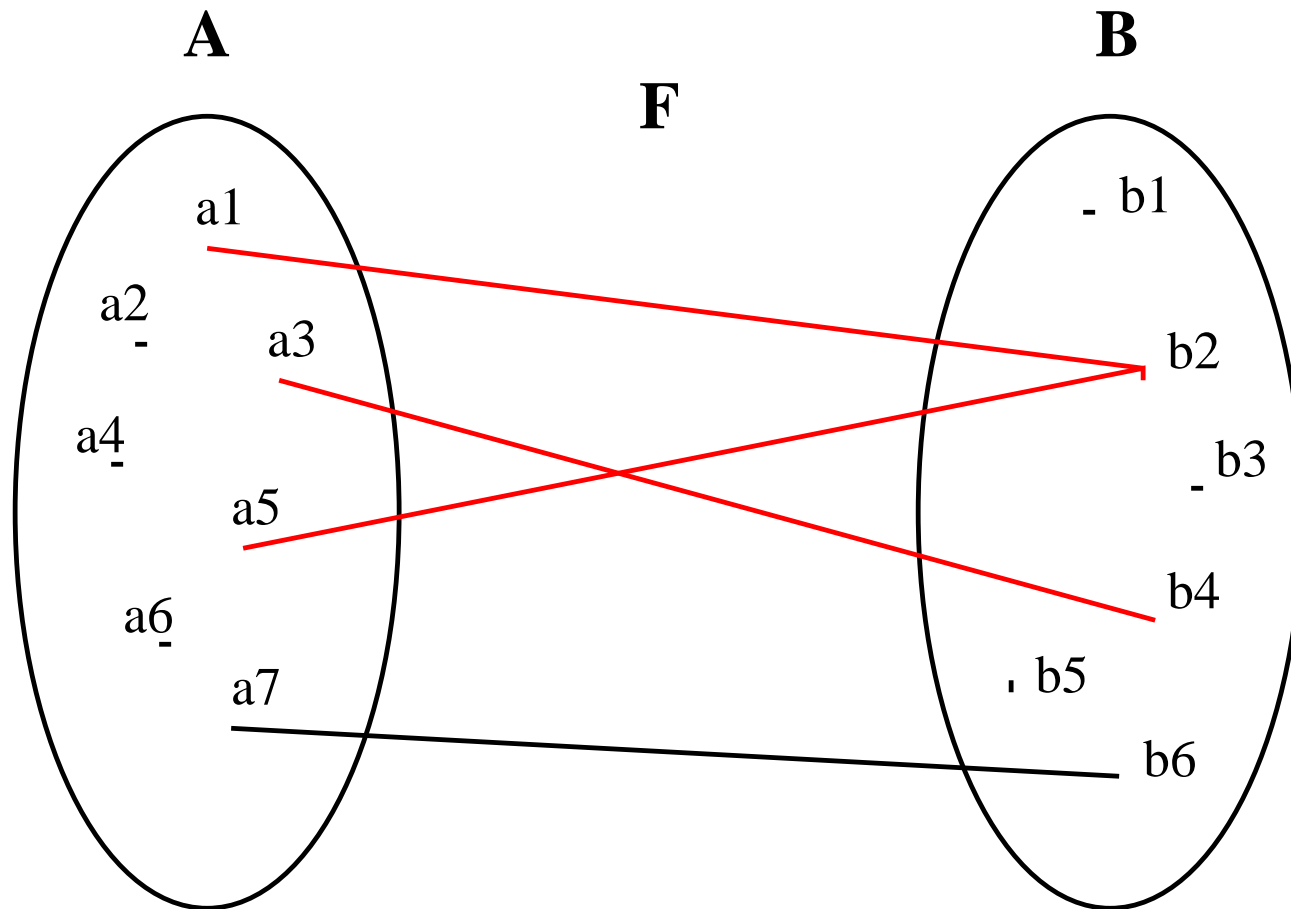
$s \triangleleft r$	domain restriction operator
$s \triangleleft r$	domain subtraction operator
$r \triangleright t$	range restriction operator
$r \triangleright t$	range subtraction operator



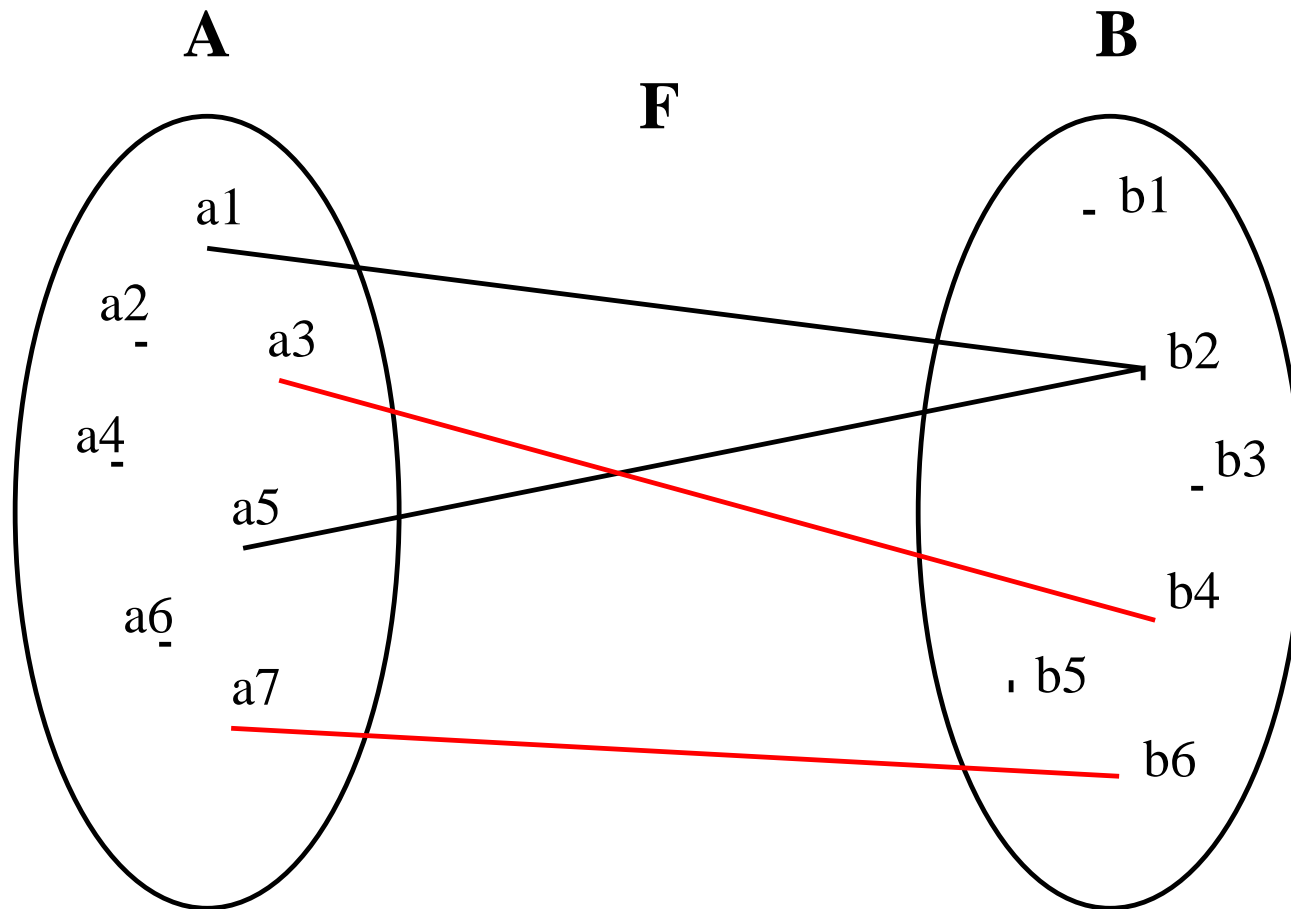
$$\{a_3, a_7\} \triangleleft F$$



$$\{a_3, a_7\} \triangleleft F$$



$$F \triangleright \{b2, b4\}$$



$$F \triangleright \{b_2\}$$

```
init
   $h := \emptyset$ 
   $r := 1$ 
```

```
receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
```

```
final
  when
     $r = n + 1$ 
  then
    skip
  end
```

- We have not established any connection between the **abstract** variable g and the **concrete** variables f and h
- It is important to do so **otherwise we cannot prove anything** concerning the refinement of event final
- We have to state that when the guard of final copy is true then h is equal to g **if g is not equal to the empty set**

$$\mathbf{inv1_3:} \quad r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$$

- **Event init** refines its abstraction (Rule **INI_INV_REF**)
- **Event final** refines its abstraction (Rules **GRD_REF** and **INV_REF**)
- **Event receive** refines skip (Rule **INV_REF**)
- **Event receive** does not diverge (Rules **WFD_REF1** and **WFD_REF2**)
- **Relative deadlock freeness** (Rule **DLF_REF**)

- For init event only

Properties of the constants \vdash Modified concrete invariant	INI_INV_REF
------------------------------------------------------------------------	-------------

(abstract_)init
 $g := \emptyset$

(concrete_)init
 $h := \emptyset$
 $r := 1$

inv1_1: $r \in 1..n+1$

prp0_1
prp0_2
prp0_3
⊢
mod. inv1_1

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1..n \rightarrow D$
⊢
 $1 \in 1..n+1$

inv1_1 / INI_INV_REF

(abstract_)init
 $g := \emptyset$

(concrete_)init
 $h := \emptyset$
 $r := 1$

inv1_2: $h = (1 .. r - 1) \triangleleft f$

prp0_1
 prp0_2
 prp0_3
 \vdash
 mod. inv1_2

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1 .. n \rightarrow D$
 \vdash
 $\emptyset = (1 .. 1 - 1) \triangleleft f$

inv1_2 / INI_INV_REF

(abstract_)init

$g := \emptyset$

(concrete_)init

$h := \emptyset$

$r := 1$

inv1_3: $r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

prp0_1
prp0_2
prp0_3

\vdash
mod. **inv1_3**

$n \in \mathbb{N}$

$0 < n$

$f \in 1 .. n \rightarrow D$

\vdash

$1 = n + 1 \wedge \emptyset \neq \emptyset \Rightarrow \emptyset = \emptyset$

inv1_3 / INI_INV_REF

(abstract_)final
 $g := f$

(concrete_)final
when
 $r = n + 1$
then
 skip
end

inv1_1: $r \in 1 .. n + 1$

inv1_2: $h = (1 .. r - 1) \triangleleft f$

inv1_3: $r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

- Abstract event final has no guard
- So there is no applicaton of rule GRD_REF
- We only have to apply rule INV_REF

- For all events (except init)
- New events refine an implicit non-guarded event with skip action

Properties of the constants Abstract invariants Concrete invariants Concrete guards \vdash Modified concrete invariant	INV_REF
-----------------------------------------------------------------------------------------------------------------------------------------	---------

inv1_1 / **final** / **INV_REF**

prp0_1
prp0_2
prp0_3
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
 guard of **final**
 \vdash
 mod. **inv1_1**

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \leftrightarrow D \\
 g = \emptyset \vee g = f \\
 r \in 1..n+1 \\
 \hline
 h = (1..r-1) \triangleleft f \\
 r = n+1 \wedge g \neq \emptyset \Rightarrow h = g \\
 r = n+1 \\
 \vdash \\
 r \in 1..n+1
 \end{array}$$

(abstract_)**final**
 $g := f$

(concrete_)**final**
when $r = n + 1$ **then** skip **end**

inv1_2 / final / INV_REF

prp0_1
prp0_2
prp0_3
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
 guard of final
 \vdash
 mod. **inv1_2**

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1 .. n \rightarrow D$
 $g \in 1 .. n \leftrightarrow D$
 $g = \emptyset \vee g = f$
 $r \in 1 .. n + 1$
 $h = (1 .. r - 1) \triangleleft f$
 $\frac{r = n + 1 \wedge g \neq \emptyset}{r = n + 1} \Rightarrow h = g$
 \vdash
 $h = (1 .. r - 1) \triangleleft f$

(abstract_)final
 $g := f$

(concrete_)final
when $r = n + 1$ **then** skip **end**

inv1_3 / final / INV_REF

prp0_1
prp0_2
prp0_3
inv0_1
inv0_2
inv1_1
inv1_2
inv1_3
 guard of final
 ⊢
 mod. **inv1_3**

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \leftrightarrow D \\
 g = \emptyset \vee g = f \\
 r \in 1..n+1 \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \wedge g \neq \emptyset \Rightarrow h = g \\
 r = n+1 \\
 \vdash \\
 r = n+1 \wedge f \neq \emptyset \Rightarrow h = f
 \end{array}$$

(abstract_)final
 $g := f$

(concrete_)final
when $r = n + 1$ **then** skip **end**

$$\begin{array}{l}
n \in \mathbb{N} \\
0 < n \\
f \in 1..n \rightarrow D \\
g \in 1..n \leftrightarrow D \\
g = \emptyset \vee g = f \\
r \in 1..n+1 \\
h = (1..r-1) \triangleleft f \\
r = n+1 \wedge g \neq \emptyset \Rightarrow h = g \\
r = n+1 \\
\vdash \\
r = n+1 \wedge f \neq \emptyset \Rightarrow h = f
\end{array}$$

MON

$$\begin{array}{l}
f \in 1..n \rightarrow D \\
h = (1..r-1) \triangleleft f \\
r = n+1 \\
\vdash \\
r = n+1 \wedge f \neq \emptyset \Rightarrow h = f
\end{array}$$

IMP_R

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \rightarrow D \\
 g = \emptyset \vee g = f \\
 r \in 1..n+1 \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \wedge g \neq \emptyset \Rightarrow h = g \\
 r = n+1 \\
 \vdash \\
 r = n+1 \wedge f \neq \emptyset \Rightarrow h = f
 \end{array}$$

MON

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \\
 \vdash \\
 r = n+1 \wedge f \neq \emptyset \Rightarrow h = f
 \end{array}$$

IMP_R

$$\begin{array}{l}
 \dots \\
 f \in 1..n \rightarrow D \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \\
 r = n+1 \wedge f \neq \emptyset \\
 \vdash \\
 h = f
 \end{array}$$

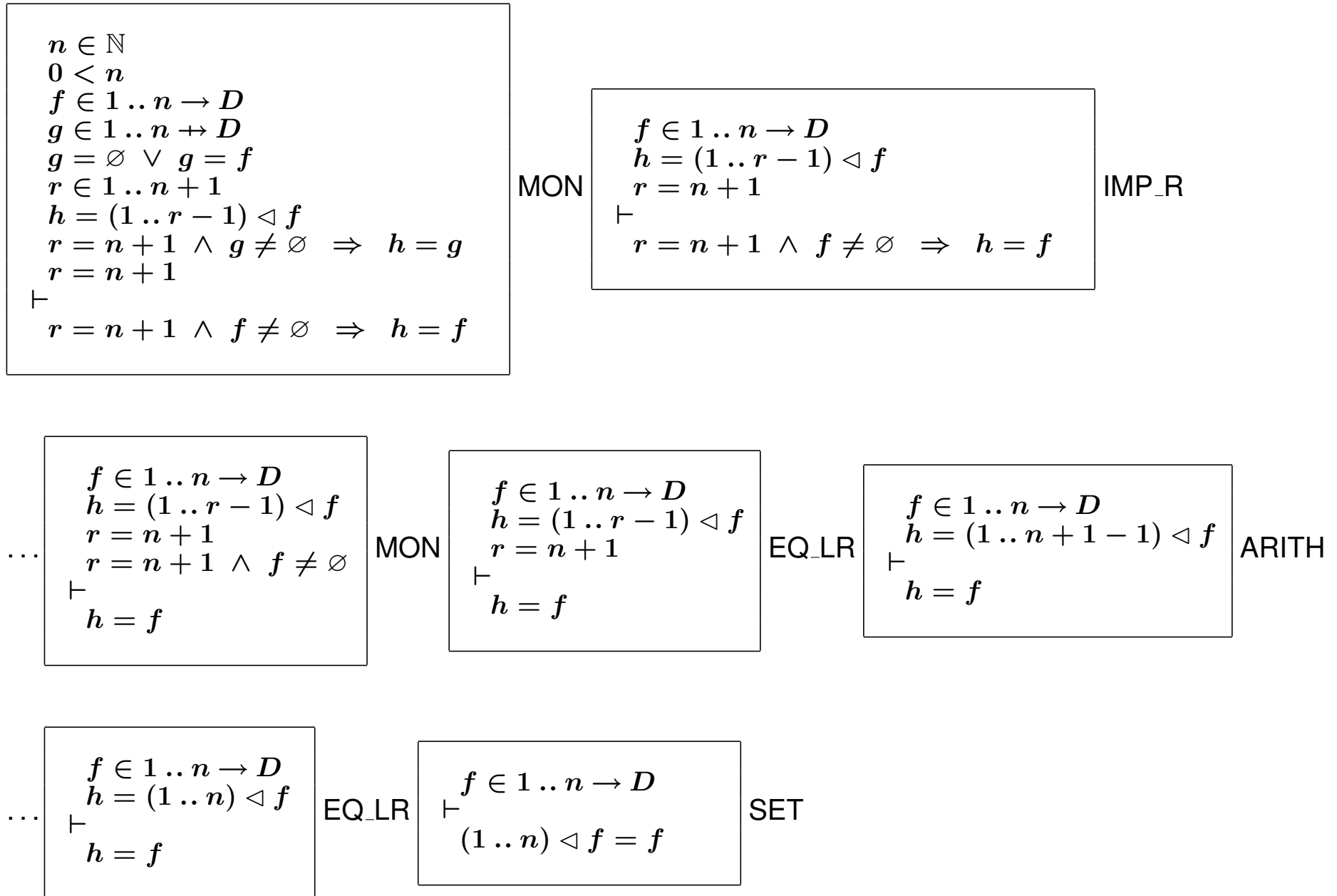
MON

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \\
 \vdash \\
 h = f
 \end{array}$$

EQ_LR

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 h = (1..n+1-1) \triangleleft f \\
 \vdash \\
 h = f
 \end{array}$$

ARITH



(implicit_abstract_)receive
skip

(new_)receive
when
 $r \leq n$
then
 $h := h \cup \{r \mapsto f(r)\}$
 $r := r + 1$
end

inv1_1: $r \in 1 .. n + 1$

inv1_2: $h = (1 .. r - 1) \triangleleft f$

inv1_3: $r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

prp0_1
 prp0_2
 prp0_3
 inv0_1
 inv0_2
 inv1_1
 inv1_2
 inv1_3
 guard of receive
 \vdash
 mod. **inv1_1**

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \leftrightarrow D \\
 g = \emptyset \vee g = f \\
 \mathbf{r} \in 1..n + 1 \\
 h = (1..r - 1) \triangleleft f \\
 \mathbf{r} = n + 1 \wedge g \neq \emptyset \Rightarrow h = g \\
 \mathbf{r} \leq n \\
 \vdash \\
 \mathbf{r} + 1 \in 1..n + 1
 \end{array}$$

```

receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
    
```

prp0_1
 prp0_2
 prp0_3
 inv0_1
 inv0_2
 inv1_1
 inv1_2
 inv1_3
 grd of receive
 \vdash
 mod. **inv1_2**

$$\begin{aligned}
 &n \in \mathbb{N} \\
 &0 < n \\
 &f \in 1..n \rightarrow D \\
 &g \in 1..n \leftrightarrow D \\
 &g = \emptyset \vee g = f \\
 &r \in 1..n + 1 \\
 &h = (1..r - 1) \triangleleft f \\
 &r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g \\
 &r \leq n \\
 \vdash & \\
 &h \cup \{r \mapsto f(r)\} = (1..r + 1 - 1) \triangleleft f
 \end{aligned}$$

```

receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
    
```

$$n \in \mathbb{N}$$

$$0 < n$$

$$f \in 1..n \rightarrow D$$

$$g \in 1..n \rightarrow D$$

$$g = \emptyset \vee g = f$$

$$r \in 1..n+1$$

$$h = (1..r-1) \triangleleft f$$

$$r = n+1 \wedge g \neq \emptyset \Rightarrow h = g$$

$$r \leq n$$

 \vdash

$$h \cup \{r \mapsto f(r)\} = (1..r+1-1) \triangleleft f$$

MON

$$f \in 1..n \rightarrow D$$

$$h = (1..r-1) \triangleleft f$$

$$r \leq n$$

 \vdash

$$h \cup \{r \mapsto f(r)\} = (1..r+1-1) \triangleleft f$$

ARITH

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \rightarrow D \\
 g = \emptyset \vee g = f \\
 r \in 1..n+1 \\
 h = (1..r-1) \triangleleft f \\
 r = n+1 \wedge g \neq \emptyset \Rightarrow h = g \\
 r \leq n \\
 \vdash \\
 h \cup \{r \mapsto f(r)\} = (1..r+1-1) \triangleleft f
 \end{array}$$

MON

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 h = (1..r-1) \triangleleft f \\
 r \leq n \\
 \vdash \\
 h \cup \{r \mapsto f(r)\} = (1..r+1-1) \triangleleft f
 \end{array}$$

ARITH

...

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 h = (1..r-1) \triangleleft f \\
 r \leq n \\
 \vdash \\
 h \cup \{r \mapsto f(r)\} = (1..r) \triangleleft f
 \end{array}$$

EQ_LR

$$\begin{array}{l}
 f \in 1..n \rightarrow D \\
 r \leq n \\
 \vdash \\
 (1..r-1) \triangleleft f \cup \{r \mapsto f(r)\} = (1..r) \triangleleft f
 \end{array}$$

SET

prp0_1
 prp0_2
 prp0_3
 inv0_1
 inv0_2
 inv1_1
 inv1_2
 inv1_3
 grd of receive
 \vdash
 mod. **inv1_3**

$$\begin{array}{l}
 n \in \mathbb{N} \\
 0 < n \\
 f \in 1..n \rightarrow D \\
 g \in 1..n \leftrightarrow D \\
 g = \emptyset \vee g = f \\
 r \in 1..n+1 \\
 h = (1..r-1) \triangleleft f \\
 \mathbf{r} = n+1 \wedge g \neq \emptyset \Rightarrow \mathbf{h} = g \\
 r \leq n \\
 \vdash \\
 \mathbf{r+1} = n+1 \wedge g \neq \emptyset \Rightarrow \mathbf{h \cup \{r \mapsto f(r)\}} = g
 \end{array}$$

```

receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
    
```

$$n \in \mathbb{N}$$

$$0 < n$$

$$f \in 1..n \rightarrow D$$

$$g \in 1..n \leftrightarrow D$$

$$g = \emptyset \vee g = f$$

$$r \in 1..n+1$$

$$h = (1..r-1) \triangleleft f$$

$$r = n+1 \wedge g \neq \emptyset \Rightarrow h = g$$

$$r \leq n$$

⊢

$$r+1 = n+1 \wedge g \neq \emptyset \Rightarrow$$

$$h \cup \{r \mapsto f(r)\} = g$$

MON

$$f \in 1..n \rightarrow D$$

$$g = \emptyset \vee g = f$$

$$h = (1..r-1) \triangleleft f$$

⊢

$$r+1 = n+1 \wedge g \neq \emptyset \Rightarrow$$

$$h \cup \{r \mapsto f(r)\} = g$$

IMP_R

$n \in \mathbb{N}$
 $0 < n$
 $f \in 1..n \rightarrow D$
 $g \in 1..n \leftrightarrow D$
 $g = \emptyset \vee g = f$
 $r \in 1..n+1$
 $h = (1..r-1) \triangleleft f$
 $r = n+1 \wedge g \neq \emptyset \Rightarrow h = g$
 $r \leq n$
 \vdash
 $r+1 = n+1 \wedge g \neq \emptyset \Rightarrow$
 $h \cup \{r \mapsto f(r)\} = g$

MON

$f \in 1..n \rightarrow D$
 $g = \emptyset \vee g = f$
 $h = (1..r-1) \triangleleft f$
 \vdash
 $r+1 = n+1 \wedge g \neq \emptyset \Rightarrow$
 $h \cup \{r \mapsto f(r)\} = g$

IMP_R

$f \in 1..n \rightarrow D$
 $g = \emptyset \vee g = f$
 $h = (1..r-1) \triangleleft f$
 $r+1 = n+1 \wedge g \neq \emptyset$
 \vdash
 $h \cup \{r \mapsto f(r)\} = g$

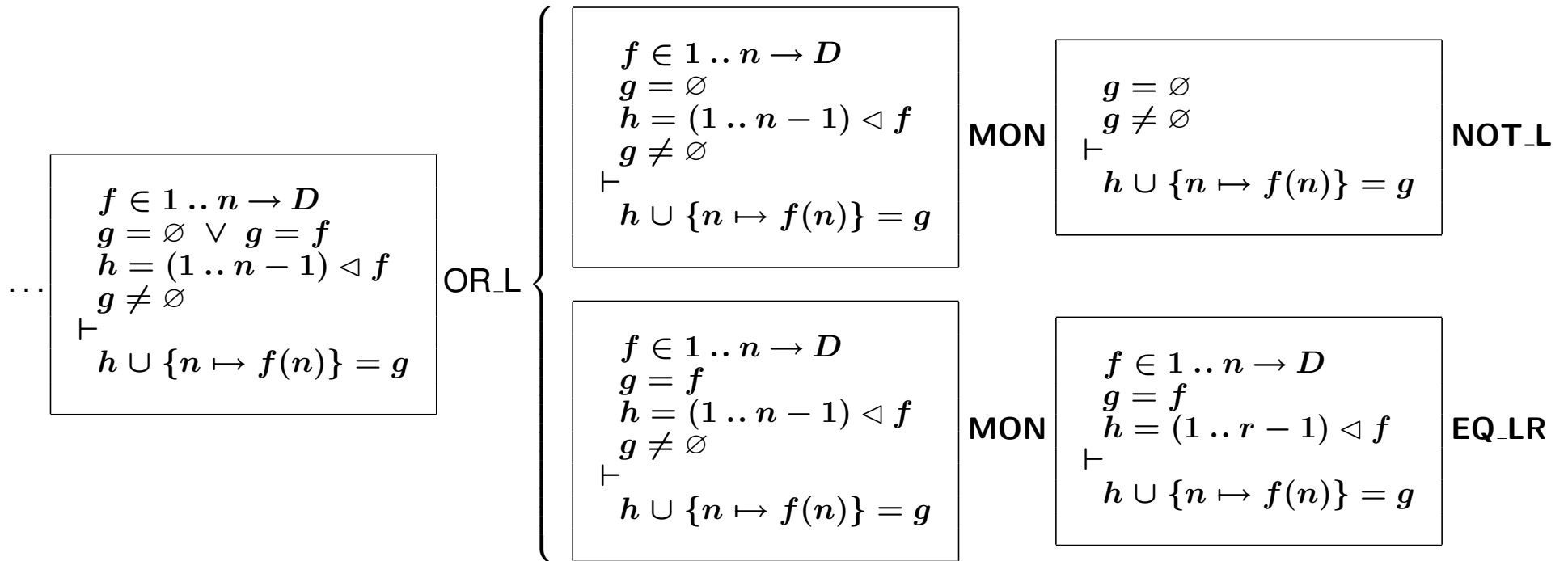
AND_L

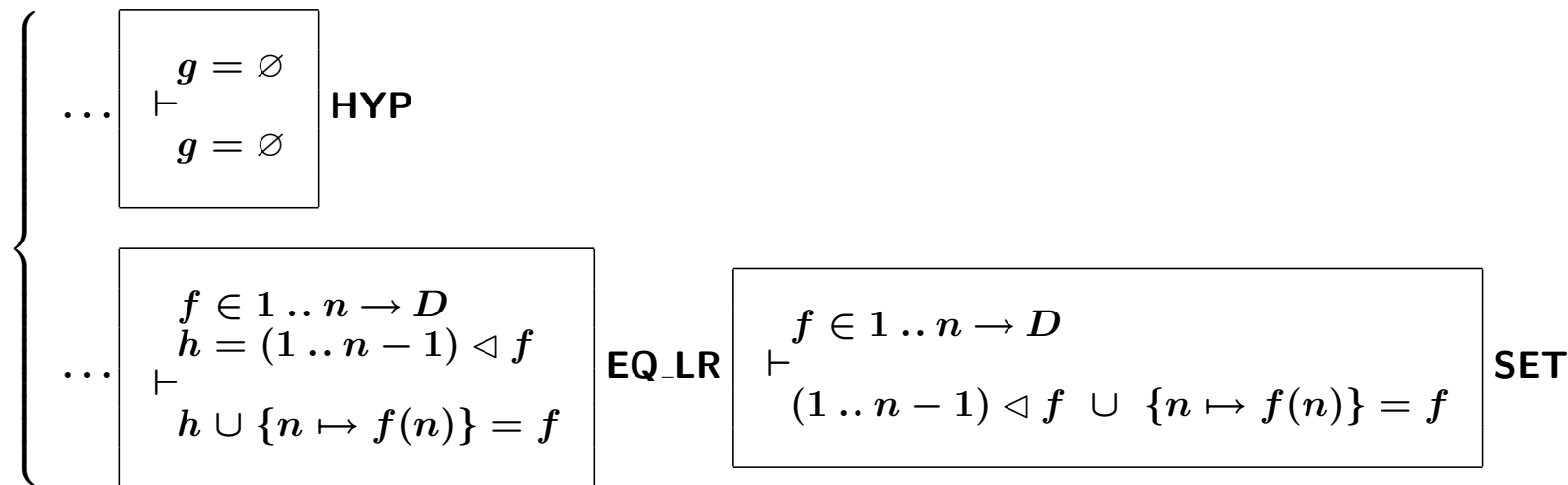
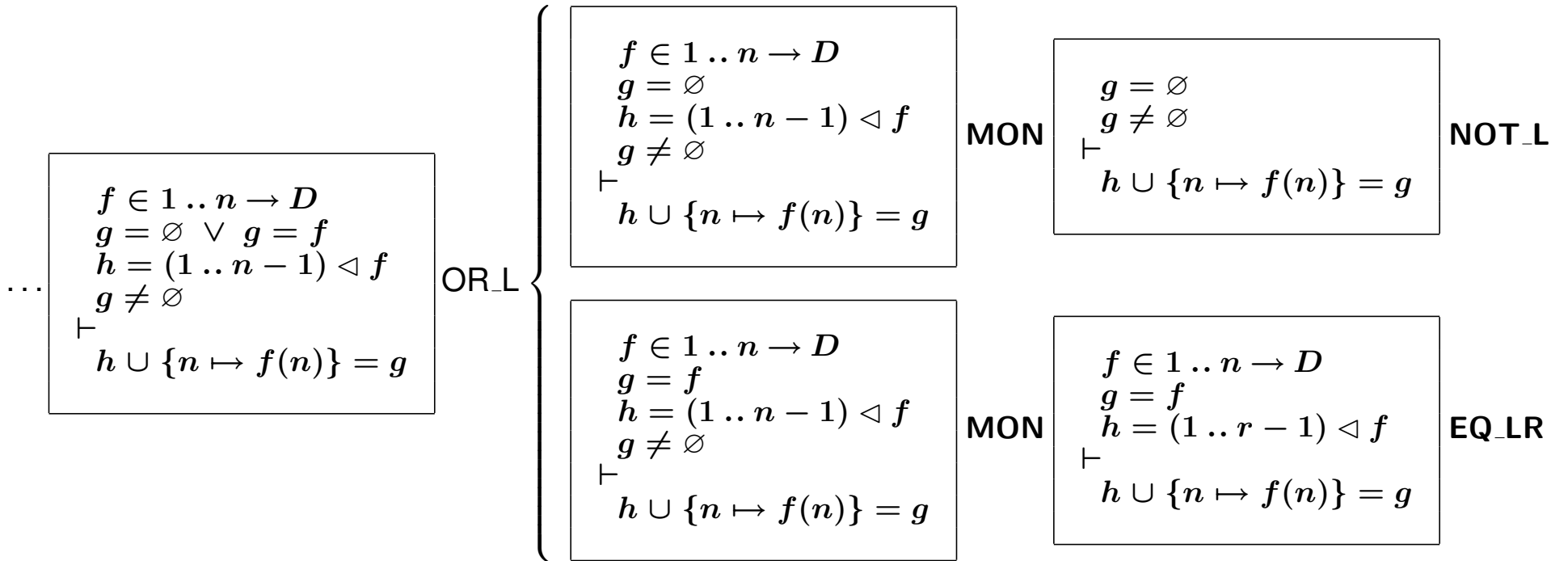
$f \in 1..n \rightarrow D$
 $g = \emptyset \vee g = f$
 $h = (1..r-1) \triangleleft f$
 $r+1 = n+1$
 $g \neq \emptyset$
 \vdash
 $h \cup \{r \mapsto f(r)\} = g$

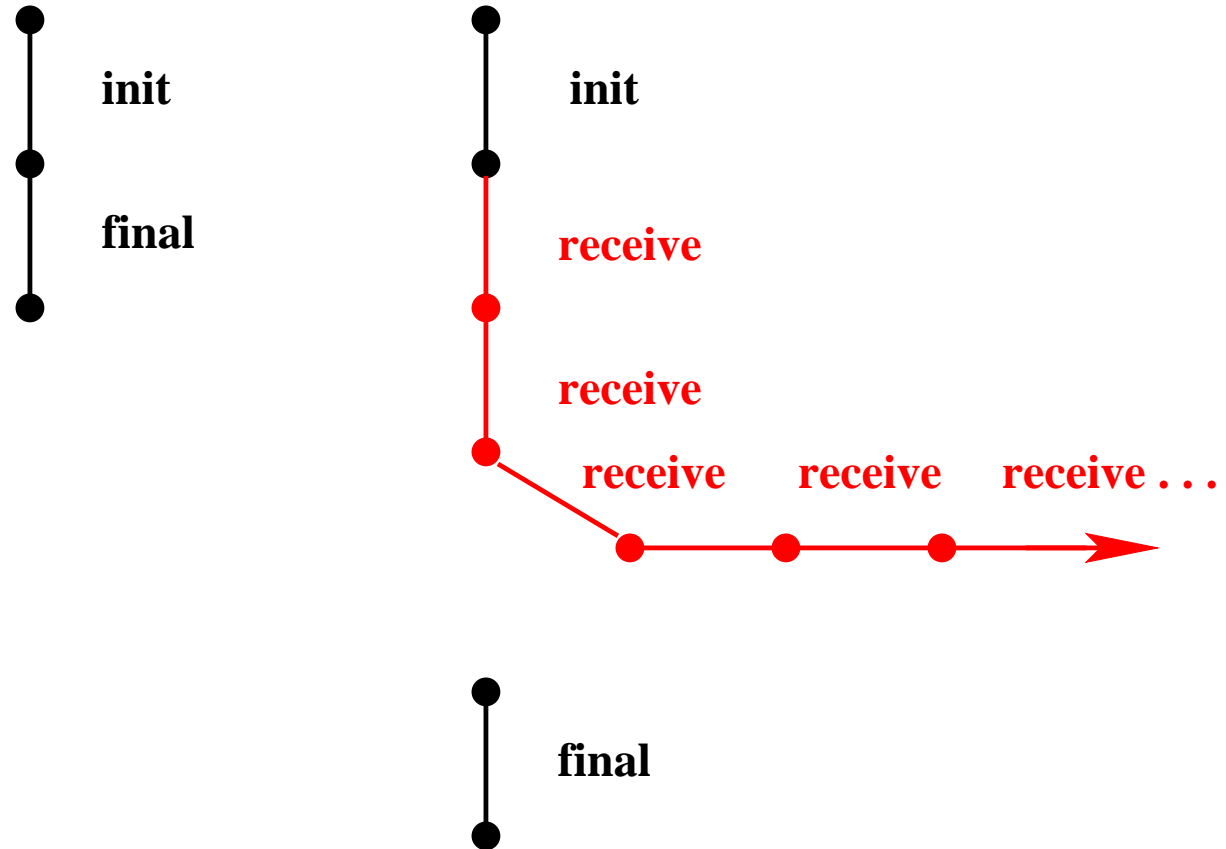
ARITH

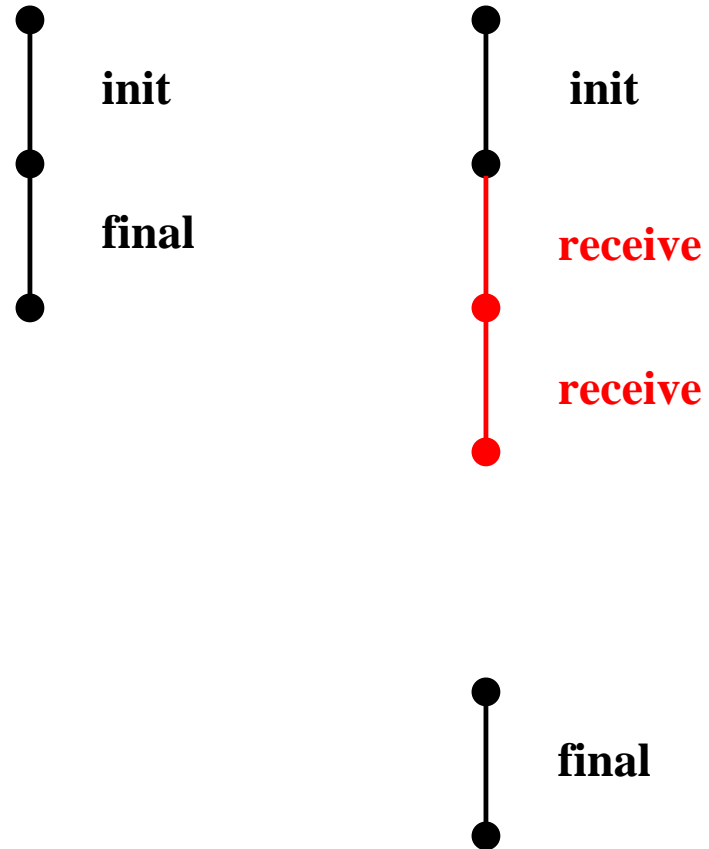
$f \in 1..n \rightarrow D$
 $g = \emptyset \vee g = f$
 $h = (1..r-1) \triangleleft f$
 $r = n$
 $g \neq \emptyset$
 \vdash
 $h \cup \{r \mapsto f(r)\} = g$

EQ_LR









- No divergence of new event receive (rules WFD_REF1,2)

$$\text{variant1: } n + 1 - r$$

- This variant **must be decreased** by the new event:

```
receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
```

- For new events only

Properties of the constants Abstract invariants Concrete invariants Concrete guards of a new event \vdash Variant $\in \mathbb{N}$	WFD_REF1
-----------------------------------------------------------------------------------------------------------------------------------------------------	----------

- Applying rule **WFD_REF1**

prp0_1

prp0_2

prp0_3

inv0_1

inv0_2

inv1_1

inv1_2

inv1_3

guard of receive

⊢

variant belongs to \mathbb{N}

$n \in \mathbb{N}$

$0 < n$

$f \in 1 .. n \rightarrow D$

$g \in 1 .. n \leftrightarrow D$

$g = \emptyset \vee g = f$

$r \in 1 .. n + 1$

$h = (1 .. r - 1) \triangleleft f$

$r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

$r \leq n$

⊢

$n + 1 - r \in \mathbb{N}$

- For new events only

Properties of the constants Abstract invariants Concrete invariants Concrete guards of a new event ⊢ Modified variant < Variant	WFD_REF2
------------------------------------------------------------------------------------------------------------------------------------------------	----------

- Applying rule **WFD_REF2**

prp0_1

prp0_2

prp0_3

inv0_1

inv0_2

inv1_1

inv1_2

inv1_3

guard of receive

⊢

variant is decreased

$$n \in \mathbb{N}$$

$$0 < n$$

$$f \in 1..n \rightarrow D$$

$$g \in 1..n \leftrightarrow D$$

$$g = \emptyset \vee g = f$$

$$r \in 1..n + 1$$

$$h = (1..r - 1) \triangleleft f$$

$$r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$$

$$r \leq n$$

⊢

$$n + 1 - (r + 1) < n + 1 - r$$

- Global proof rule

Properties of the constants Abstract invariants Concrete invariants Disjunction of abstract guards \vdash Disjunction of concrete guards	DLF_REF
-----------------------------------------------------------------------------------------------------------------------------------------------------------	---------

- Abstract Events

```
final
   $g := f$ 
```

- Concrete Events

```
receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
```

```
final
  when
     $r = n + 1$ 
  then
    skip
  end
```

- Applying rule **DLF_REF**

prp0_1

prp0_2

prp0_3

inv0_1

inv0_2

inv1_1

inv1_2

inv1_3

⊢

disj. of conc. guards

$n \in \mathbb{N}$

$0 < n$

$f \in 1 .. n \rightarrow D$

$g \in 1 .. n \leftrightarrow D$

$g = \emptyset \vee g = f$

$r \in 1 .. n + 1$

$h = (1 .. r - 1) \triangleleft f$

$r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

⊢

$r \leq n \vee r = n + 1$

variables: h, r

inv1_1: $r \in 1 .. n + 1$

inv1_2: $h = (1 .. r - 1) \triangleleft f$

inv1_3: $r = n + 1 \wedge g \neq \emptyset \Rightarrow h = g$

varaint1: $n + 1 - r$

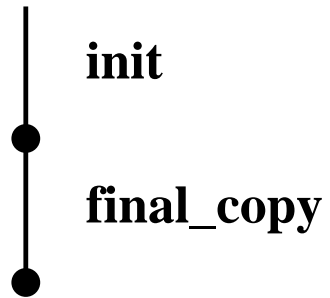
init
 $h := \emptyset$
 $r := 1$

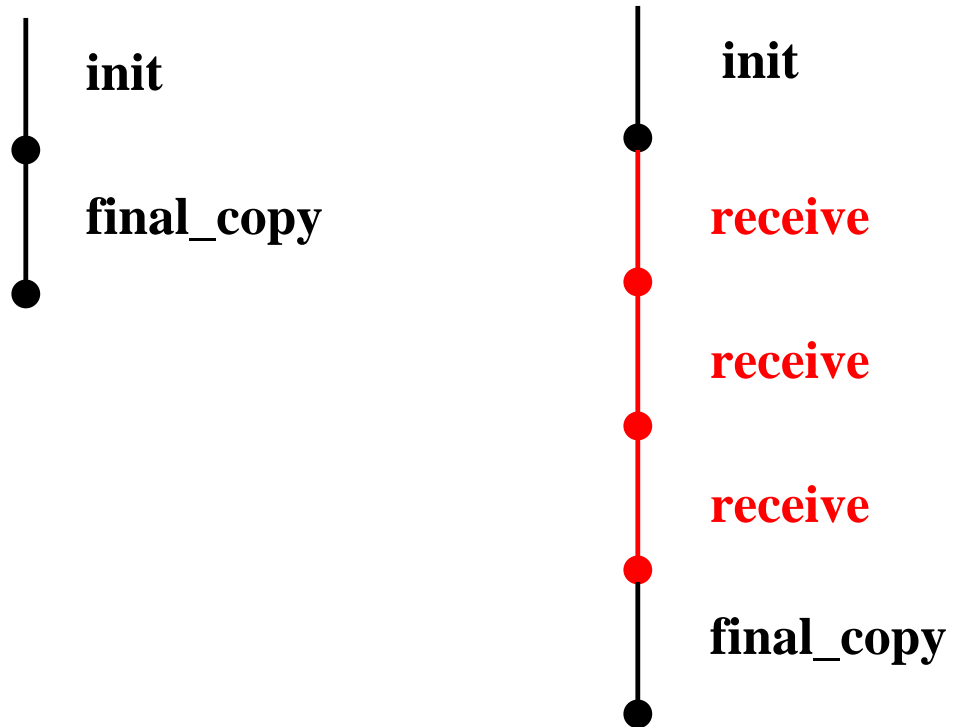
receive
when
 $r \leq n$
then
 $h := h \cup \{r \mapsto f(r)\}$
 $r := r + 1$
end

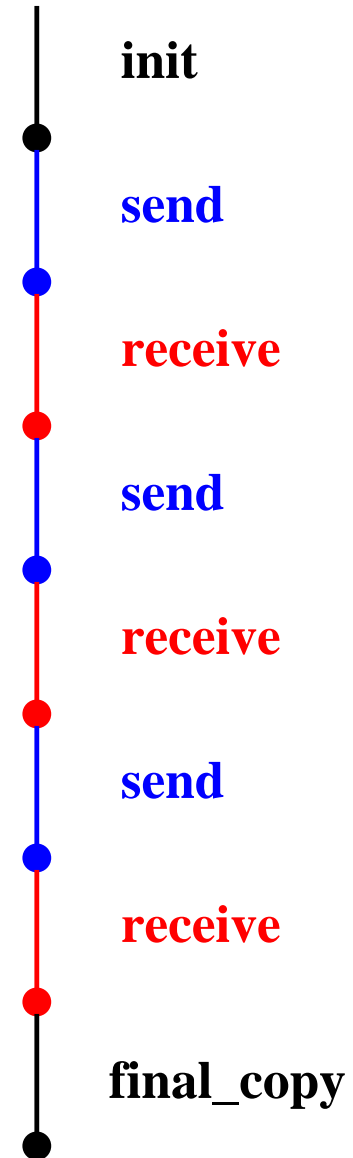
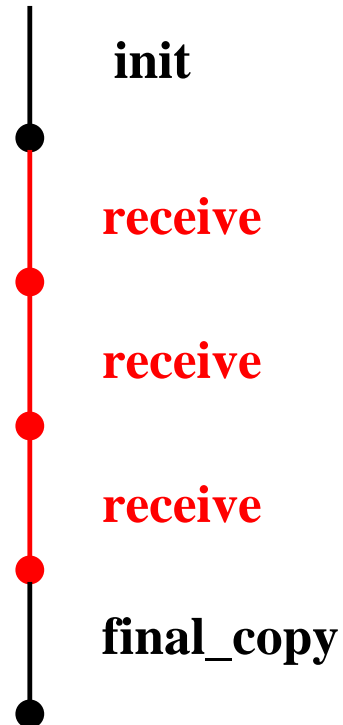
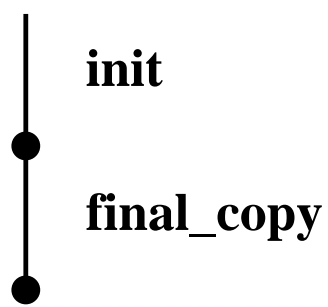
final
when
 $r = n + 1$
then
 skip
end

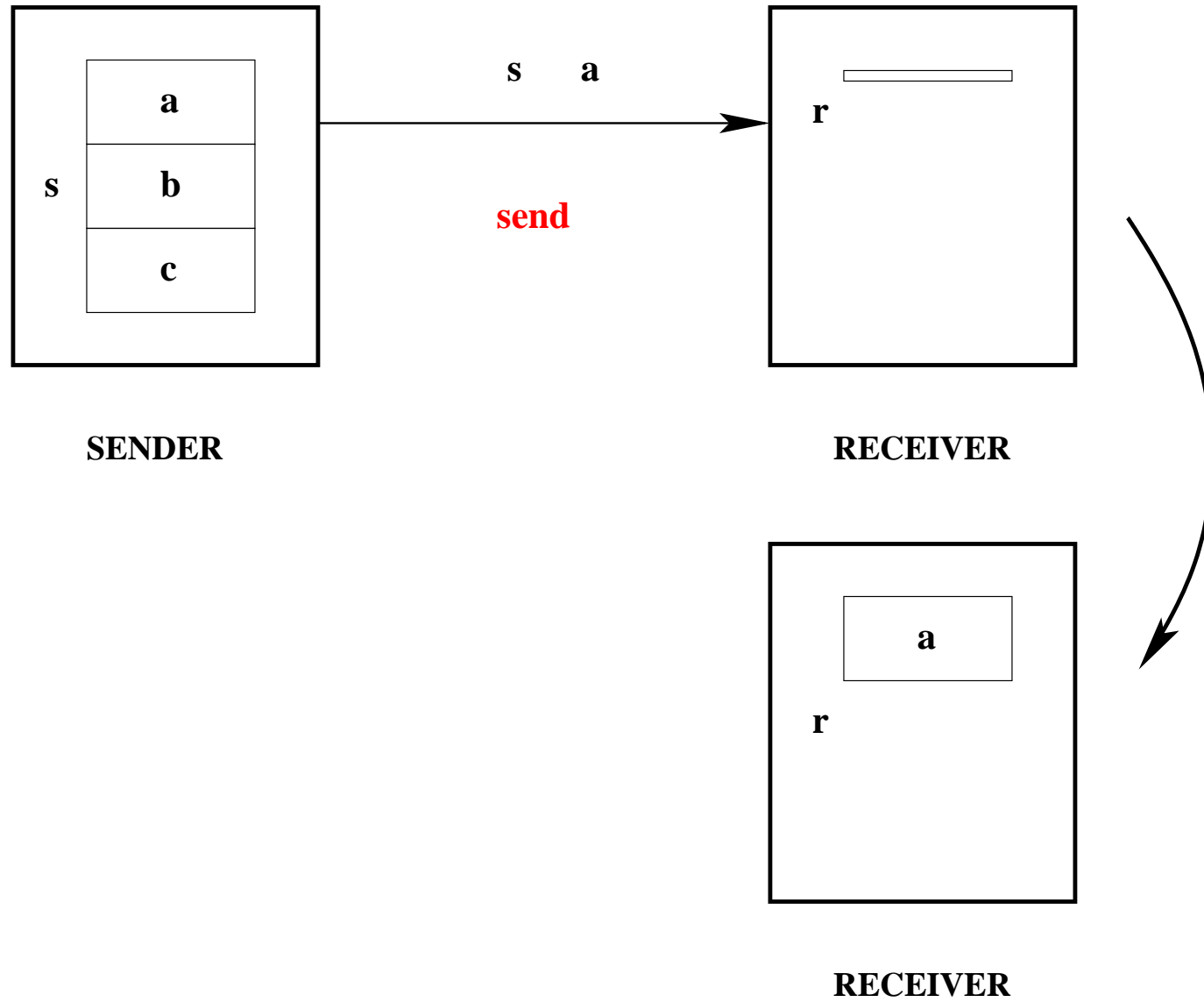
- This model is not satisfactory: event receive **accesses file f**

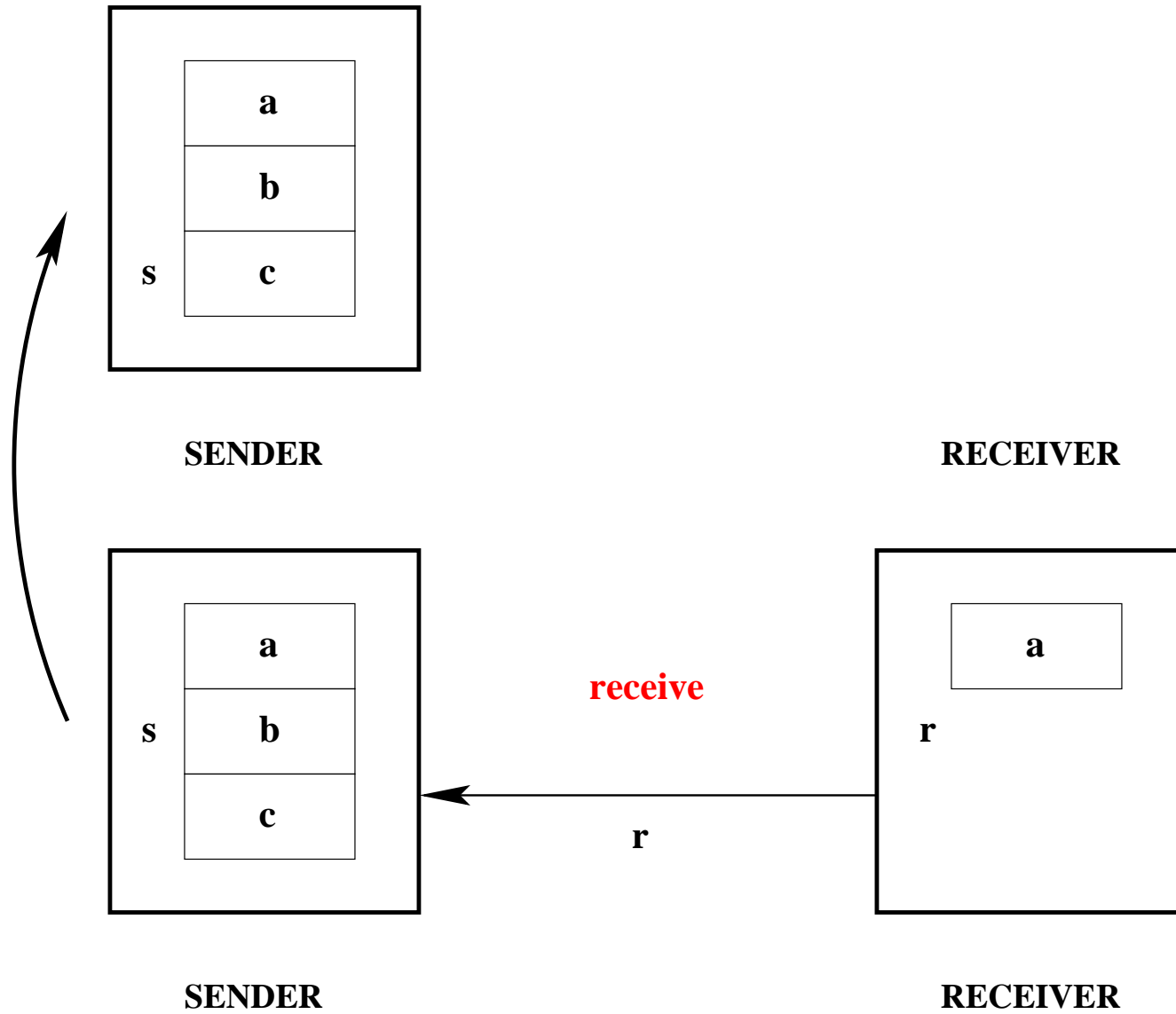
- **Initial model**: The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement**: The file is transmitted gradually (FUN3)
- **Second refinement**: The two agents are separated
- **Third refinement**: Towards an implementation
- **Decomposition**

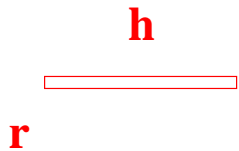
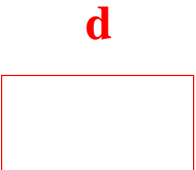
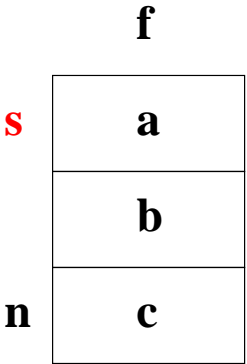


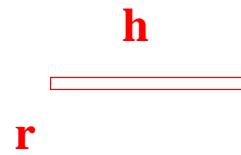
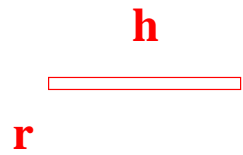
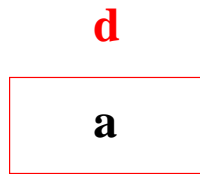
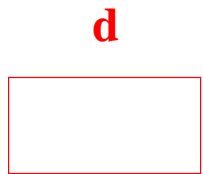
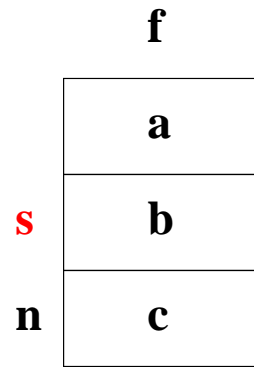
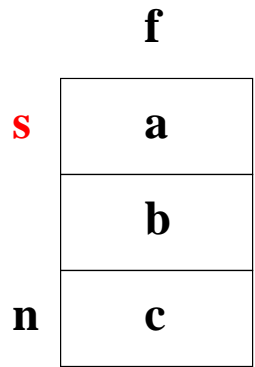


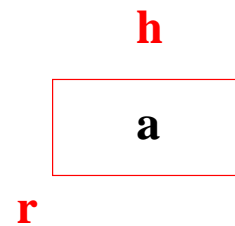
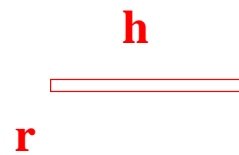
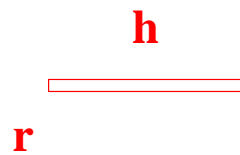
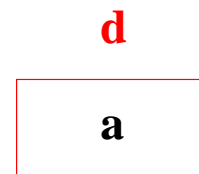
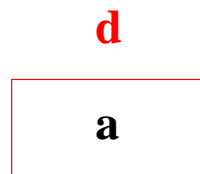
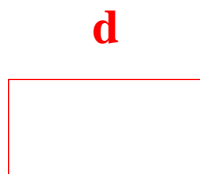
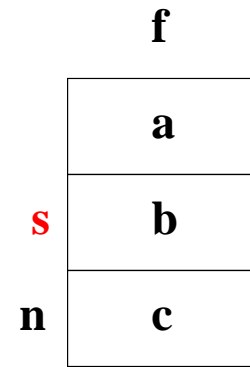
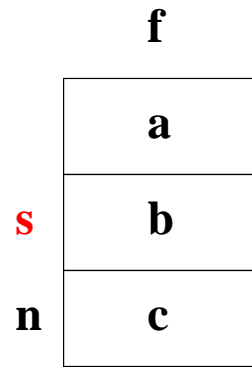
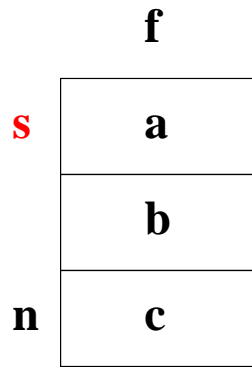


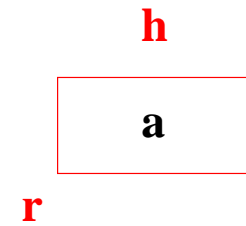
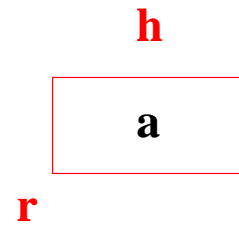
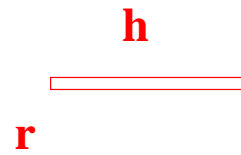
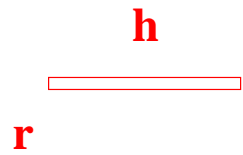
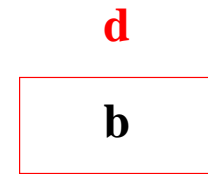
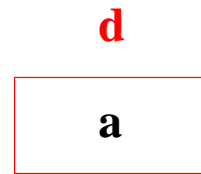
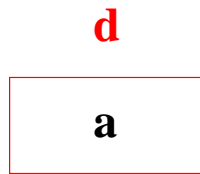
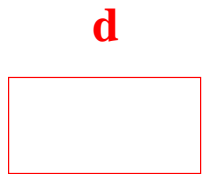
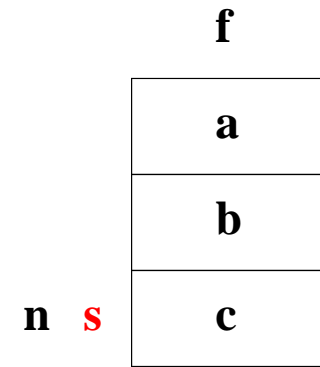
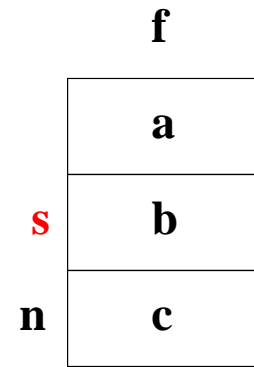
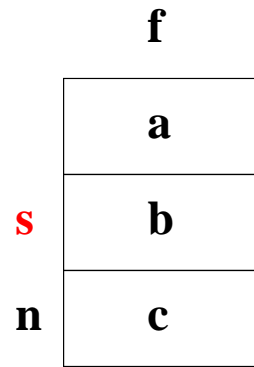
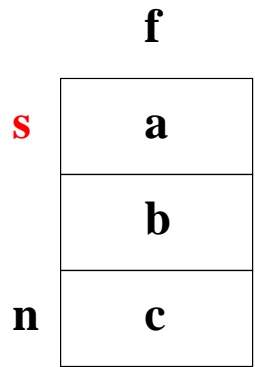


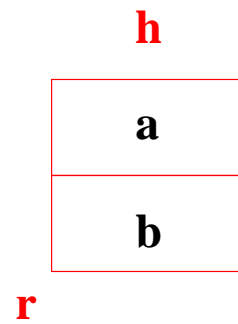
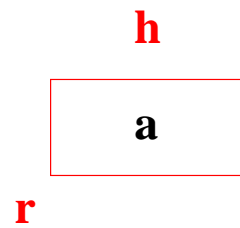
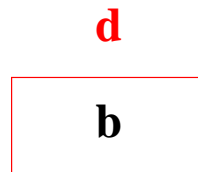
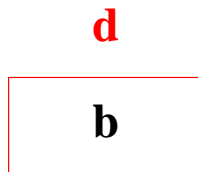
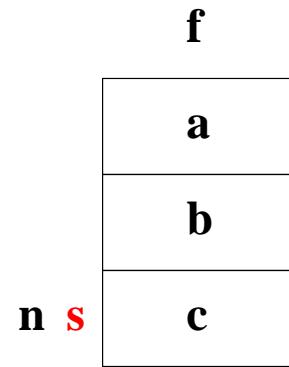
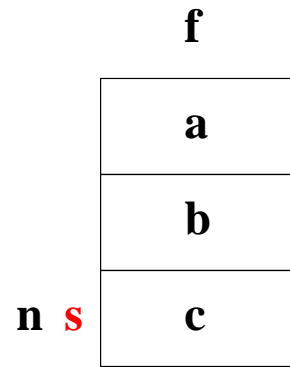


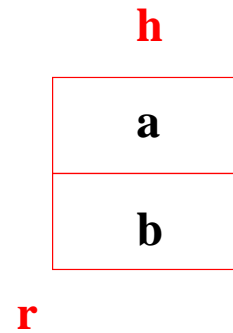
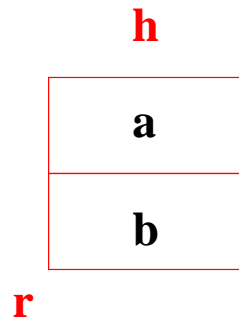
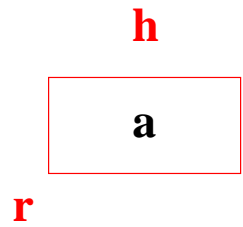
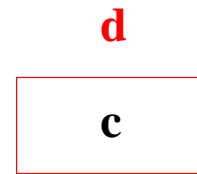
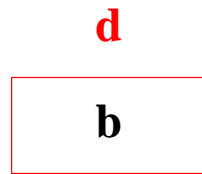
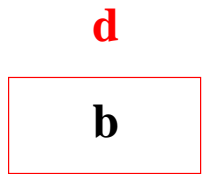
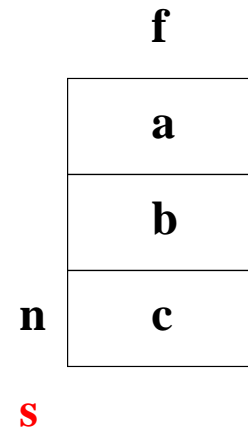
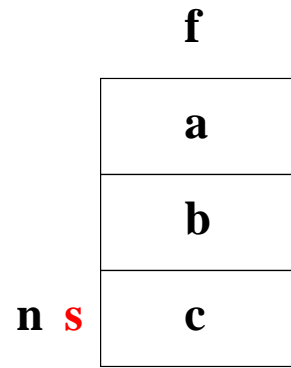
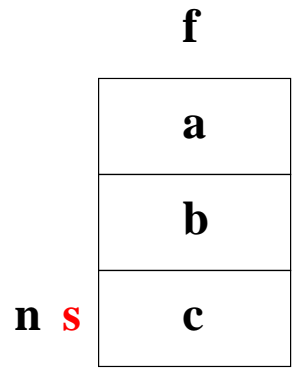


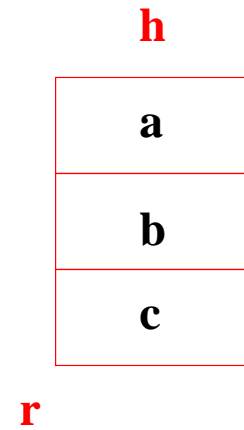
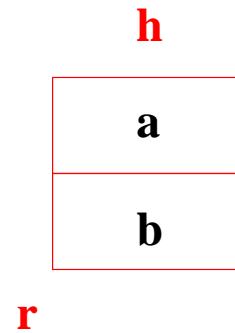
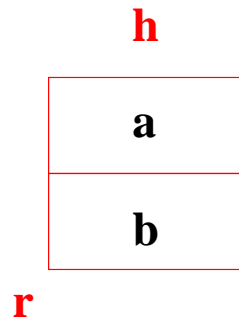
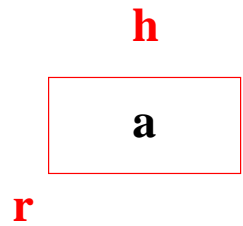
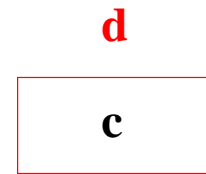
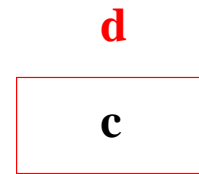
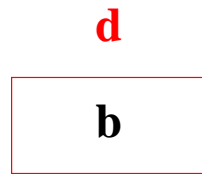
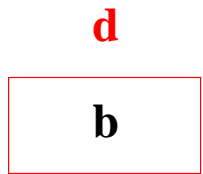
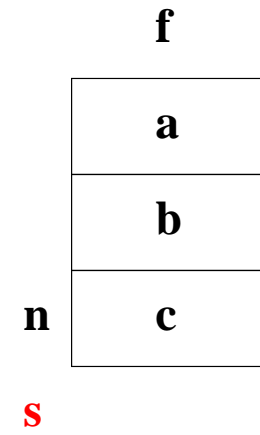
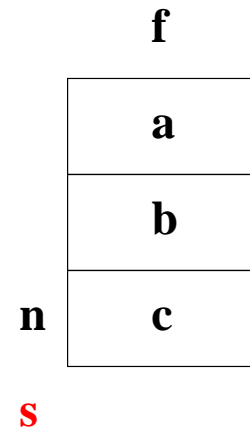
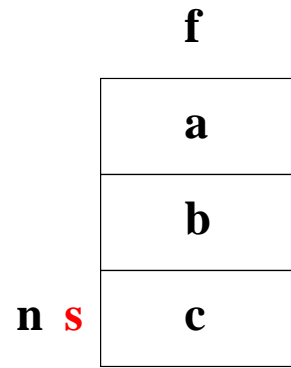
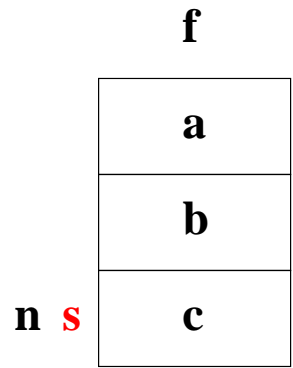


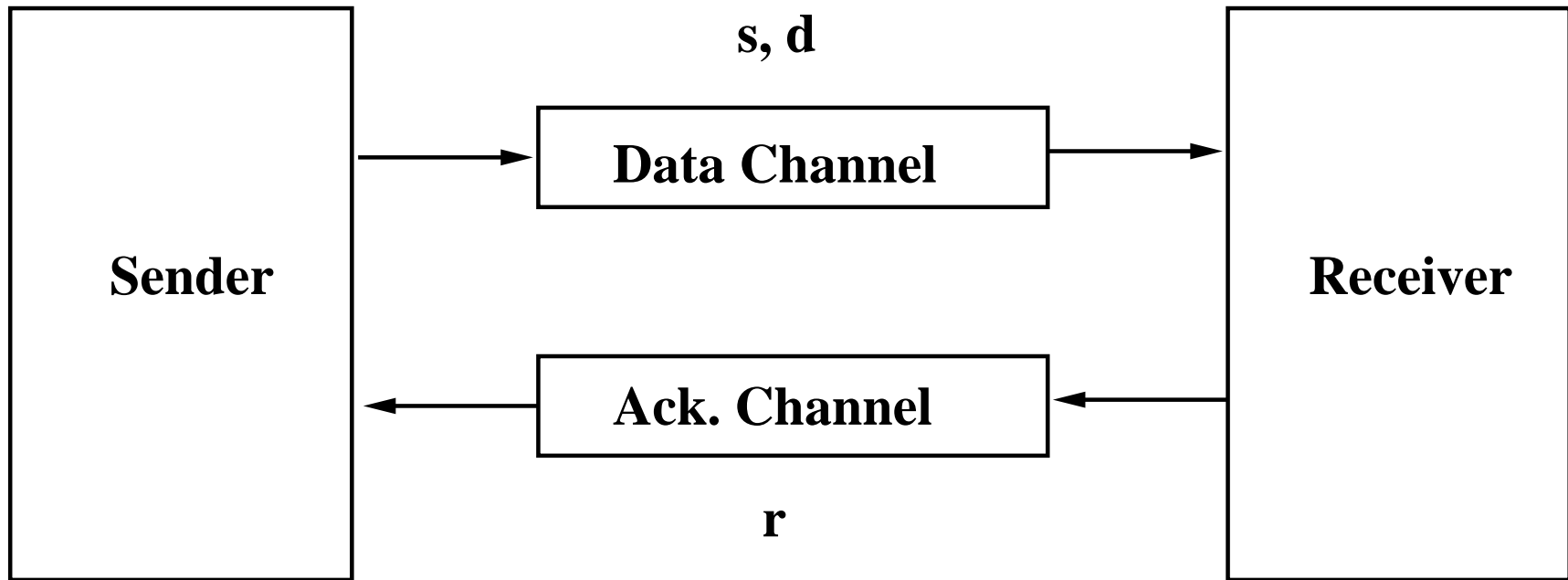












- We introduce an additional variable s , and a data item d

carrier sets: D

constants: n, f, d_0

variables: h, r, s, d

inv2_1: $s \in 1 .. n + 1$

inv2_2: $s \in r .. r + 1$

inv2_3: $d \in D$

inv2_4: $s = r + 1 \Rightarrow d = f(r)$

prp2_1: $d_0 \in D$

init

$h := \emptyset$

$s := 1$

$r := 1$

$d := d0$

send

when

$s = r$

$s \neq n + 1$

then

$d, s := f(s), s + 1$

end

receive

when

$s = r + 1$

then

$h := h \cup \{r \mapsto d\}$

$r := r + 1$

end

final

when

$s = r$

$r = n + 1$

then

skip

end

- **Event init** refines its abstraction (**INI_INV_REF**)
- **Event final** refines its abstraction (**GRD_REF** and **INV_REF**)
- **Event receive** refines its abstraction (**GRD_REF**, **INV_REF**, **EQL_REF**)
- **Event send** refines skip (**INV_REF**)
- **Event send** does not diverge (**WFD_REF1** and **WFD_REF2**)
- **Relative deadlock freeness** (**DLF_REF**)

- Applying rule EQL_REF

```

receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
    
```

```

receive
  when
     $s = r + 1$ 
  then
     $h := h \cup \{r \mapsto d\}$ 
     $r := r + 1$ 
  end
    
```

...
 invariant **inv2_2**
 invariant **inv2_4**
 conc guard of receive
 \vdash
 equality of actions

```

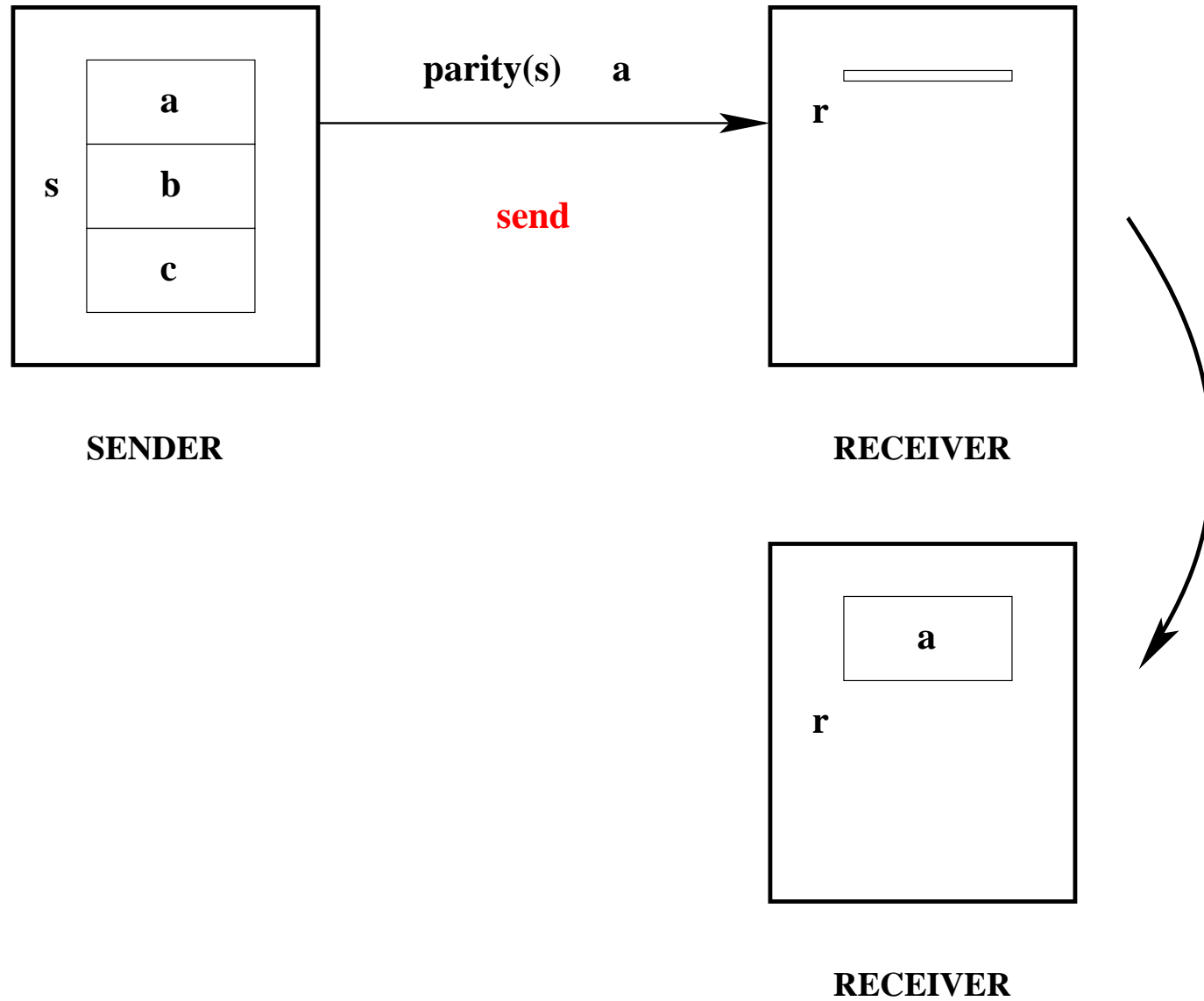
...
 $s = r + 1 \Rightarrow d = f(r)$ 
 $s = r + 1$ 
 $\vdash$ 
 $h \cup \{r \mapsto f(r)\} = h \cup \{r \mapsto d\}$ 
    
```

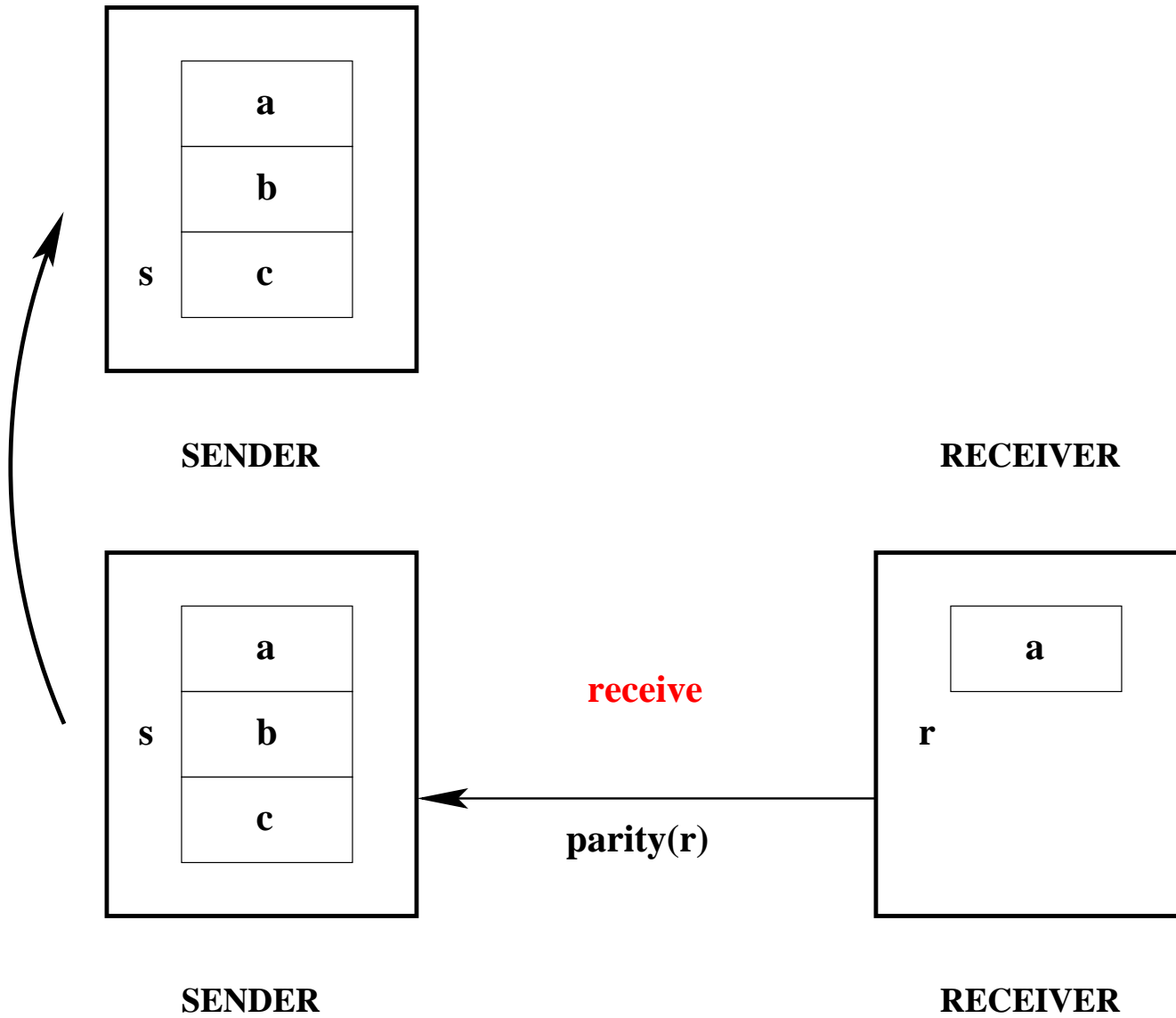
- **Initial model**: The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement**: The file is transmitted gradually (FUN3)
- **Second refinement**: The two agents are separated
- **Third refinement**: Towards an implementation
- **Decomposition**

```
send
  when
     $s = r$ 
     $s \neq n + 1$ 
  then
     $d := f(s)$ 
     $s := s + 1$ 
end
```

```
receive
  when
     $s = r + 1$ 
  then
     $h := h \cup \{r \mapsto d\}$ 
     $r := r + 1$ 
end
```

```
inv2_2:  $s \in r .. r + 1$ 
```





prp3_1: $parity \in \mathbb{N} \rightarrow \{0, 1\}$

prp3_2: $parity(0) = 0$

prp3_3: $\forall x \cdot (x \in \mathbb{N} \Rightarrow parity(x + 1) = 1 - parity(x))$

thm3_1: $\forall x, y \cdot \left(\begin{array}{l} x \in \mathbb{N} \\ y \in \mathbb{N} \\ x \in y .. y + 1 \\ parity(x) = parity(y) \\ \Rightarrow \\ x = y \end{array} \right)$

carrier sets: D

constants: $n, f, parity$

variables: h, s, r, d, p, q

inv3_1: $p = parity(s)$

inv3_2: $q = parity(r)$

prp3_1: $parity \in \mathbb{N} \rightarrow \{0, 1\}$

prp3_2: $parity(0) = 0$

prp3_3: $\forall x \cdot \left(\begin{array}{l} x \in \mathbb{N} \\ \Rightarrow \\ parity(x + 1) = 1 - parity(x) \end{array} \right)$

init

$h := \emptyset$
 $s := 1$
 $r := 1$
 $p := 1$
 $q := 1$
 $d := d0$

final

when
 $p = q$
 $r = n + 1$
then
skip
end

send

when

$p = q$
 $s \neq n + 1$

then

$d := f(s)$
 $s := s + 1$
 $p := 1 - p$

end

receive

when

$p \neq q$

then

$h := h \cup \{r \mapsto d\}$
 $r := r + 1$
 $q := 1 - q$

end

- The proofs are left as **exercises**

	Total	Interactive
Initial Model	1	0
1st Refinement	13	3
2nd Refinement	12	1
3rd Refinement	13	9
Total	39	13

-
- **Initial model**: The file is transmitted in one shot (FUN1 and FUN2)
 - **First refinement**: The file is transmitted gradually (FUN3)
 - **Second refinement**: The two agents are separated
 - **Third refinement**: Towards an implementation
 - **Decomposition** (outlook only)

- As **refinements proceed**, models can become **quite complicated**
- The **state** becomes quite large and there are **many events**
- Formal **proofs** may become **quite heavy** (noise)
- Even simply **reading** a model becomes **quite hard**
- Solution: **cutting the model into smaller pieces.**

- Variables of the model to decompose are **partitioned**
- Each sub-model has two kinds of variables: **internal** and **external**
- **Internal** variables are **not shared** between sub-models
- **External** variables are **shared** between sub-models
- **External** variables allows sub-models to **communicate**

carrier sets: D
constants: n, f
internal variables: s
external variables: d, p, q

prpS0_1: $n \in \mathbb{N}_1$

prpS0_2: $0 < n$

prpS0_3: $f \in 1 .. n \rightarrow D$

invS0_1: $s \in 1 .. n + 1$

invS0_2: $d \in D$

invS0_3: $p \in \{0, 1\}$

invS0_4: $q \in \{0, 1\}$

carrier sets: D

internal variables: h, r

external variables: d, p, q

invR0_1: $h \in \mathbb{N} \leftrightarrow D$

invR0_2: $r \in \mathbb{N}$

invR0_3: $d \in D$

invR0_4: $p \in \{0, 1\}$

invR0_5: $q \in \{0, 1\}$

-
- Events of the model to decomposed are **partitioned between the sub-models**
 - Partitioned events are called **internal events** in each sub-model
 - **External events** are added to each sub-model
 - They **simulate** the behavior of the events of the other sub-models
 - An external event is an **abstraction** of some internal event
 - Events can be **renamed** before being partitioned

- Event **send** is renamed **send_data**: it goes in the **Sender**
- Event **receive** is renamed **send_acknowledgment**: it goes in the **Receiver**
- External event **receive_data** in the Receiver simulates event **send_data** in the Sender
- External event **receive acknowledgment** in the Sender simulates event **send_acknowledgment** in the Receiver

send_data

when

$p = q$

$s \neq n + 1$

then

$d := f(s)$

$s := s + 1$

$p := 1 - p$

end

↑

receive_data

when

$p = q$

then

$d \in D$

$p \in \{0, 1\}$

end

↓

d, p
⇒

receive_acknowledgment

when

$p \neq q$

then

$q \in \{0, 1\}$

end

q
⇐

send_acknowledgment

when

$p \neq q$

then

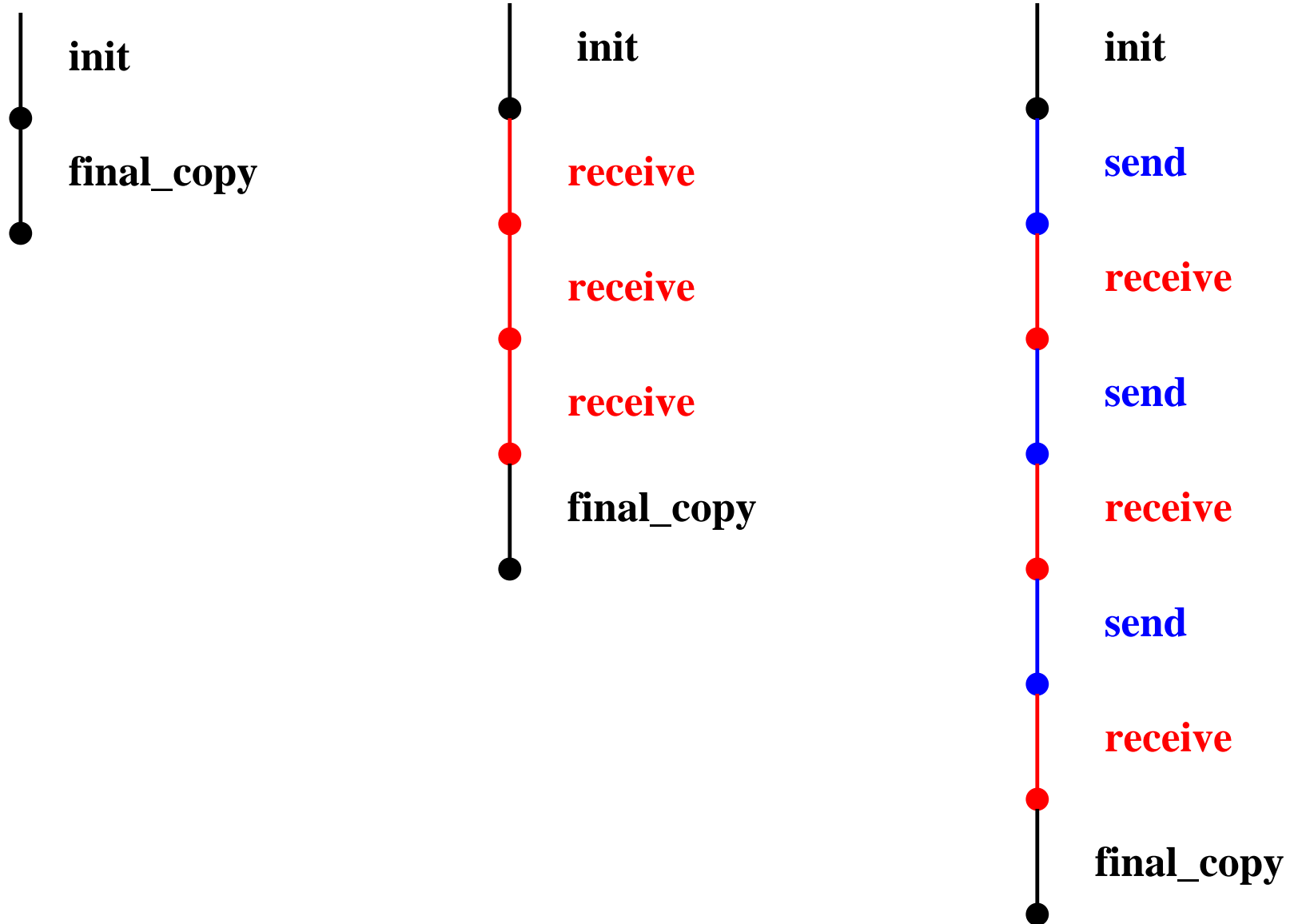
$h := h \cup \{r \mapsto d\}$

$r := r + 1$

$q := 1 - q$

end

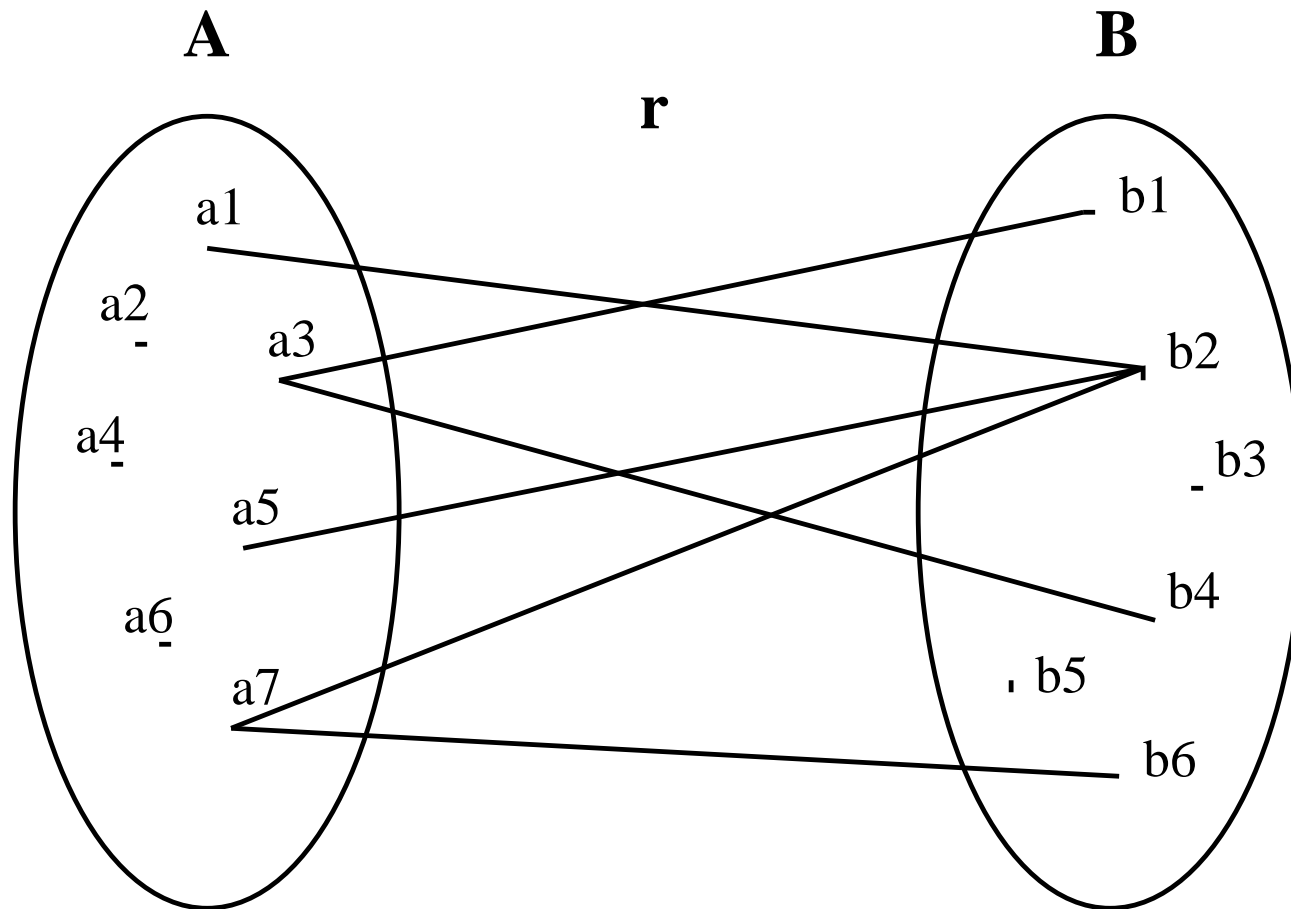
-
- More mathematical **conventions**
 - **How to write a model**
 - What kind of things we have **to prove**
 - How the proof can **help finding invariants**
 - Many things can be done by **tools**
 - A small **theory of parities**

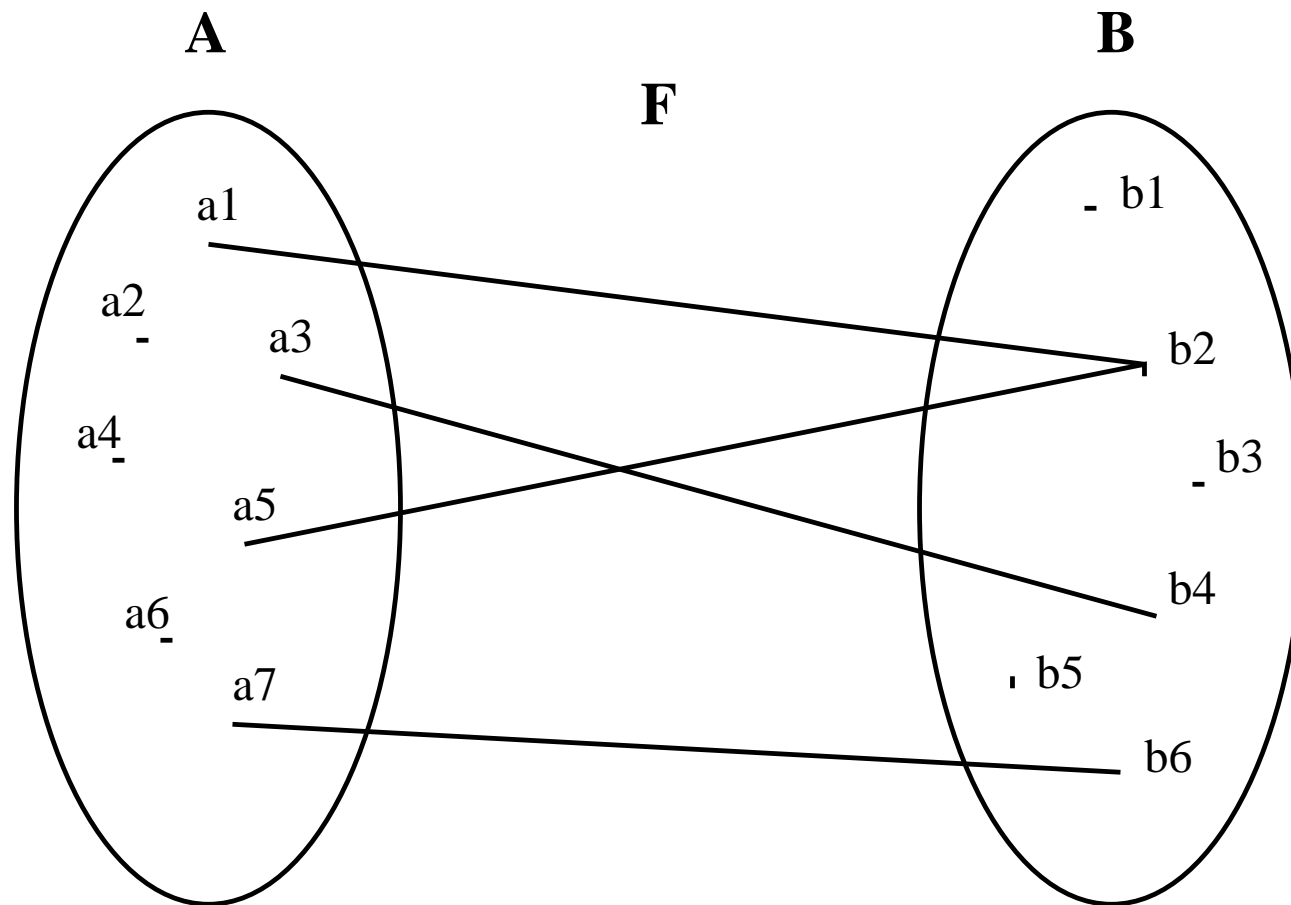


$x \in S$	Set membership operator
\mathbb{N}	set of Natural Numbers: $\{0, 1, 2, 3, \dots\}$
$a .. b$	Interval from a to b : $\{a, a + 1, \dots, b\}$ (empty when $b < a$)
$a \mapsto b$	pair constructing operator
$S \times T$	Cartesian product operator
$S \subseteq T$	set inclusion operator
$\mathbb{P}(S)$	power set operator

$S \leftrightarrow T$	Set of binary relations from S to T
$S \rightarrow T$	Set of total functions from S to T
$S \twoheadrightarrow T$	Set of partial functions from S to T
$\text{dom}(r)$	Domain of a relation r
$\text{ran}(r)$	Range of a relation r

$s \triangleleft r$	domain restriction operator
$s \triangleleft r$	domain subtraction operator
$r \triangleright t$	range restriction operator
$r \triangleright t$	range subtraction operator

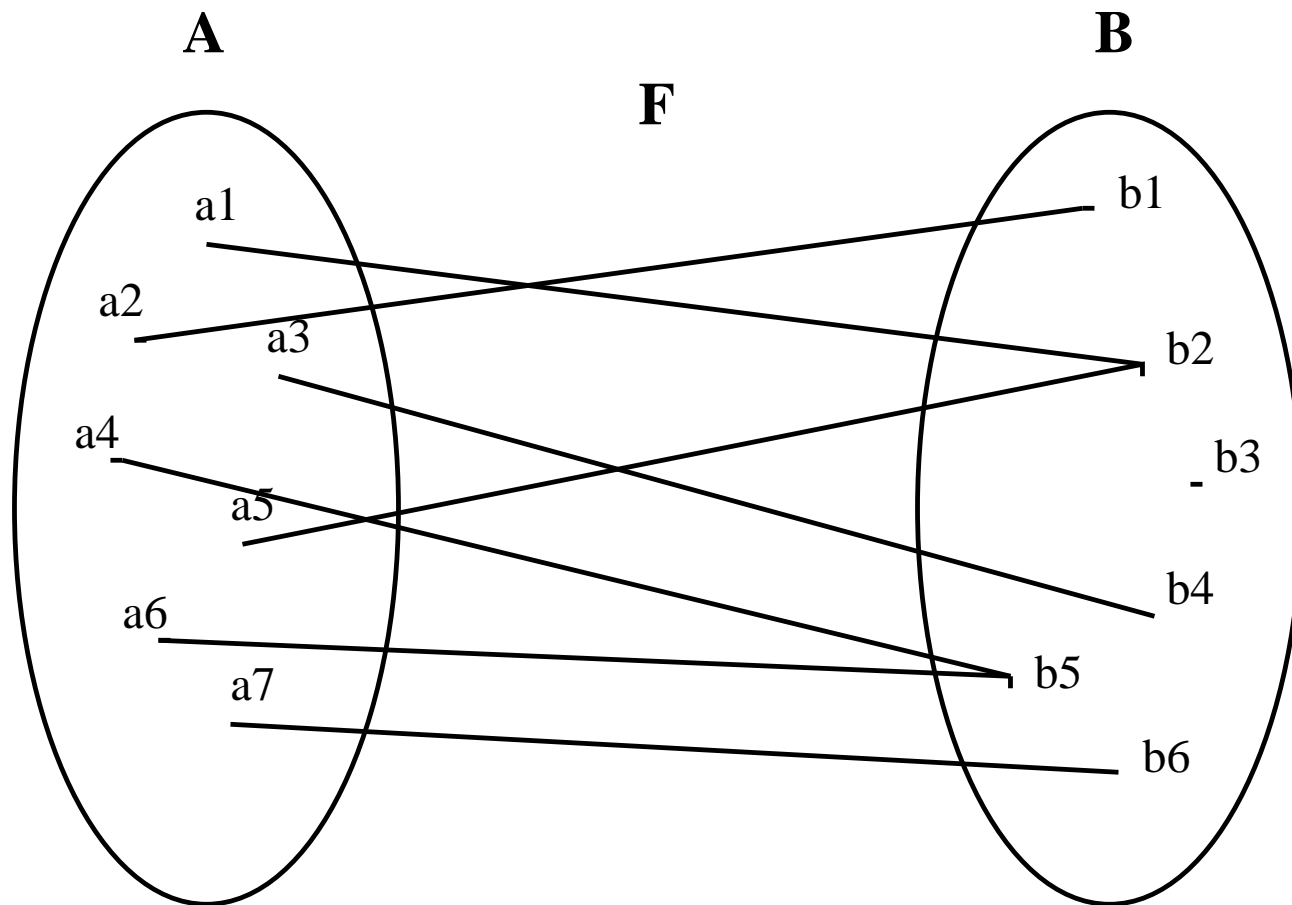




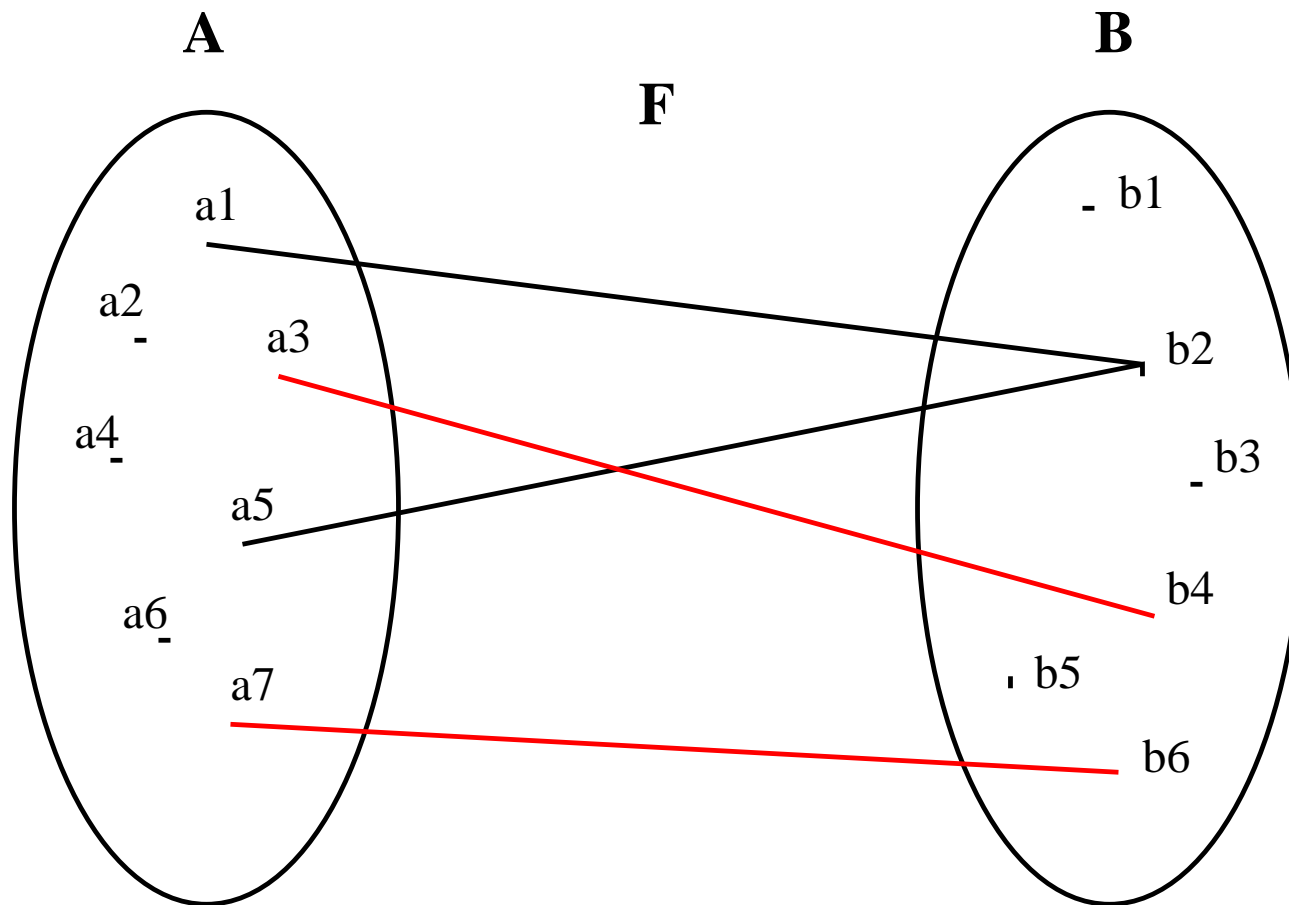
$$F = \{a1 \mapsto b2, a3 \mapsto b4, a5 \mapsto b2, a7 \mapsto b6\}$$

$$\text{dom}(F) = \{a1, a3, a5, a7\}$$

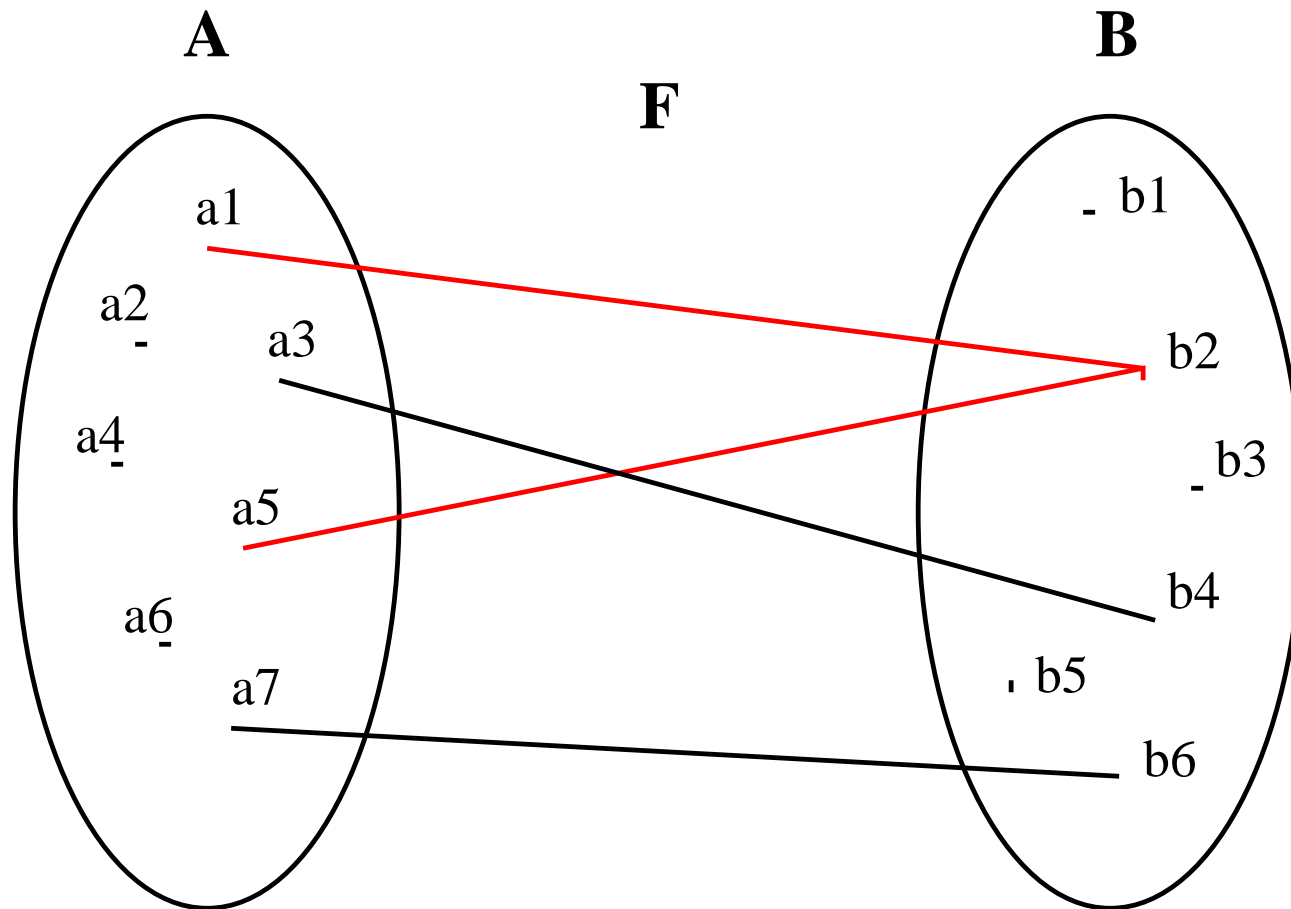
$$\text{ran}(F) = \{b2, b4, b6\}$$



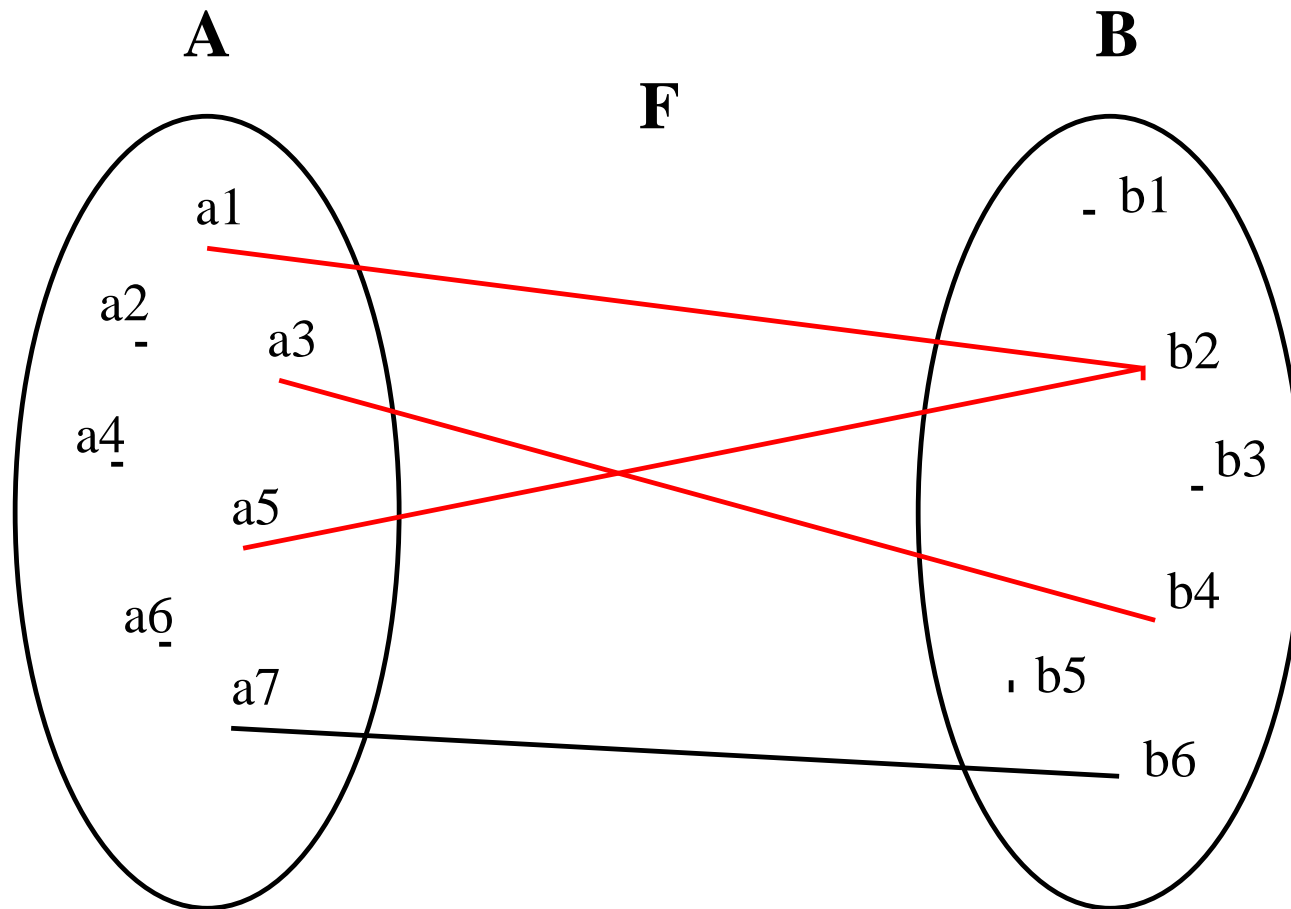
$$\text{dom}(F) = A$$



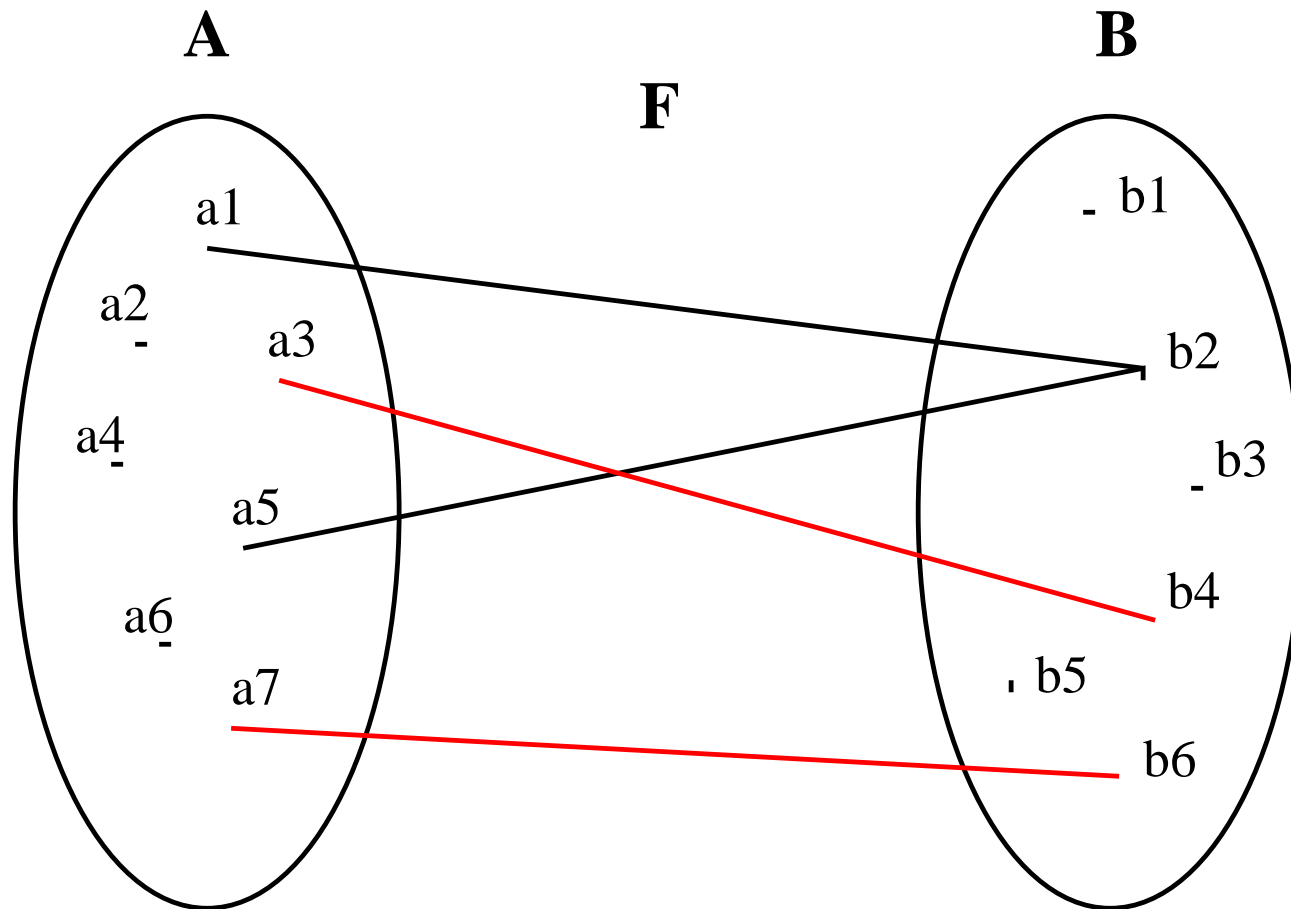
$$\{a_3, a_7\} \triangleleft F$$



$$\{a_3, a_7\} \triangleleft F$$



$$F \triangleright \{b2, b4\}$$



$$F \triangleright \{b_2\}$$

-
- List of **Carrier Sets** (identifiers)
 - List of **Constants** (identifiers)
 - List of **Properties** (predicates built on sets and constants)
 - List of **Variables** (identifiers)
 - List of **Invariants** (predicates built on sets, constants, and variables)
 - List of **Events**