

# Event-B Course

## 4. File Transfer Protocol

Jean-Raymond Abrial

September-October-November 2011

- 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 sent piece by piece between the two sites

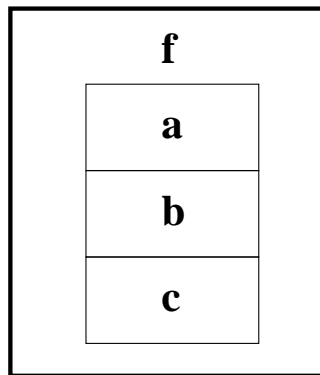
FUN-3

- 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**

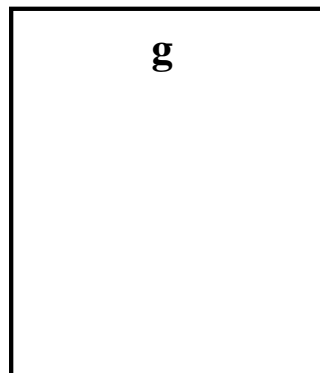
- **Initial model**: The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement**: The file is transmitted gradually (FUN3)
- **Second refinement**: The two agents are separated
- **Third refinement**: Towards an implementation

## INITIAL SITUATION

### SENDER

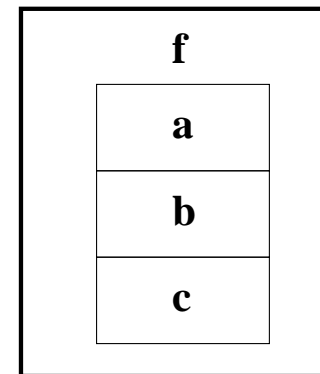


### RECEIVER

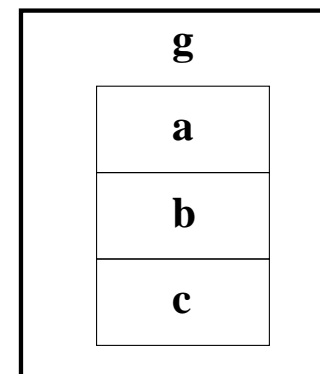


## FINAL SITUATION

### SENDER



### RECEIVER



	<b>f</b>
<b>1</b>	<b>a</b>
	<b>b</b>
<b>n</b>	<b>c</b>



**sets:**  $D$

**constants:**  $n$   
 $f$

**axm0\_1:**  $0 < n$

**axm0\_2:**  $f \in 1 .. n \rightarrow D$

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

**variables:**  $g$   
 $b$

**inv0\_1:**  $g \in 1 .. n \rightarrow D$

**inv0\_2:**  $b = \text{FALSE} \Rightarrow g = \emptyset$

**inv0\_3:**  $b = \text{TRUE} \Rightarrow g = f$

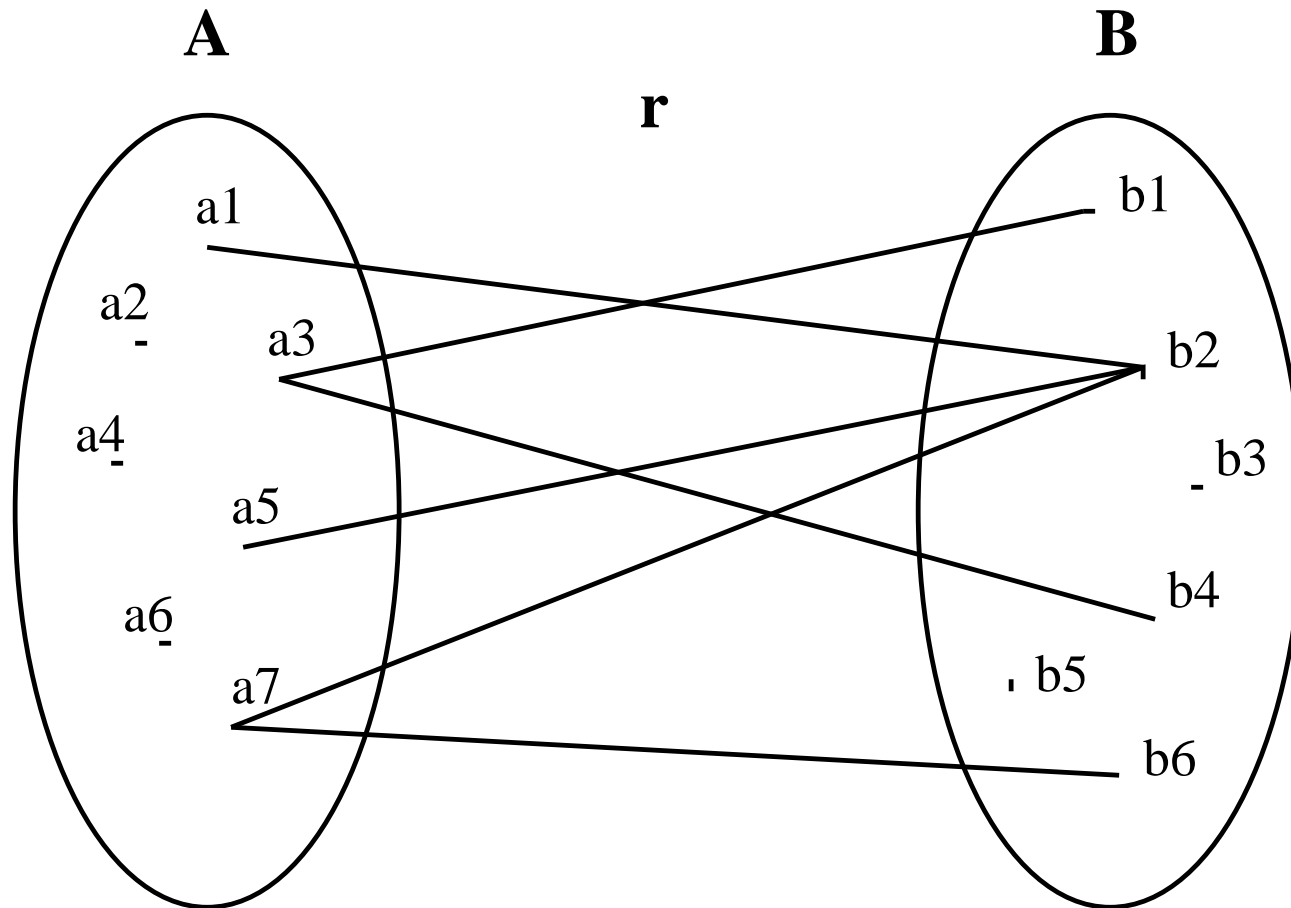
$b = \text{TRUE}$  means protocol is finished

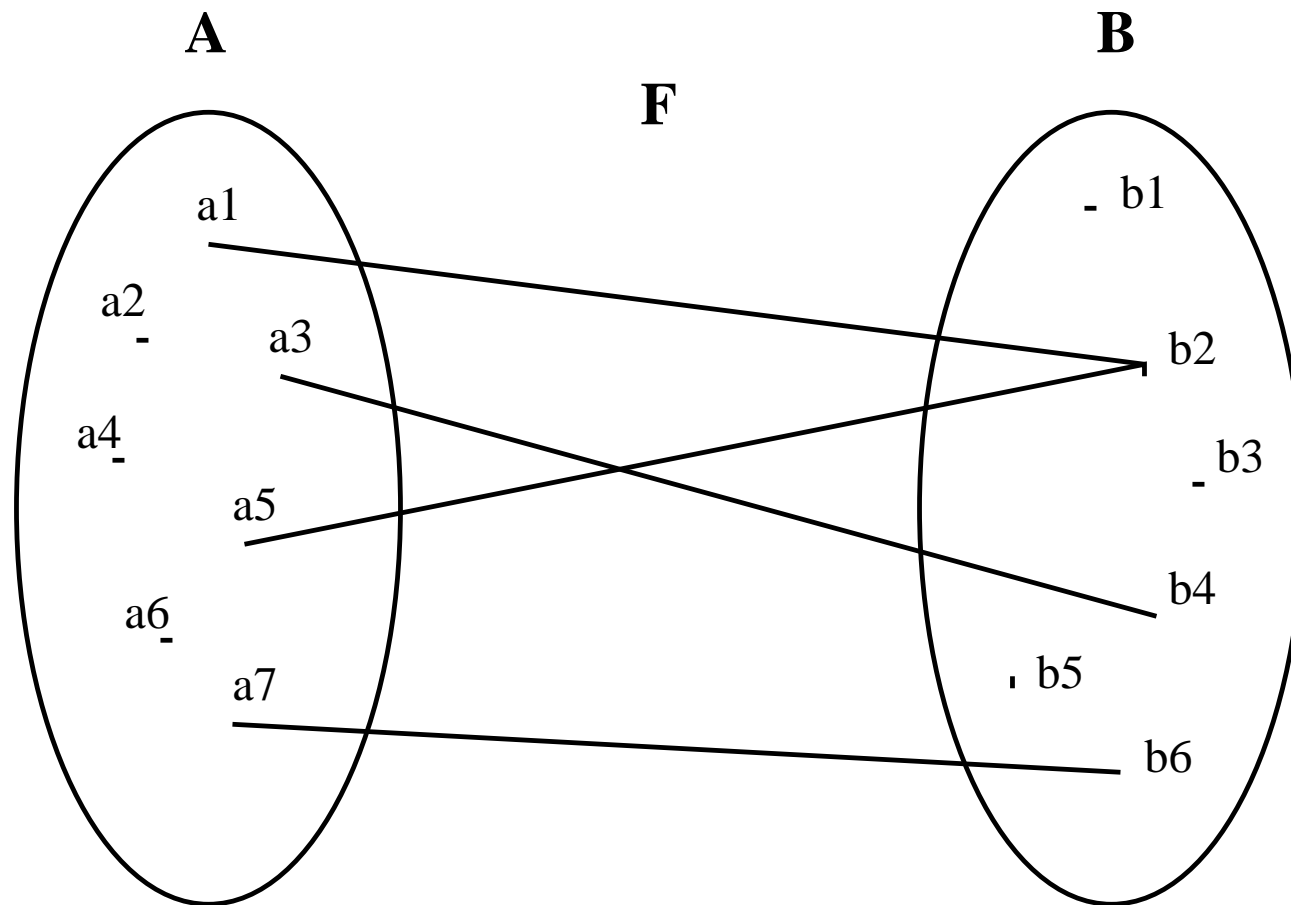
---

$x \in S$	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 $r$

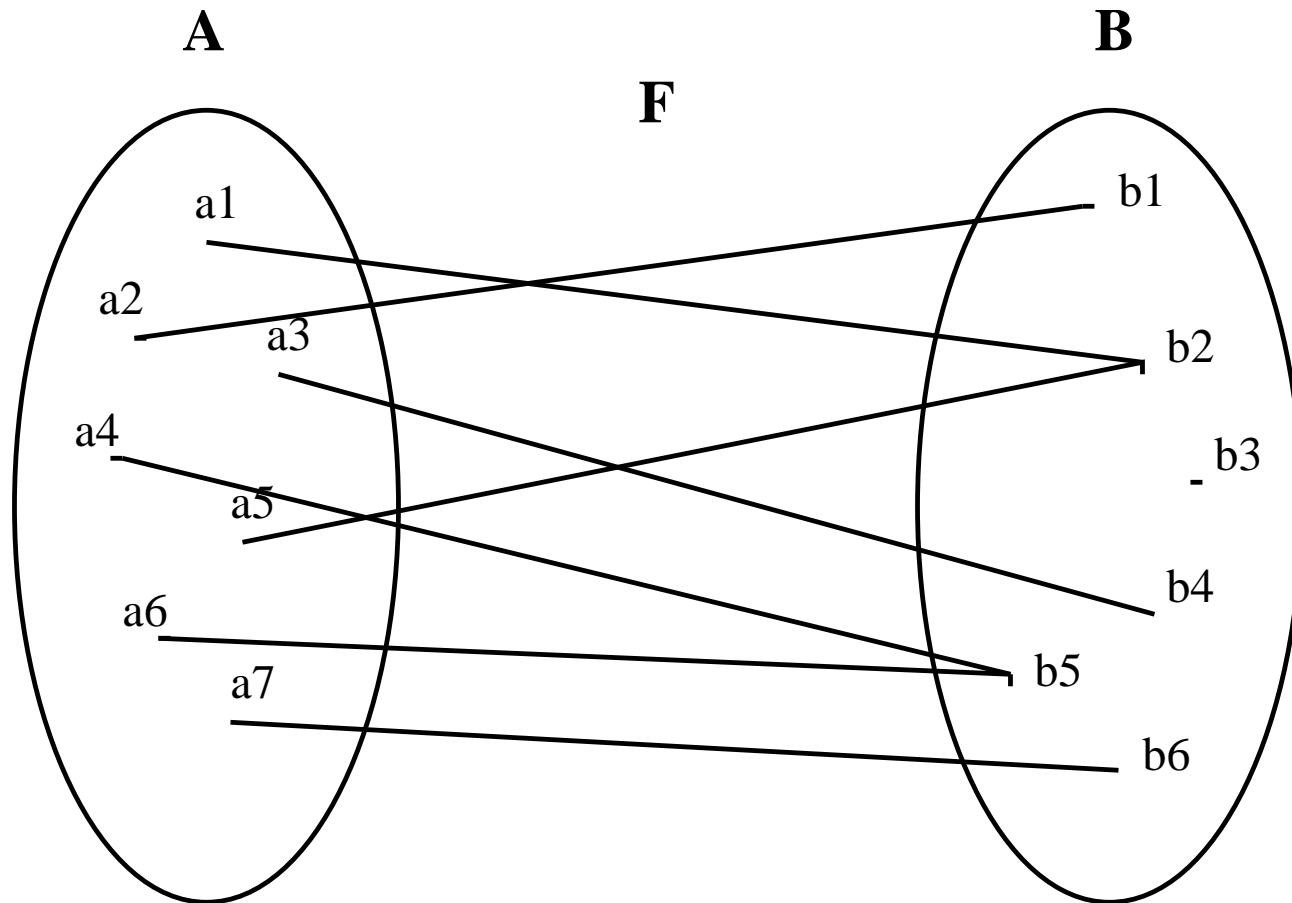




$$F = \{a_1 \mapsto b_2, a_3 \mapsto b_4, a_5 \mapsto b_2, a_7 \mapsto b_6\}$$

$$\text{dom}(F) = \{a_1, a_3, a_5, a_7\}$$

$$\text{ran}(F) = \{b_2, b_4, b_6\}$$



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

init

$g := \emptyset$

$b := \text{FALSE}$

final

**when**

$b = \text{FALSE}$

**then**

$g := f$

$b := \text{TRUE}$

**end**



sets:  $D$

constants:  $n$   
 $f$

axm0\_1:  $0 < n$

axm0\_2:  $f \in 1 .. n \rightarrow D$

variables:  $g$   
 $b$

inv0\_1:  $g \in 1 .. n \rightarrow D$

inv0\_2:  $b = \text{FALSE} \Rightarrow g = \emptyset$

inv0\_3:  $b = \text{TRUE} \Rightarrow g = f$

init

$g := \emptyset$   
 $b := \text{FALSE}$

final

**when**

$b = \text{FALSE}$

**then**

$g := f$

$b := \text{TRUE}$

**end**

- Event **init establishes** invariants **inv0\_1** to **inv0\_3** (Rule INV)
- Event **final preserves** invariants **inv0\_1** to **inv0\_3** (Rule INV)

**inv0\_1:**  $g \in 1 .. n \leftrightarrow D$

**inv0\_2:**  $b = \text{FALSE} \Rightarrow g = \emptyset$

**inv0\_3:**  $b = \text{TRUE} \Rightarrow g = f$

init

$g := \emptyset$   
 $b := \text{FALSE}$

final

**when**

$b = \text{FALSE}$

**then**

$g := f$

$b := \text{TRUE}$

**end**

- For the init event in the initial model

Properties ⊢ Modified Invariant	INV
---------------------------------------	-----

- Applying Rule INV to invariant **inv0\_1**

init

$g := \emptyset$   
 $b := \text{FALSE}$

**inv0\_1**:  $g \in 1 .. n \rightarrow D$

**axm0\_1**

**axm0\_2**

⊢

modified **inv0\_1**

$0 < n$

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

⊢

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

**inv0\_1** / INV

- Applying Rule INV to invariant **inv0\_2**

init

$g := \emptyset$

$b := \text{FALSE}$

**inv0\_2**:  $b = \text{FALSE} \Rightarrow g = \emptyset$

**axm0\_1**  
**axm0\_2**

⊢

modified **inv0\_2**

$0 < n$

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

⊢

$\text{FALSE} = \text{FALSE} \Rightarrow \emptyset = \emptyset$

**inv0\_2** / INV

- Applying Rule INV to invariant **inv0\_3**

init  
 $g := \emptyset$   
 $b := \mathbf{FALSE}$

**inv0\_3**:  $b = \mathbf{TRUE} \Rightarrow g = f$

**axm0\_1**  
**axm0\_2**  
 $\vdash$   
 modified **inv0\_3**

$0 < n$   
 $f \in 1 .. n \rightarrow D$   
 $\vdash$   
 $\mathbf{FALSE} = \mathbf{TRUE} \Rightarrow \emptyset = f$

**inv0\_3** / INV

- For other events in the initial model

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

- Applying Rule INV

```

final
  when
    b = FALSE
  then
    g := f
    b := TRUE
  end
    
```

axm0\_1  
 axm0\_2  
 inv0\_1  
 ...  
 grd  
 ⊢  
 modified inv0\_1

```

0 < n
f ∈ 1 .. n → D
g ∈ 1 .. n ⇔ D
...
b = FALSE
⊢
f ∈ 1 .. n ⇔ D
    
```

final / inv0\_1 / INV



- Applying Rule INV

```

final
  when
    b = FALSE
  then
    g := f
    b := TRUE
  end
    
```

axm0\_1  
 axm0\_2  
 inv0\_2  
 ...  
 grd  
 ⊢  
 modified inv0\_2

```

0 < n
f ∈ 1 .. n → D
b = FALSE ⇒ g = ∅
...
b = FALSE
⊢
TRUE = FALSE ⇒ f = ∅
    
```

final / inv0\_2 / INV

- Applying Rule INV

```

final
  when
    b = FALSE
  then
    g := f
    b := TRUE
  end
    
```

axm0\_1  
 axm0\_2  
 inv0\_3  
 ...  
 grd  
 ⊢  
 modified inv0\_3

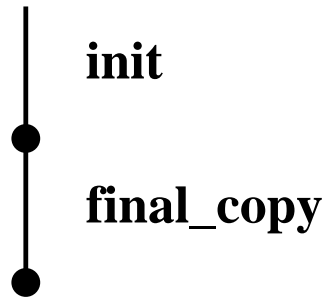
```

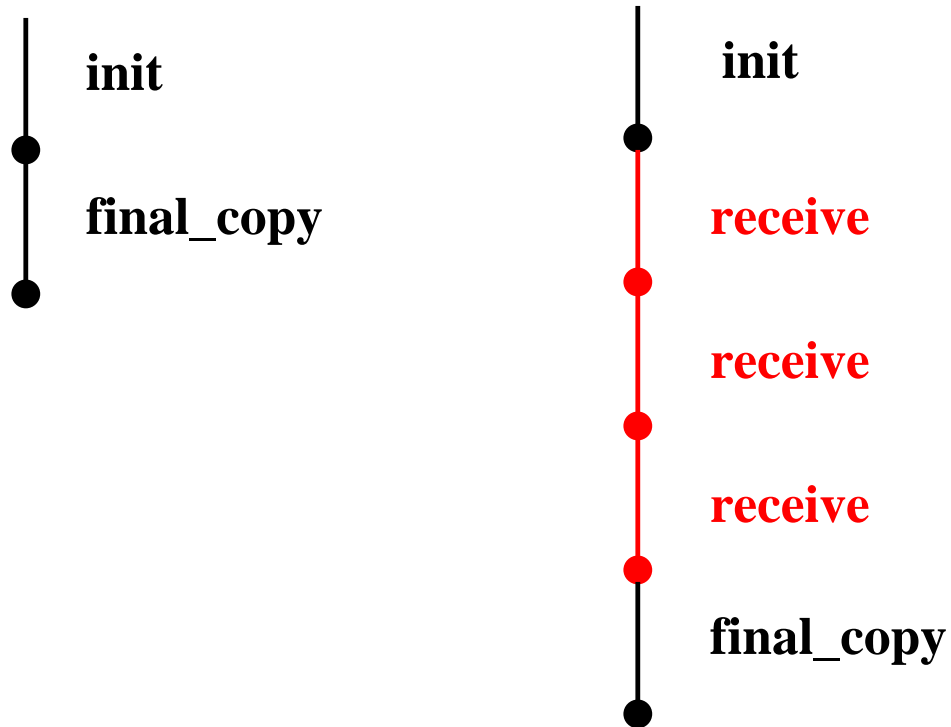
0 < n
f ∈ 1 .. n → D
b = TRUE ⇒ g = f
...
b = FALSE
⊢
TRUE = TRUE ⇒ f = f
    
```

final / inv0\_3 / INV

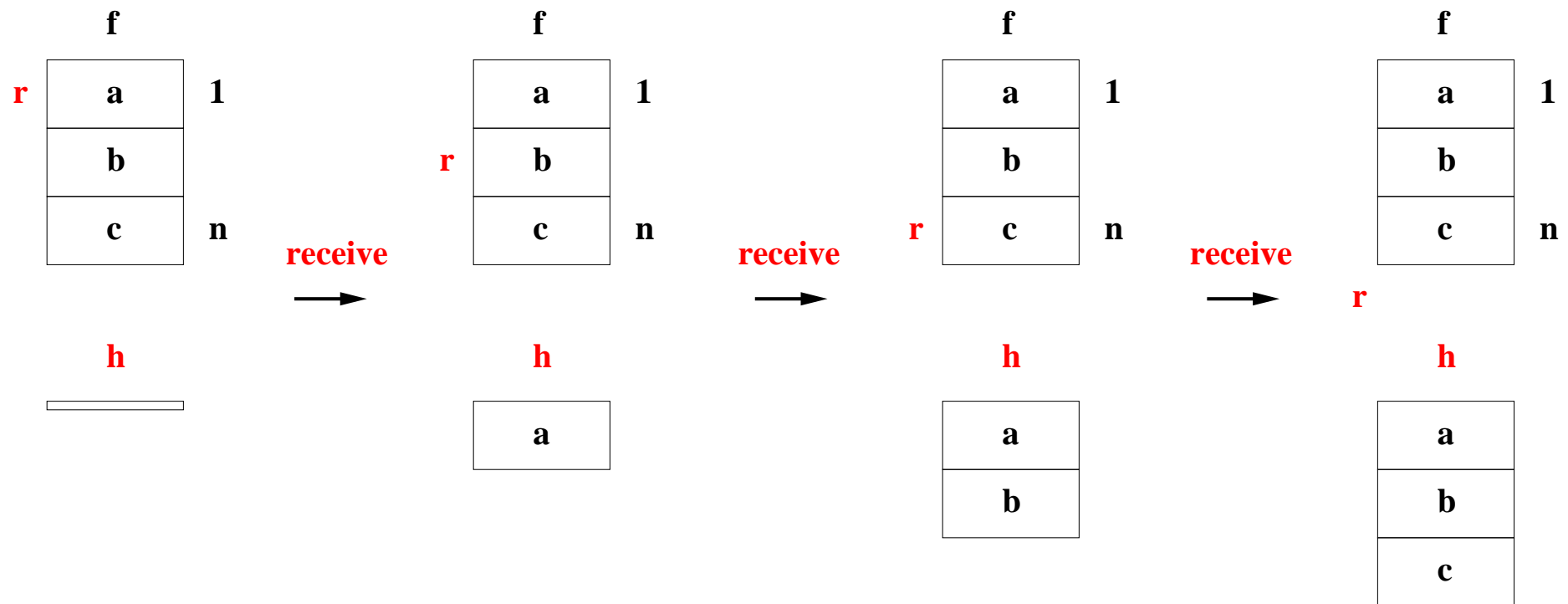
- **Initial model:** The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement:** The file is transmitted gradually (FUN3)
- **Second refinement:** The two agents are separated
- **Third refinement:** Towards an implementation

- 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:**  $b$   
 $h$   
 $r$

**inv1\_1:**  $r \in 1 .. n + 1$

**inv1\_2:**  $h = (1 .. r - 1) \triangleleft f$

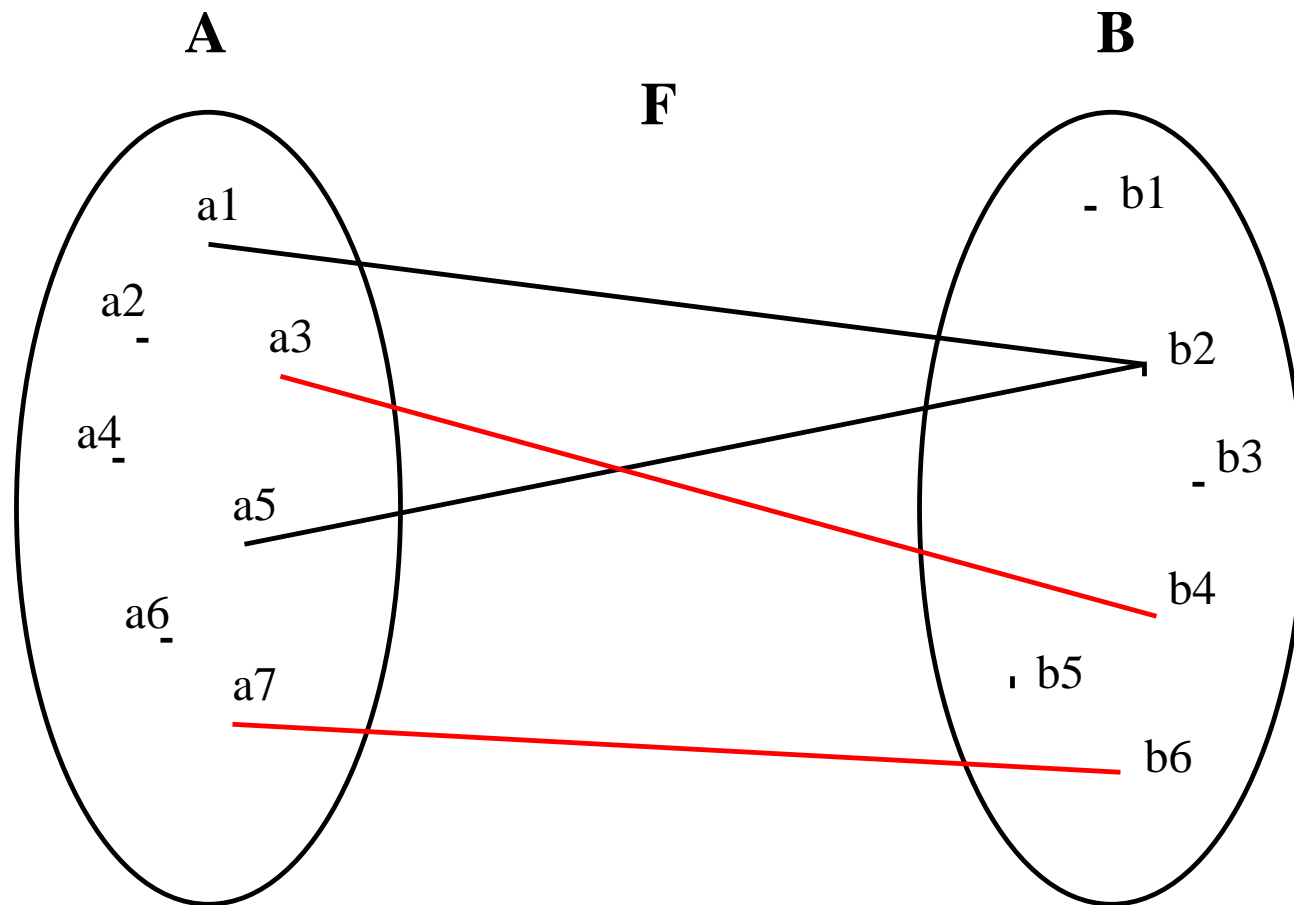
**inv1\_3:**  $b = \text{TRUE} \Rightarrow r = n + 1$

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

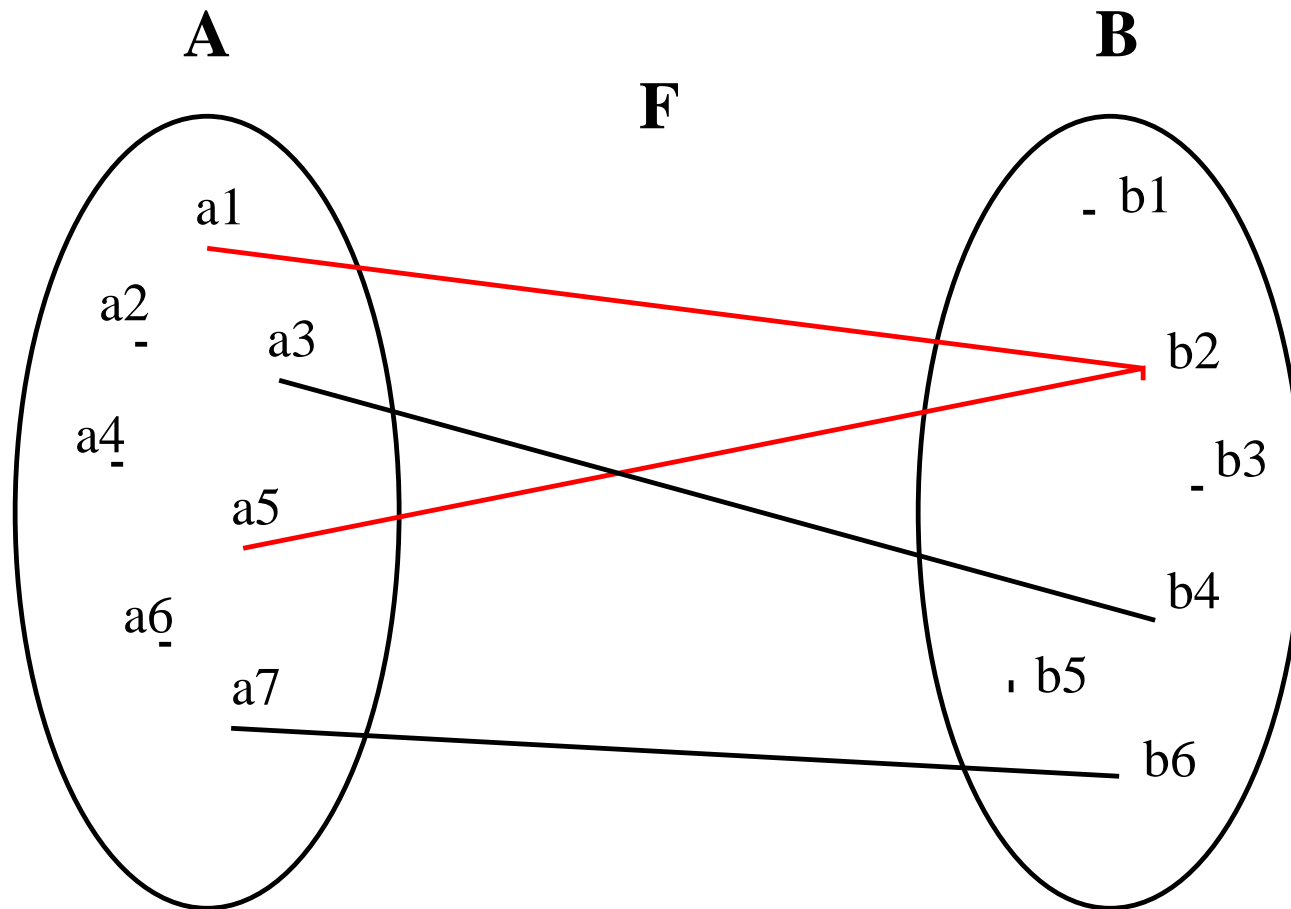


---

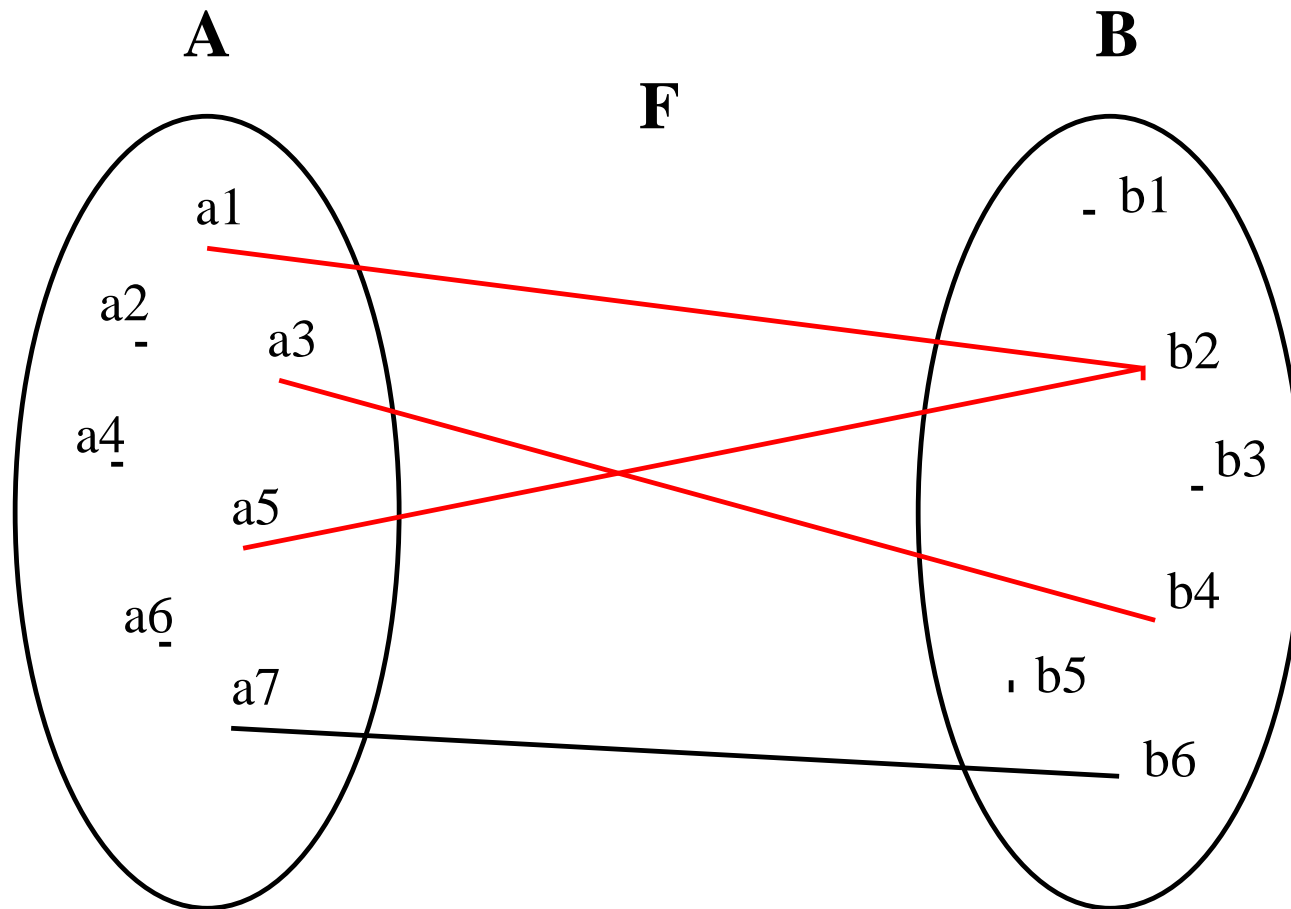
$s \triangleleft r$	domain restriction operator
$s \triangleleft r$	domain subtraction operator
$r \triangleright t$	range restriction operator
$r \triangleright t$	range subtraction operator



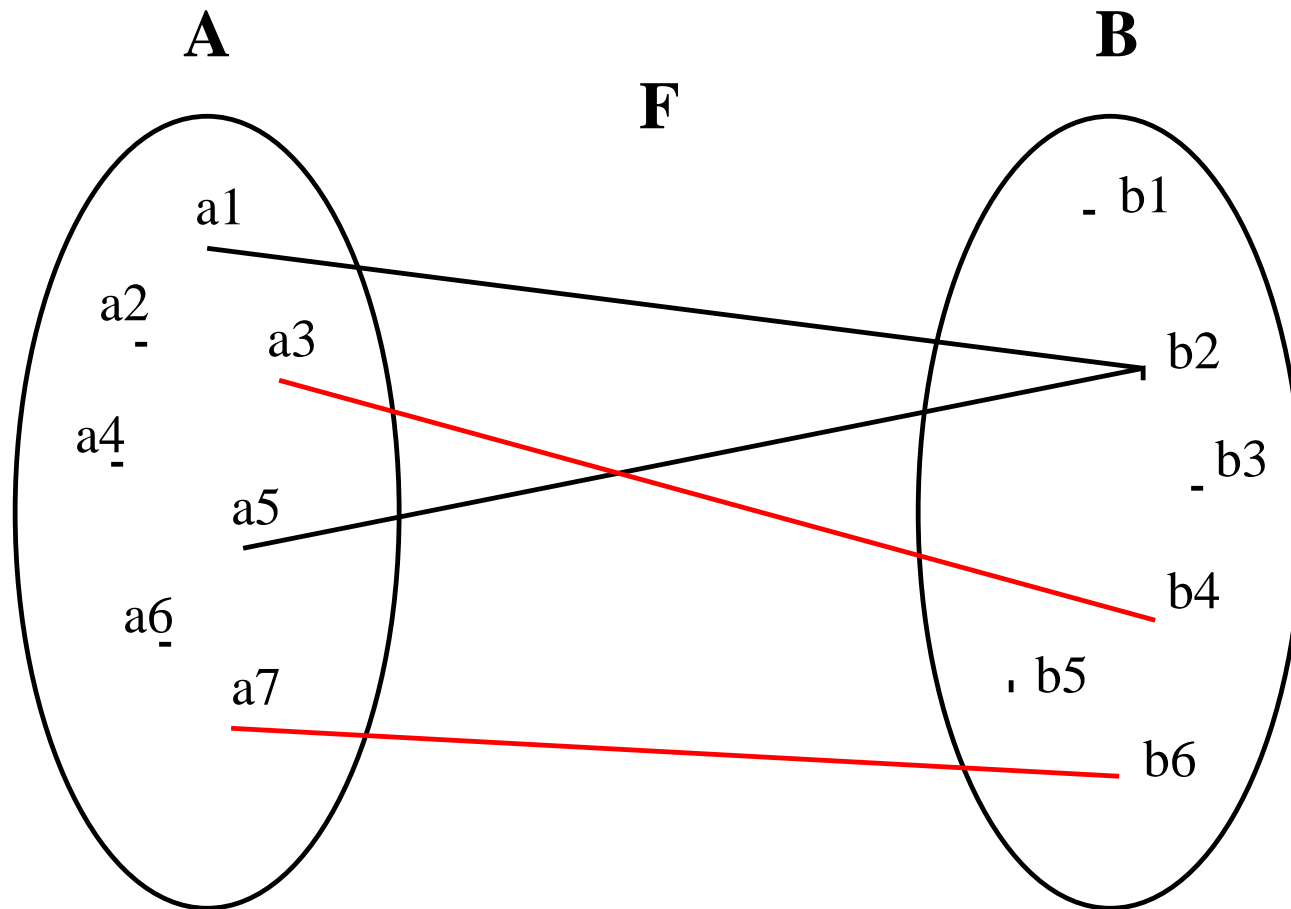
$$\{a_3, a_7\} \triangleleft F$$



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



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



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

init

$h := \emptyset$

$r := 1$

$b := \text{FALSE}$

receive

**when**

$r \leq n$

**then**

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

$r := r + 1$

**end**

final

**when**

$r = n + 1$

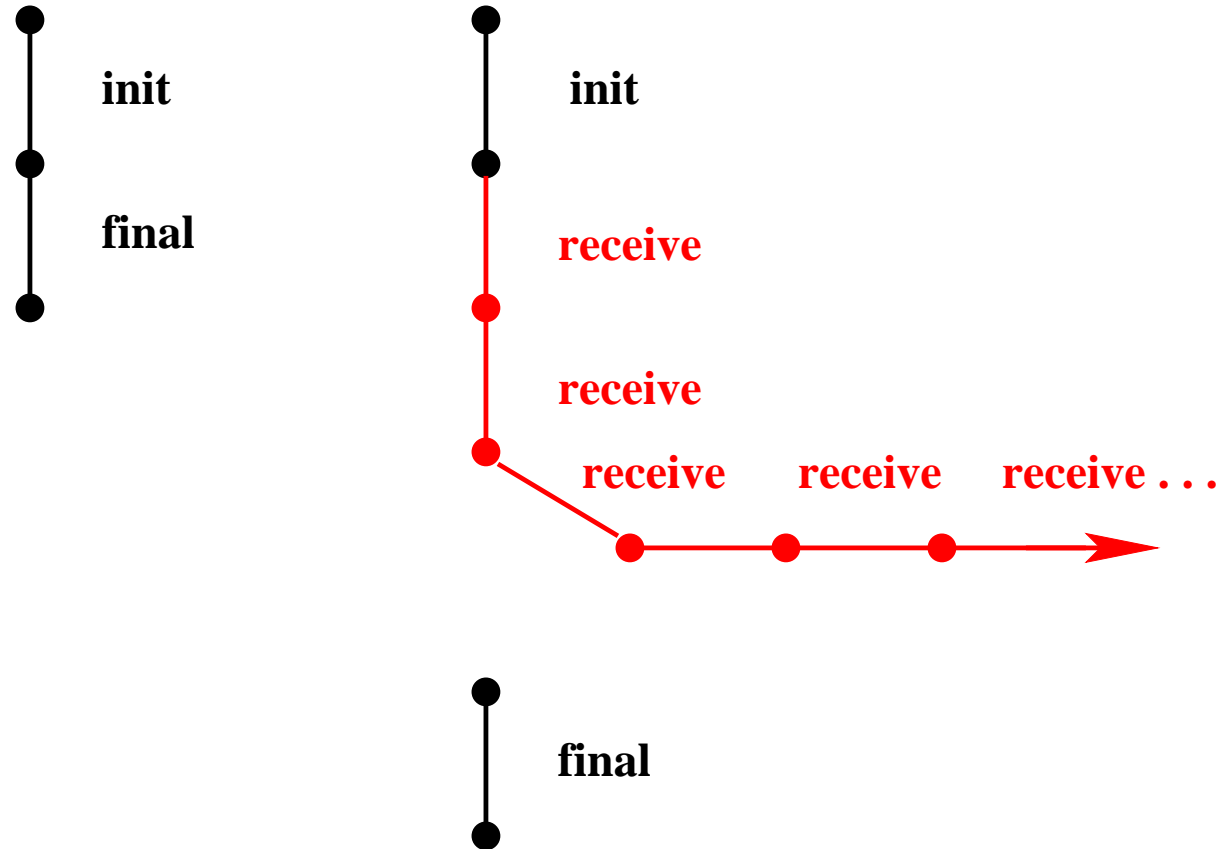
$b = \text{FALSE}$

**then**

$b := \text{TRUE}$

**end**

- **Event init** refines its abstraction
- **Event final** refines its abstraction
- **Event receive** refines skip
- **Event receive** does not diverge
- **Relative deadlock freeness**





- No divergence of new event receive (rules NAT and VAR)

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

- 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 $\in \mathbb{N}$	NAT
---	-----

- Applying rule **NAT**

...

**inv1\_1**

...

⊢

variant belongs to  $\mathbb{N}$

...

$r \in 1 .. n + 1$

...

⊢

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

- For new events only

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

- Applying rule **VAR**

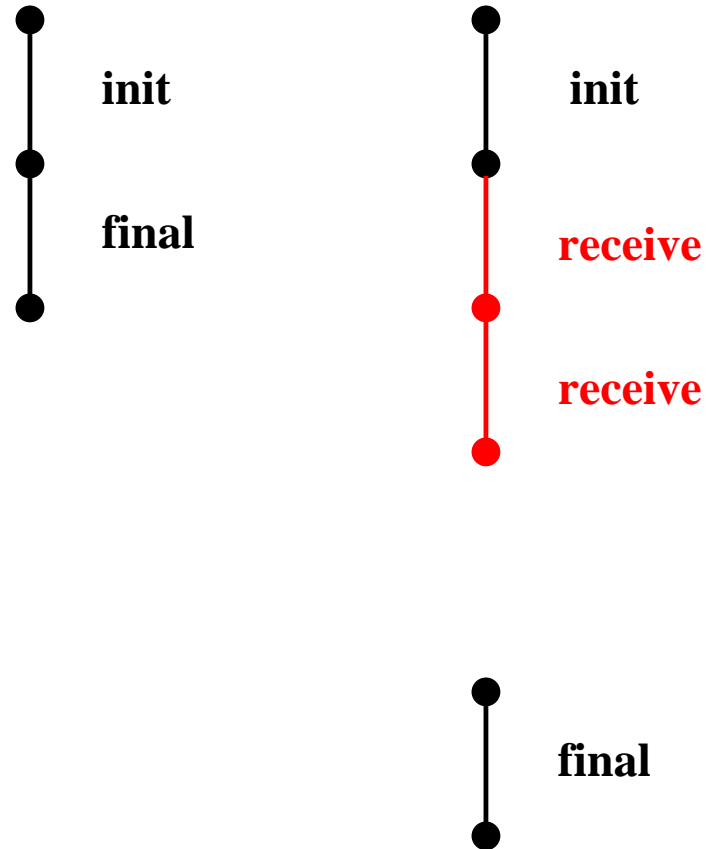
```
receive
  when
     $r \leq n$ 
  then
     $h := h \cup \{r \mapsto f(r)\}$ 
     $r := r + 1$ 
  end
```

...

variant is decreased

...

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



- Global proof rule

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

## - Abstract Events

```
final
  when
     $b = \text{FALSE}$ 
  then
     $g := f$ 
     $b := \text{TRUE}$ 
  end
```

## - Concrete Events

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

```
final
  when
     $b = \text{FALSE}$ 
     $r = n + 1$ 
  then
     $b := \text{TRUE}$ 
  end
```



- Applying rule **DLF**

⋮  
**inv1\_1**  
disj. of abs. guards  
⊢  
disj. of conc. guards

⋮  
 $r \in 1 .. n + 1$   
 $b = \text{FALSE}$   
⊢  
 $r \leq n \vee (b = \text{FALSE} \wedge r = n + 1)$

**variables:**  $b$   
 $h$   
 $r$

**inv1\_1:**  $r \in 1..n + 1$

**inv1\_2:**  $h = (1..r - 1) \triangleleft f$

**inv1\_3:**  $b = \text{TRUE} \Rightarrow r = n + 1$

**variant1:**  $n + 1 - r$

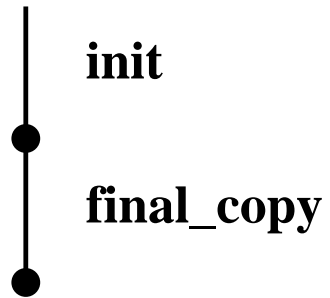
init  
 $b := \text{FALSE}$   
 $h := \emptyset$   
 $r := 1$

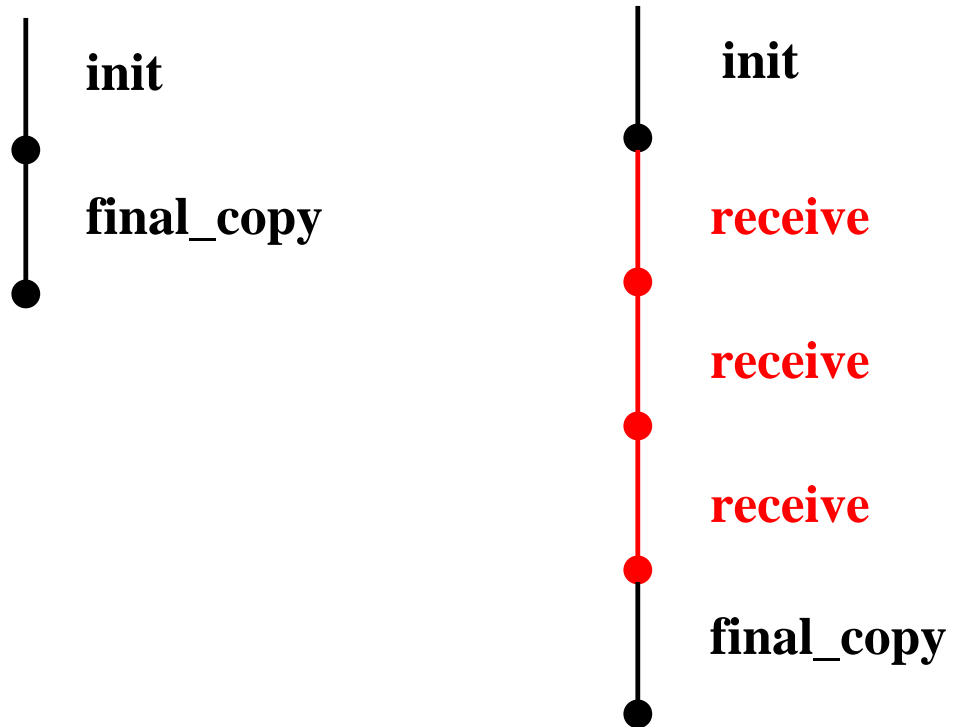
receive  
**when**  
 $r \leq n$   
**then**  
 $h := h \cup \{r \mapsto f(r)\}$   
 $r := r + 1$   
**end**

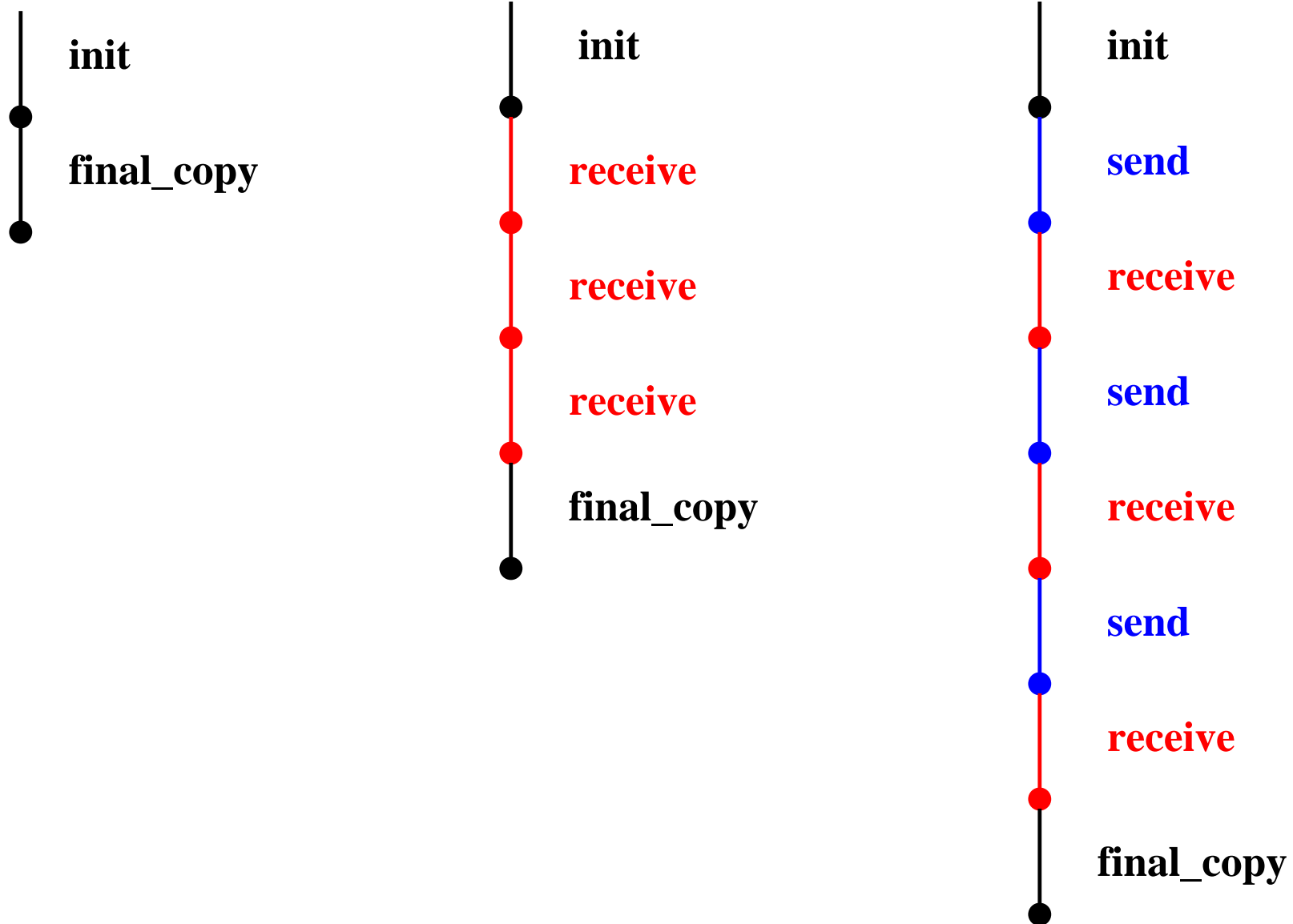
final  
**when**  
 $b = \text{FALSE}$   
 $r = n + 1$   
**then**  
 $b := \text{TRUE}$   
**end**

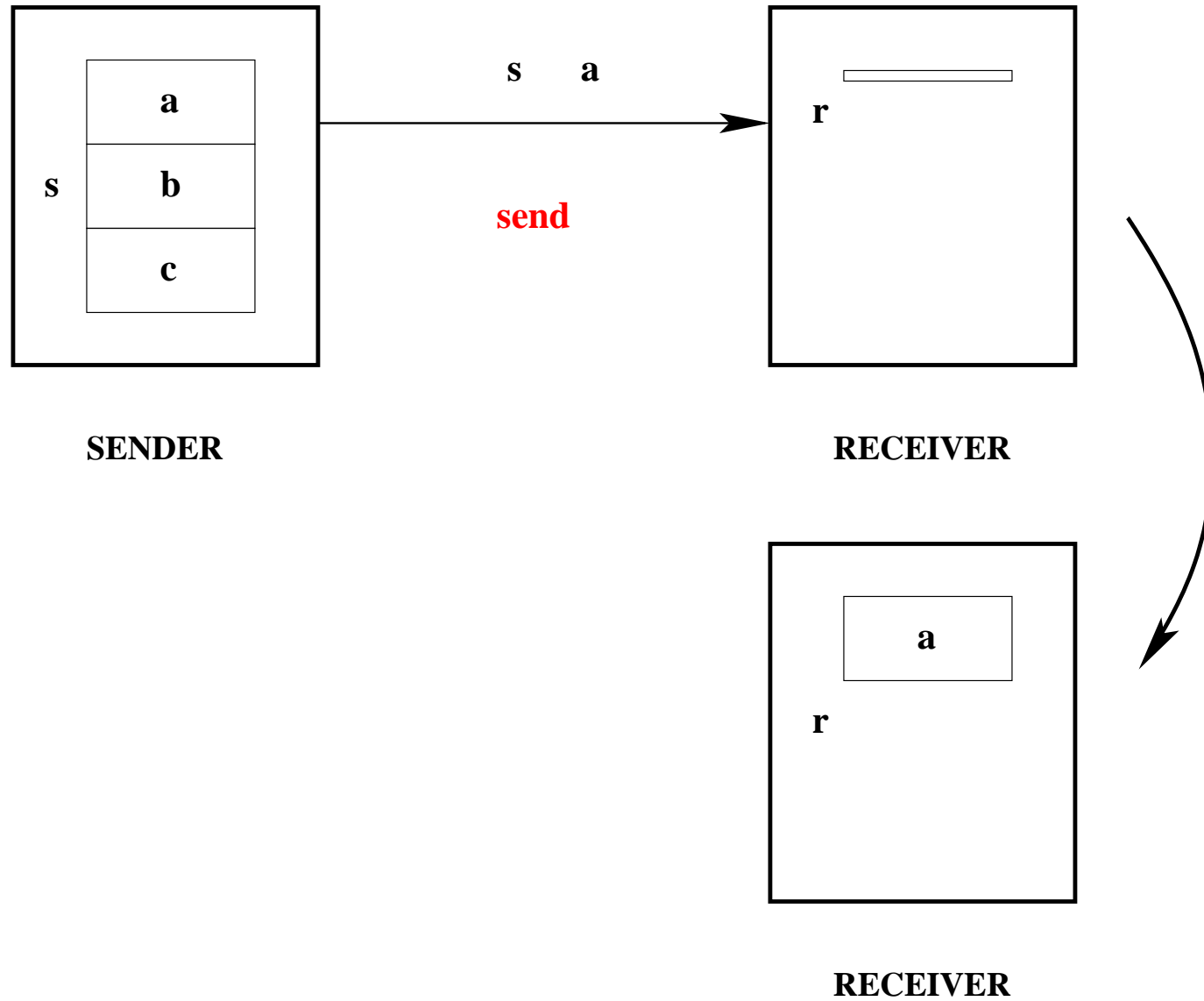
- This model is not satisfactory: **event receive accesses file  $f$**

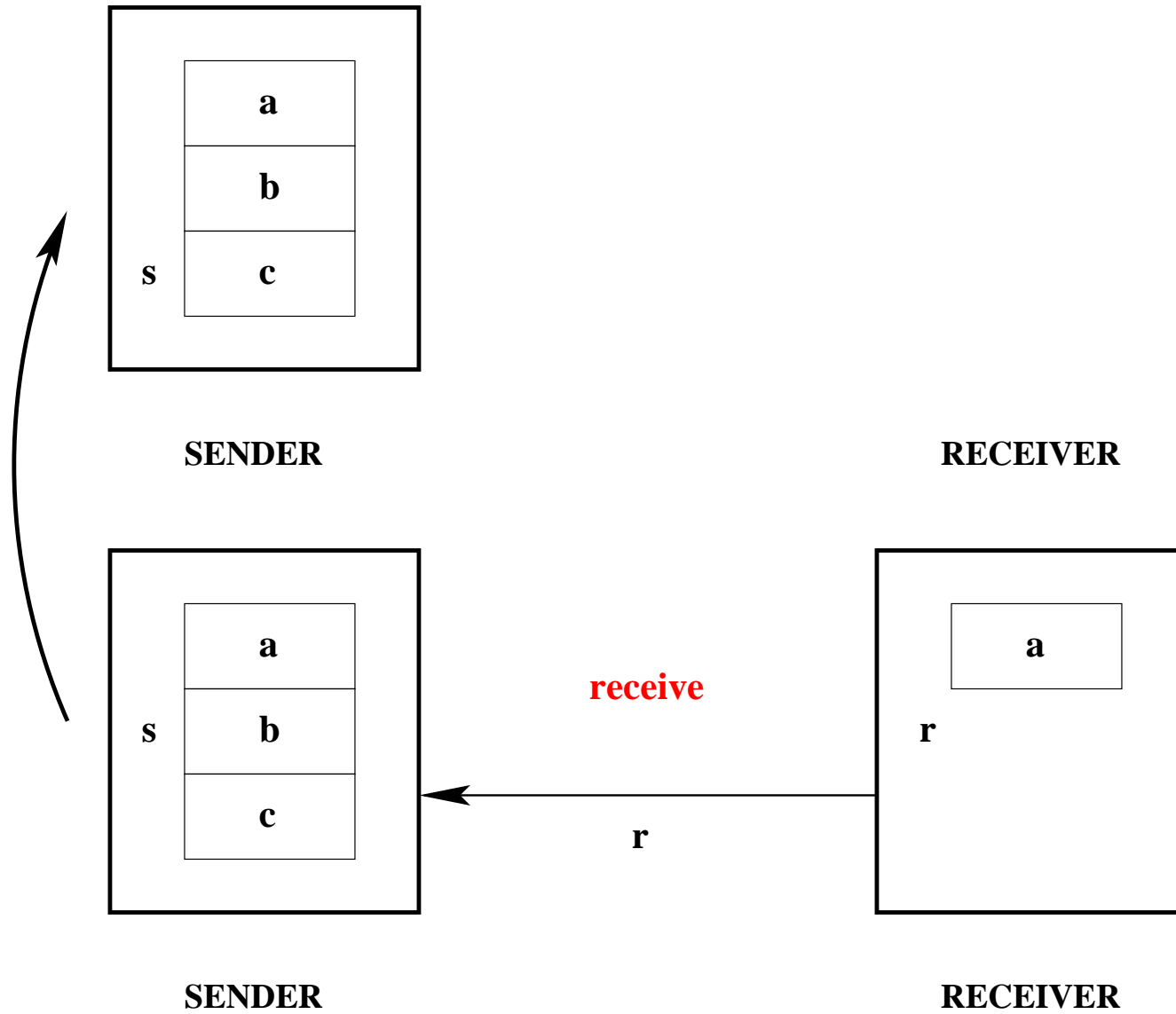
- **Initial model:** The file is transmitted in one shot (FUN1 and FUN2)
- **First refinement:** The file is transmitted gradually (FUN3)
- **Second refinement:** The two agents are separated
- **Third refinement:** Towards an implementation



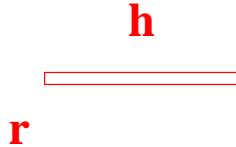
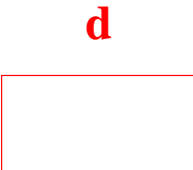
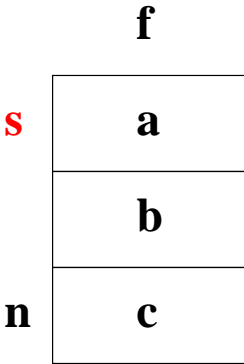


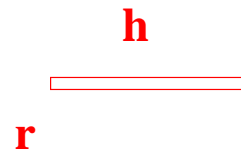
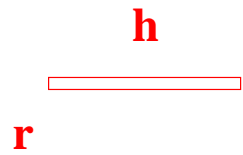
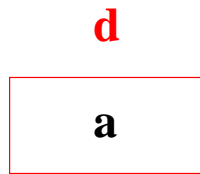
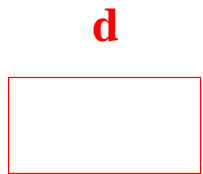
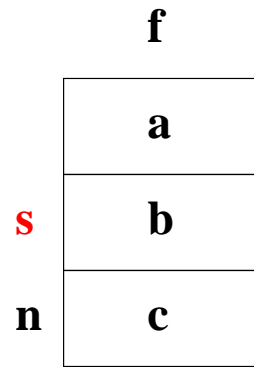
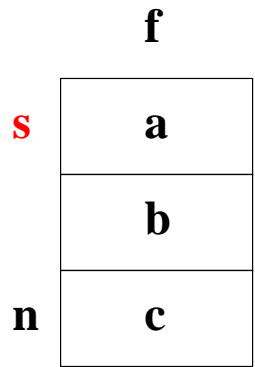


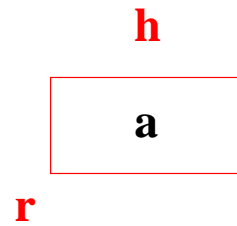
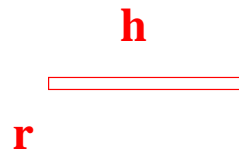
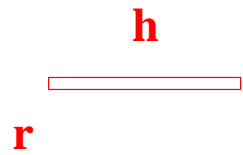
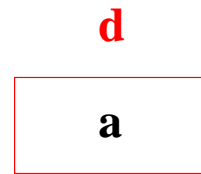
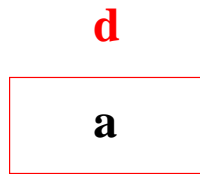
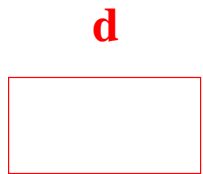
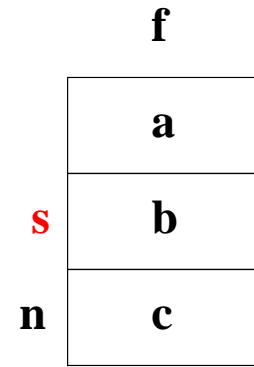
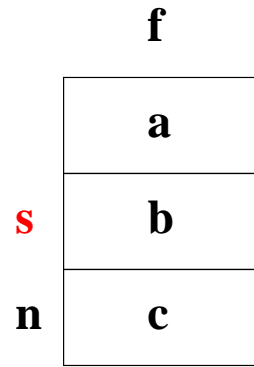
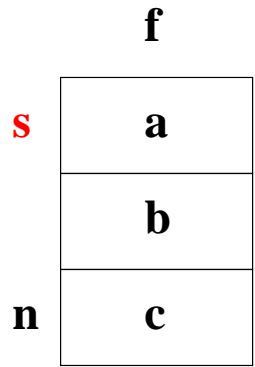


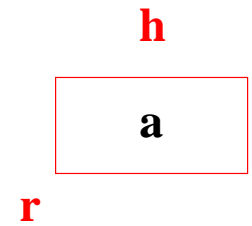
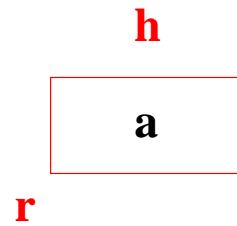
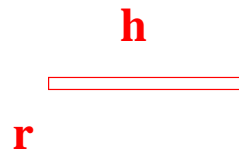
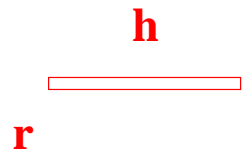
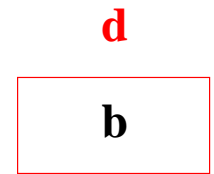
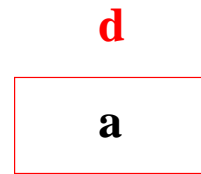
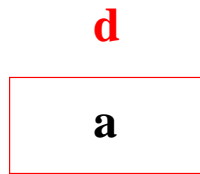
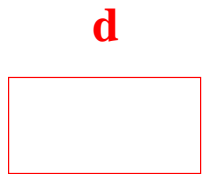
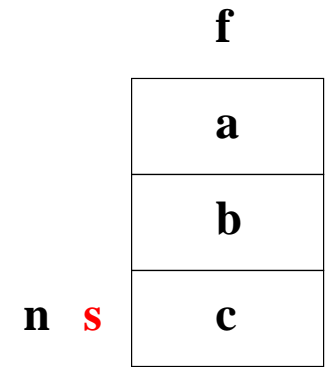
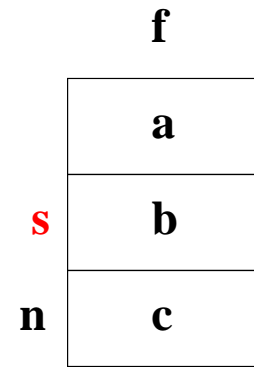
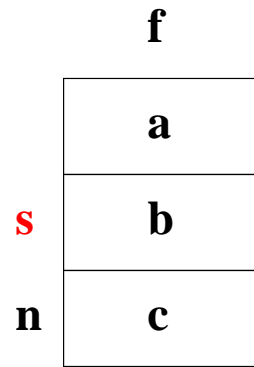
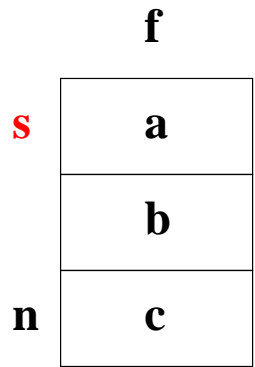


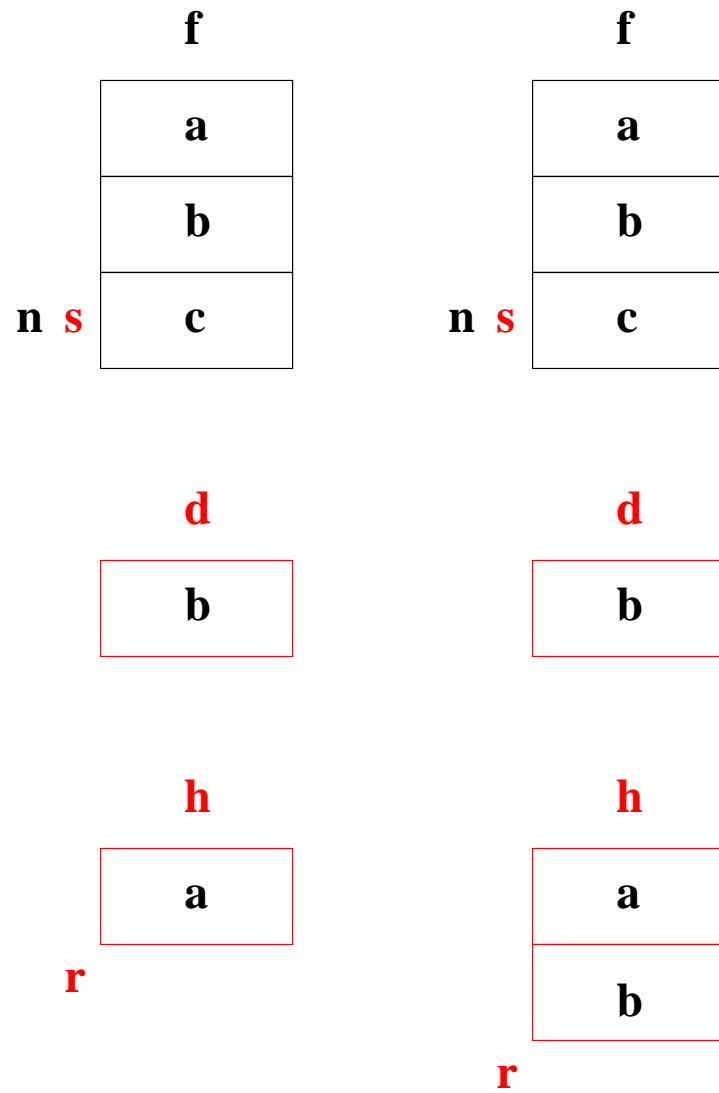


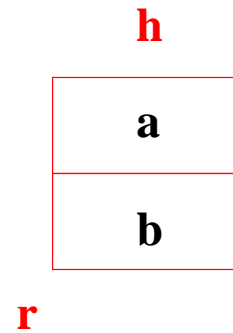
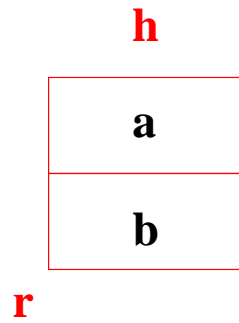
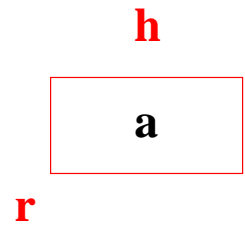
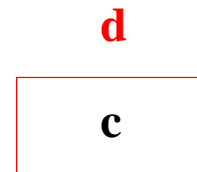
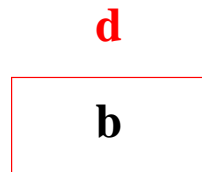
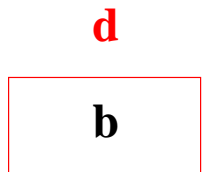
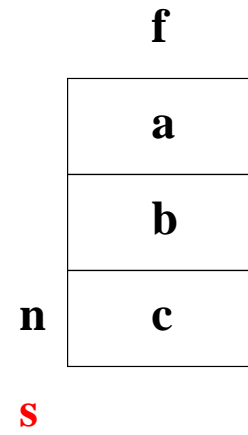
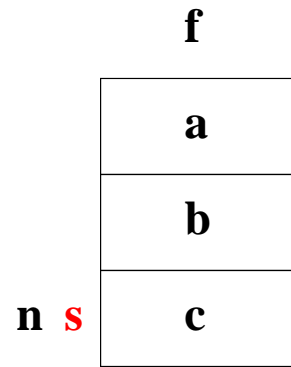
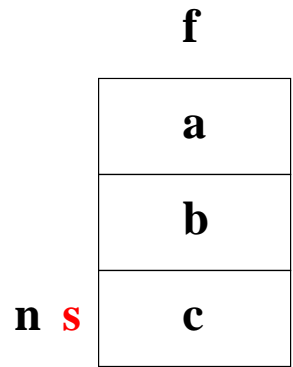


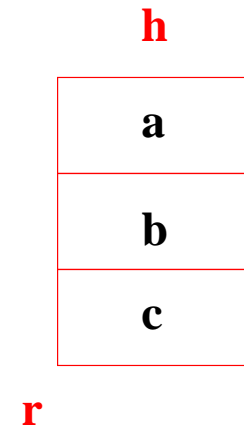
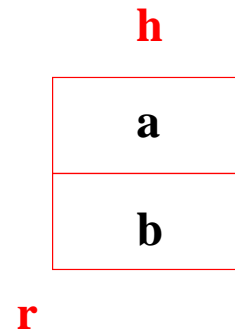
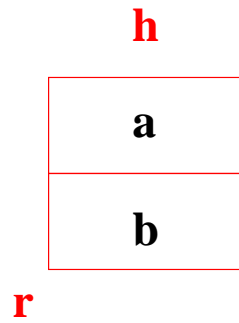
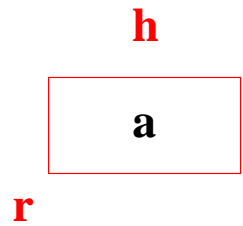
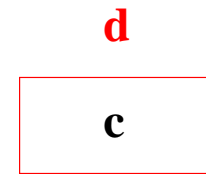
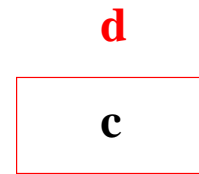
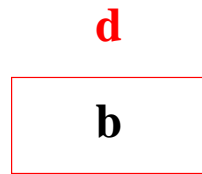
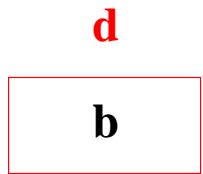
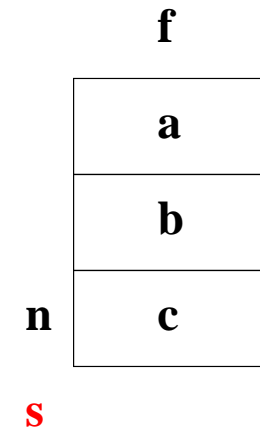
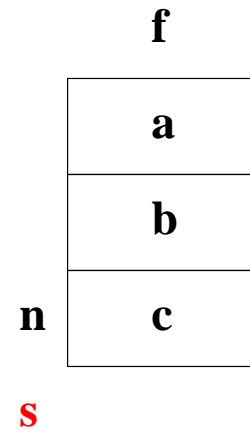
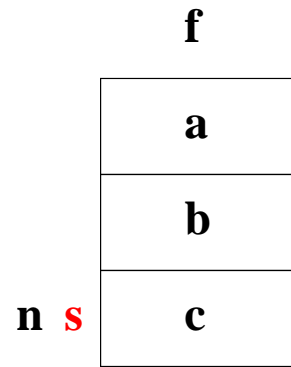
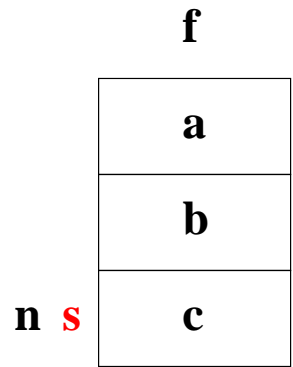


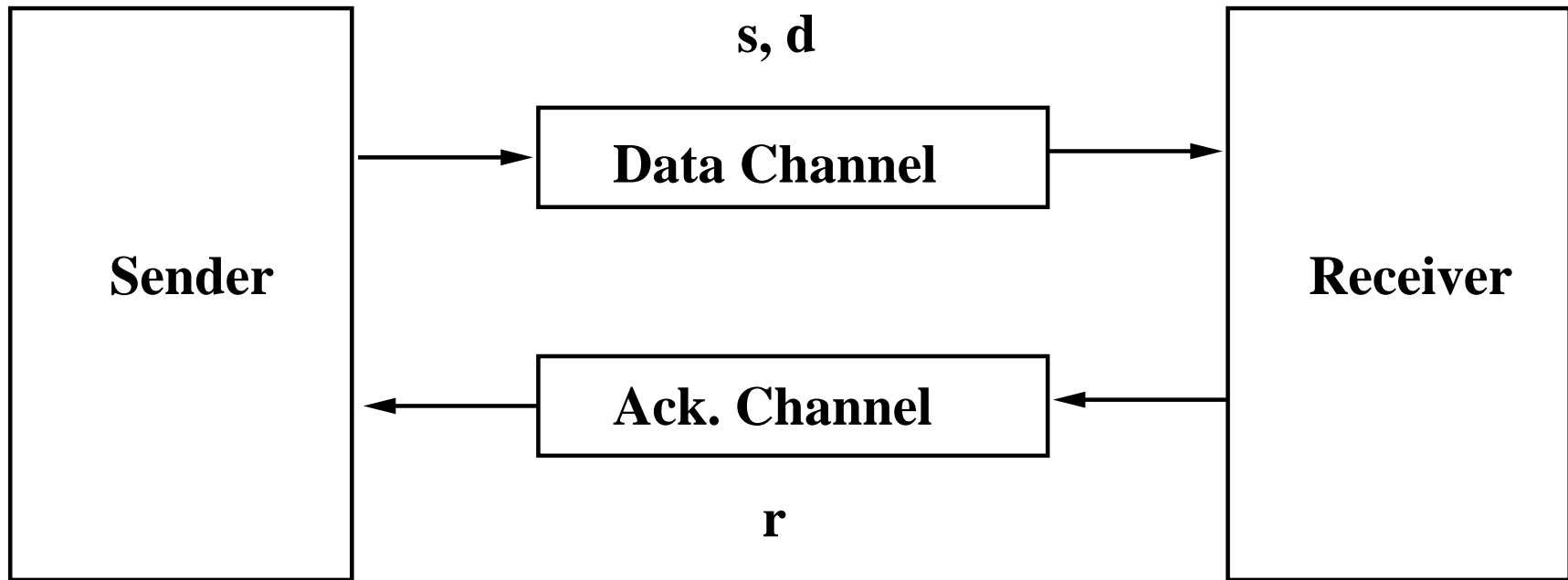














- We introduce an additional variable  $s$ , and a data item  $d$

**variables:**  $b$   
 $h$   
 $r$   
 $s$   
 $d$

**inv2\_1:**  $s \leq n + 1$

**inv2\_2:**  $s \in r .. r + 1$

**inv2\_3:**  $s = r + 1 \Rightarrow d = f(r)$

```
init
   $b := \text{FALSE}$ 
   $h := \emptyset$ 
   $r := 1$ 
   $s := 1$ 
   $d \in D$ 
```

```
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
```

```
final
  when
     $b = \text{FALSE}$ 
     $r = n + 1$ 
  then
     $b := \text{TRUE}$ 
  end
```

- **Event init** refines its abstraction
- **Event final** refines its abstraction
- **Event receive** refines its abstraction
- **Event send** refines skip
- **Event send** does not diverge
- **Relative deadlock freeness**

- **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

```
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
```

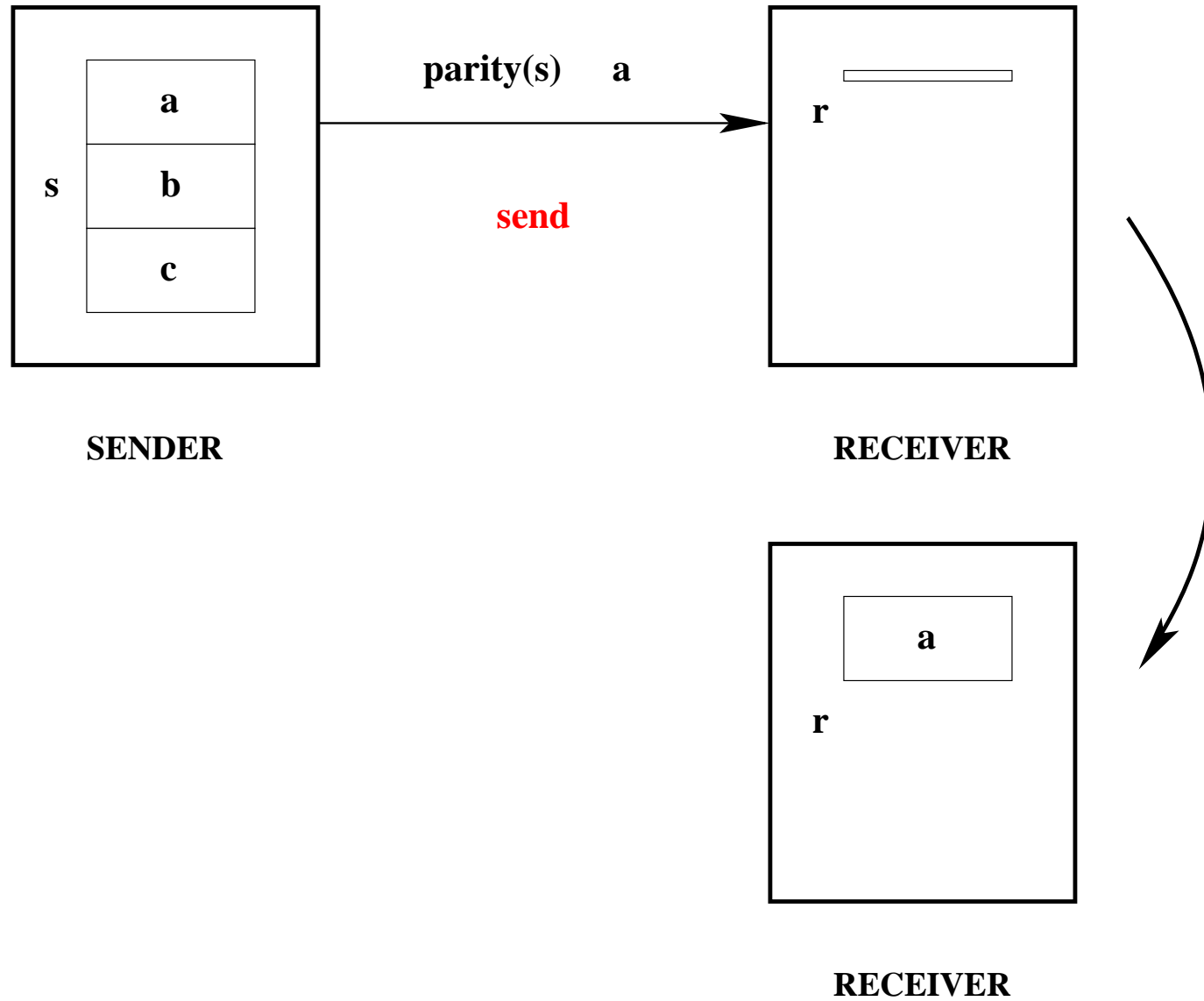
```
inv2_2:  $s \in r .. r + 1$ 
```

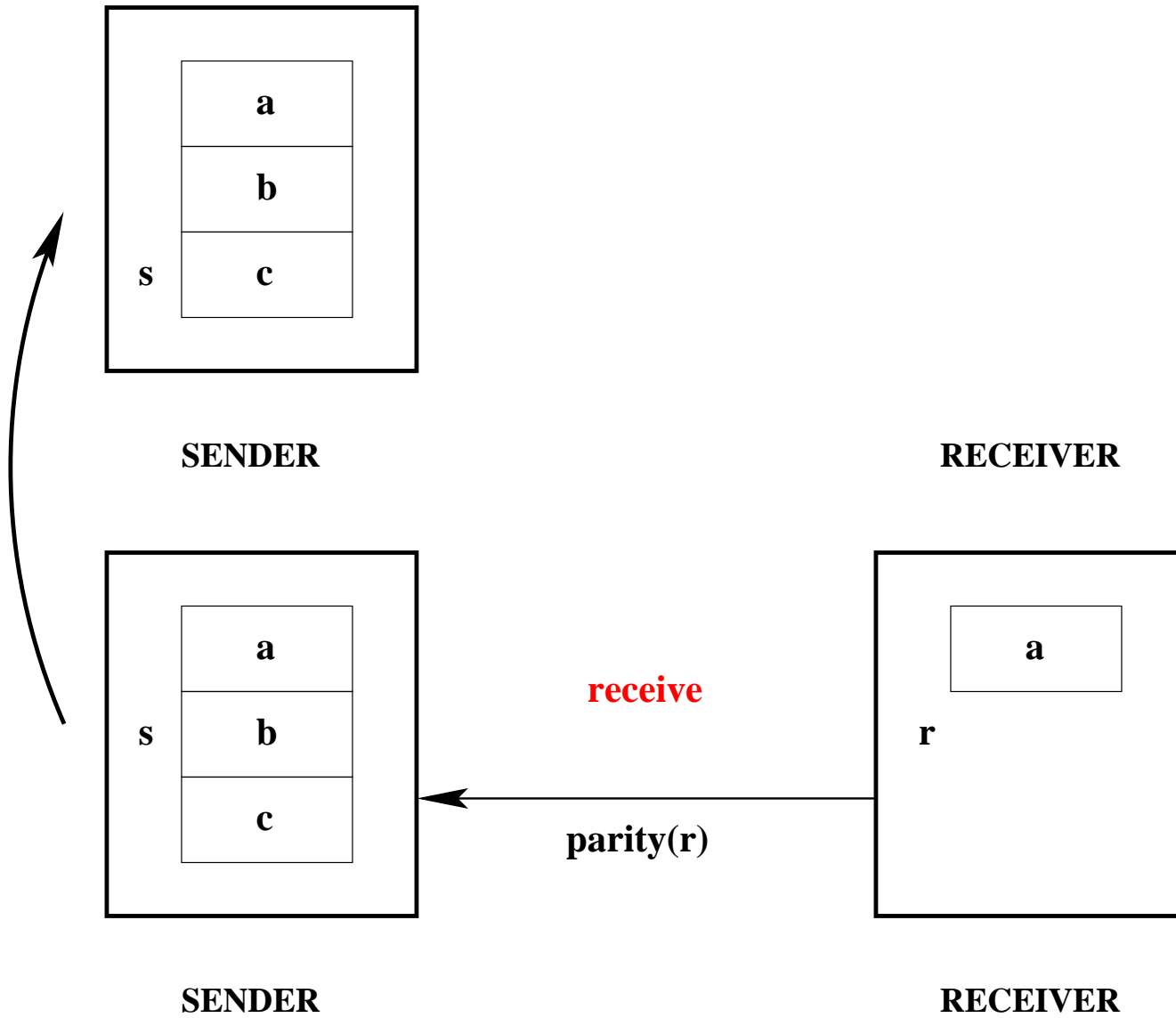
```
send
  when
     $s = r$ 
     $s \neq n + 1$ 
  then
     $d := f(s)$ 
     $s := s + 1$ 
  end
```

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

**inv2\_2:**  $s \in r .. r + 1$

- In order to compare  $r$  and  $s$ , it is sufficient to compare their parities







$$\mathbf{axm3\_1:} \quad \mathit{parity} \in \mathbb{N} \rightarrow \{0, 1\}$$

$$\mathbf{axm3\_2:} \quad \mathit{parity}(0) = 0$$

$$\mathbf{axm3\_3:} \quad \forall x \cdot x \in \mathbb{N} \Rightarrow \mathit{parity}(x + 1) = 1 - \mathit{parity}(x)$$

$$\begin{aligned} \mathbf{thm3\_1:} \quad & \forall x, y \cdot x \in \mathbb{N} \\ & y \in \mathbb{N} \\ & x \in y .. y + 1 \\ & \mathit{parity}(x) = \mathit{parity}(y) \\ & \Rightarrow \\ & x = y \end{aligned}$$

**variables:** *b*  
*h*  
*r*  
*s*  
*d*  
*p*  
*q*

**inv3\_1:**  $p = \textit{parity}(s)$

**inv3\_2:**  $q = \textit{parity}(r)$

```

init
   $b := \text{FALSE}$ 
   $h := \emptyset$ 
   $r := 1$ 
   $s := 1$ 
   $d \in D$ 
   $p := 1$ 
   $q := 1$ 

```

```

final
  when
     $b = \text{FALSE}$ 
     $r = n + 1$ 
  then
     $b := \text{TRUE}$ 
  end

```

```

send
  when
     $p = q$ 
     $s \neq n + 1$ 
  then
     $d := f(s)$ 
     $s := s + 1$ 
     $p := 1 - p$ 
  end

```

```

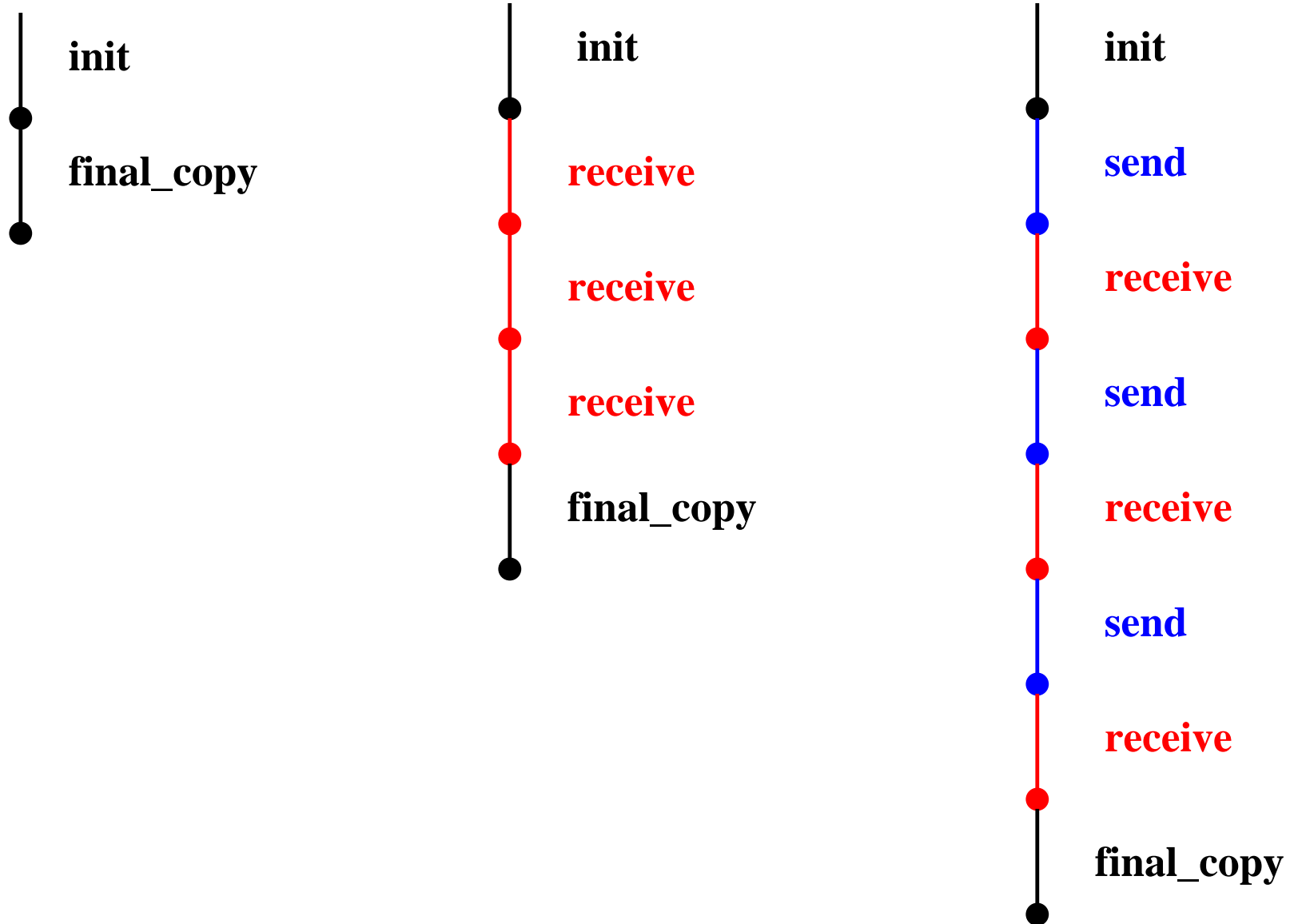
receive
  when
     $p \neq q$ 
  then
     $h := h \cup \{r \mapsto d\}$ 
     $r := r + 1$ 
     $q := 1 - q$ 
  end

```

- The proofs are left as **exercises**

	Total	Interactive
Initial Model	6	0
1st Refinement	13	0
2nd Refinement	15	0
3rd Refinement	8	5
Total	42	5

- 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

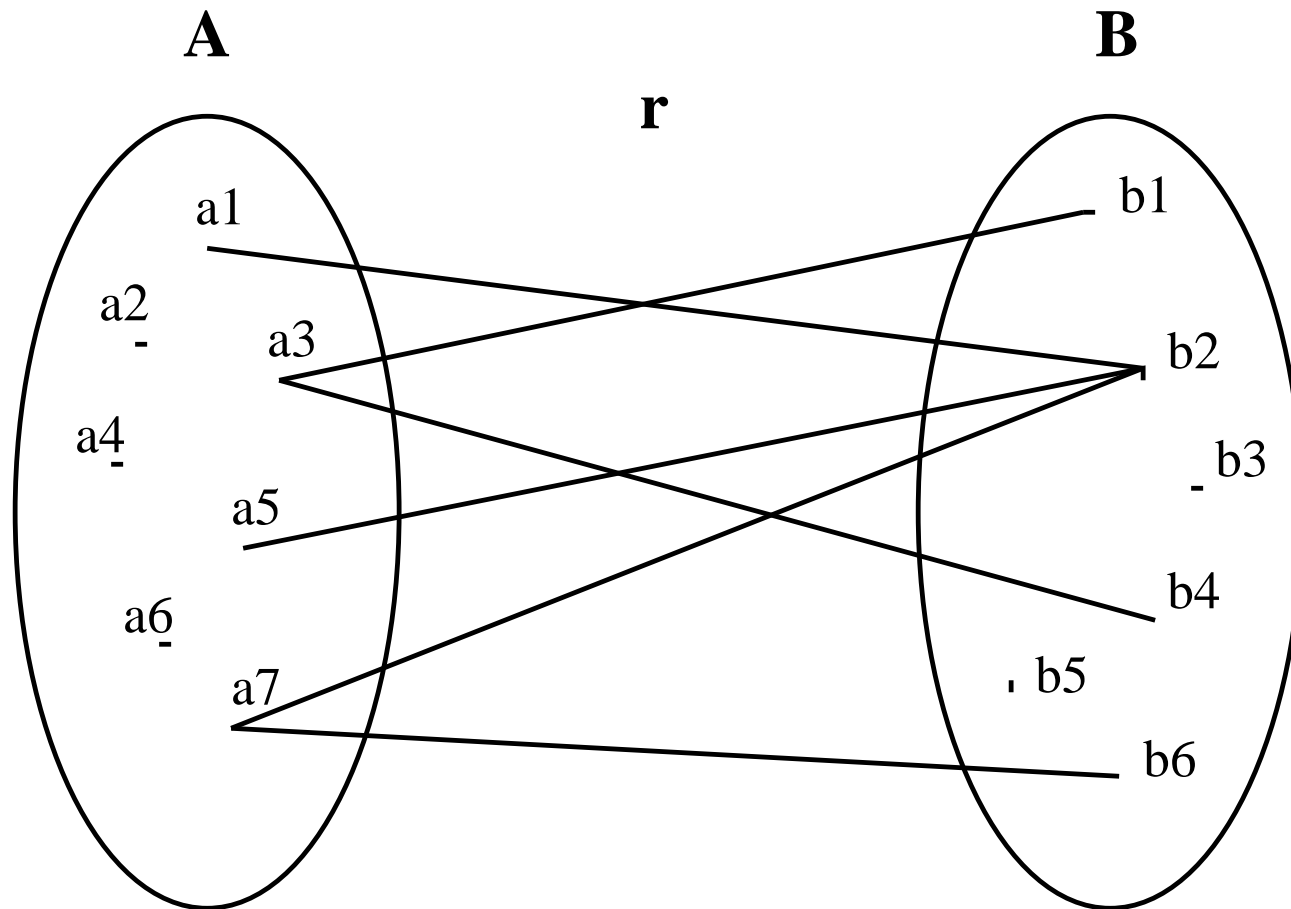


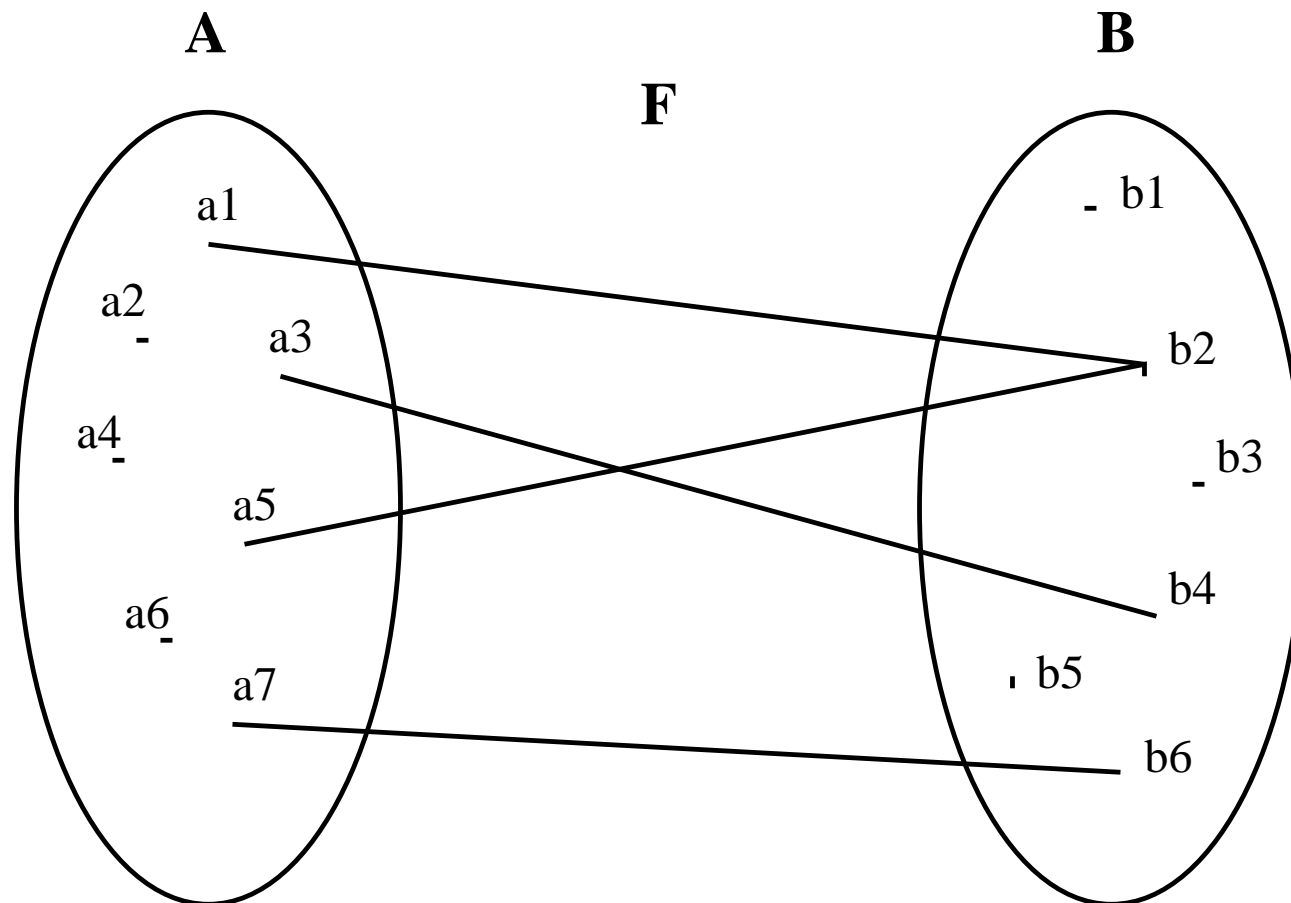
---

$S \leftrightarrow T$	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

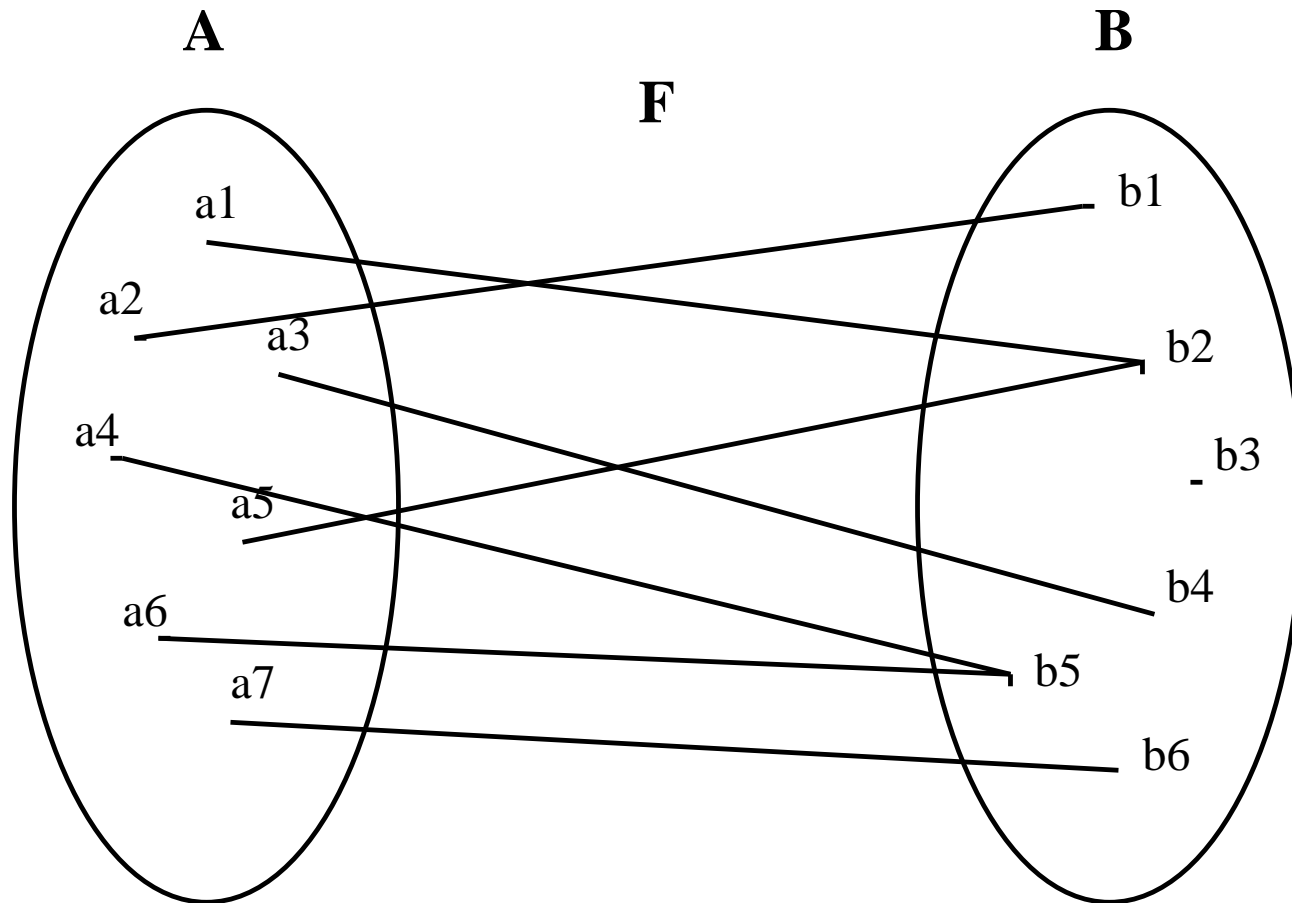




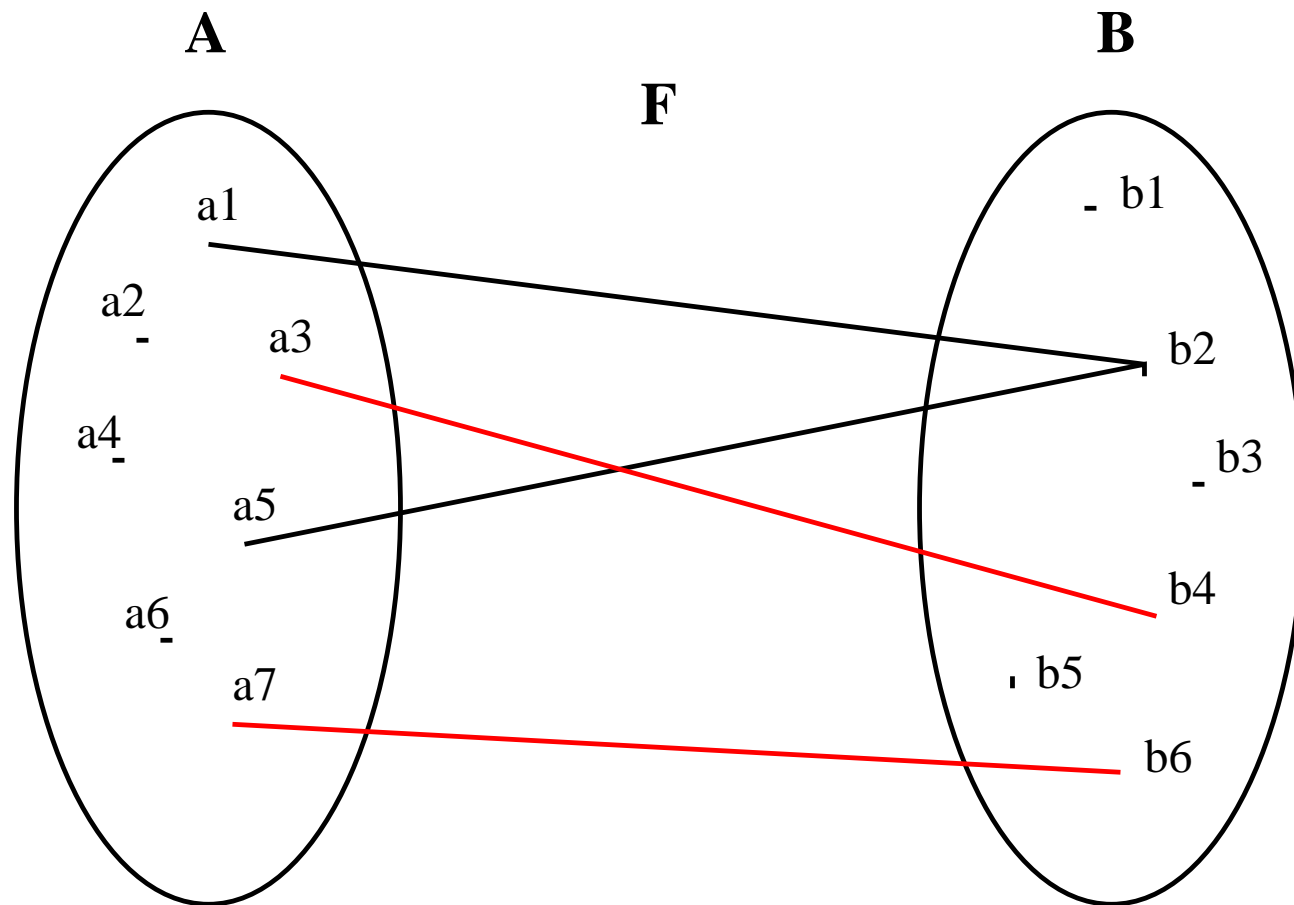
$$F = \{a_1 \mapsto b_2, a_3 \mapsto b_4, a_5 \mapsto b_2, a_7 \mapsto b_6\}$$

$$\text{dom}(F) = \{a_1, a_3, a_5, a_7\}$$

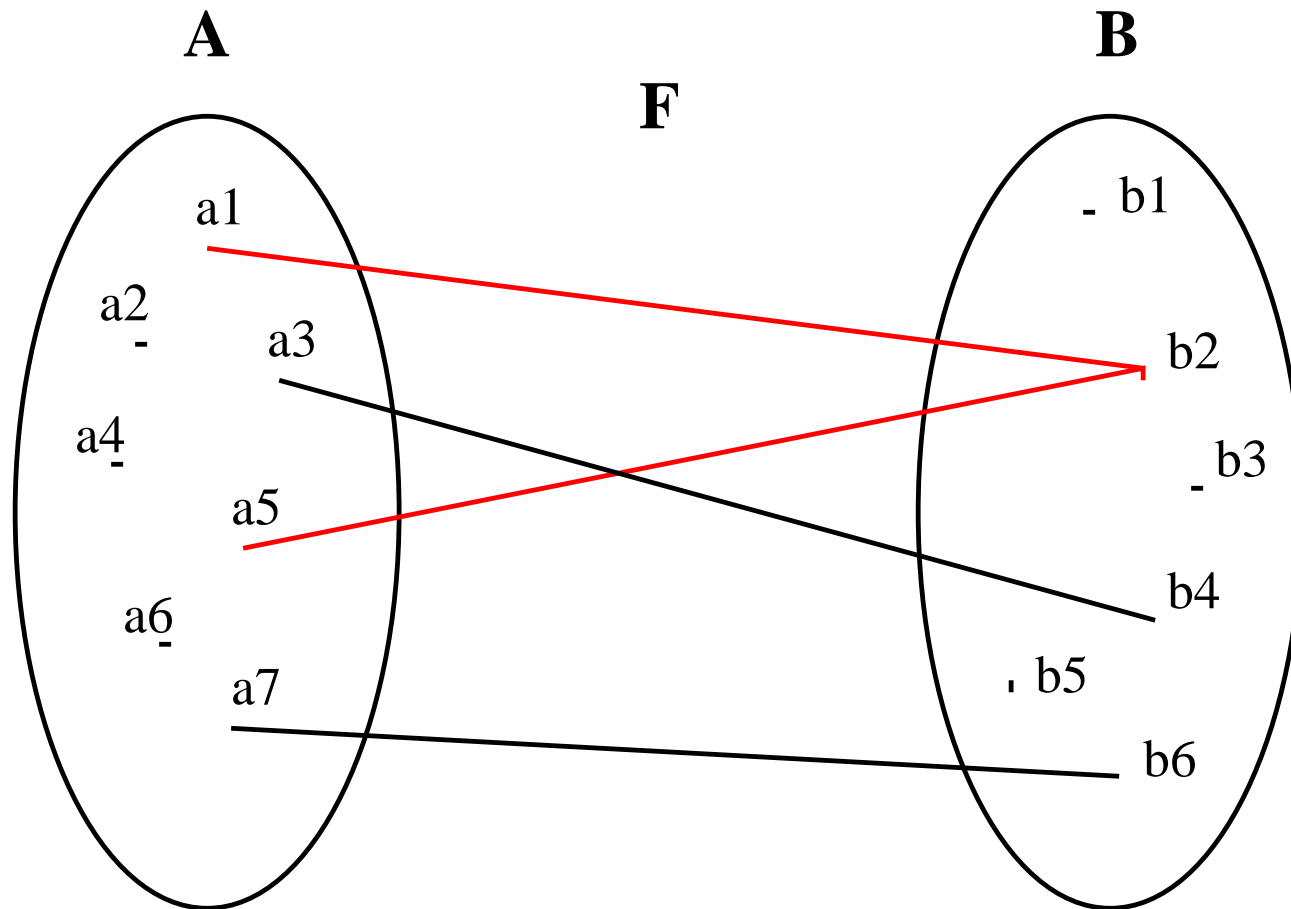
$$\text{ran}(F) = \{b_2, b_4, b_6\}$$



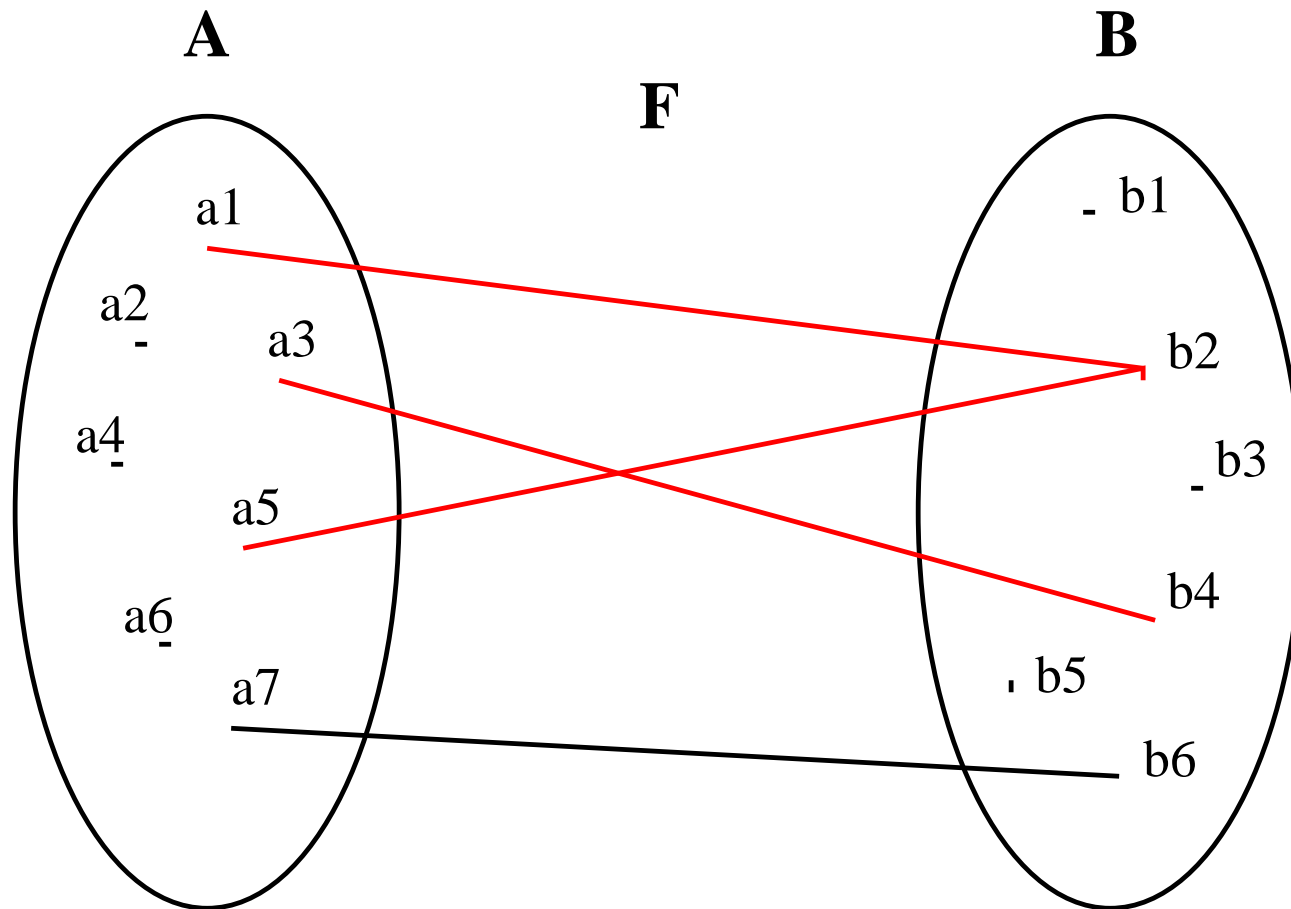
$$\text{dom}(F) = A$$



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

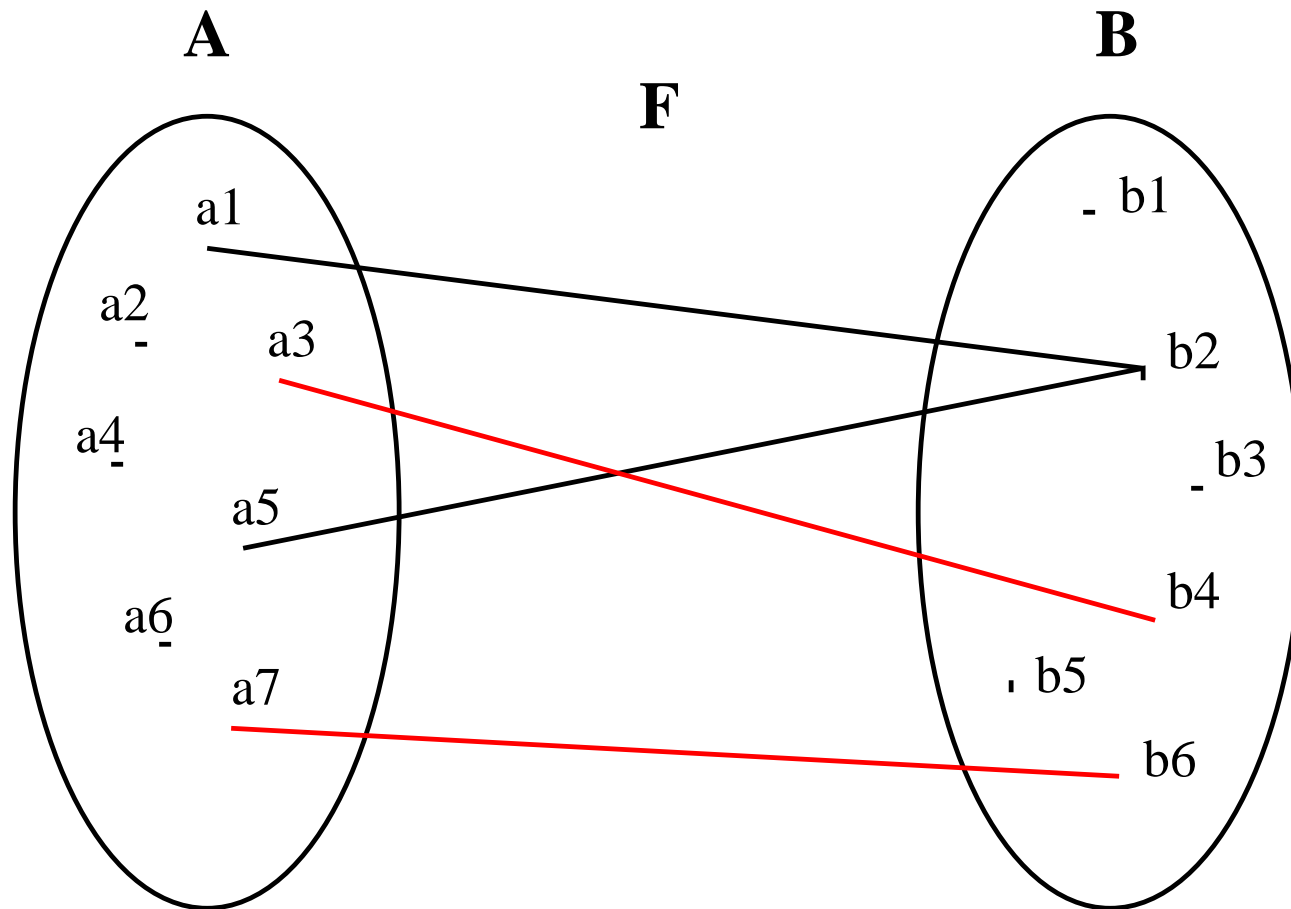


$$\{a_3, a_7\} \triangleleft F$$



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





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

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

sets:  $D$

constants:  $n$   
 $f$

axm0\_1:  $0 < n$

axm0\_2:  $f \in 1 .. n \rightarrow D$

variables:  $g$   
 $b$

inv0\_1:  $g \in 1 .. n \leftrightarrow D$

inv0\_2:  $b = \text{FALSE} \Rightarrow g = \emptyset$

inv0\_3:  $b = \text{TRUE} \Rightarrow g = f$

init

$g := \emptyset$   
 $b := \text{FALSE}$

final

**when**

$b = \text{FALSE}$

**then**

$g := f$

$b := \text{TRUE}$

**end**

variables:  $b$   
 $h$   
 $r$

inv1\_1:  $r \in 1..n + 1$

inv1\_2:  $h = (1..r - 1) \triangleleft f$

inv1\_3:  $b = \text{TRUE} \Rightarrow r = n + 1$

thm1\_1:  $b = \text{TRUE} \Rightarrow h = g$

variant1:  $n + 1 - r$

init  
 $b := \text{FALSE}$   
 $h := \emptyset$   
 $r := 1$

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

final  
**when**  
 $b = \text{FALSE}$   
 $r = n + 1$   
**then**  
 $b := \text{TRUE}$   
**end**

- Variable  $g$  has disappeared: it is not satisfactory

- We want to **keep the variable  $g$**  in the first refinement
- That seems **impossible** since event **receive** has to refine **skip**
- For this, we introduce the notion of **anticipated event**
- Such an event will be **later** proved to be **conveergent**

sets:  $D$

constants:  $n$   
 $f$

axm0\_1:  $0 < n$

axm0\_2:  $f \in 1..n \rightarrow D$

variables:  $g$

inv0\_1:  $g \in 1..n \rightarrow D$

init  
 $g := \emptyset$

receive  
**status**  
anticipated  
**when**  
 $g \neq f$   
**then**  
 $g \in 1..n \rightarrow D$   
**end**

final  
**when**  
 $g = f$   
**then**  
skip  
**end**

- Event receive is highly non-deterministic.
- Notice the event final
- Variable  $b$  is not useful anymore

variables:  $g$   
 $r$

inv1\_1:  $r \in 1..n + 1$

inv1\_2:  $g = (1..r - 1) \triangleleft f$

variant1:  $n + 1 - r$

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

receive  
**status**  
**convergent**  
**when**  
 $r \neq n + 1$   
**then**  
 $g(r) := f(r)$   
 $r := r + 1$   
**end**

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

Note: the event receive works now with variable  $g$