

11. Synchronizing Processes on a Tree Network

Jean-Raymond Abrial

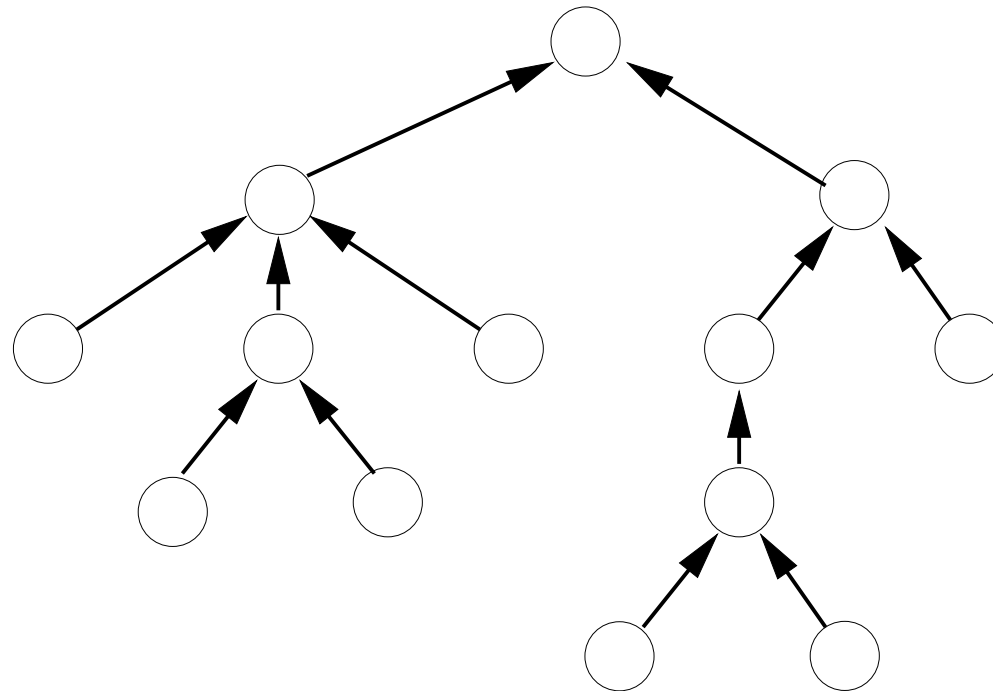
2009

- Learning a few more **modeling conventions**
- Learning more about **abstraction**
- Learning how to **formalize** an interesting structure: a **tree**
- Study a more complicated problem in **distributed computing**
- Example studied in the following book:
[W.H.J. Feijen and A.J.M. van Gasteren.](#)
On a Method of Multi-programming Springer Verlag 1999.

- Define the **informal requirements**
- Define the **refinement strategy**
- Construct the various **more and more concrete models**

We have a fixed set of processes forming a tree

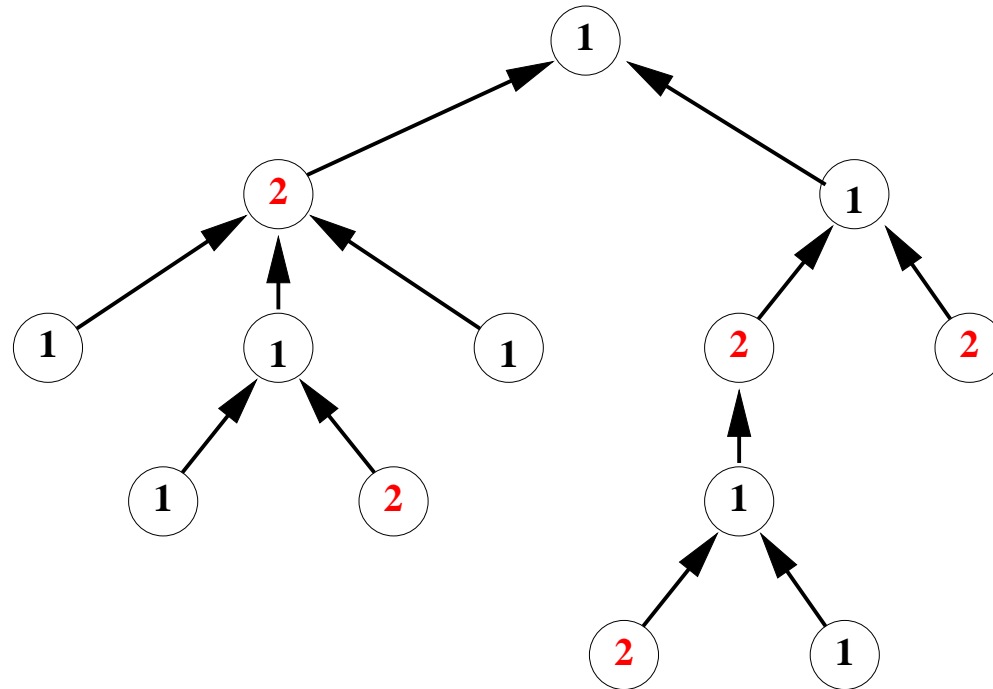
ENV-1



- All processes are supposed to execute **for ever the same code**
- But processes must **remain synchronized**
- For this, we assign a **counter** to each process

Each process has a counter, which is a natural number	ENV-2
---	-------

- The counter of a process represents its “phase”
- The difference between any two counters is not greater than 1
- Each process is thus at most one phase ahead of the others



The difference between any two counters is at most equal to 1

FUN-1

- **Reading** the counters

Each process can read the counters of its immediate neighbors only	FUN-2
--	-------

- **Modifying** the counters

The counter of a process can be modified by this process only	FUN-3
---	-------

- Construct an abstract **initial model dealing with FUN-1 and FUN-3**
- **Improve** the design to (partially) take care of **FUN-2**
- **Improve** the design to better take care of **FUN-2**
- **Simplify** the final design to obtain an **efficient implementation**

The difference between any two counters is at most equal to 1	FUN-1
---	-------

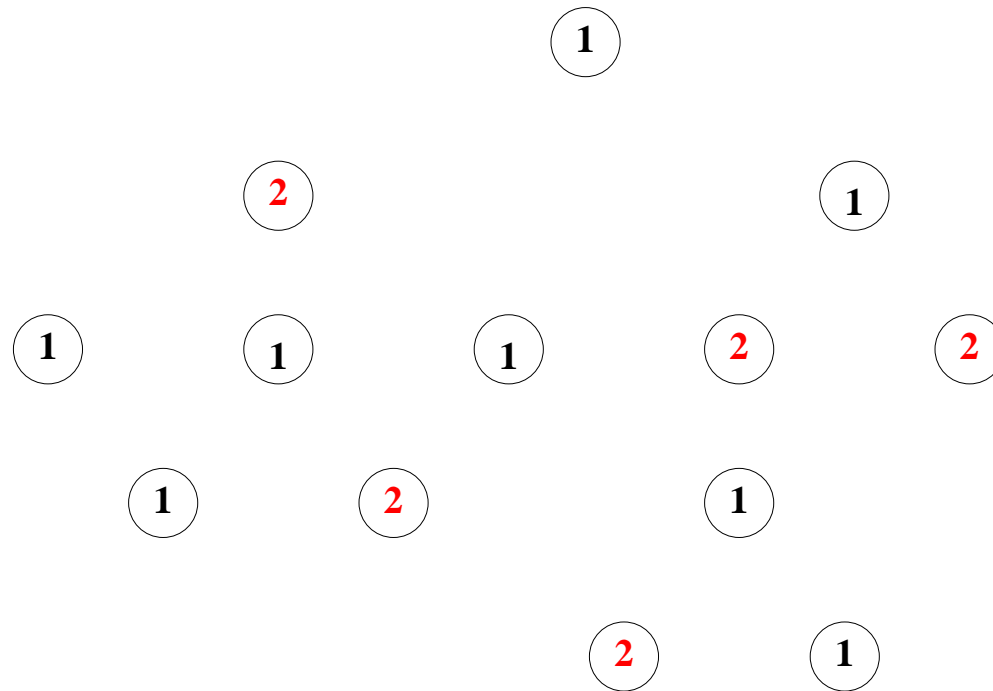
Each process can read the counters of its immediate neighbors only	FUN-2
--	-------

The counter of a process can be modified by this process only	FUN-3
---	-------

- We simplify the situation: **we forget about the tree**
- We just define the counters and **express the main property: FUN-1**

The difference between any two counters is at most equal to 1	FUN-1
---	-------

- The initial model is always **far more abstract** than the final system
- **Other requirements** are probably **not fulfilled**



The difference between any two counters is at most equal to 1

FUN-1

carrier set: P

axm0_1: $\text{finite}(P)$

variable: c

inv0_1: $c \in P \rightarrow \mathbb{N}$

inv0_2: $\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ c(x) \leq c(y) + 1 \end{array} \right)$

```
init
  c := P × {0}
```

```
ascending
  any n where
    n ∈ P
    ∀m · m ∈ P ⇒ c(n) ≤ c(m)
  then
    c(n) := c(n) + 1
  end
```

- A process counter is incremented **only when \leq to all other counters**
- Notice the **non-determinacy**

$$c \in P \rightarrow \mathbb{N}$$

inv0_1

$$\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ c(x) \leq c(y) + 1 \end{array} \right)$$

inv0_2

$$n \in P$$

Guards of event
ascending

$$\forall m \cdot (m \in P \Rightarrow c(n) \leq c(m))$$

⊢

$$\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ (c \triangleleft \{n \mapsto c(n) + 1\})(x) \leq (c \triangleleft \{n \mapsto c(n) + 1\})(y) + 1 \end{array} \right)$$

↑

Modified invariant **inv0_2**

$$c \in P \rightarrow \mathbb{N}$$

$$\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ c(x) \leq c(y) + 1 \end{array} \right)$$

$$n \in P$$

$$\forall m \cdot (m \in P \Rightarrow c(n) \leq c(m))$$

$$x \in P$$

$$y \in P$$

⊢

$$(c \triangleleft \{n \mapsto c(n) + 1\})(x) \leq (c \triangleleft \{n \mapsto c(n) + 1\})(y) + 1$$

- We perform then an easy proof by cases: $\left\{ \begin{array}{l} x = n, y = n \\ x \neq n, y = n \\ x = n, y \neq n \\ x \neq n, y \neq n \end{array} \right.$

- Initialisation and **invariant establishment**
- **Liveness**: a forgotten requirement

Once started, the system must work for ever

FUN-4

ascending

any n **where**

$n \in P$

$\forall m \cdot m \in P \Rightarrow c(n) \leq c(m)$

then

$c(n) := c(n) + 1$

end

- Requirement **FUN-2** is not fulfilled:

Each node can read the counters of its immediate neighbors only	FUN-2
---	-------

- We introduce a **special process** r
- We suppose that the **counter of r is always minimal**

$$\forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$$

- This is a **new invariant** (for the moment)

- We simplify the guard

(abstract-)ascending

any n **where**

$n \in P$

$\forall m \cdot m \in P \Rightarrow c(n) \leq c(m)$

then

$c(n) := c(n) + 1$

end

(concrete-)ascending

any n **where**

$n \in P$

$c(n) = c(r)$

then

$c(n) := c(n) + 1$

end

- We have then to prove **guard strengthening**

$$c \in P \rightarrow \mathbb{N}$$

$$\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ c(x) \leq c(y) + 1 \end{array} \right)$$

$$\forall m \cdot (m \in P \Rightarrow \boxed{c(r)} \leq c(m))$$

$$n \in P$$

$$\boxed{c(n) = c(r)}$$

⊢

$$n \in P$$

$$\forall m \cdot (m \in P \Rightarrow \boxed{c(n)} \leq c(m))$$

inv0_1

inv0_2

new invariant

Guards of **concrete**
event ascending

Guards of **abstract**
event ascending

```
ascending
  any  $n$  where
     $n \in P$ 
     $c(n) = c(r)$ 
  then
     $c(n) := c(n) + 1$ 
  end
```

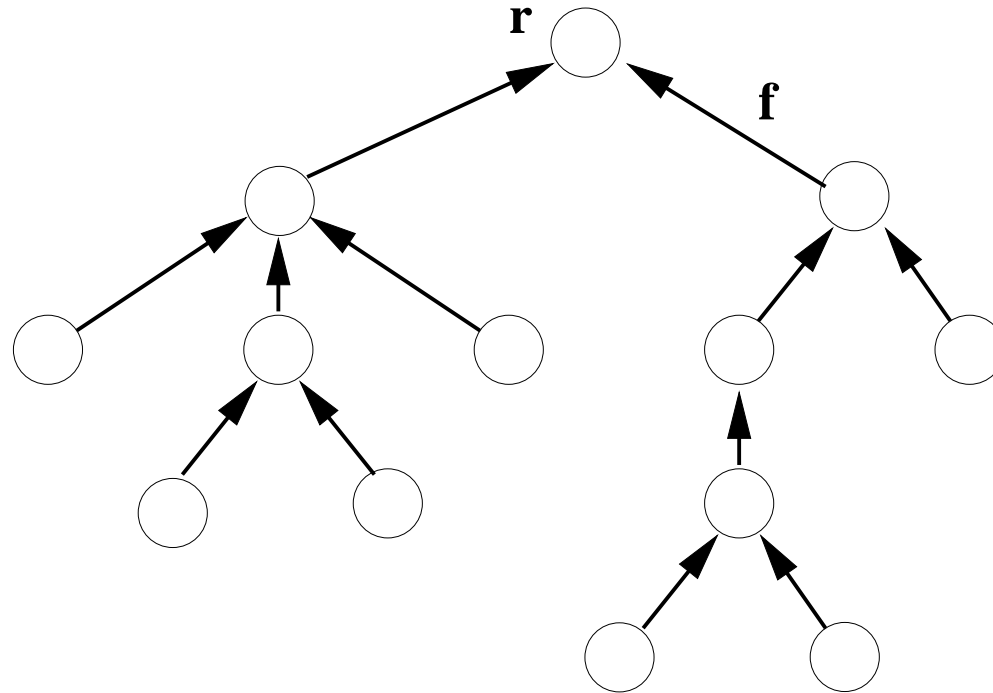
$$\forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$$

1. We have to prove that the **new invariant is preserved** by the event
2. The guard of the event **still does not fulfill requirement FUN-2**

Each node **can read** the counters of its
immediate neighbors only

FUN-2

- **Problem 1 solved in this refinement**, problem 2 solved later



- A tree has got a **root r** and a **parent function f**
- This is **not sufficient to defined a tree** (but enough for the moment)

- We define **the root r** of the tree
- And the **parent function f** (defined everywhere except at the root)

carrier set: P

constants: r, f

axm1_1: $r \in P$

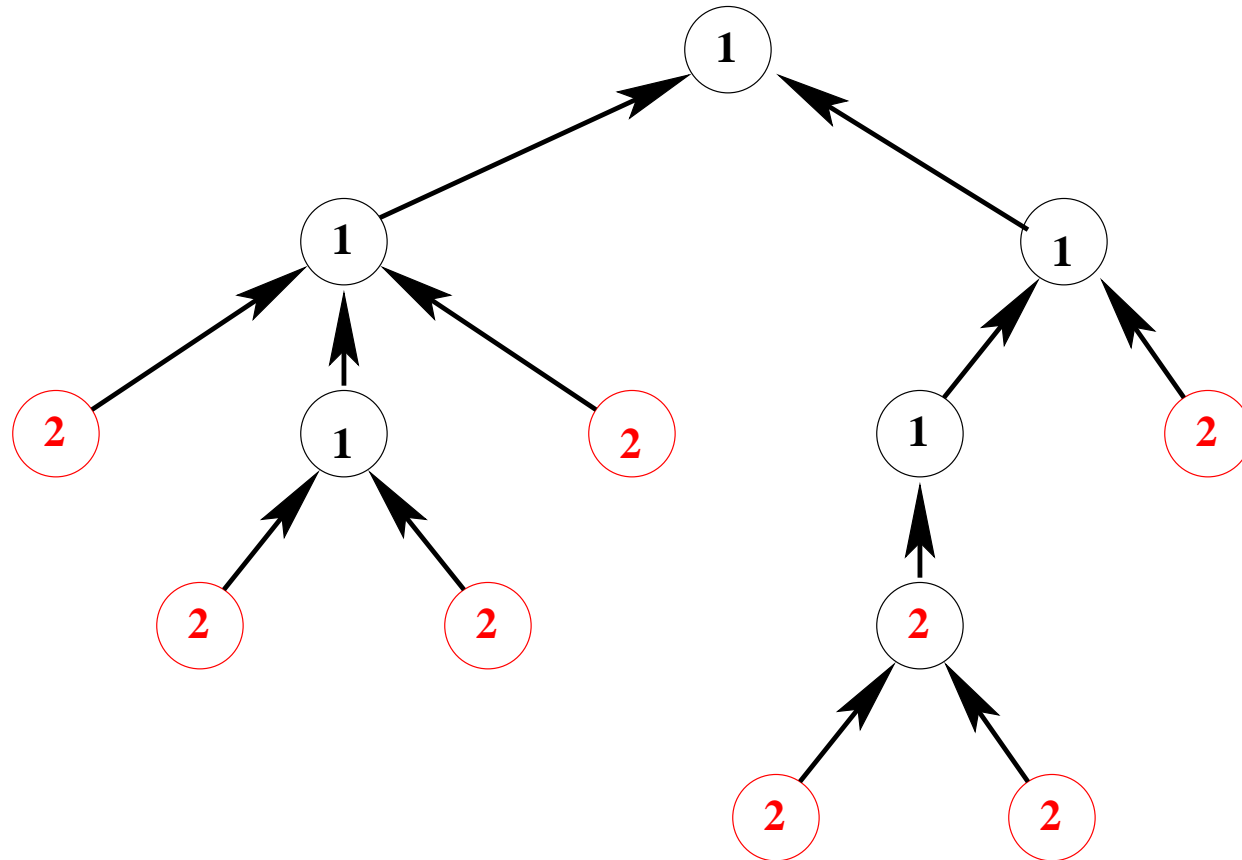
axm1_2: $f \in P \setminus \{r\} \rightarrow P$

- We define a **weaker invariant**
- The counter of the parent of each node m is \leq than that of m

$$\mathbf{inv1_1:} \quad \forall m \cdot m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m)$$

- The **minimality of the counter** at the root **can be proved**:

$$\mathbf{thm1_1:} \quad \forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$$



inv1_1 : $\forall m \cdot m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m)$

thm1_1 : $\forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$

- Adding a guard

ascending

any n **where**

$n \in P$

$c(r) = c(n)$

$\forall m \cdot m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)$

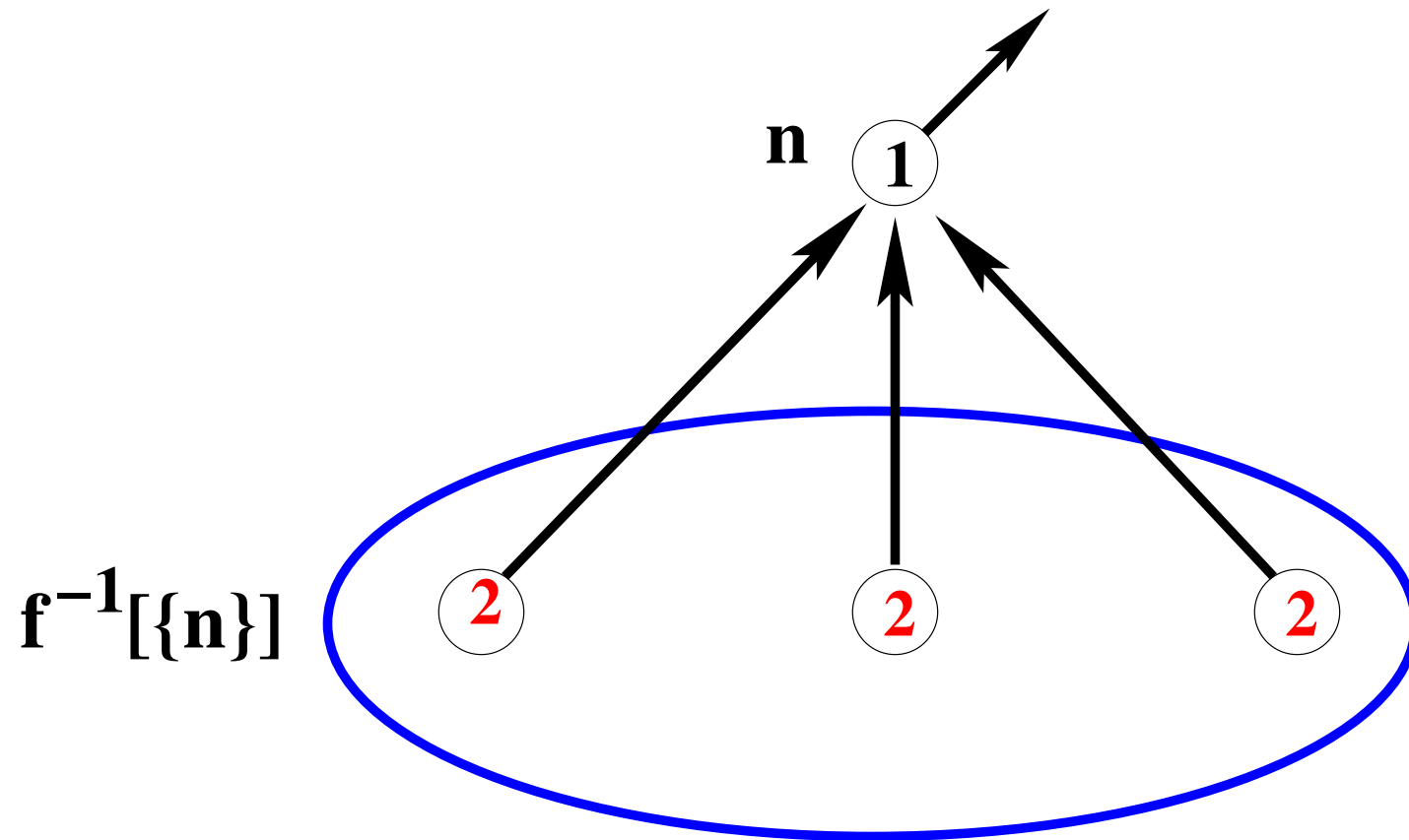
then

$c(n) := c(n) + 1$

end

- This will allow us to prove **inv1_1** easily (again, a **proof by cases**)

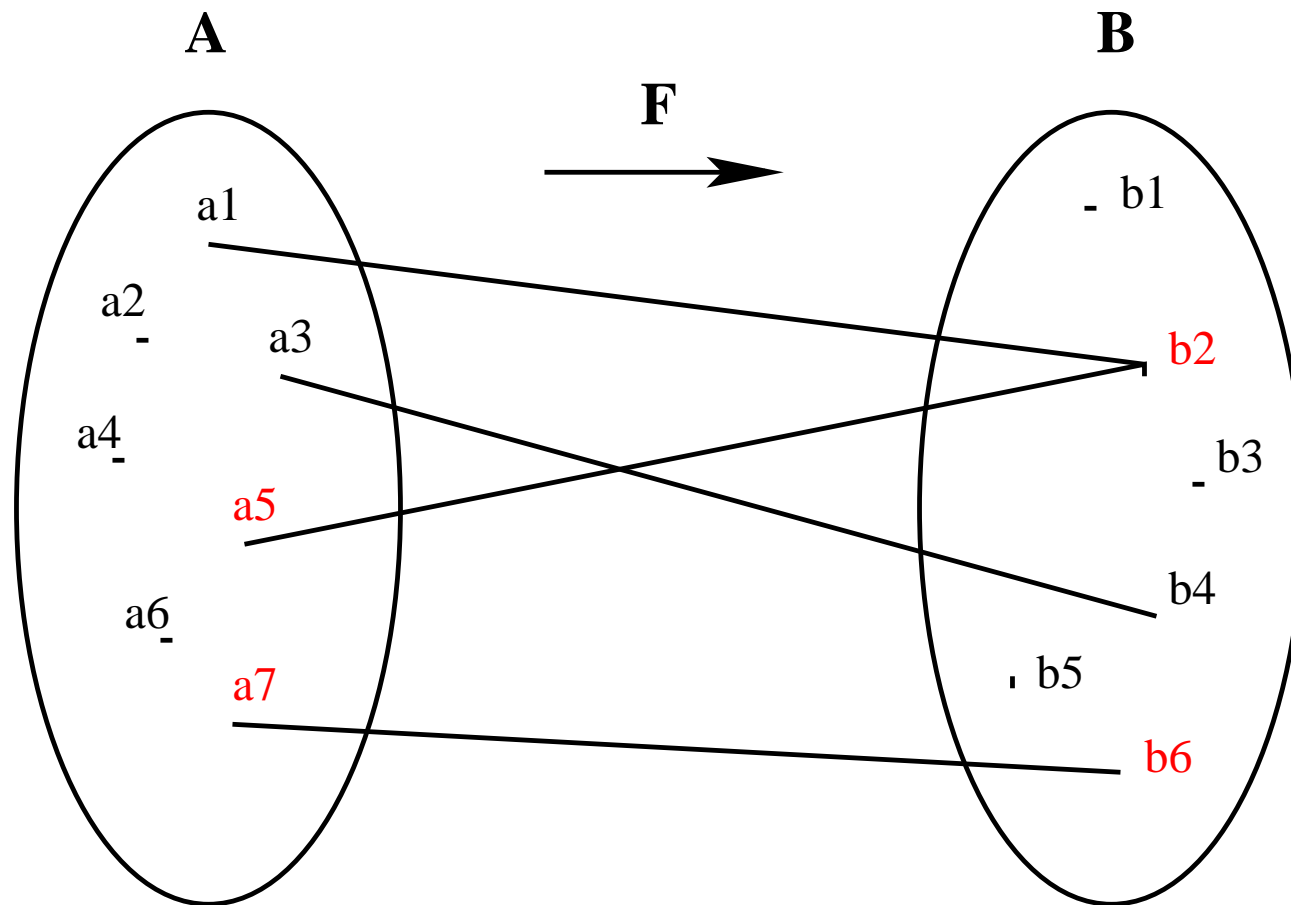
inv1_1 : $\forall m \cdot m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m)$



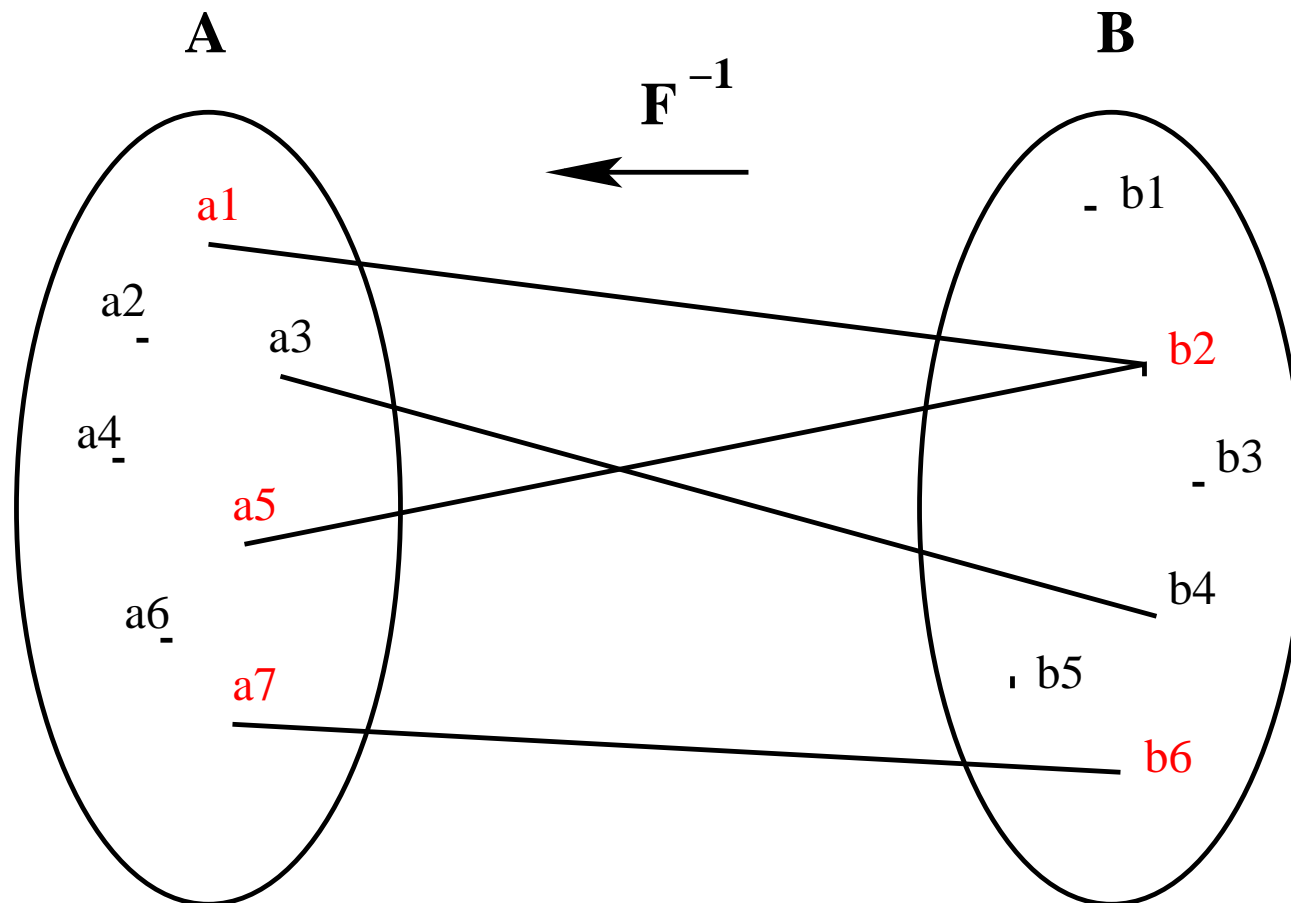
$$\forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(n) < c(m))$$

f^{-1}	converse of function f
$f[S]$	image of set S under f

- Can be generalized to **binary relations**



$$F[\{a_5, a_7\}] = \{b_2, b_6\}$$



$$F^{-1}[\{b_2, b_6\}] = \{a_1, a_5, a_7\}$$

- Suppose we have a function f and the following sets:

$$f \in S \rightarrow T \quad A \subseteq S \quad C \subseteq T$$

- Then we have the following equivalences:

$$F \in f[A] \quad \Leftrightarrow \quad \exists x \cdot x \in A \quad \wedge \quad x \mapsto F \in f$$

$$E \in f^{-1}[C] \quad \Leftrightarrow \quad \exists y \cdot y \in C \quad \wedge \quad E \mapsto y \in f$$

- **WARNING**

$$a \mapsto b \in f \quad \Leftrightarrow \quad a \in \text{dom}(f) \quad \wedge \quad b = f(a)$$

- Suppose we have a function f and the following sets:

$$f \in S \rightarrow T \quad A \subseteq S \quad B \subseteq S \quad C \subseteq T \quad D \subseteq T$$

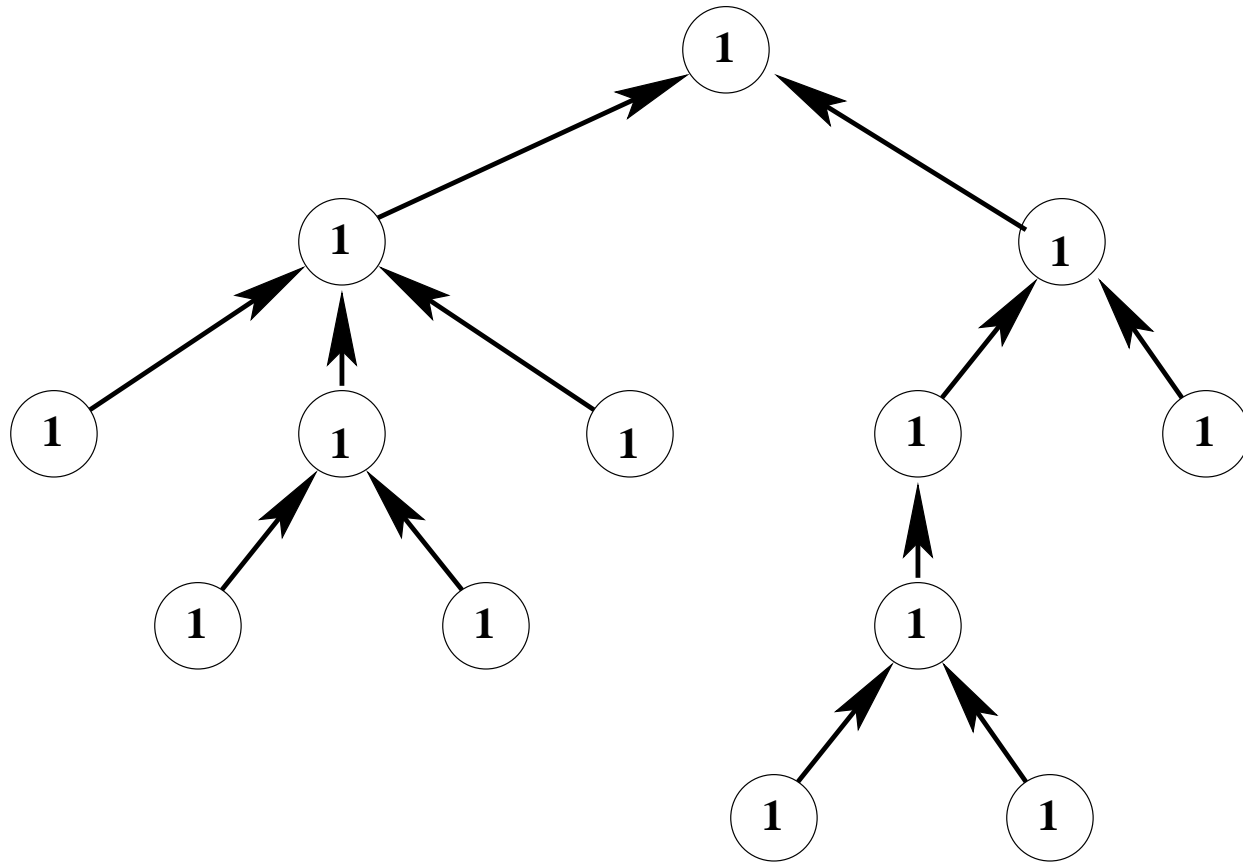
$$f[A \cup B] = f[A] \cup f[B]$$

$$f^{-1}[C \cap D] = f^{-1}[C] \cap f^{-1}[D]$$

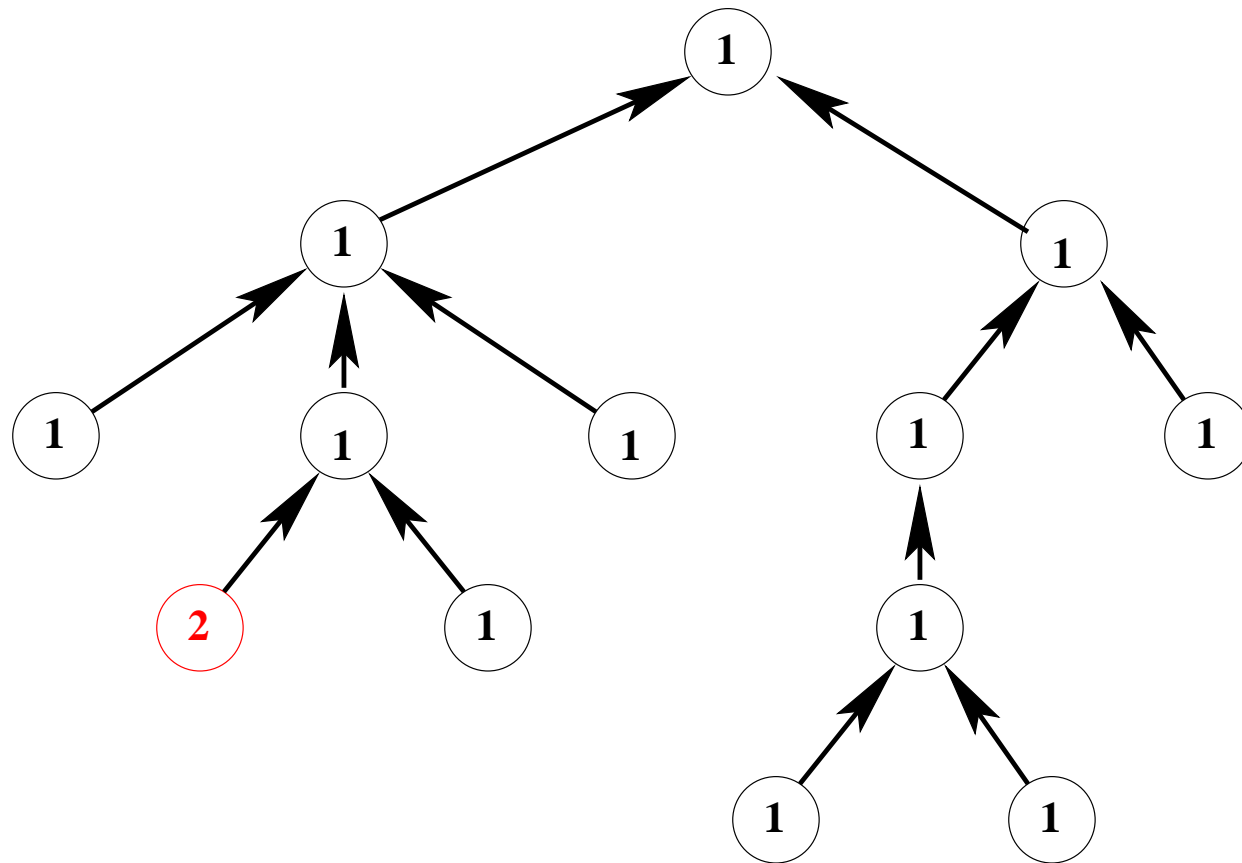
$$f^{-1}[C \setminus D] = f^{-1}[C] \setminus f^{-1}[D]$$

$$f[S] = \text{ran}(f)$$

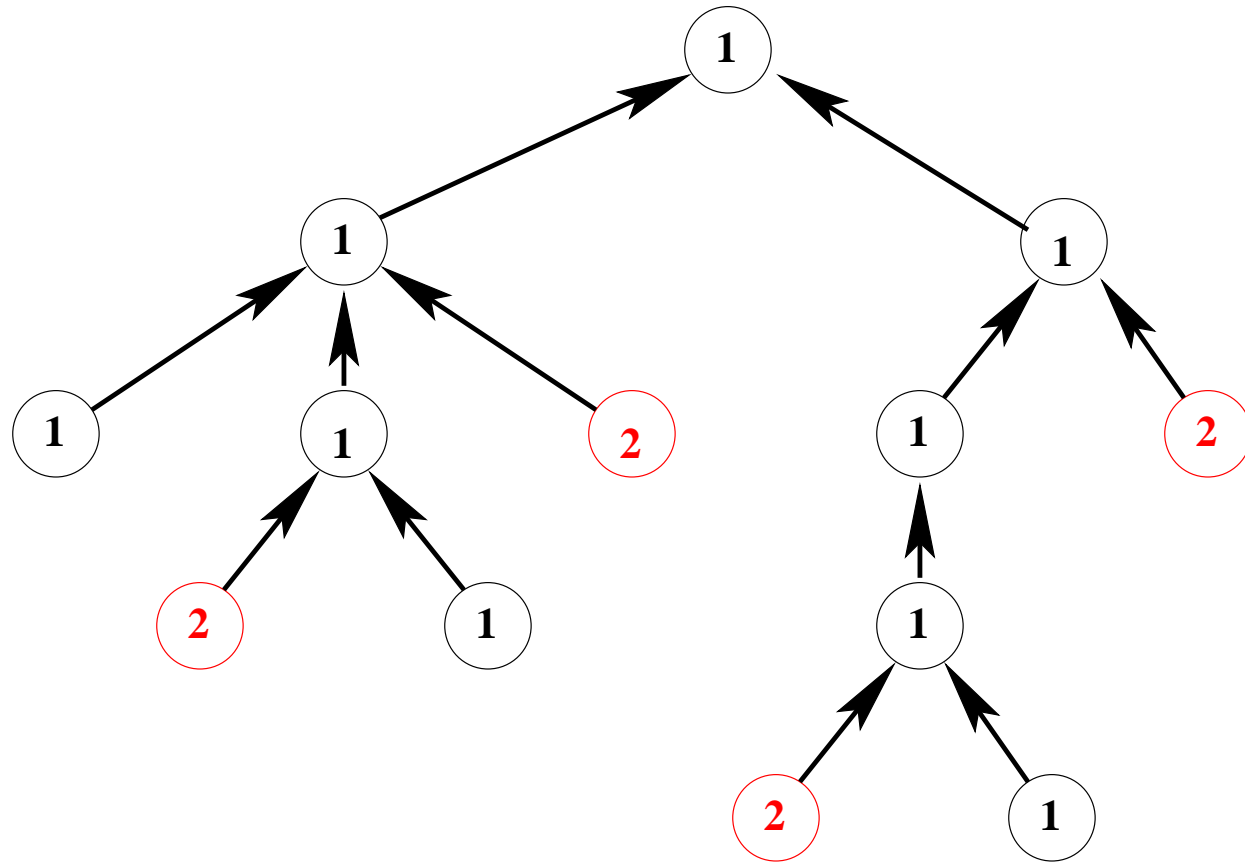
$$f^{-1}[T] = \text{dom}(f)$$



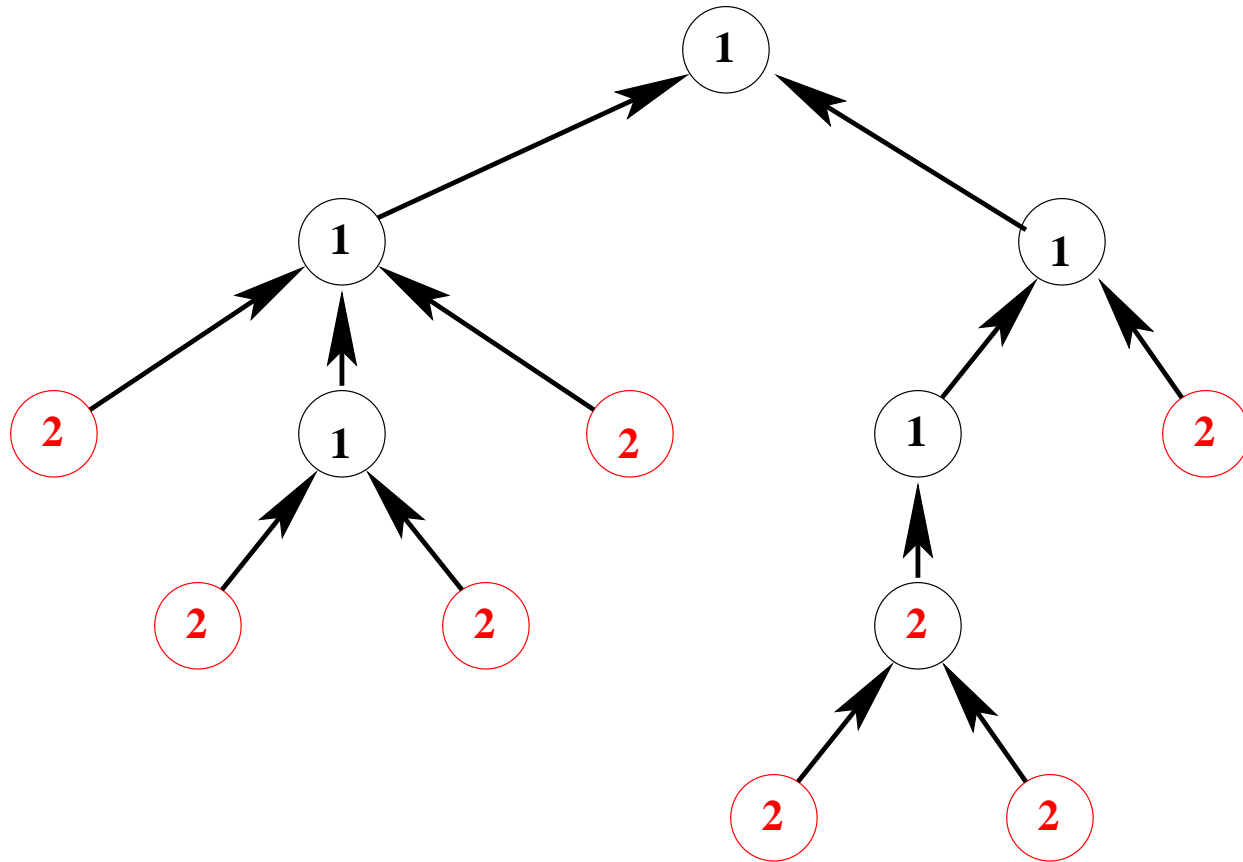
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



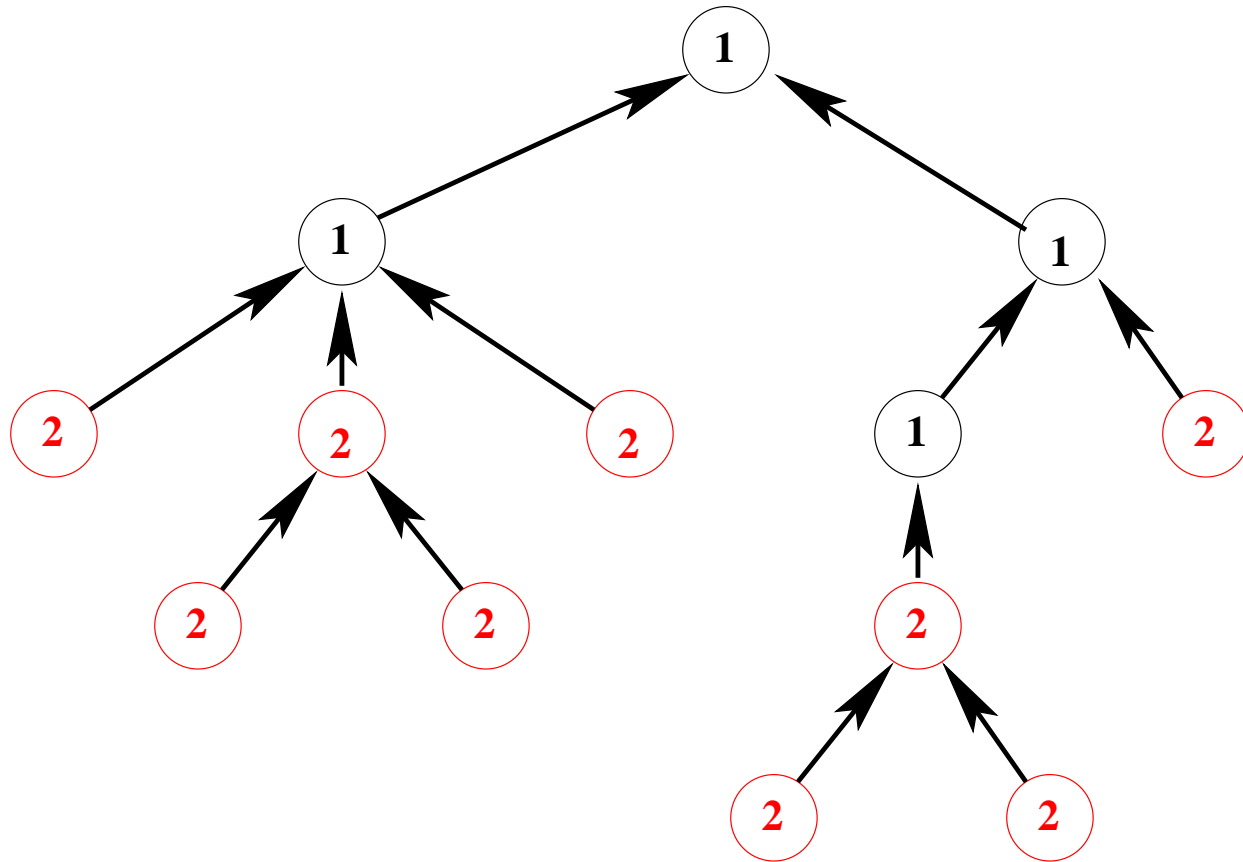
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



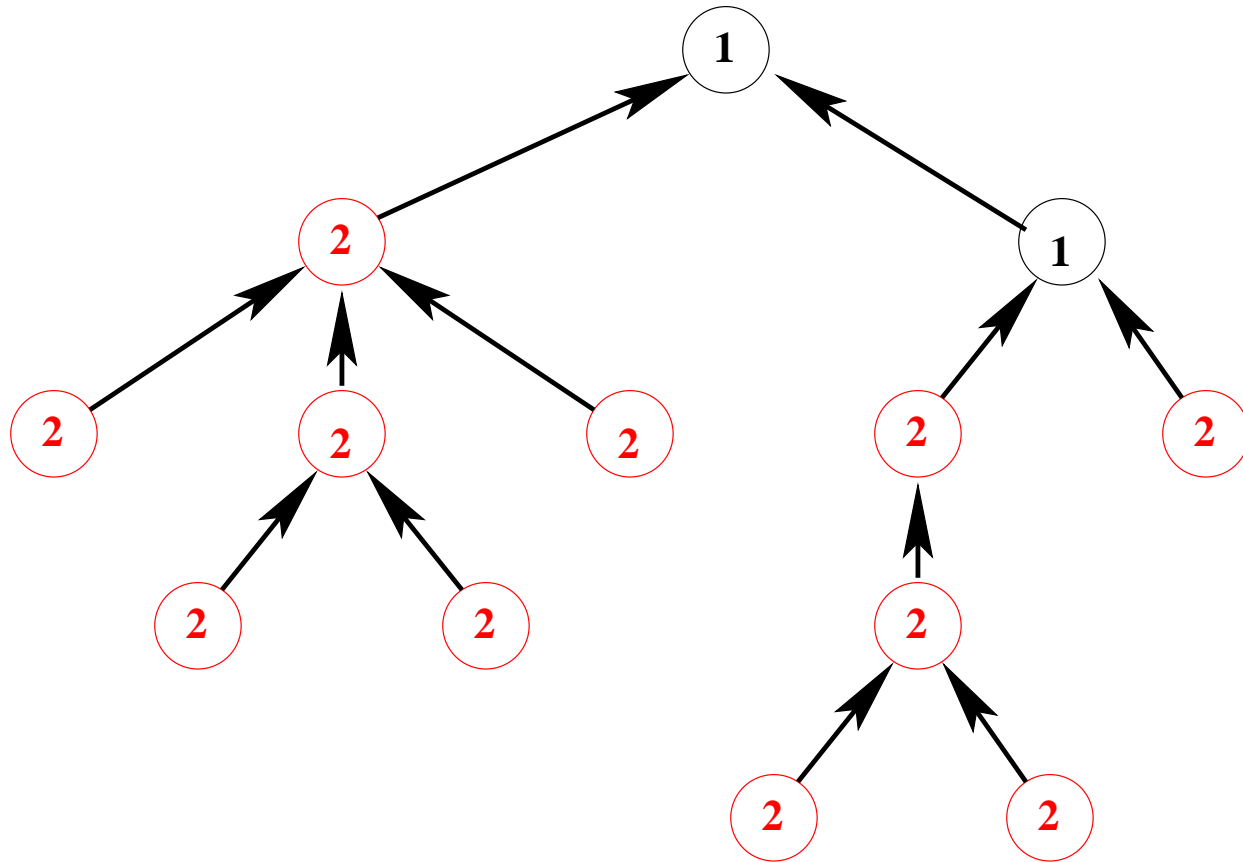
- the guards: $\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$



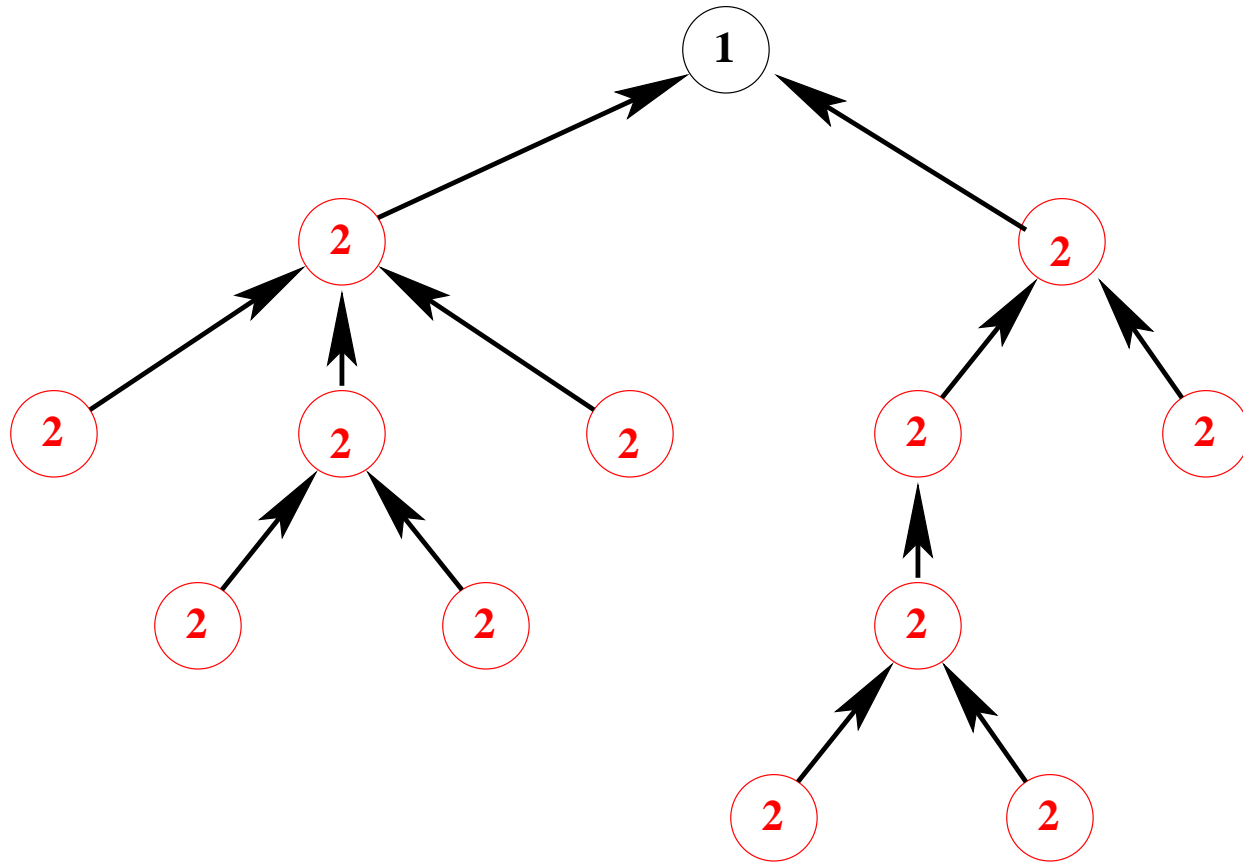
- the guards: $\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$



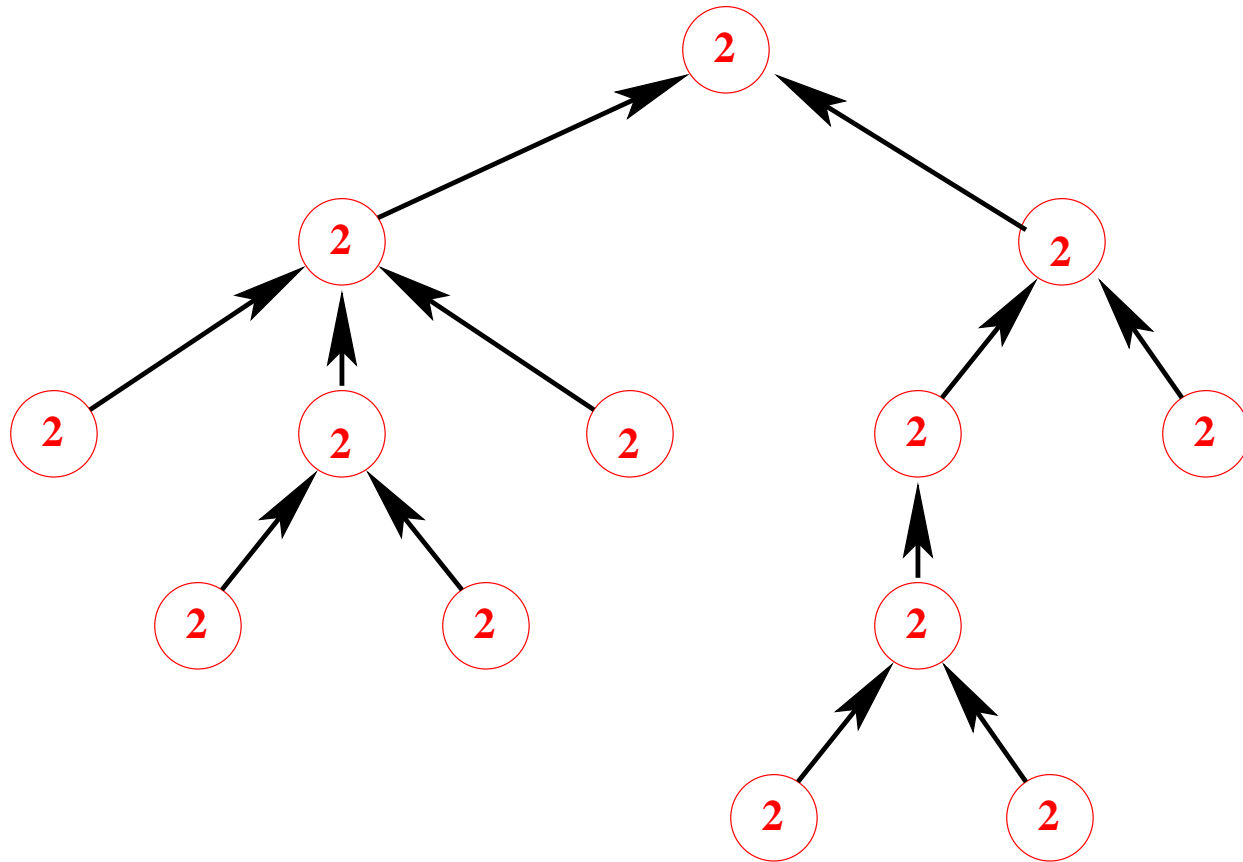
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



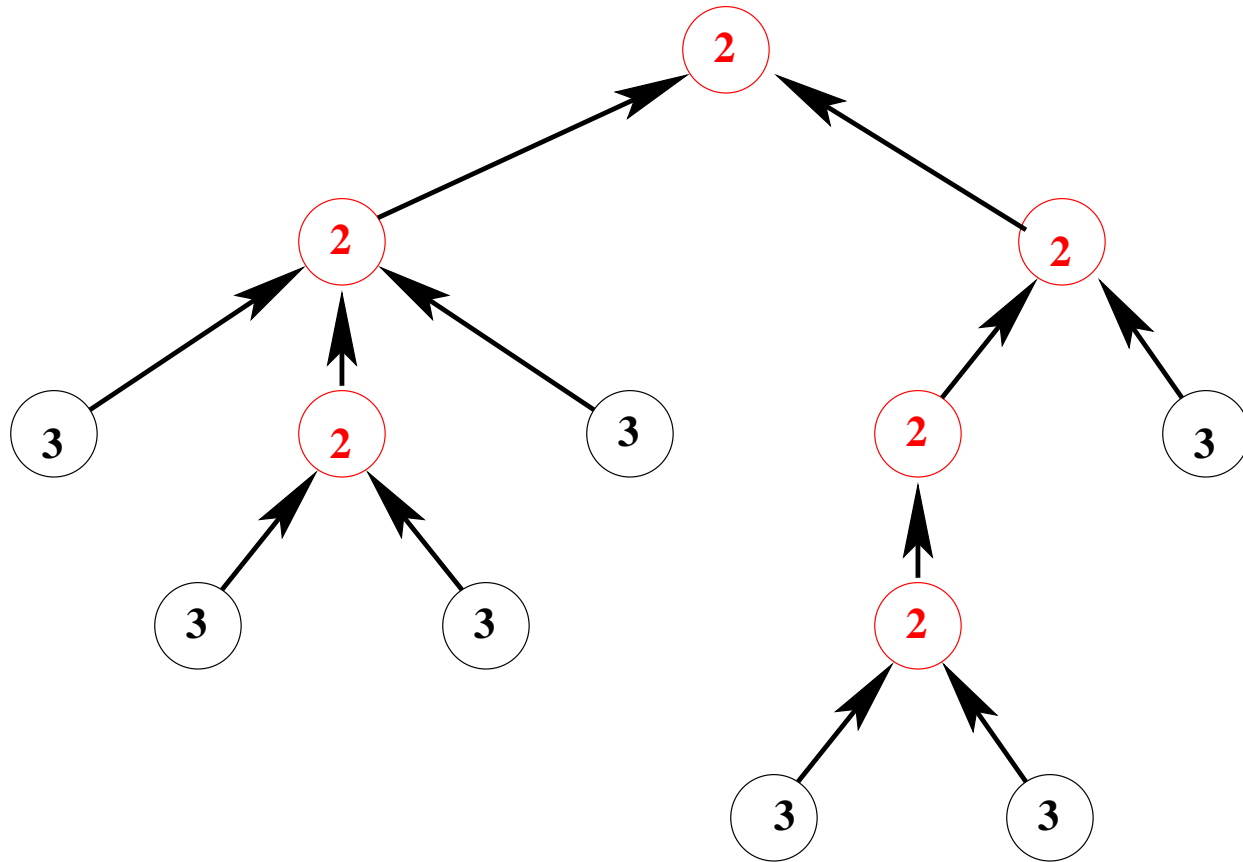
- the guards:
$$\left\{ \begin{array}{l} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{array} \right.$$



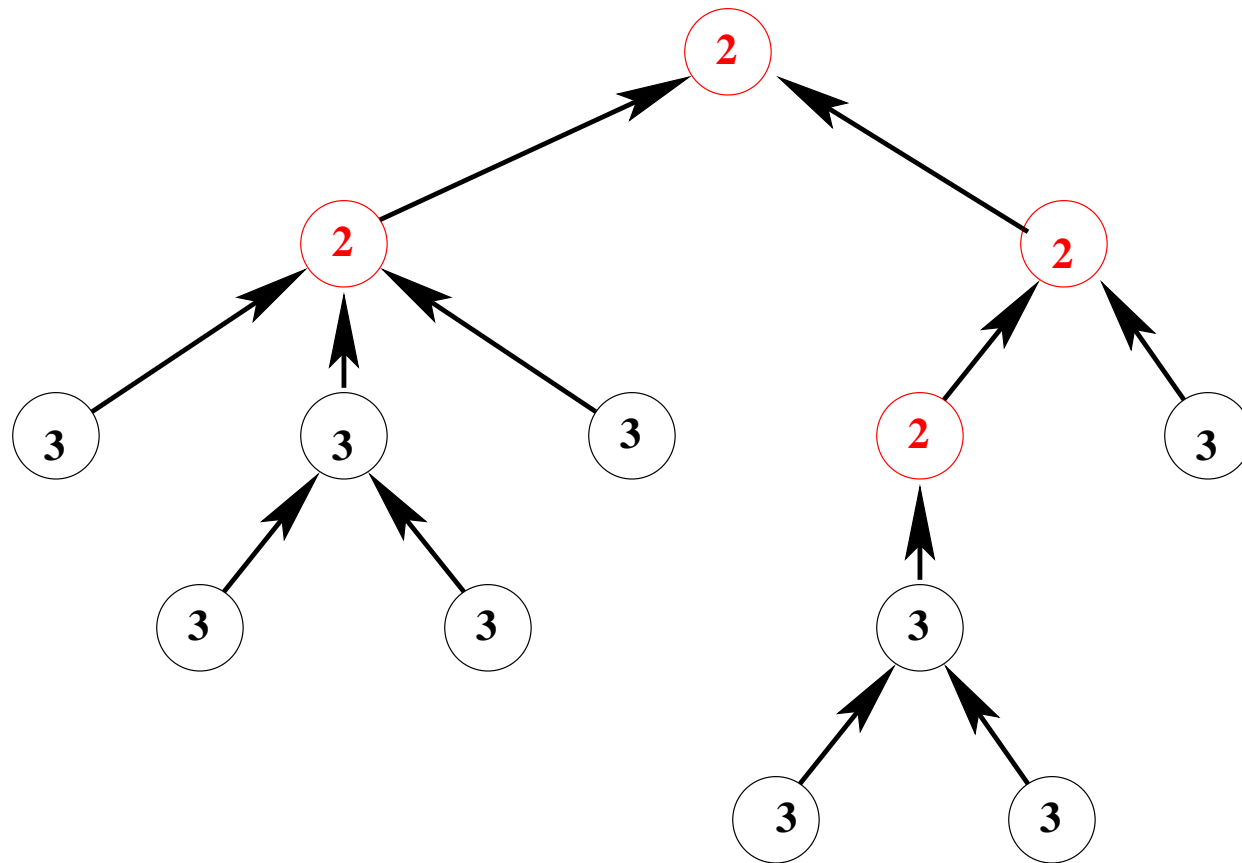
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



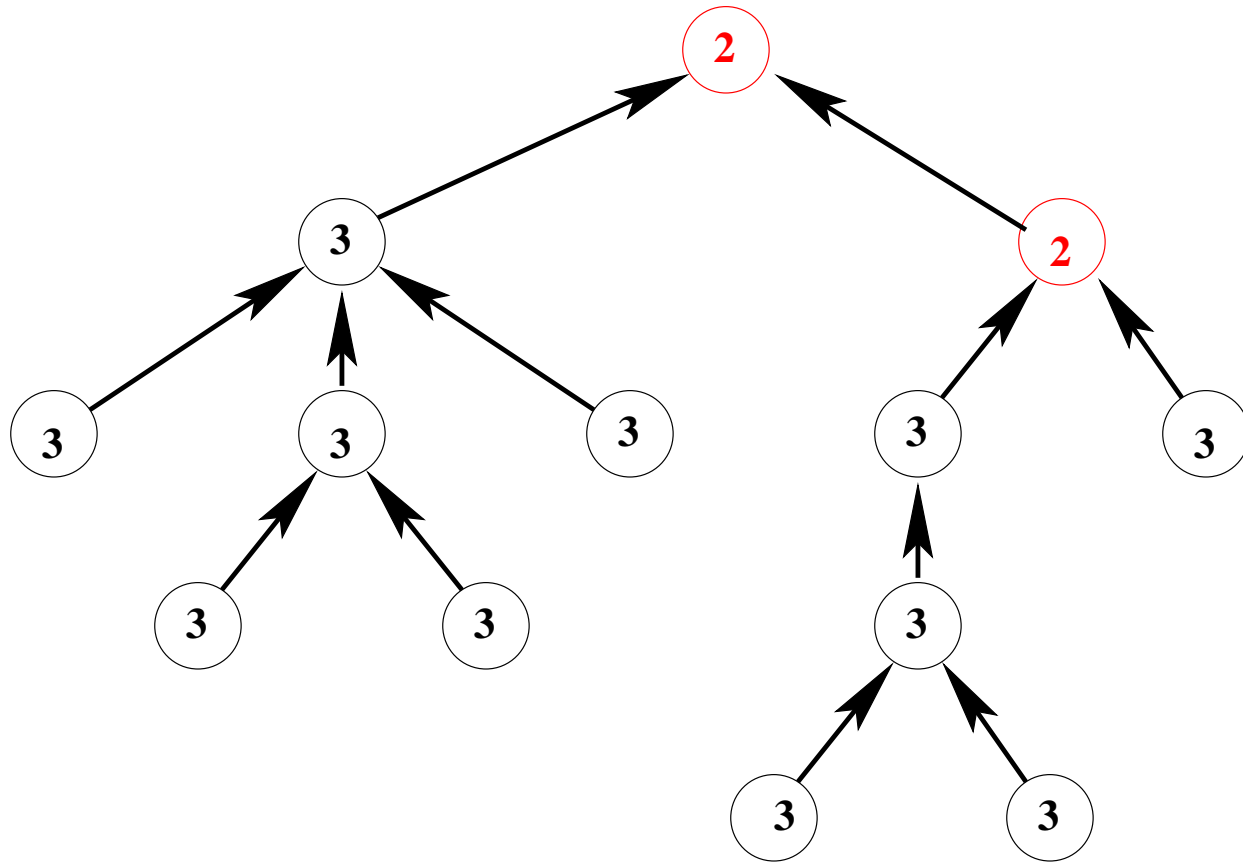
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



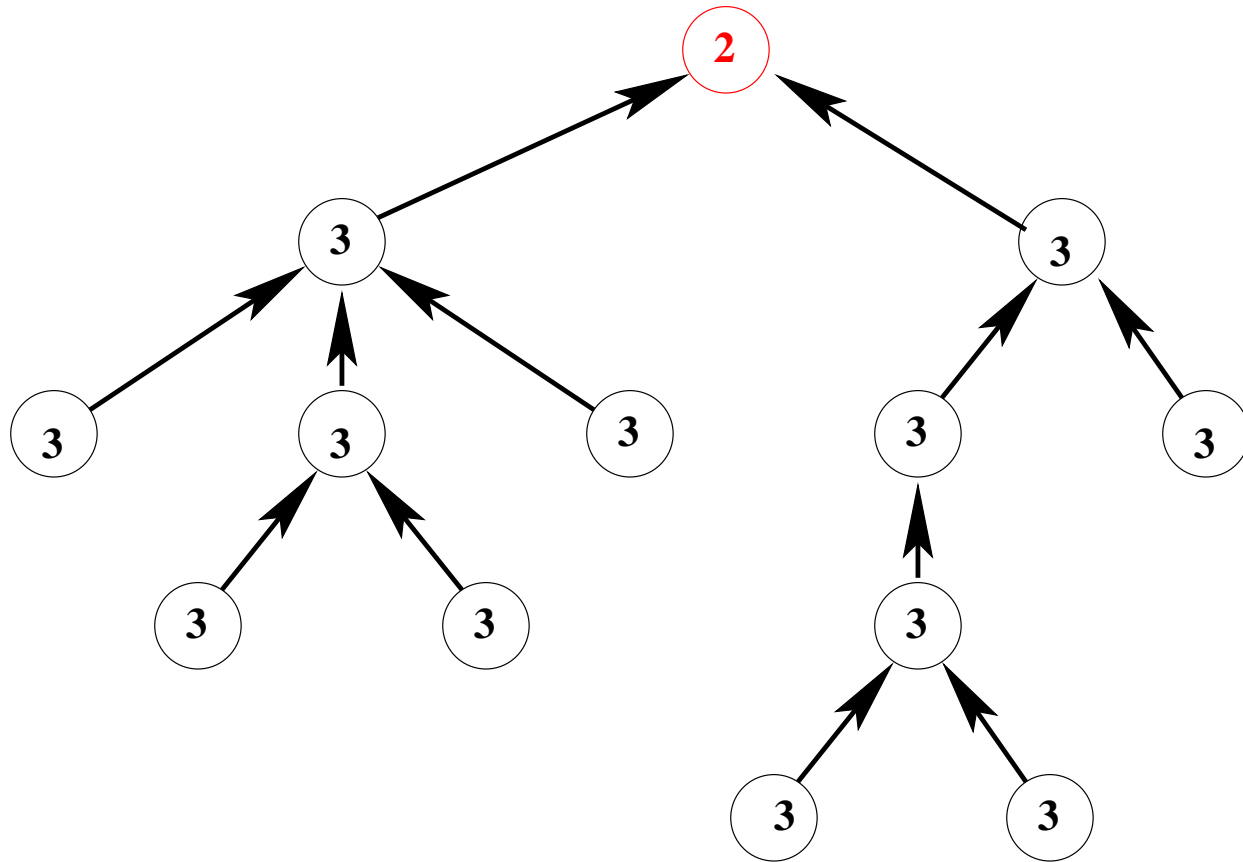
- the guards: $\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$



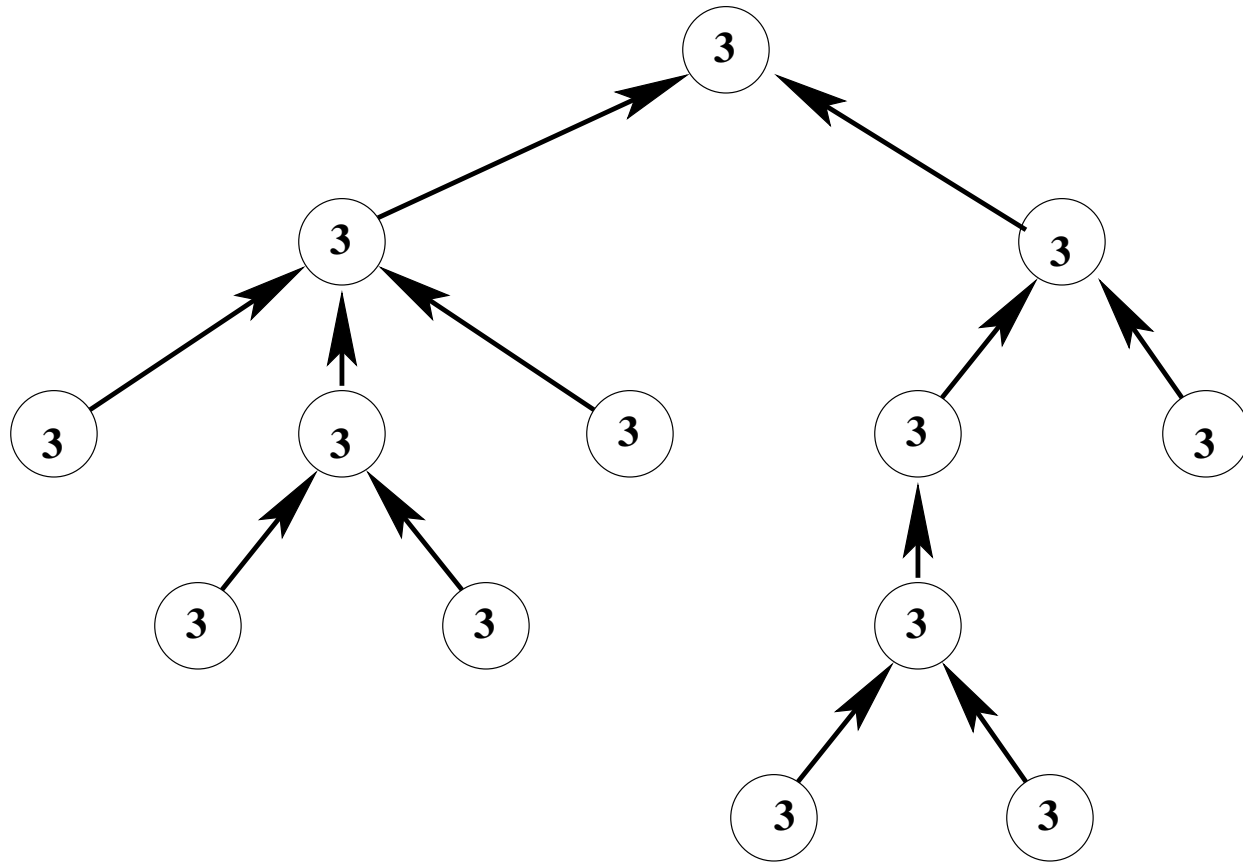
- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$



- the guards:
$$\begin{cases} c(r) = c(n) \\ \forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)) \end{cases}$$

$$\text{inv0_1: } c \in P \rightarrow \mathbb{N}$$

$$\text{inv0_2: } \forall x, y \cdot x \in P \wedge y \in P \Rightarrow c(x) \leq c(y) + 1$$

$$\text{axm1_1: } r \in P$$

$$\text{axm1_2: } f \in P \setminus \{r\} \rightarrow P$$

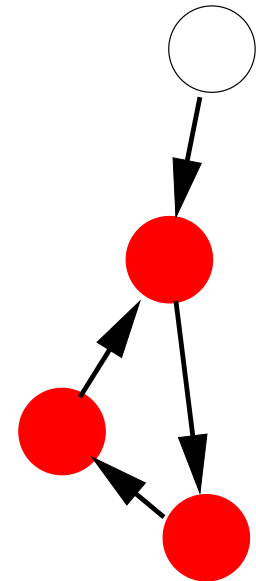
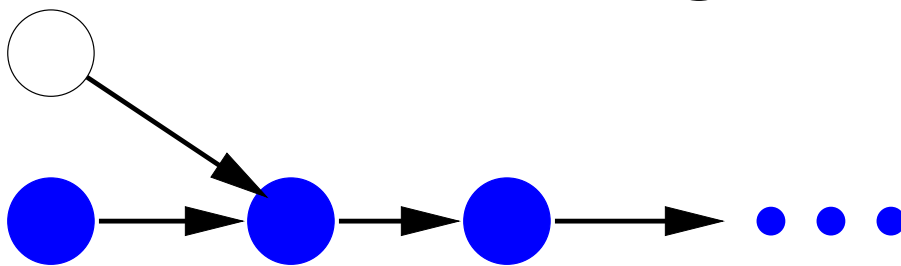
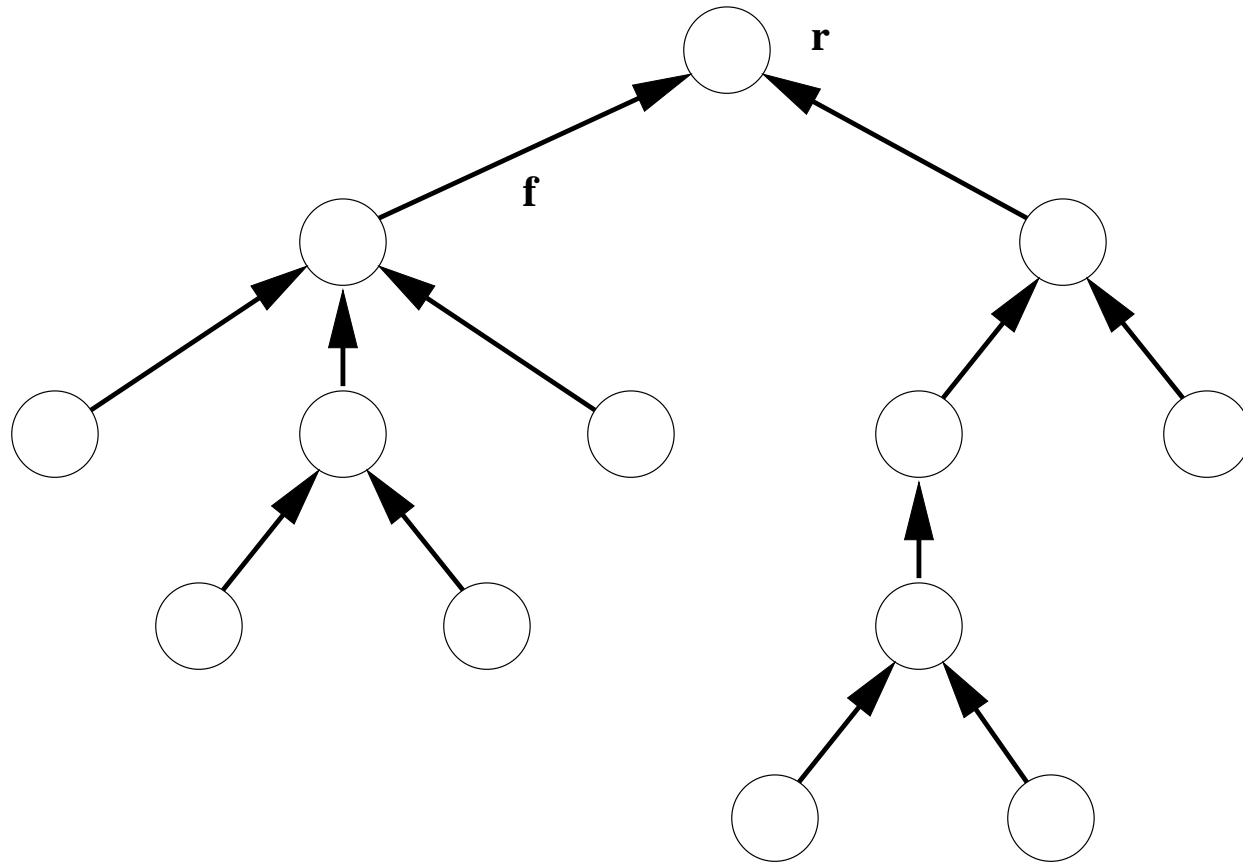
$$\text{inv1_1: } \forall m \cdot m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m)$$

$$\text{thm1_1: } \forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$$

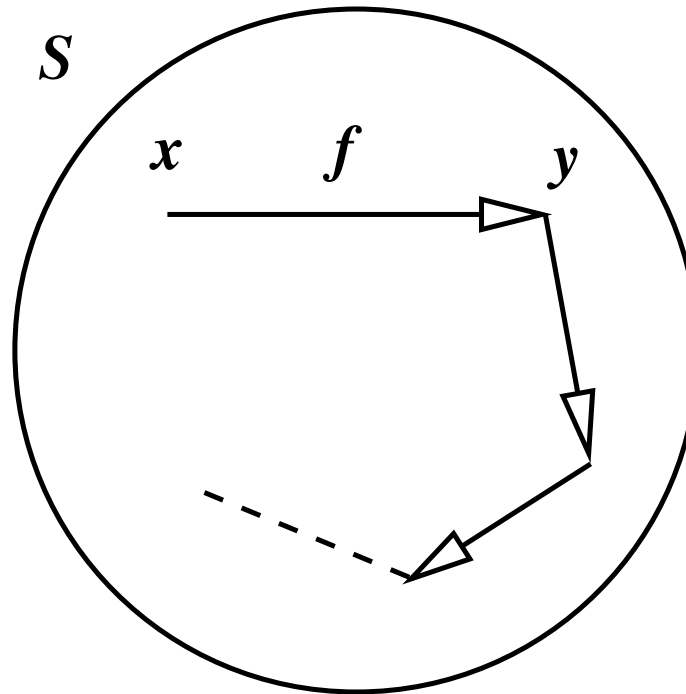
- Properties and invariants are **not sufficient** to prove **thm1_1**

Problems with the **parent** Function: Cycles and Infinite Chains

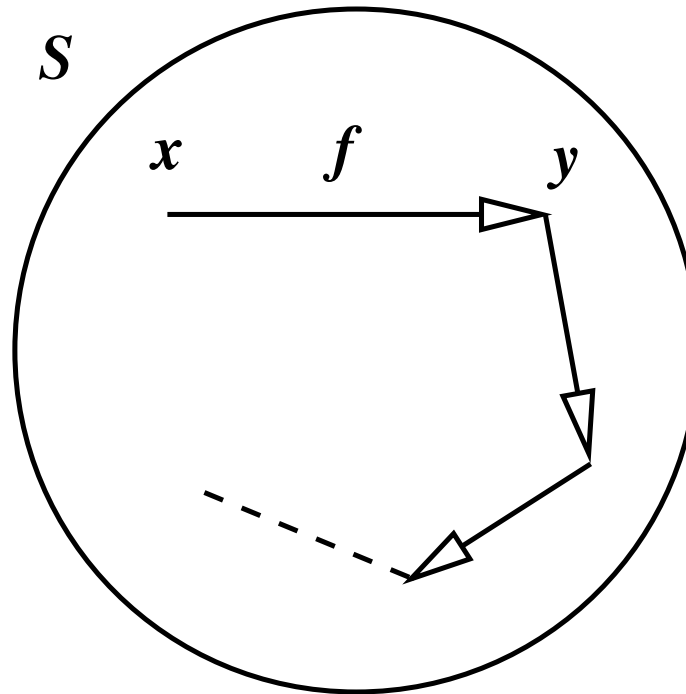
46



- The set S is made of **cycles** or **infinite chains**

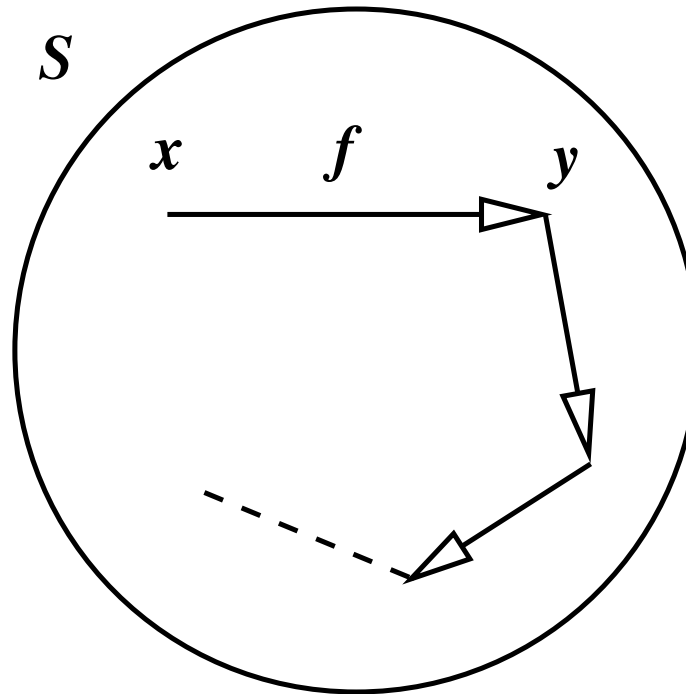


- The set S is made of **cycles** or **infinite chains**



$$\forall x \cdot x \in S \Rightarrow (\exists y \cdot y \in S \wedge x \mapsto y \in f)$$

- The set S is made of **cycles** or **infinite chains**



$$\forall x \cdot x \in S \Rightarrow (\exists y \cdot y \in S \wedge x \mapsto y \in f)$$

$$S \subseteq f^{-1}[S]$$

- The **root** (**axm1_1**)
- The **parent function** (**axm1_2**)
- There are **no cycles** and **no infinite chains** (**axm1_3**)

$$\text{axm1_1 : } r \in P$$

$$\text{axm1_2 : } f \in P \setminus \{r\} \rightarrow P$$

$$\text{axm1_3 : } \forall S . \left(\begin{array}{l} S \subseteq P \\ S \subseteq f^{-1}[S] \\ \Rightarrow \\ S = \emptyset \end{array} \right)$$

$$\text{axm1_1 : } r \in P$$

$$\text{axm1_2 : } f \in P \setminus \{r\} \rightarrow P$$

$$\text{axm1_3 : } \forall S . \left(\begin{array}{l} S \subseteq P \\ S \subseteq f^{-1}[S] \\ \Rightarrow \\ S = \emptyset \end{array} \right)$$

$$\text{thm1_2 : } \forall T . \left(\begin{array}{l} T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \subseteq T \end{array} \right)$$

$$\begin{aligned} & \forall S . \left(\begin{array}{l} S \subseteq P \\ S \subseteq f^{-1}[S] \\ \Rightarrow \\ S = \emptyset \end{array} \right) \\ \Rightarrow & \left(\begin{array}{l} T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \subseteq T \end{array} \right) \end{aligned}$$

$$\begin{aligned} & \forall S . \left(\begin{array}{l} S \subseteq P \\ S \subseteq f^{-1}[S] \\ \Rightarrow \\ S = \emptyset \end{array} \right) \\ & T \subseteq P \\ & r \in T \\ & f^{-1}[T] \subseteq T \\ \Rightarrow & P \setminus T = \emptyset \end{aligned}$$

- Removing the second universal quantification
- Replacing $P \subseteq T$ by $P \setminus T = \emptyset$

$$\begin{array}{l}
 \forall S. \left(\begin{array}{l} S \subseteq P \\ S \subseteq f^{-1}[S] \\ \Rightarrow \\ S = \emptyset \end{array} \right) \\
 T \subseteq P \\
 r \in T \\
 f^{-1}[T] \subseteq T \\
 \Rightarrow \\
 P \setminus T = \emptyset
 \end{array}
 \qquad
 \begin{array}{l}
 \left(\begin{array}{l} P \setminus T \subseteq P \\ P \setminus T \subseteq f^{-1}[P \setminus T] \\ \Rightarrow \\ P \setminus T = \emptyset \end{array} \right) \\
 T \subseteq P \\
 r \in T \\
 f^{-1}[T] \subseteq T \\
 \Rightarrow \\
 P \setminus T = \emptyset
 \end{array}$$

- Instantiating S with $P \setminus T$ in universal quantification

$$P \setminus T \subseteq f^{-1}[P \setminus T]$$

$$P \setminus T \subseteq f^{-1}[P] \setminus f^{-1}[T]$$

$$P \setminus T \subseteq \text{dom}(f) \setminus f^{-1}[T]$$

$$P \setminus T \subseteq (P \setminus \{r\}) \setminus f^{-1}[T]$$

$$P \setminus T \subseteq P \setminus (\{r\} \cup f^{-1}[T])$$

$$\{r\} \cup f^{-1}[T] \subseteq T$$

$$\{r\} \subseteq T \quad \wedge \quad f^{-1}[T] \subseteq T$$

$$r \in T \quad \wedge \quad f^{-1}[T] \subseteq T$$

$$\left(\begin{array}{l} P \setminus T \subseteq P \\ P \setminus T \subseteq f^{-1}[P \setminus T] \\ \Rightarrow \\ P \setminus T = \emptyset \end{array} \right)$$

$$\begin{array}{l} T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \setminus T = \emptyset \end{array}$$

$$\left(\begin{array}{l} P \setminus T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \setminus T = \emptyset \end{array} \right)$$

$$\begin{array}{l} T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \setminus T = \emptyset \end{array}$$

$$\forall T . \left(\begin{array}{l} T \subseteq P \\ r \in T \\ f^{-1}[T] \subseteq T \\ \Rightarrow \\ P \subseteq T \end{array} \right)$$

$$\forall T . \left(\begin{array}{l} T \subseteq P \\ r \in T \\ \forall m . \left(\begin{array}{l} m \in P \setminus \{r\} \\ f(m) \in T \\ \Rightarrow \\ m \in T \end{array} \right) \\ \Rightarrow \\ P \subseteq T \end{array} \right)$$

- To prove a “property” T for **all nodes**, it is sufficient to prove that
 - the **root r** has property T
 - a **node m** has property T provided **its parent $f(m)$** has it
- It is an **Induction Rule**

axm1_1 : $r \in P$

Root

axm1_2 : $f \in P \setminus \{r\} \rightarrow P$

Parent function

thm1_3 : $\forall T . \left(\begin{array}{l} T \subseteq P \\ r \in T \\ \forall m . \left(\begin{array}{l} m \in P \setminus \{r\} \\ f(m) \in T \end{array} \right) \\ \Rightarrow \\ m \in T \end{array} \right) \Rightarrow P \subseteq T$ Induction

We have to prove:

$$\forall x \cdot (x \in P \Rightarrow c(r) \leq c(x))$$

or, equivalently

$$P \subseteq \{x \mid x \in P \wedge c(r) \leq c(x)\}$$

We instantiate T with

$$\{x \mid x \in P \wedge c(r) \leq c(x)\}$$

in **thm1_3**

$$\forall T \cdot \left(\begin{array}{l} T \subseteq P \\ r \in T \\ \forall m \cdot \left(\begin{array}{l} m \in P \setminus \{r\} \\ f(m) \in T \end{array} \Rightarrow m \in T \right) \end{array} \right) \Rightarrow P \subseteq T$$

We have thus to prove

$$\{x \mid x \in P \wedge c(r) \leq c(x)\} \subseteq P$$

$$r \in \{x \mid x \in P \wedge c(r) \leq c(x)\}$$

$$\forall m \cdot \left(\begin{array}{l} m \in P \setminus \{r\} \\ f(m) \in \{x \mid x \in P \wedge c(r) \leq c(x)\} \\ \Rightarrow \\ m \in \{x \mid x \in P \wedge c(r) \leq c(x)\} \end{array} \right)$$

The only “difficult” part is the **third one**, reducing to

$$\begin{array}{l} m \in P \setminus \{r\} \\ c(r) \leq c(f(m)) \end{array}$$

$$\Rightarrow$$

$$c(r) \leq c(m)$$

It remains to prove

$$\begin{aligned} & m \in P \setminus \{r\} \\ & c(r) \leq c(f(m)) \\ \Rightarrow \\ & c(r) \leq c(m) \end{aligned}$$

But we have (this is invariant **inv1_1**)

$$\forall m \cdot (m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m))$$

- The result follows by **transitivity**

ascending

any n **where**

$n \in P$

$c(r) = c(n)$

$\forall m \cdot \left(\begin{array}{l} m \in f^{-1}[\{n\}] \\ \Rightarrow \\ c(n) < c(m) \end{array} \right)$

then

$c(n) := c(n) + 1$

end

- The **third** guard is **correct** (n uses its **children counters** only)
- The **second** guard is **not correct** (n uses the **root counter**)

- Relative deadlock freedom
- The **refinement does not deadlock** (since the abstraction does not)

ascending

any n **where**

$n \in P$

$c(r) = c(n)$

$\forall m \cdot m \in f^{-1}[\{n\}] \Rightarrow c(f(m)) < c(m)$

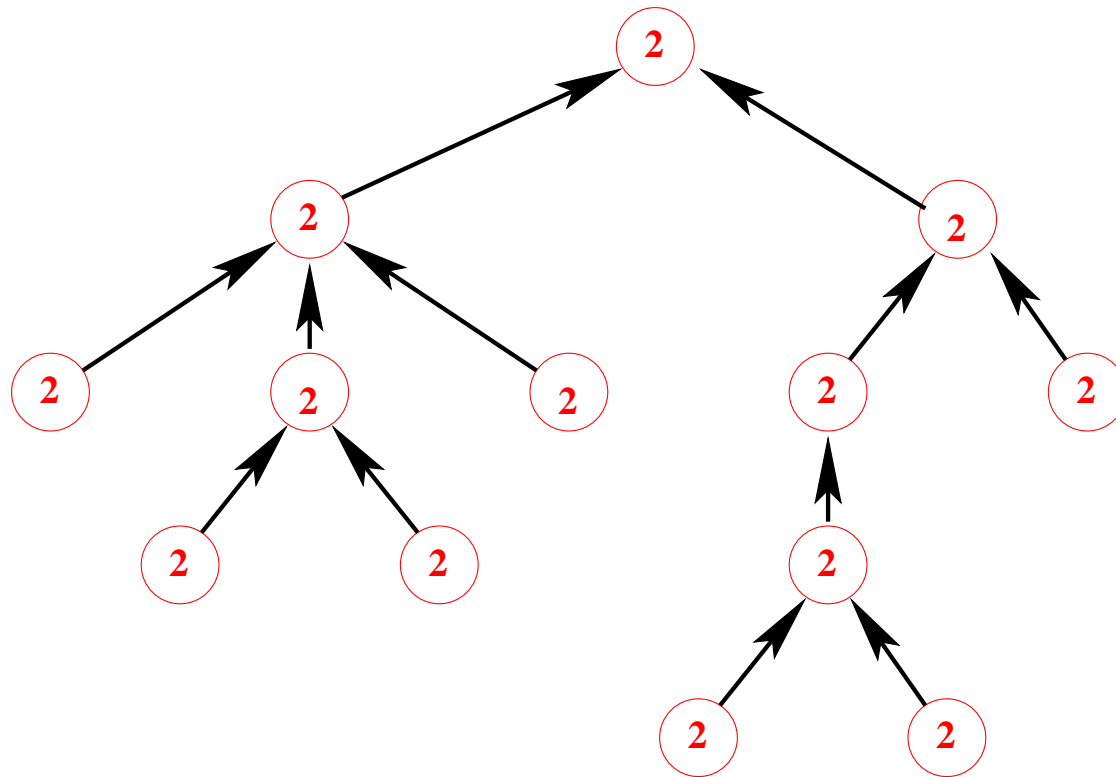
then

$c(n) := c(n) + 1$

end

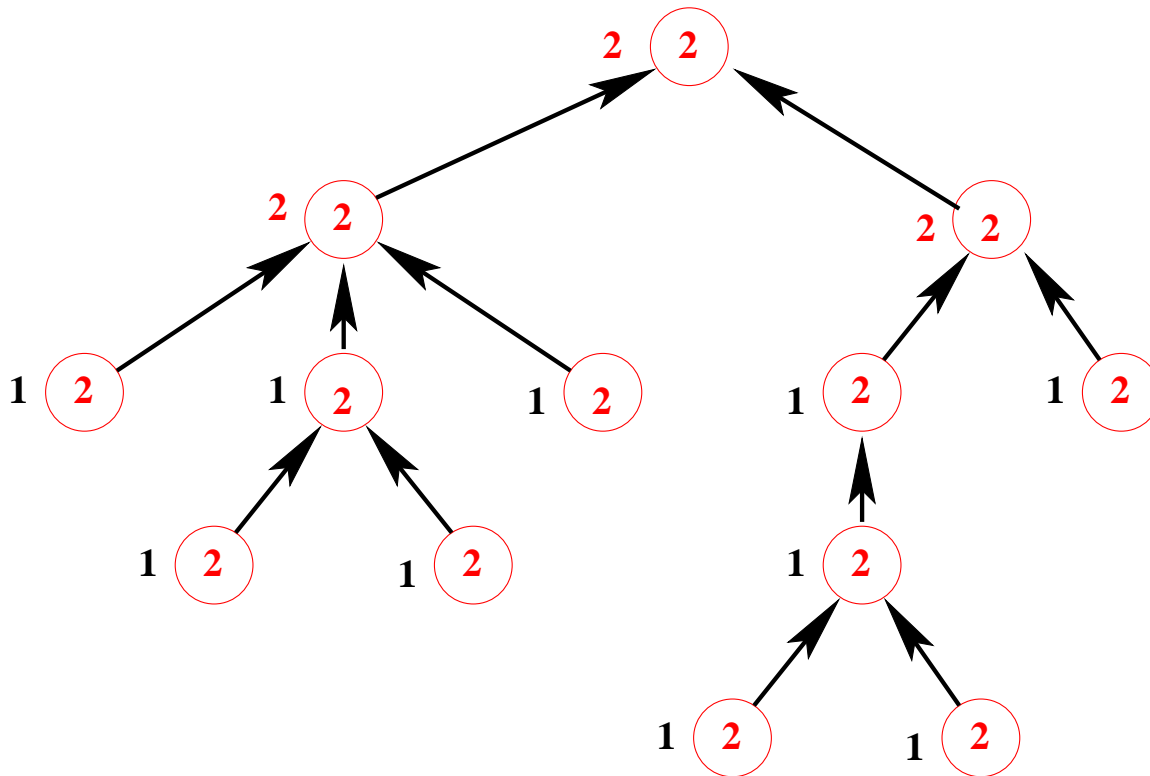
- The **second** guard is **not correct**: n uses the **root counter** $c(r)$

- We want to **replace the guard $c(r) = c(n)$** in event "ascending"



- Processes must be aware when this situation does occur

We add a **second counter d** at each node



- The second counter d has properties which are **similar to those of c**

carrier set: P

constants: r, f

variables: c, d

Invariant **inv2_2**
is as **inv0_2**

inv2_1: $d \in P \rightarrow \mathbb{N}$

inv2_2: $\forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ d(x) \leq d(y) + 1 \end{array} \right)$

ascending

any n **where**

$$n \in P$$

$$c(r) = c(n)$$

$$\forall m \cdot \left(\begin{array}{l} m \in f^{-1}[\{n\}] \\ \Rightarrow \\ c(f(m)) < c(m) \end{array} \right)$$

then

$$c(n) := c(n) + 1$$

end

descending

any n **where**

$$n \in P$$

$$\forall m \cdot \left(\begin{array}{l} m \in P \\ \Rightarrow \\ d(n) \leq d(m) \end{array} \right)$$

then

$$d(n) := d(n) + 1$$

end

- Proof of the preservation of **inv2_2** by event "descending" is easy

$$\mathbf{inv2_2:} \quad \forall x, y \cdot \left(\begin{array}{l} x \in P \\ y \in P \\ \Rightarrow \\ d(x) \leq d(y) + 1 \end{array} \right)$$

- It is similar to that of the preservation of **inv0_2** by event "ascending"

- Relative deadlock freedom
- The new event does not take control for ever

- We extend the **invariant of counter d**
- We establish the **relationship between both counters c and d**
- This will allow us to refine event **ascending**
- We construct the **descending wave** (by refining event descending)
- Remark: this is the **most difficult refinement**

$$\mathbf{inv3_1:} \quad \forall m \cdot m \in P \setminus \{r\} \Rightarrow d(m) \leq d(f(m))$$

$$\mathbf{inv3_2:} \quad d(r) \leq c(r)$$

$$\mathbf{thm3_1:} \quad \forall m \cdot m \in P \Rightarrow d(m) \leq d(r)$$

- **thm3_1** can be proved by using the **tree Induction (axm1_3)**
- **inv3_1** and **thm3_1** have to be compared to **inv1_1** and **thm1_1**

$$\mathbf{inv1_1:} \quad \forall m \cdot m \in P \setminus \{r\} \Rightarrow c(f(m)) \leq c(m)$$

$$\mathbf{thm1_1:} \quad \forall m \cdot m \in P \Rightarrow c(r) \leq c(m)$$

(abstract-)ascending

any n **where** $n \in P$ $c(n) = c(r)$

$$\forall m \cdot \left(\begin{array}{l} m \in f^{-1}[\{n\}] \\ \Rightarrow \\ c(f(m)) < c(m) \end{array} \right)$$
then $c(n) := c(n) + 1$ **end**

(concrete-)ascending

any n **where** $n \in P$ $c(n) = d(n)$

$$\forall m \cdot \left(\begin{array}{l} m \in f^{-1}[\{n\}] \\ \Rightarrow \\ c(f(m)) < c(m) \end{array} \right)$$
then $c(n) := c(n) + 1$ **end**

concrete guard
 according to **thm3_1**
 invariant **inv3_2**
 according to **thm1_1**

abstract guard

$$\begin{array}{l} c(n) = d(n) \\ d(n) \leq d(r) \\ d(r) \leq c(r) \\ c(r) \leq c(n) \\ \vdash \\ c(n) = c(r) \end{array}$$

- We have reached our goal: **event ascending indeed fulfills FUN-2**

(abstract-)descending

any n **where**

$n \in P$

$\forall m \cdot \left(\begin{array}{l} m \in P \\ \Rightarrow \\ d(n) \leq d(m) \end{array} \right)$

then

$d(n) := d(n) + 1$

end

(concrete-)descending_1

any n **where**

$n \in P \setminus \{r\}$

$d(n) \neq d(f(n))$

then

$d(n) := d(n) + 1$

end

Guard strengthening:

$n \in P \setminus \{r\}$

$d(n) \neq d(f(n))$

$m \in P$

\Rightarrow

$d(n) \leq d(m)$

- In order to prove guard strengthening, we need the theorems:

thm3_2: $\forall n \cdot n \in P \setminus \{r\} \Rightarrow d(f(n)) \in d(n) .. d(n) + 1$

thm3_3: $\forall n \cdot n \in P \Rightarrow d(r) \in d(n) .. d(n) + 1$

$$\begin{array}{l}
 n \in P \setminus \{r\} \\
 d(n) \neq d(f(n)) \\
 m \in P \\
 \Rightarrow \\
 d(n) \leq d(m)
 \end{array}
 \quad \rightsquigarrow \quad
 \begin{array}{l}
 d(r) \in d(n) + 1 .. d(n) + 2 \\
 d(r) \in d(m) .. d(m) + 1 \\
 \Rightarrow \\
 d(n) \leq d(m)
 \end{array}$$

$$d(f(n)) = d(n) + 1 \quad (\text{thm3_2 and } d(n) \neq d(f(n)))$$

$$d(r) \in d(f(n)) .. d(f(n)) + 1 \quad (\text{thm3_3})$$

(abstract-)descending

any n **where**

$n \in P$

$\forall m \cdot \left(\begin{array}{l} m \in P \\ \Rightarrow \\ d(n) \leq d(m) \end{array} \right)$

then

$d(n) := d(n) + 1$

end

(concrete-)descending_2

when

$d(r) \neq c(r)$

then

$d(r) := d(r) + 1$

end

- Here we need a witness for n : the root r is the obvious choice

Guard strengthening

$d(r) \neq c(r)$

$m \in P$

\Rightarrow

$d(r) \leq d(m)$

- In order to prove guard strengthening, we need the theorem

thm3_4: $\forall n \cdot n \in P \Rightarrow c(r) \in d(n) .. d(n) + 1$

- We instantiate n with r leading to $c(r) \in d(r) .. d(r + 1)$
- Then we instantiate n to m leading to $c(r) \in d(m) .. d(m + 1)$

$$c(r) \in d(r) .. d(r + 1)$$

$$c(r) \in d(m) .. d(m + 1)$$

$$d(r) \neq c(r)$$

$$m \in P$$

$$\Rightarrow d(r) \leq d(m)$$

$$d(r) + 1 \in d(m) .. d(m) + 1$$

$$\Rightarrow d(r) \leq d(m)$$

- In order to prove the previous theorem

$$\mathbf{thm3_4:} \quad \forall n \cdot n \in P \Rightarrow c(r) \in d(n) .. d(n) + 1$$

- We need the following additional invariant

$$\mathbf{inv3_3:} \quad \forall n \cdot n \in P \Rightarrow c(n) \in d(n) .. d(n) + 1$$

- We have thus to prove that this invariant is preserved by the three events: ascending, descending_1, and descending_2.

inv3_1: $\forall m \cdot (m \in P \setminus \{r\} \Rightarrow d(m) \leq d(f(m)))$

inv3_2: $d(r) \leq c(r)$

inv3_3: $\forall n \cdot (n \in P \Rightarrow c(n) \in d(n) .. d(n) + 1)$

thm3_1: $\forall m \cdot (m \in P \Rightarrow d(m) \leq d(r))$

thm3_2: $\forall n \cdot (n \in P \setminus \{r\} \Rightarrow d(f(n)) \in d(n) .. d(n) + 1)$

thm3_3: $\forall n \cdot (n \in P \Rightarrow d(r) \in d(n) .. d(n) + 1)$

thm3_4: $\forall n \cdot (n \in P \Rightarrow c(r) \in d(n) .. d(n) + 1)$

ascending

any n **where**

$$n \in P$$

$$c(n) = d(n)$$

$$\forall m \cdot (m \in f^{-1}[\{n\}] \Rightarrow c(n) \neq c(m))$$

then

$$c(n) := c(n) + 1$$

end

descending_1

any n **where**

$$n \in P \setminus \{r\}$$

$$d(n) \neq d(f(n))$$

then

$$d(n) := d(n) + 1$$

end

descending_2

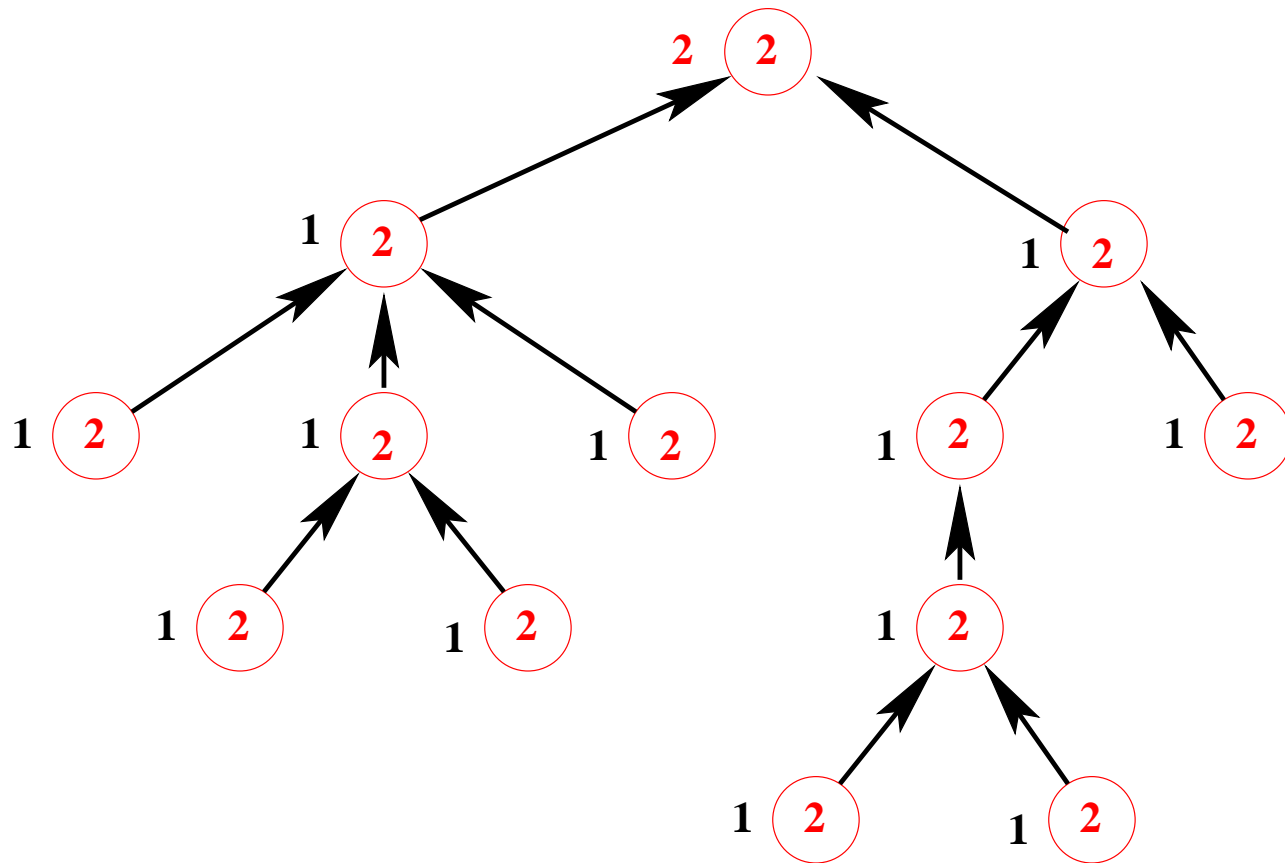
when

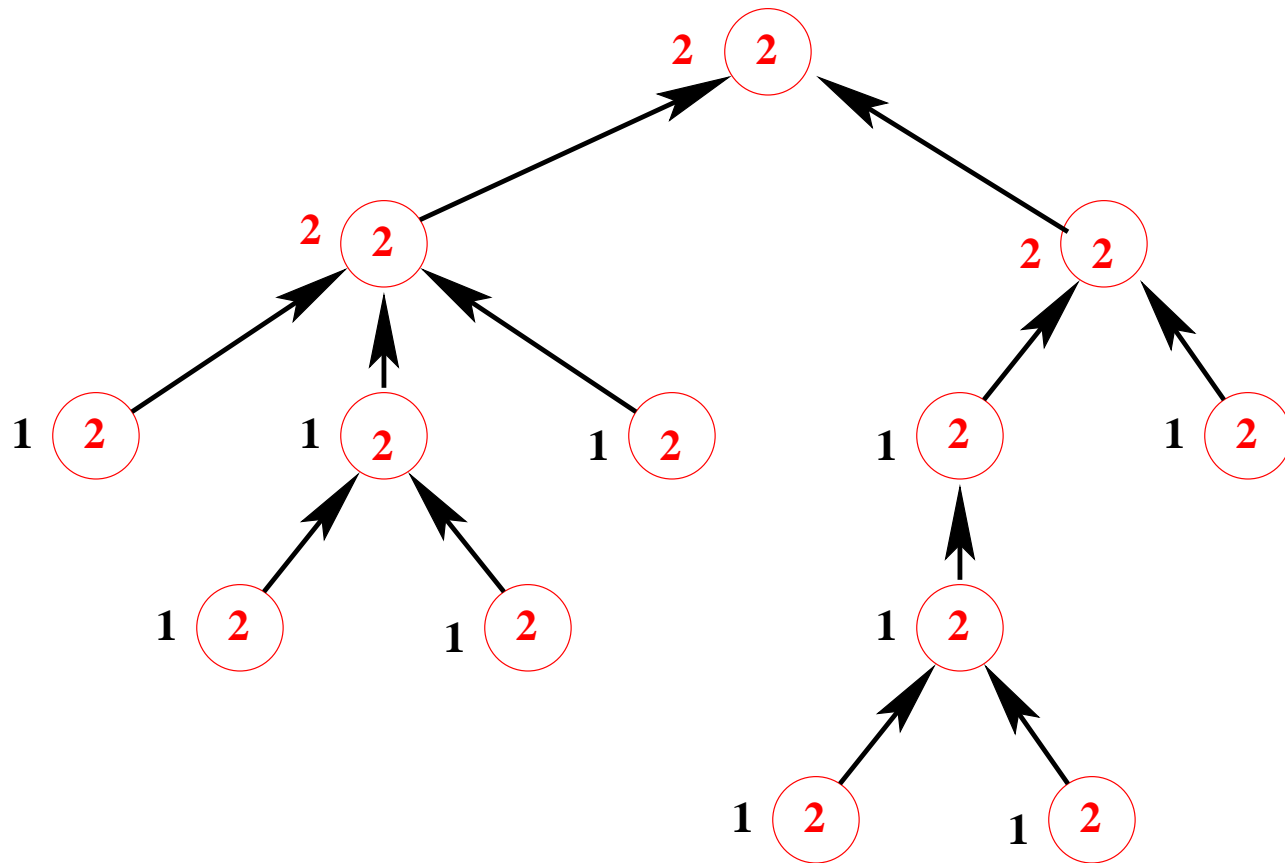
$$d(r) \neq c(r)$$

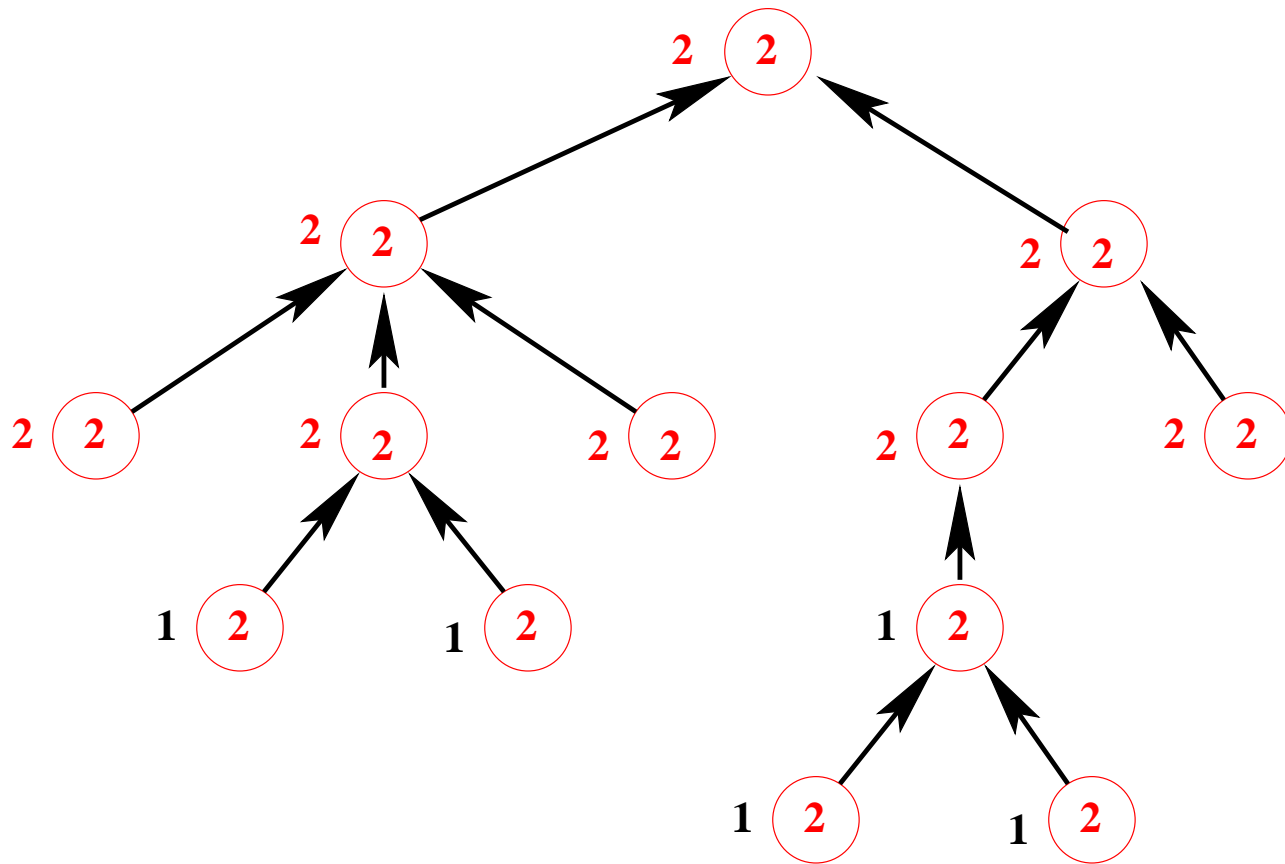
then

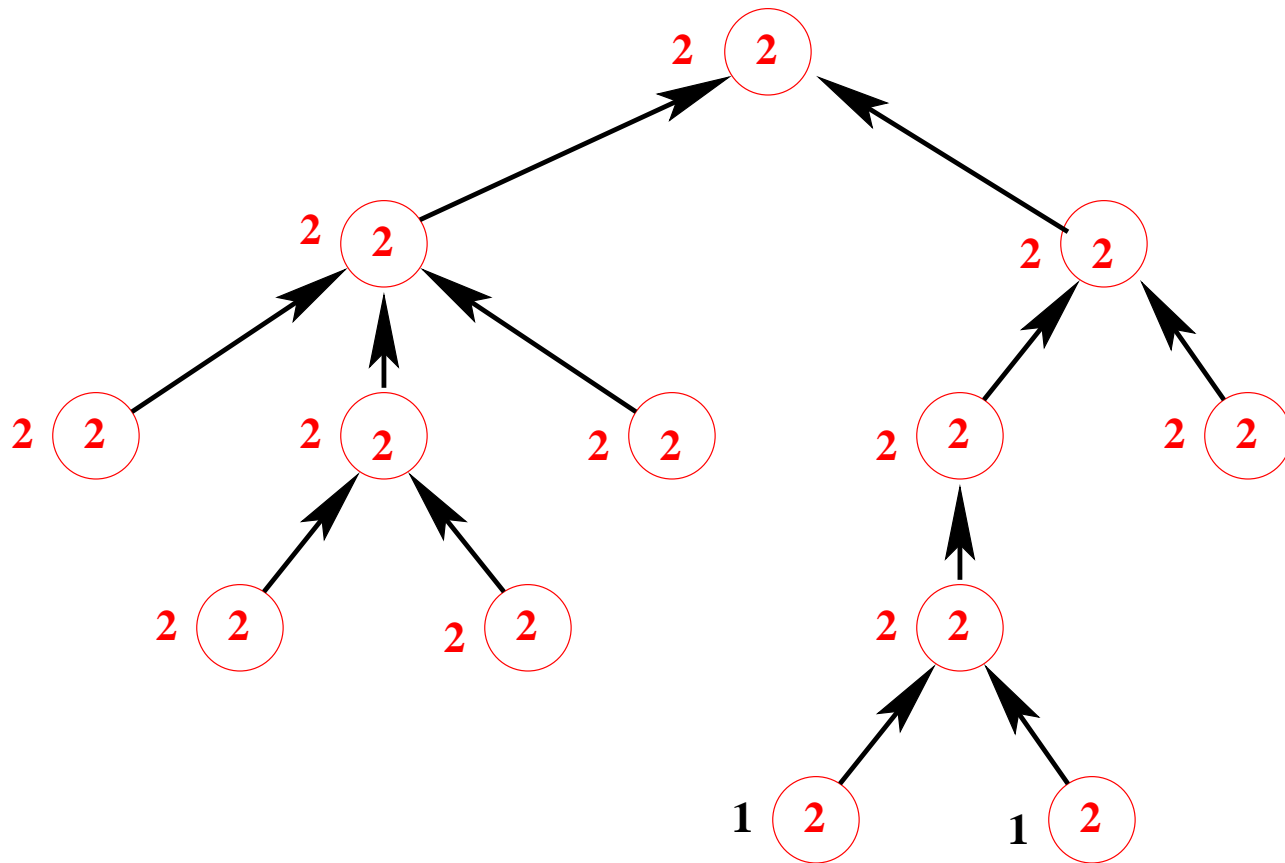
$$d(r) := d(r) + 1$$

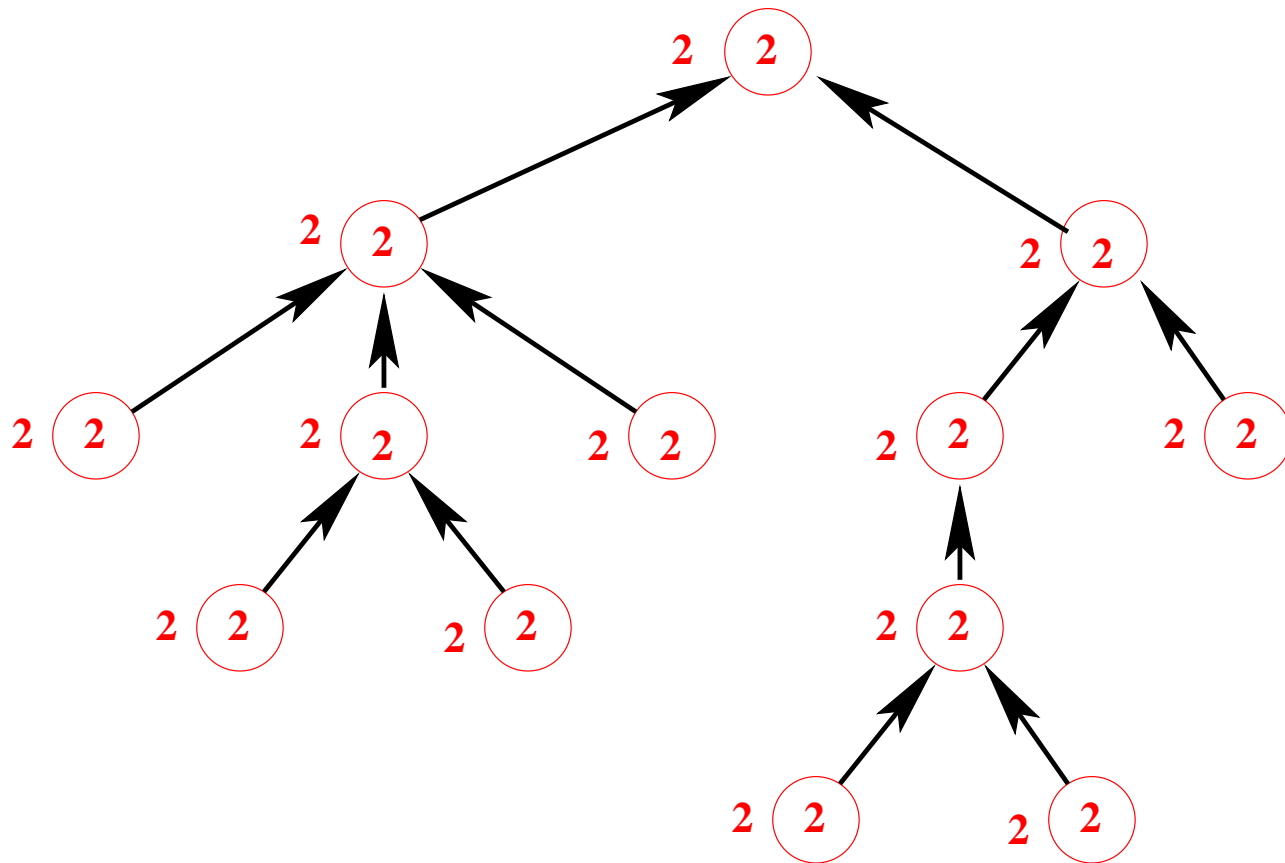
end

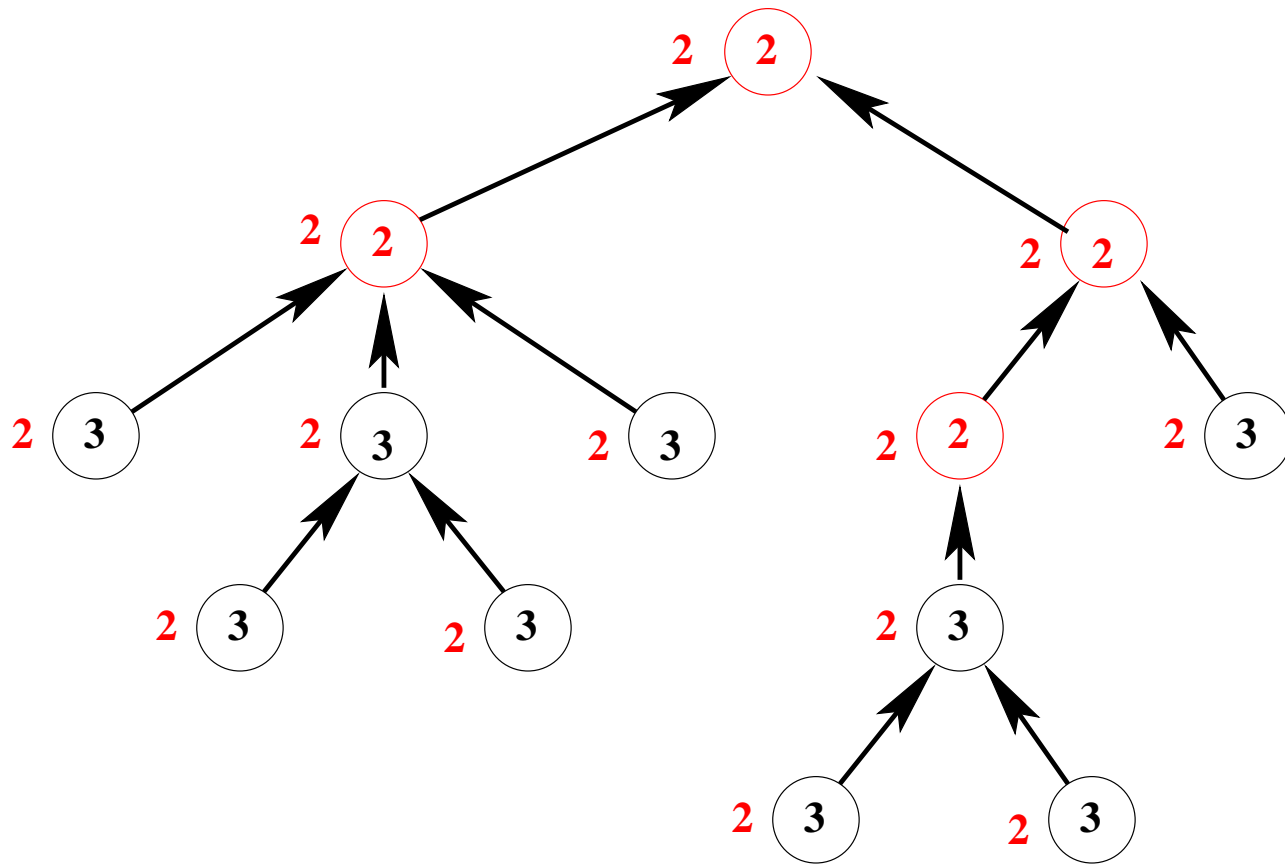


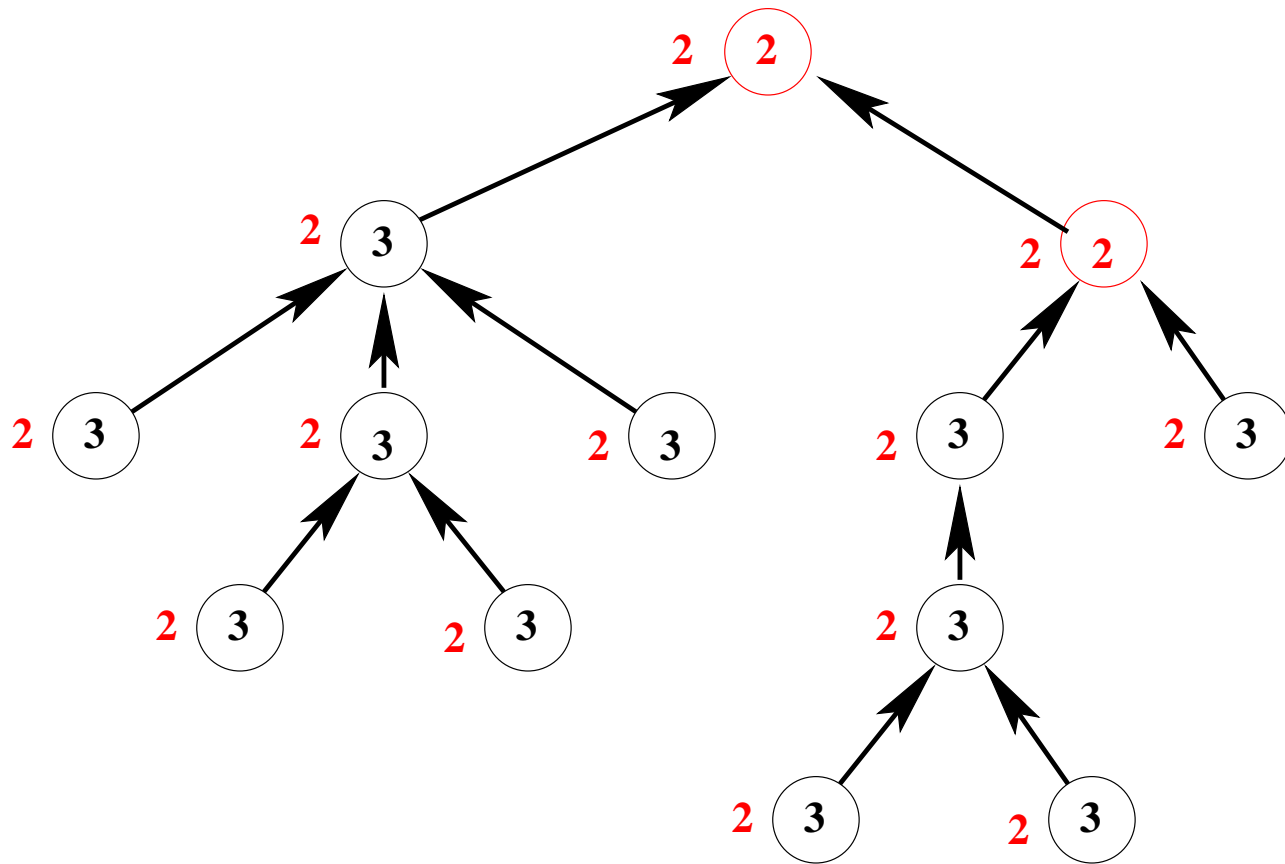


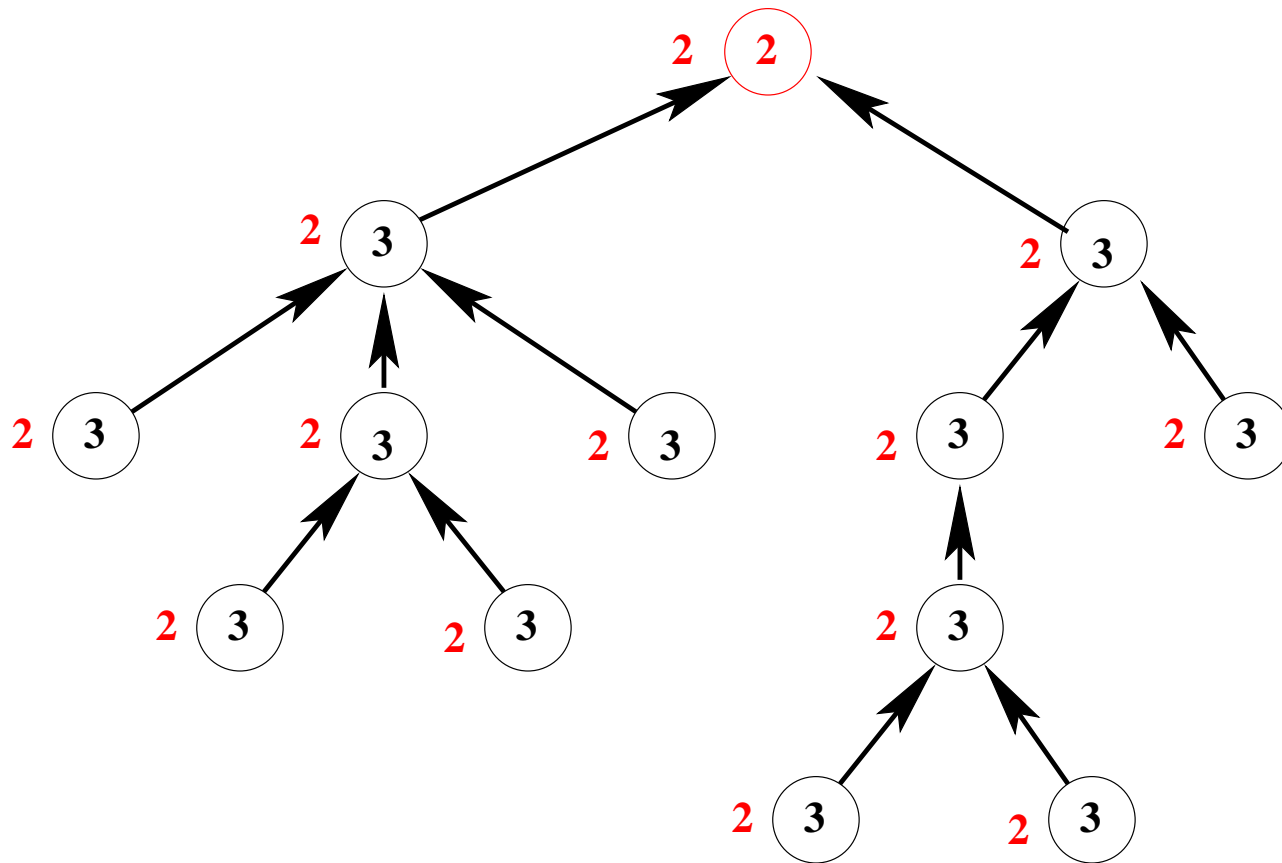


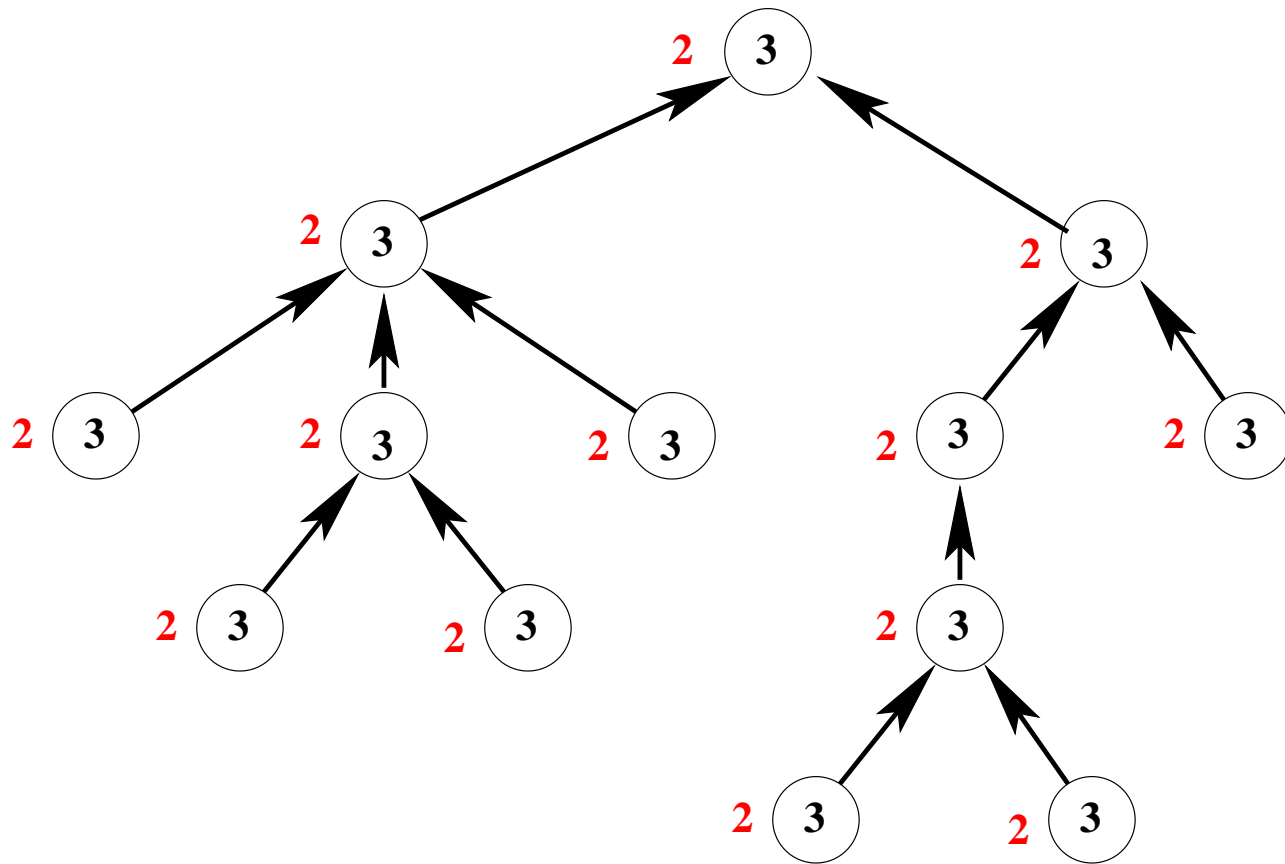












- We replace the counters by their parities
- we add the constant *parity*

carrier set: P

constants: r, f, \textit{parity}

axm4_1: $\textit{parity} \in \mathbb{N} \rightarrow \{0, 1\}$

axm4_2: $\textit{parity}(0) = 0$

axm4_2: $\forall x . (x \in \mathbb{N} \Rightarrow \textit{parity}(x + 1) = 1 - \textit{parity}(x))$

$$\mathbf{thm4_1:} \quad \forall x, y . \left(\begin{array}{l} x \in \mathbb{N} \\ y \in \mathbb{N} \\ x \in y .. y + 1 \\ \Rightarrow \\ \mathit{parity}(x) = \mathit{parity}(y) \Leftrightarrow x = y \end{array} \right)$$

- We replace c and d by p and q

variables: p, q

inv4_1: $p \in P \rightarrow \{0, 1\}$

inv4_2: $q \in P \rightarrow \{0, 1\}$

inv4_3: $\forall n . (n \in P \Rightarrow p(n) = \textit{parity}(c(n)))$

inv4_4: $\forall n . (n \in P \Rightarrow q(n) = \textit{parity}(d(n)))$

```
ascending
  any  $n$  where
     $n \in P$ 
     $p(n) = q(n)$ 
     $\forall m \cdot ( m \in f^{-1}[\{n\}] \Rightarrow p(m) \neq p(n) )$ 
  then
     $p(n) := 1 - p(n)$ 
  end
```

```
descending_1
  any  $n$  where
     $n \in P \setminus \{r\}$ 
     $q(n) \neq q(f(n))$ 
  then
     $q(n) := 1 - q(n)$ 
  end
```

```
descending_2
  when
     $p(r) \neq q(r)$ 
  then
     $q(r) := 1 - q(r)$ 
  end
```

Models	Total	Interactive
Initial Model	7	0
Refinement 1	12	0
Refinement 2	7	0
Refinement 3	28	0
Refinement 4	23	0
TOTAL	77	0