Transition system

Definition 1.3.1 A ***transition system*** S is a pair of the form

$$S = (C, T)$$

where C is the set of ***configurations*** and $T \subseteq C \times C$ is a relation, the ***transition relation***. $\square$

Sequences generated by a transition system

Definition 1.3.3 Let $S = (C, T)$ be a transition system. S ***generates a set of sequences***, $\mathcal{S}(S)$, defined as follows:

1. the finite sequence $c_0, c_1, \ldots, c_n$ (for $n \geq 0$) belongs to $\mathcal{S}(S)$ if
   - (ι) $c_0 \in C$
   - ($\iota\iota$) for all i with $1 \leq i \leq n$: $(c_{i-1}, c_i) \in T$
2. the infinite sequence $c_0, c_1, \ldots, c_n, \ldots$ belongs to $\mathcal{S}(S)$ if
   - (ι) $c_0 \in C$
   - ($\iota\iota$) for all $i \geq 1$: $(c_{i-1}, c_i) \in T$

$\square$

Processes generated by a transition system

Definition 1.3.5 Let $S = (C, T)$ be a transition system. The set of ***processes generated by*** S, written $\mathcal{P}(S)$, is the subset of $\mathcal{S}(S)$ containing

1. all infinite sequences of $\mathcal{S}(S)$
2. all finite sequences $c_0, c_1, \ldots, c_n$ $(n \geq 0)$ of $\mathcal{S}(S)$ for which it holds that there is no $c \in C$ with $(c_n, c) \in T$.

The final configuration of a finite process is called a ***dead configuration***. □

Football

Transition system Football

Configurations: $\{[t, X, a, b] \mid 0 \leq t \leq 90, X \in \{A, B, R\}, a, b \in \mathbf{N}\}$

$$
\begin{array}{lcll}
[t, A, a, b] & \triangleright & [t+2, B, a, b] & \textbf{if } t \leq 88 \\
[t, A, a, b] & \triangleright & [t+2, B, a+1, b] & \textbf{if } t \leq 88 \\
[t, A, a, b] & \triangleright & [t+1, B, a, b] & \textbf{if } t \leq 89 \\
[t, A, a, b] & \triangleright & [t+1, B, a+1, b] & \textbf{if } t \leq 89 \\
[90, A, a, b] & \triangleright & [90, R, a, b] & \\
[t, B, a, b] & \triangleright & [t+2, A, a, b] & \textbf{if } t \leq 88 \\
[t, B, a, b] & \triangleright & [t+2, A, a, b+1] & \textbf{if } t \leq 88 \\
[t, B, a, b] & \triangleright & [t+1, A, a, b] & \textbf{if } t \leq 89 \\
[t, B, a, b] & \triangleright & [t+1, A, a, b+1] & \textbf{if } t \leq 89 \\
[90, B, a, b] & \triangleright & [90, R, a, b] &
\end{array}
$$

Induction principle

Induction principle Let $P(0), P(1), \ldots, P(n), \ldots$ be statements. If

a) $P(0)$ is true

b) for all $n \geq 0$ it holds that $P(n)$ implies $P(n+1)$,

then $P(n)$ is true for all $n \geq 0$.

Invariance principle

Invariance principle for transition systems Let $S = (C, T)$ be a transition system and let $c_0 \in C$ be a configuration. If $I(c)$ is a statement about the configurations of the system, the following holds. If

a) $I(c_0)$ is true

b) for all $(c, c') \in T$ it holds that $I(c)$ implies $I(c')$

then $I(c)$ is true for any configuration c that occurs in a sequence starting with c_0.

Termination principle

Termination principle for transition systems Let $S = (C, T)$ be a transition system and let $\mu : C \to \mathbf{N}$ be a function. If

$$\text{for all } (c, c') \in T \text{ it holds that } \mu(c) > \mu(c')$$

then all processes in $\mathcal{P}(S)$ are finite.

Nim

Transition system Nim

Configurations: $\{A, B\} \times \mathbf{N}$

$$
\begin{array}{lll}
[A, n] \triangleright [B, n-2] & \textbf{if } n \geq 2 \\
[A, n] \triangleright [B, n-1] & \textbf{if } n \geq 1 \\
[B, n] \triangleright [A, n-2] & \textbf{if } n \geq 2 \\
[B, n] \triangleright [A, n-1] & \textbf{if } n \geq 1
\end{array}
$$

Towers of Hanoi

Transition system $\mathsf{Hanoi}(n)$

Configurations: $\{[A,B,C] \mid \{A,B,C\} \text{ a partition of } \{1,\ldots,n\}\}$

$$
\begin{array}{lll}
[A,B,C] & \triangleright & [A\setminus\{r\}, B\cup\{r\}, C] \;\; \textbf{if } (r = \min A) \wedge (r < \min B) \\
[A,B,C] & \triangleright & [A\setminus\{r\}, B, C\cup\{r\}] \;\; \textbf{if } (r = \min A) \wedge (r < \min C) \\
[A,B,C] & \triangleright & [A\cup\{r\}, B\setminus\{r\}, C] \;\; \textbf{if } (r = \min B) \wedge (r < \min A) \\
[A,B,C] & \triangleright & [A, B\setminus\{r\}, C\cup\{r\}] \;\; \textbf{if } (r = \min B) \wedge (r < \min C) \\
[A,B,C] & \triangleright & [A\cup\{r\}, B, C\setminus\{r\}] \;\; \textbf{if } (r = \min C) \wedge (r < \min A) \\
[A,B,C] & \triangleright & [A, B\cup\{r\}, C\setminus\{r\}] \;\; \textbf{if } (r = \min C) \wedge (r < \min B)
\end{array}
$$

Euclid's algorithm

Transition system Euclid

Configurations: $\{[m,n] \mid m,n \geq 1\}$

$[m,n] \ \triangleright \ [m-n,n] \quad \textbf{if } m > n$

$[m,n] \ \triangleright \ [m,n-m] \quad \textbf{if } m < n$

Expressions

Transition system Expressions

Configurations: $\{0, 1, +, \mathbf{E}, \mathbf{T}, (,)\}^*$

$\alpha\mathbf{E}\beta \;\triangleright\; \alpha\mathbf{T}\beta$

$\alpha\mathbf{E}\beta \;\triangleright\; \alpha\mathbf{T}{+}\mathbf{E}\beta$

$\alpha\mathbf{T}\beta \;\triangleright\; \alpha 0\beta$

$\alpha\mathbf{T}\beta \;\triangleright\; \alpha 1\beta$

$\alpha\mathbf{T}\beta \;\triangleright\; \alpha(\mathbf{E})\beta$

Expressions (context-free)

Transition system Expressions

Configurations: $\{0, 1, +, \mathbf{E}, \mathbf{T}, (,)\}^*$

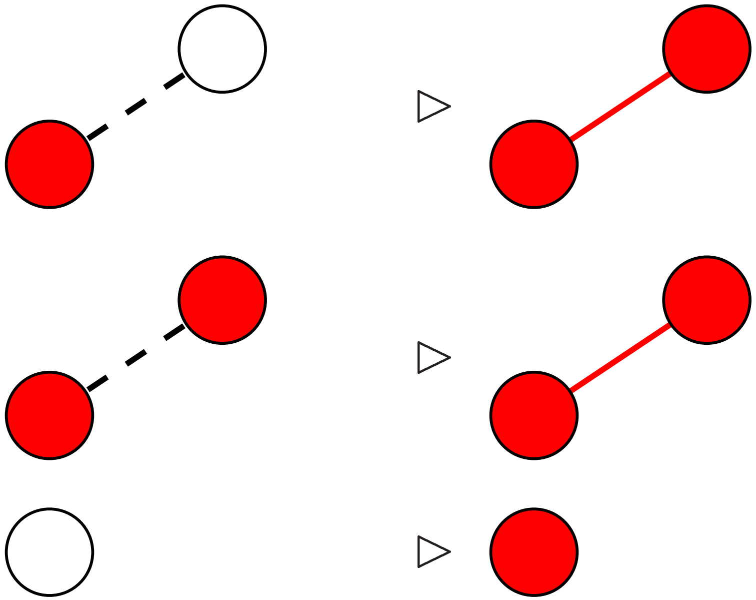
$$\mathbf{E} \;\triangleright\; \mathbf{T}, \mathbf{T}{+}\mathbf{E}$$

$$\mathbf{T} \;\triangleright\; 0, 1, (\mathbf{E})$$

Graph coloring

Transition system GraphColoring

Configurations: Danish graphs



if there are no pink nodes

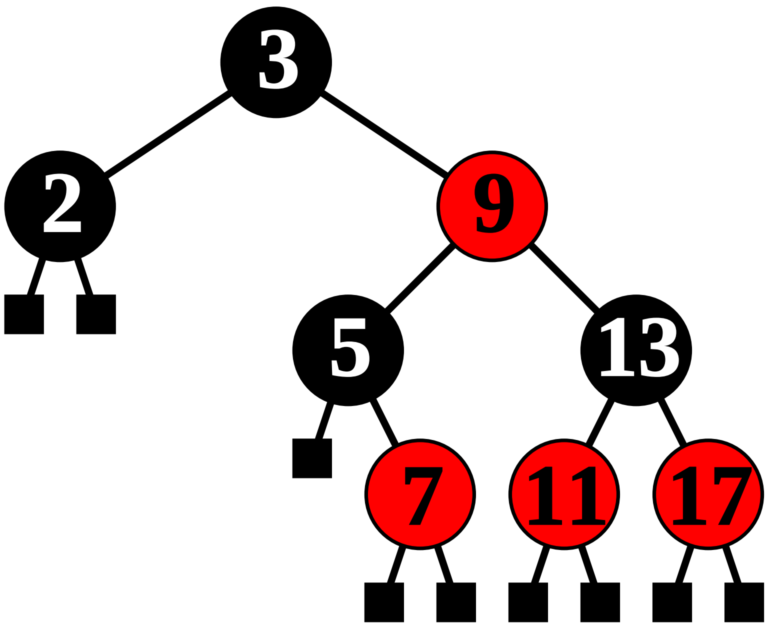
Red-black tree

Definition 1.5.7 A ***red-black tree*** is binary search tree in which all internal nodes are colored either red or black, in which the leaves are black, and

Invariant I_2 Each red node has a black parent.

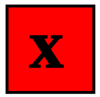
Invariant I_3 There is the same number of black nodes on all paths from the root to a leaf.

□



Insertion

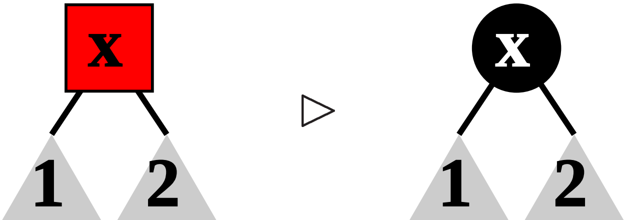
Illegitimate red node:



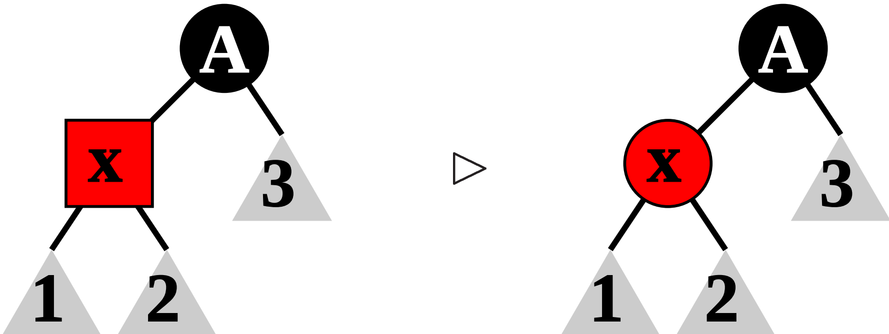
Invariant I_2'**:** Each *legitimate* red node has a black parent.

Insertion: transitions 1 and 2

The illegitimate node is the root of the tree:

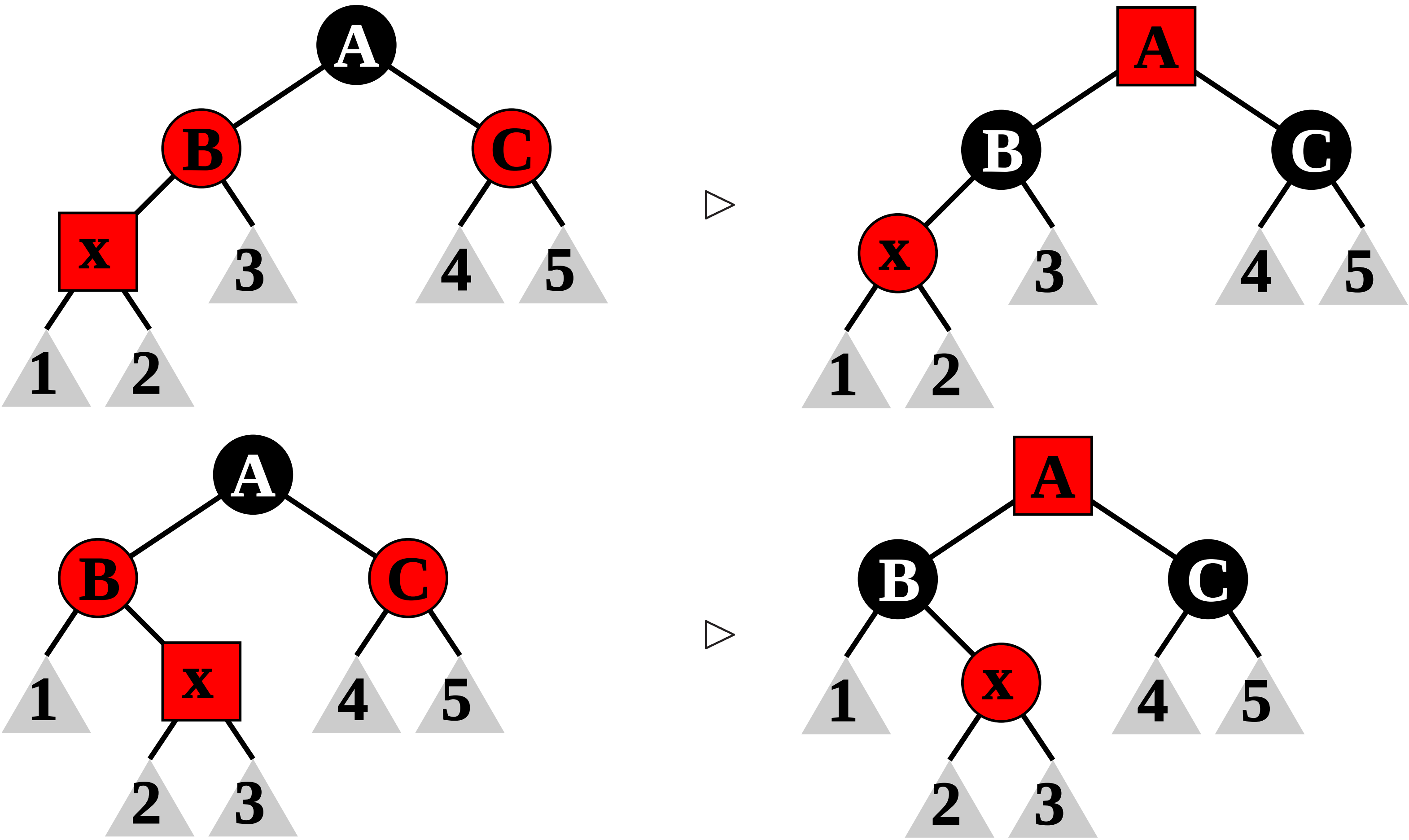


The illegitimate node has a black father:



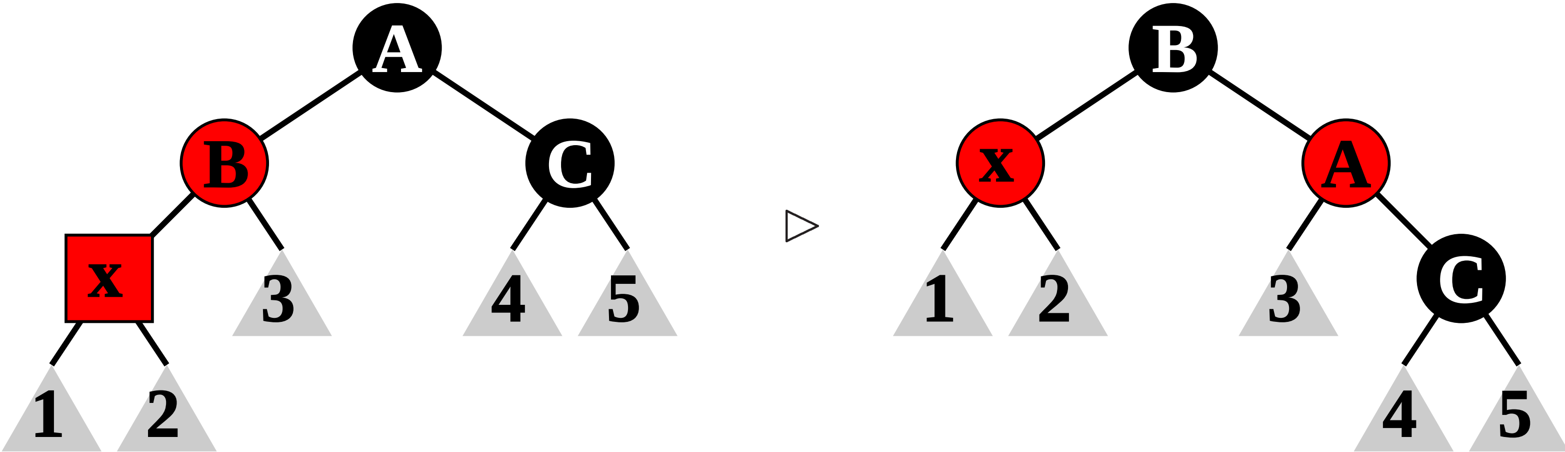
Insertion: transitions 3.1 and 3.2

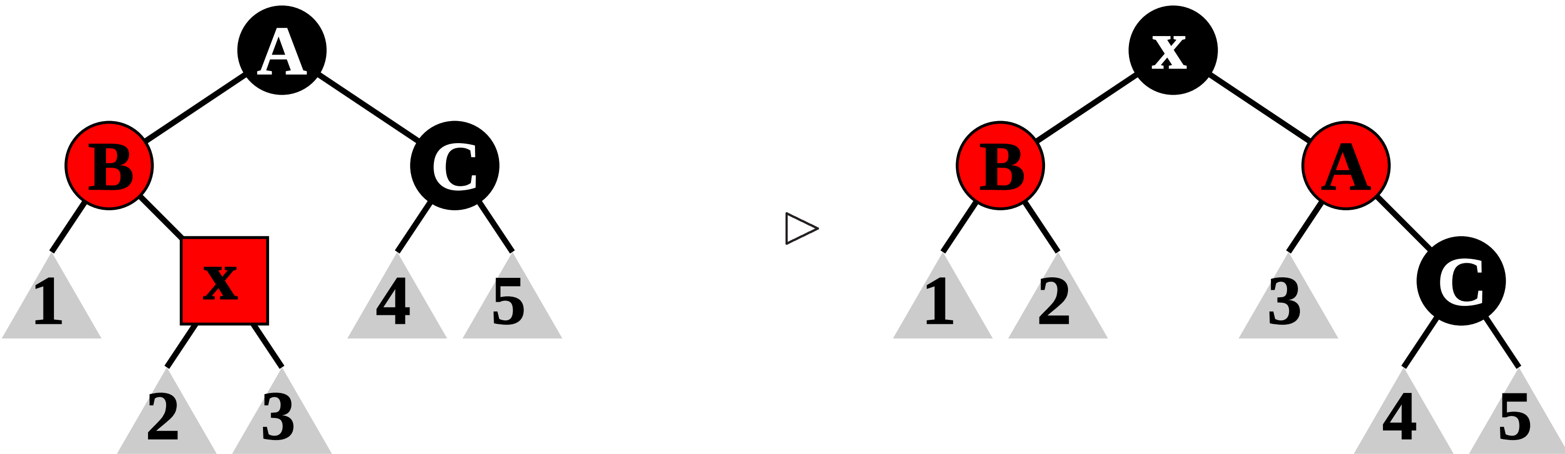
The illegitimate node has a red father and a red uncle:



Insertion: transitions 4.1 and 4.2

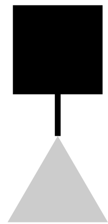
The illegitimate node has a red father and a black uncle:





Deletion

Illegitimate black node:

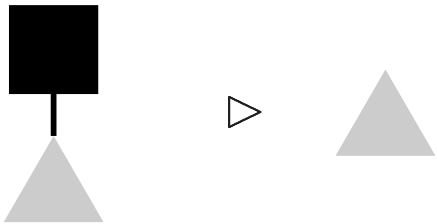


Invariant I_1' The tree satisfies I_1 if we remove the illegitimate node.

Invariant I_2' Each red node has a *legitimate* black father.

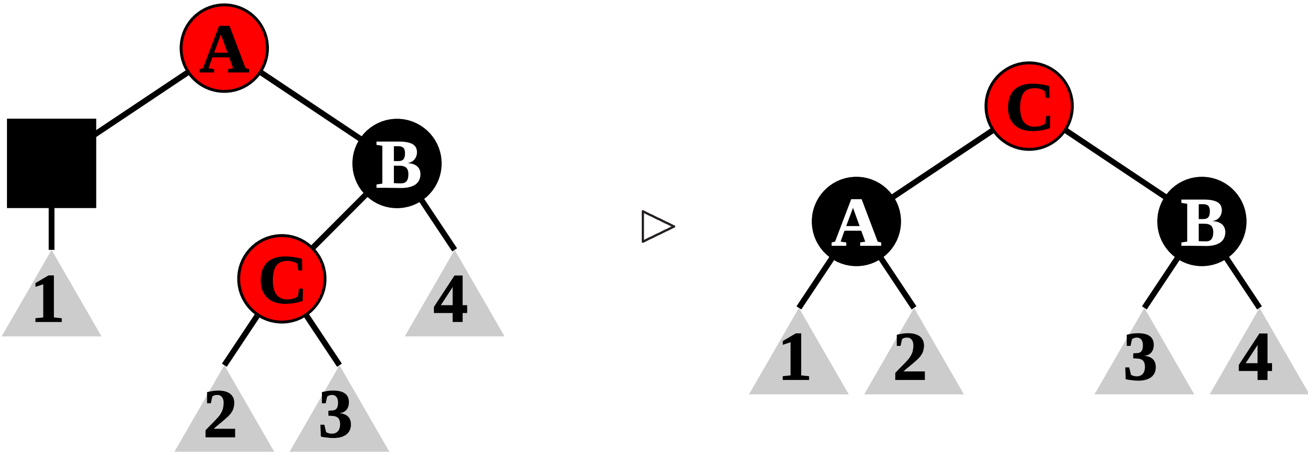
Deletion: transition 1

The illegitimate node is the root:

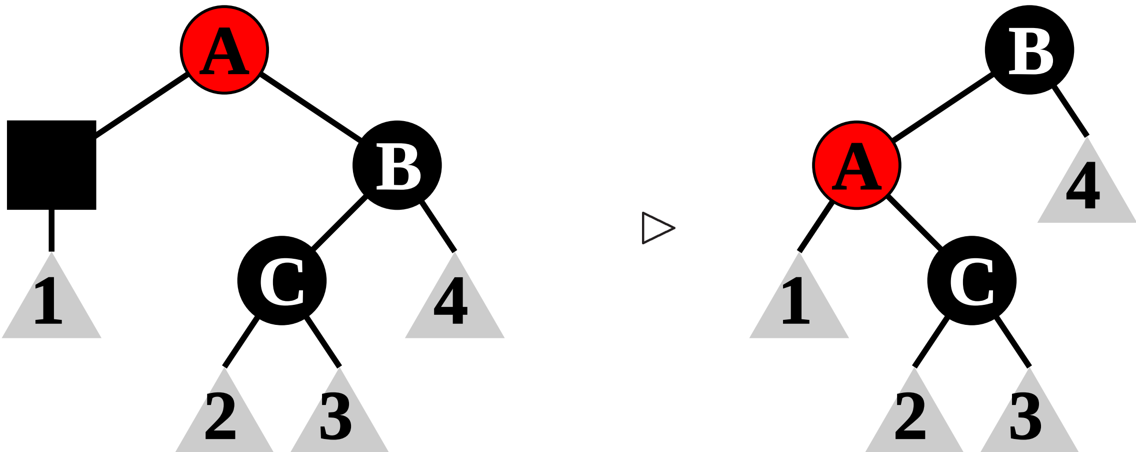


Deletion: transitions 2 and 3

The illegitimate node has a red father and a red closer nephew:

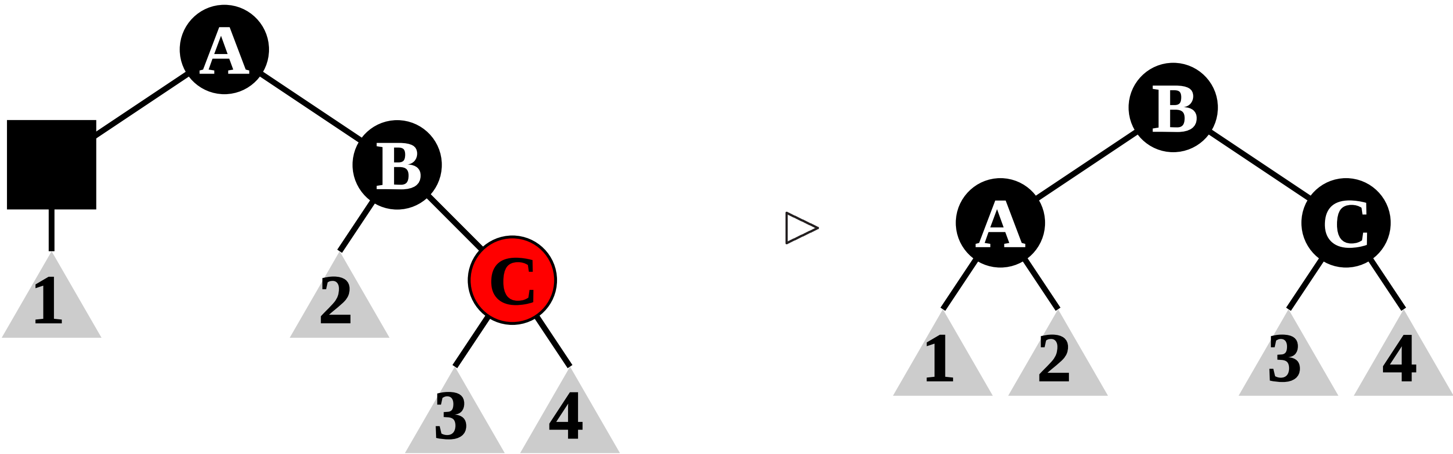


The illegitimate node has a red father and a black closer nephew:



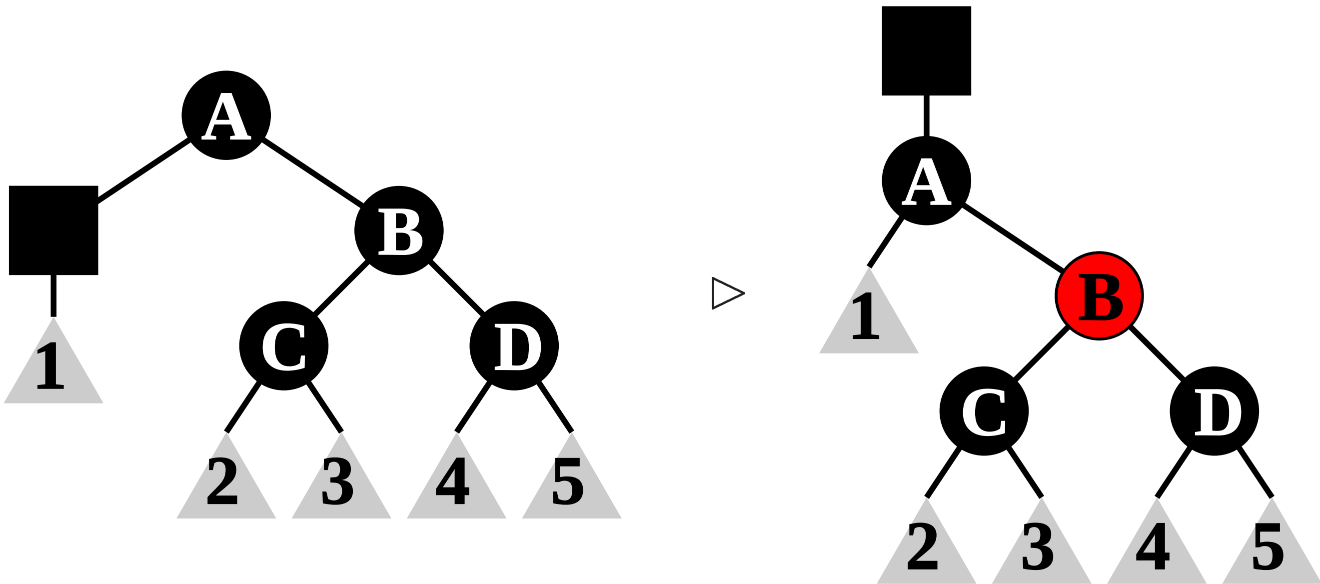
Deletion: transitions 4.1 and 4.2

The illegitimate node has a black father, a black sibling and one red nephew:



Deletion: transition 5

The illegitimate node has a black father, a black sibling and two black nephews



Deletion: transition 6

The illegitimate node has a black father and a red sibling: