THEORETICAL COMPUTER SCIENCE TUTORING (2)

Maurizio Fiusco

TOR VERGATA
UNIVERSITÀ DEGLI STUDI DI ROMA

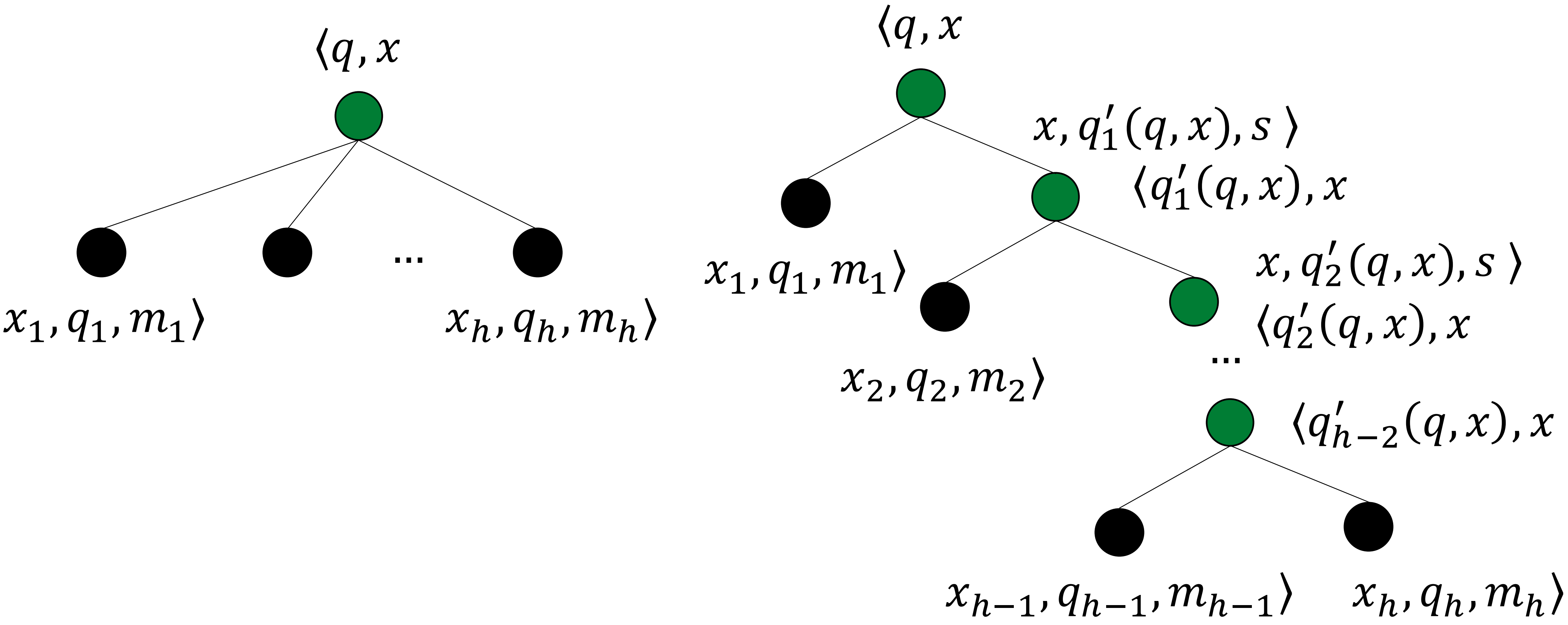
Problem 2.6 from [EsMacchineTuring.pdf (uniroma2.it)]

Let k be a constant in $\mathbb{N}$, and let $\boldsymbol{NT_k}$ be a non-deterministic Turing machine with a degree of non-determinism equal to $\boldsymbol{k}$. Define a non-deterministic Turing machine $\boldsymbol{NT_2}$ with a degree of non-determinism equal to 2 that is equivalent to $\boldsymbol{NT_k}$

$$P_k(q,x) = \langle q,x,x_1,q_1,m_1\rangle, \langle q,x,x_2,q_2,m_2\rangle, \dots, \langle q,x,x_h,q_h,m_h\rangle \qquad h \le k$$

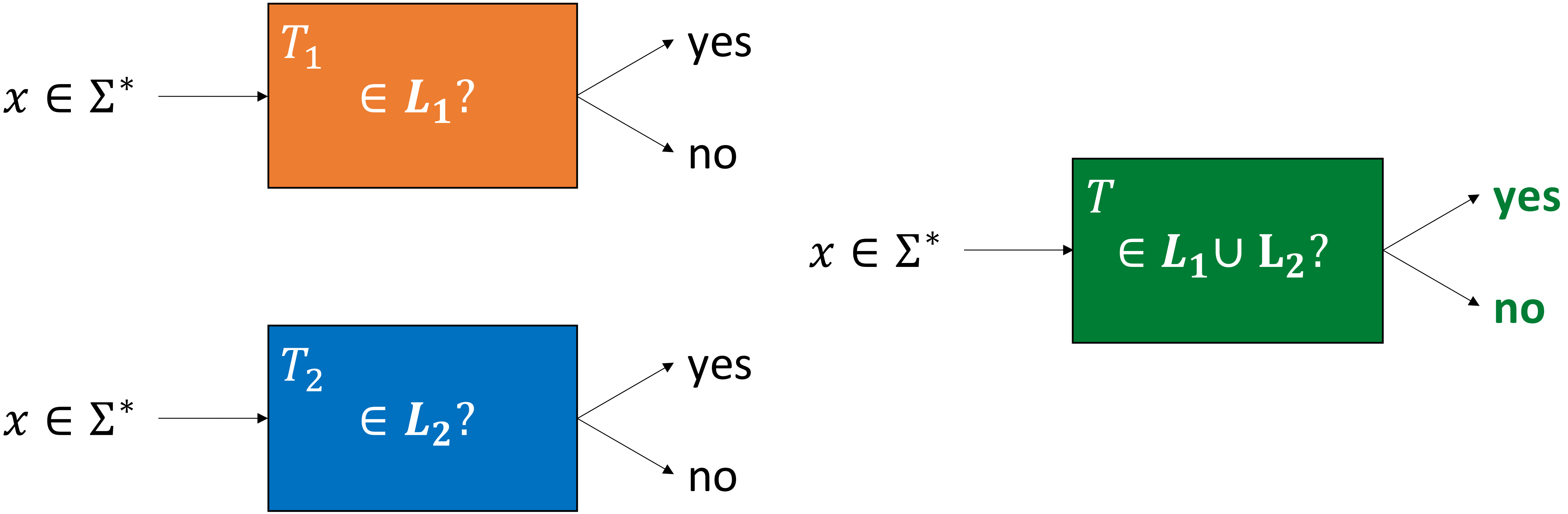
Problem 2.6 from [EsMacchineTuring.pdf (uniroma2.it)]

$$P_k(q,x) = \langle q,x,x_1,q_1,m_1\rangle, \langle q,x,x_2,q_2,m_2\rangle, \ldots, \langle q,x,x_h,q_h,m_h\rangle$$



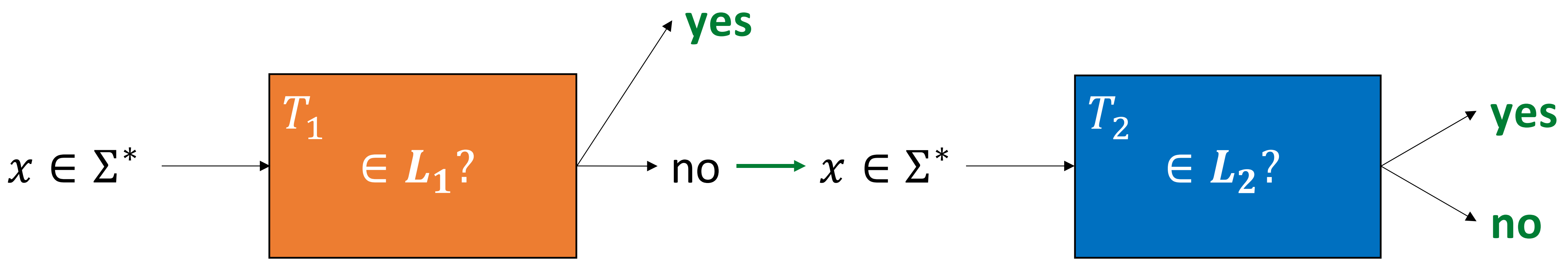
Problem 3.1 from EsLinguaggiDecidibili.pdf (uniroma2.it)

Let Σ be a finite alphabet and let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ be two **decidable** languages. Show that $L1 \cup L2$ is **decidable**.



Problem 3.1 from [EsLinguaggiDecidibili.pdf (uniroma2.it)]

We can use T_1 and T_2 as “black boxes”



We want to prove that:

$\boldsymbol{T}$ terminates for every input x, and furthermore, it terminates in the accepting state if and only if $x \in \boldsymbol{L1}$ or $x \in \boldsymbol{L2}$, that is, if and only if $x \in \boldsymbol{L1} \cup \boldsymbol{L2}$.

Problem 3.1 from EsLinguaggiDecidibili.pdf (uniroma2.it)

Let's define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

$x \in \Sigma^*$

$x \in \Sigma^*$

1. $\boldsymbol{T}$ simulates $\mathbf{T_1(x)}$ on the first tape:
   - $\mathbf{T_1(x)}$ ends in the accepting state ✅ $\boldsymbol{T}$ accepts
   - $\mathbf{T_1(x)}$ ends in the rejecting state 🤷 $\boldsymbol{T}$ begins **phase-2**

Problem 3.1 from EsLinguaggiDecidibili.pdf (uniroma2.it)

Let's define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

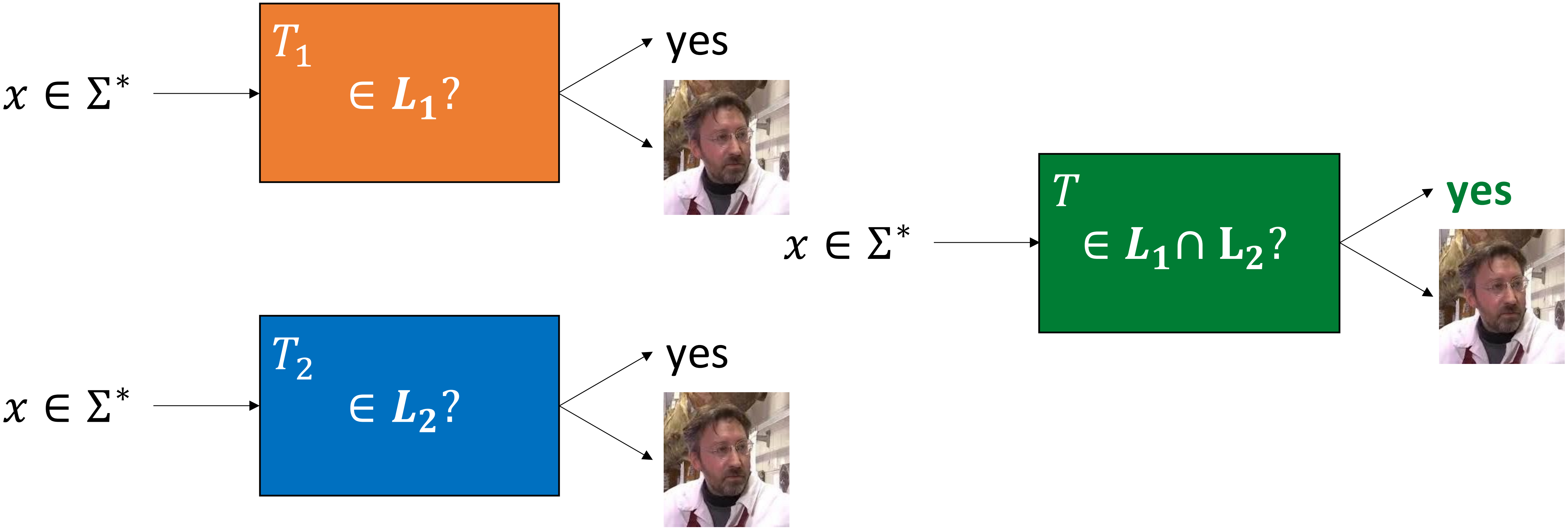
$x \in \Sigma^*$

$x \in \Sigma^*$

2. $\boldsymbol{T}$ simulates $\mathbf{T_2(x)}$ on the second tape:
   - $\mathbf{T_2(x)}$ ends in the accepting state ✓ $\boldsymbol{T}$ accepts
   - $\mathbf{T_2(x)}$ ends in the rejecting state ✗ $\boldsymbol{T}$ rejects

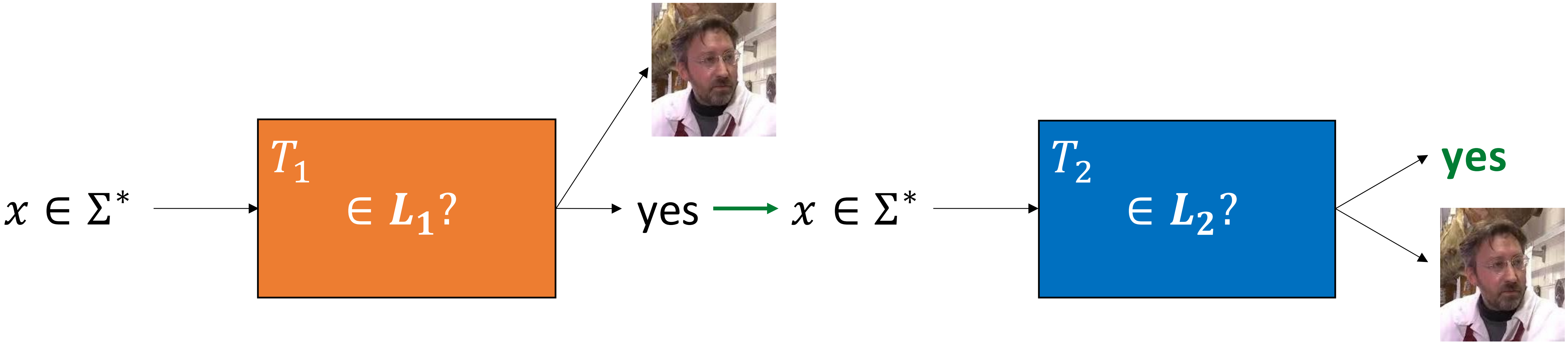
Problem 3.2 from EsLinguaggiDecidibili.pdf (uniroma2.it)

Let Σ be a finite alphabet and let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ be two **acceptable** languages. Show that $L1 \cap L2$ is **acceptable**.



Problem 3.2 from [EsLinguaggiDecidibili.pdf (uniroma2.it)]

We can use T_1 and T_2 as “black boxes”



We want to prove that:
$T(x)$ terminates in the accepting state if and only if $x \in L1 \cap L2$, It is explicitly noted that nothing can be said about the outcome of the computation of $T(x)$ for $x \notin L1 \cap L2$.

Problem 3.2 from [EsLinguaggiDecidibili.pdf (uniroma2.it)]()

Let's define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

$x \in \Sigma^*$

$x \in \Sigma^*$

1. $\boldsymbol{T}$ simulates $\mathbf{T_1(x)}$ on the first tape:
   - $\mathbf{x} \in \boldsymbol{L_1} \Leftrightarrow \mathbf{T_1(x)}$ ends in the accepting state ✔ $\boldsymbol{T}$ begins **phase-2**
   - $\mathbf{T_1(x)}$ ends in the rejecting state ✘ $\boldsymbol{T}$ rejects
   - $\mathbf{T_1(x)}$ doesn't terminate 😵 $\boldsymbol{T(x)}$ doesn't terminate

Problem 3.2 from EsLinguaggiDecidibili.pdf (uniroma2.it)

Let's define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

$x \in \Sigma^*$

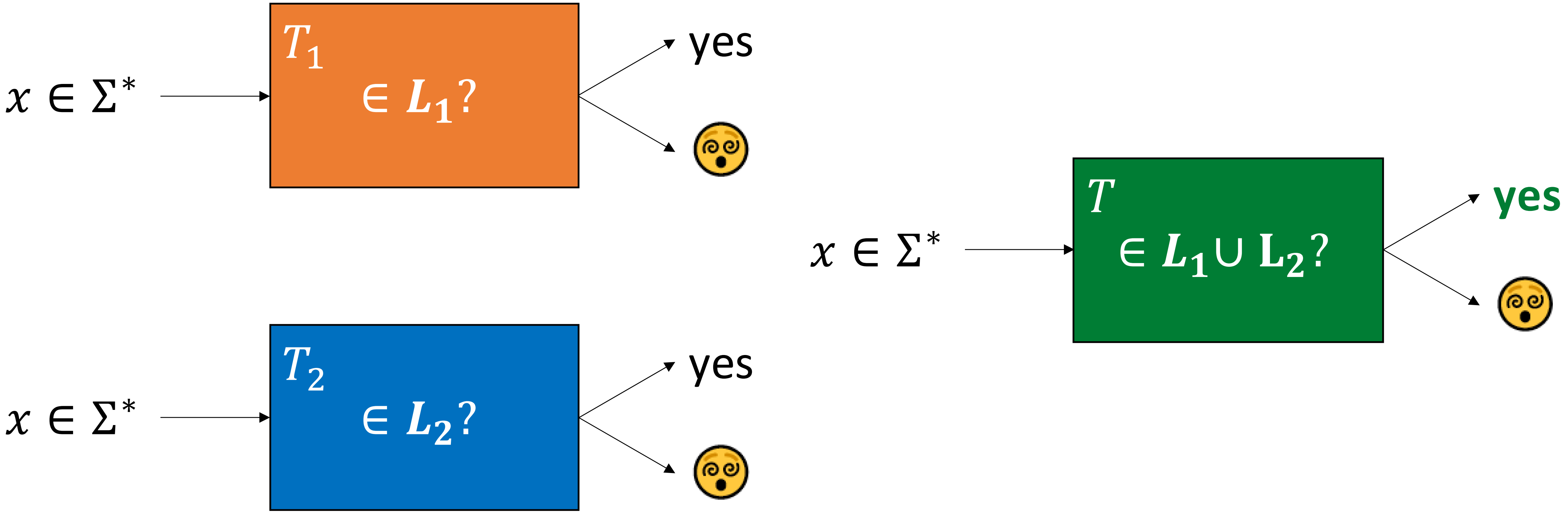
$x \in \Sigma^*$

2. $\boldsymbol{T}$ simulates $\mathbf{T_2(x)}$ on the first tape:

- $\mathbf{x} \in \boldsymbol{L_2} \Leftrightarrow \mathbf{T_2(x)}$ ends in the accepting state ✔ $\boldsymbol{T}$ accepts
- $\mathbf{T_2(x)}$ ends in the rejecting state ✘ $\boldsymbol{T}$ rejects
- $\mathbf{T_2(x)}$ doesn't terminate 😵 $\boldsymbol{T(x)}$ doesn't terminate

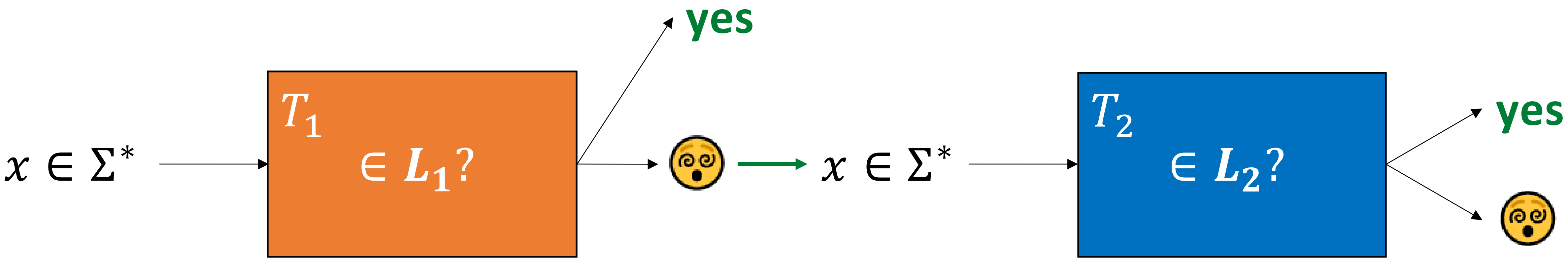
Exercise 2 from Lezione06-macchineLinguaggiFunzioni.pptx (slide 16)

Let Σ be a finite alphabet and let $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ be two **acceptable** languages. Show that $L1 \cup L2$ is **acceptable**.

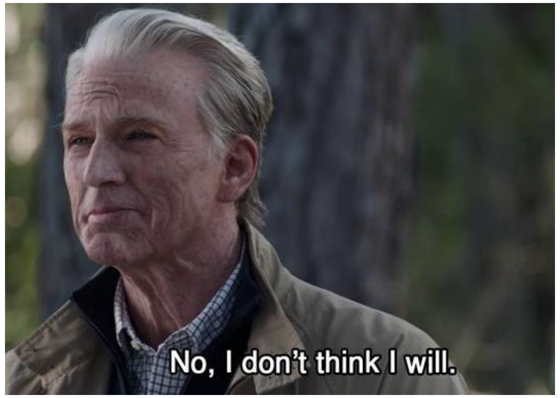


Exercise 2 from Lezione06-macchineLinguaggiFunzioni.pptx (slide 16)

Can we use T_1 and T_2 as "black boxes"?



We want to prove that:
$\boldsymbol{T}$ terminates (in the accepting state) for every input $\boldsymbol{x} \in \boldsymbol{L_1} \cup \boldsymbol{L_2}$



Exercise 2 from Lezione06-macchineLinguaggiFunzioni.pptx (slide 16)

Let's define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

$x \in \Sigma^*$

$x \in \Sigma^*$

1. $\boldsymbol{T}$ executes a single instruction of $\boldsymbol{T_1}$ on the first tape
   - $\boldsymbol{T_1}$ halts in accepting state $\Leftrightarrow \boldsymbol{x \in L_1}$ ✔️ $\boldsymbol{T}$ accepts
   - $\boldsymbol{T_1}$ halts in rejecting state $\Longrightarrow \boldsymbol{x \notin L_1}$ 🤷‍♀️ **2 or** ❌ if $\boldsymbol{T_2}$ has rejected
   - $\boldsymbol{T_1}$ doesn't halt 🤷‍♀️ **2**

Exercise 2 from [Lezione06-macchineLinguaggiFunzioni.pptx]() (slide 16)

Let’s define $\boldsymbol{T}$

$$\boldsymbol{Q} = \boldsymbol{Q_1} \cup \boldsymbol{Q_2} \cup \{\boldsymbol{q_{A_T}}, \boldsymbol{q_{R_T}}\}$$

$x \in \Sigma^*$

$x \in \Sigma^*$

2. $\boldsymbol{T}$ executes a single instruction of $\boldsymbol{T_2}$ on the first tape
   - $\boldsymbol{T_2}$ halts in accepting state $\Leftrightarrow \boldsymbol{x \in L_2}$ ✔️ $\boldsymbol{T}$ accepts
   - $\boldsymbol{T_2}$ halts in rejecting state $\Longrightarrow \boldsymbol{x \notin L_2}$ 🤷‍♀️ **1 or** ❌ if $\boldsymbol{T_1}$ has rejected
   - $\boldsymbol{T_2}$ doesn’t halt 🤷‍♀️ **1**

Problem 1 from the exam held on July 4, 2019

Remember how Turing machines can be encoded as integers. Let $f: \mathbb{N} \rightarrow \mathbb{N}$ be a function defined as follows:

$$\boldsymbol{f(i)} = \begin{cases} \mathbf{0 \text{ if } \boldsymbol{i} \text{ is the encoding of the Turing machine}} \\ \mathbf{1 \text{ if } \boldsymbol{i} \text{ isn't the encoding of the Turing machine}} \end{cases}$$

After **defining the concept of computability** of a function, **discuss the computability of $\boldsymbol{f(n)}$** by demonstrating your claims.