Chapter 2 : Turing machines

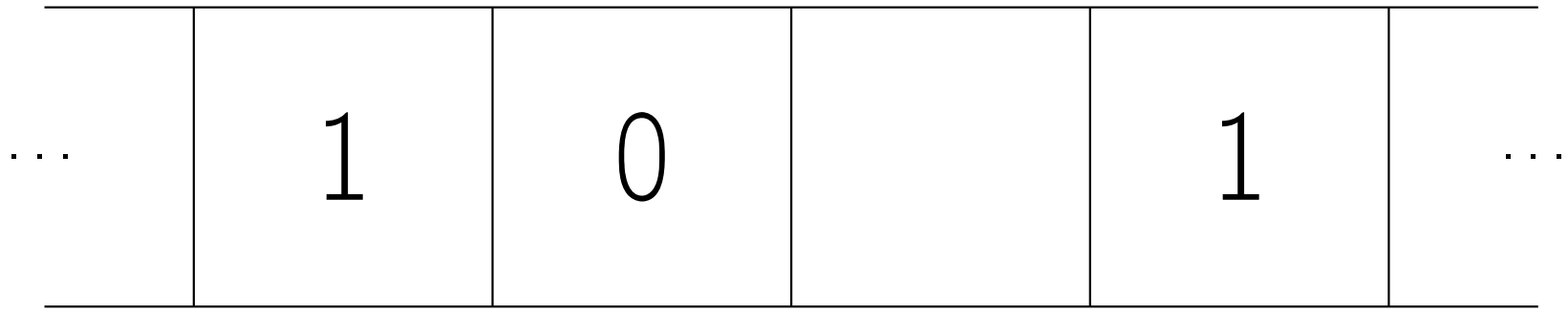
ENSIIE - Computational complexity theory

Dimitri Watel (dimitri.watel@ensiie.fr)

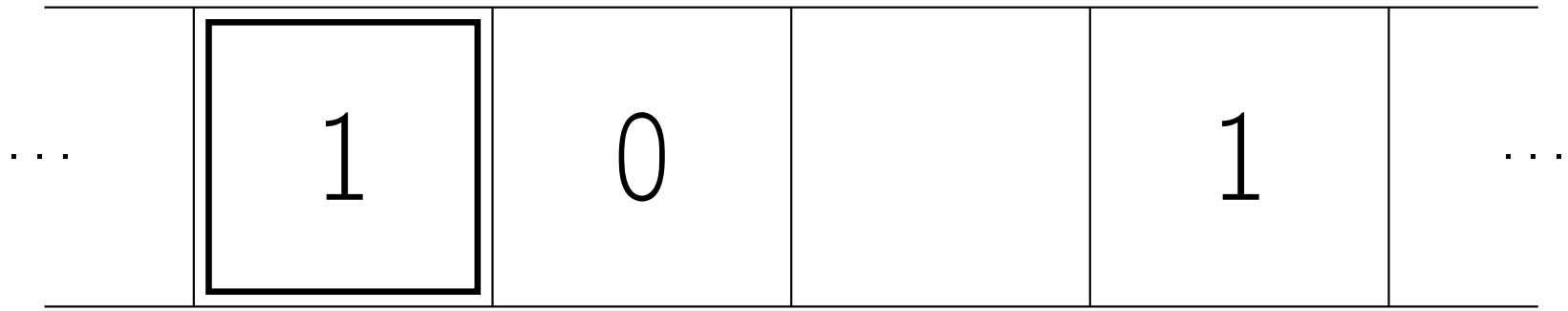
2022

Turing machine

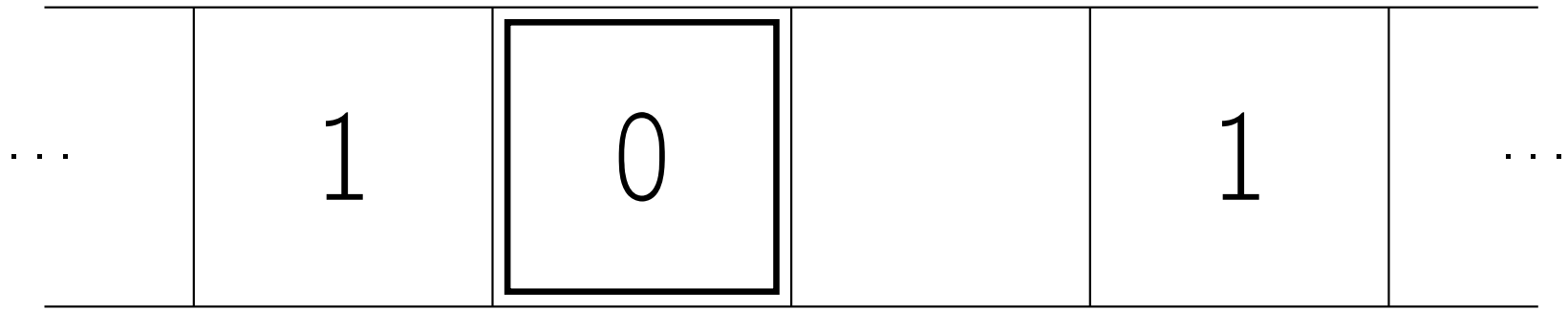
Turing machine - the tape



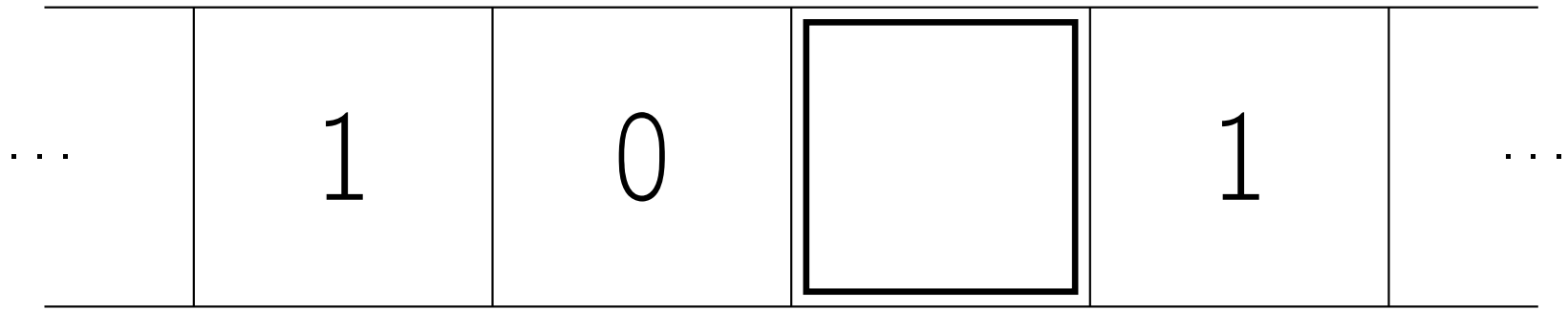
Turing machine - the head



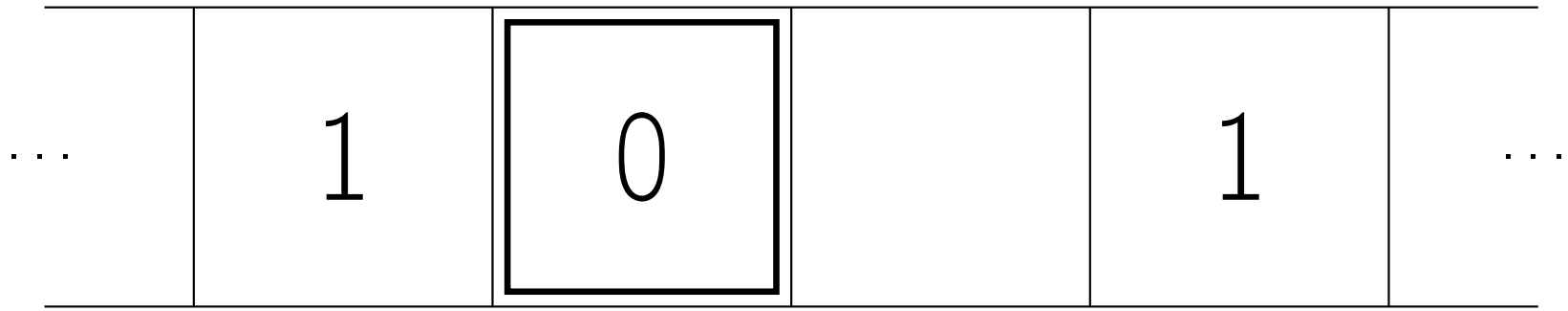
Turing machine - the head



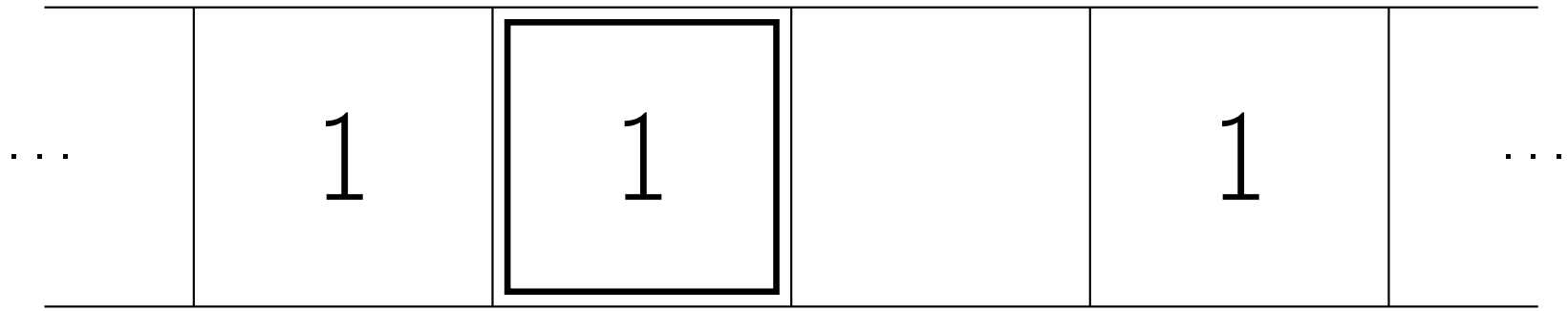
Turing machine - the head



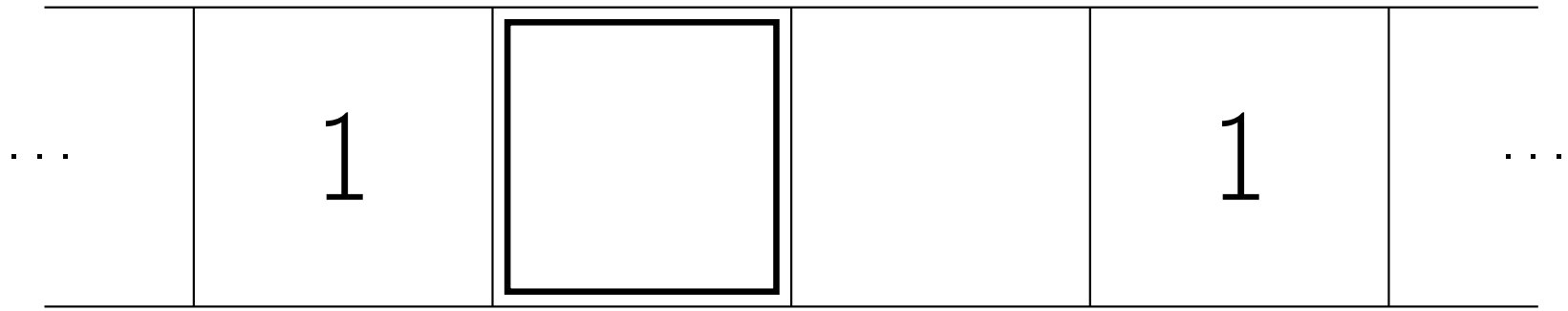
Turing machine - the head



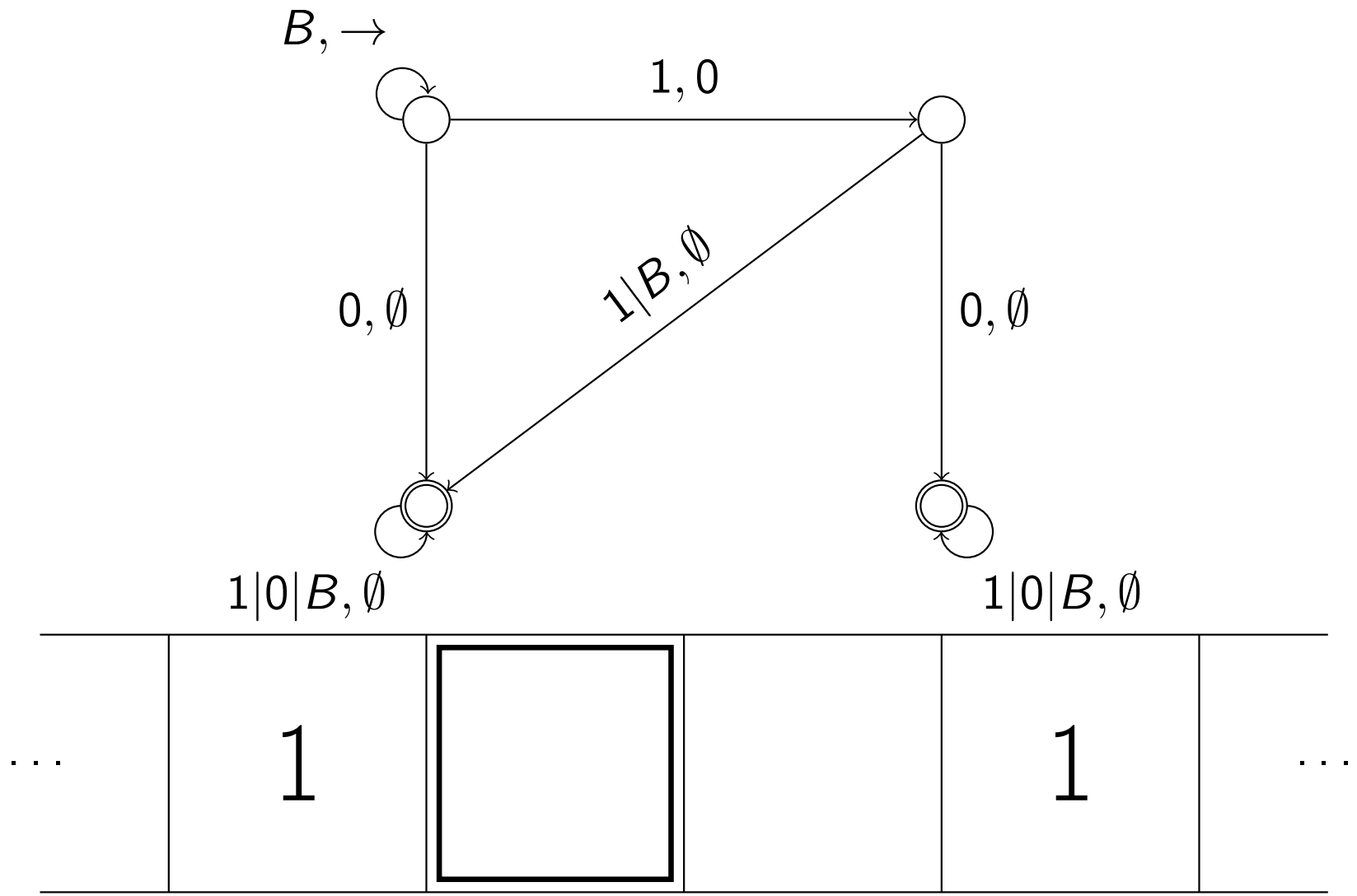
Turing machine - the head



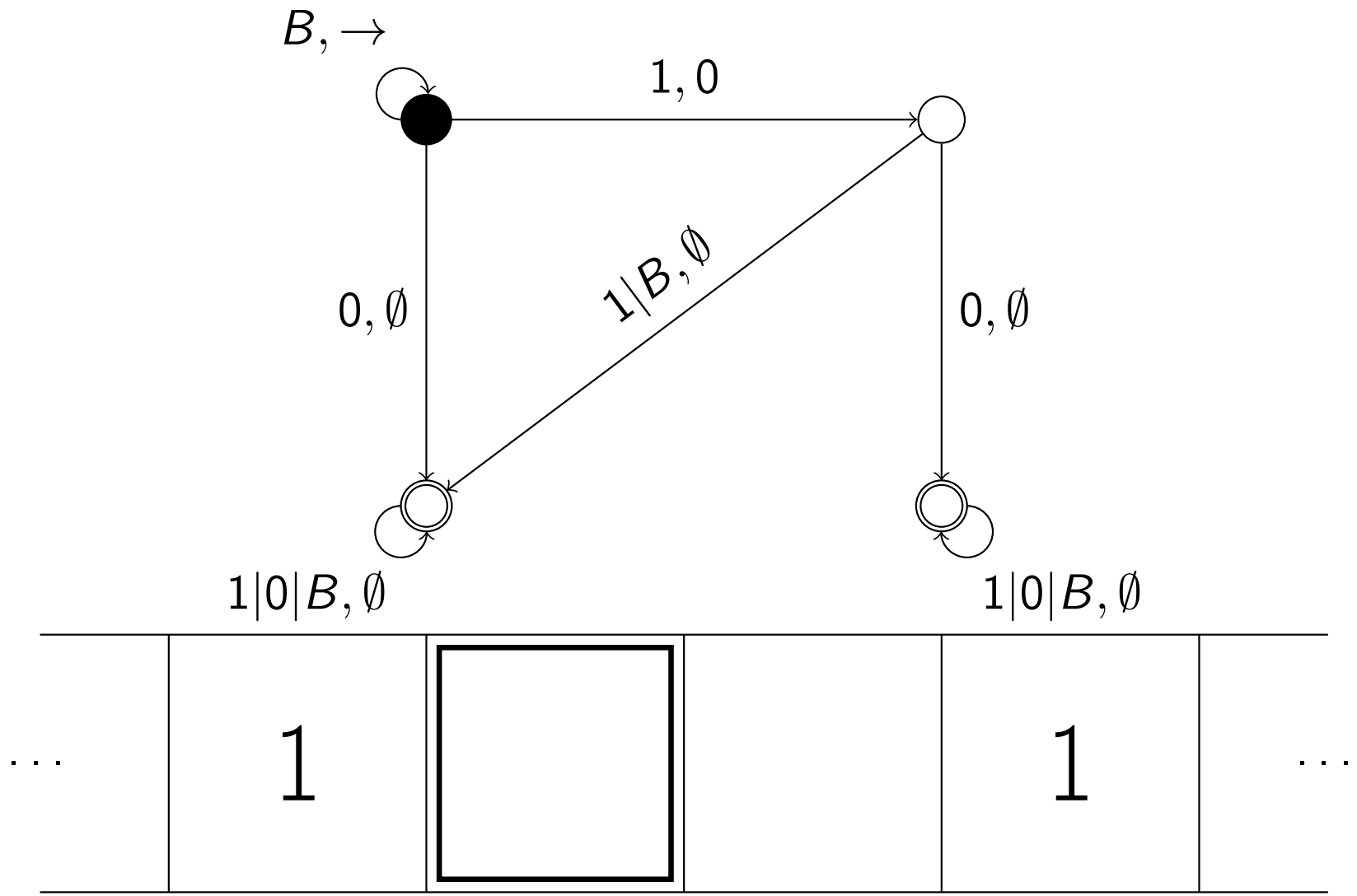
Turing machine - the head



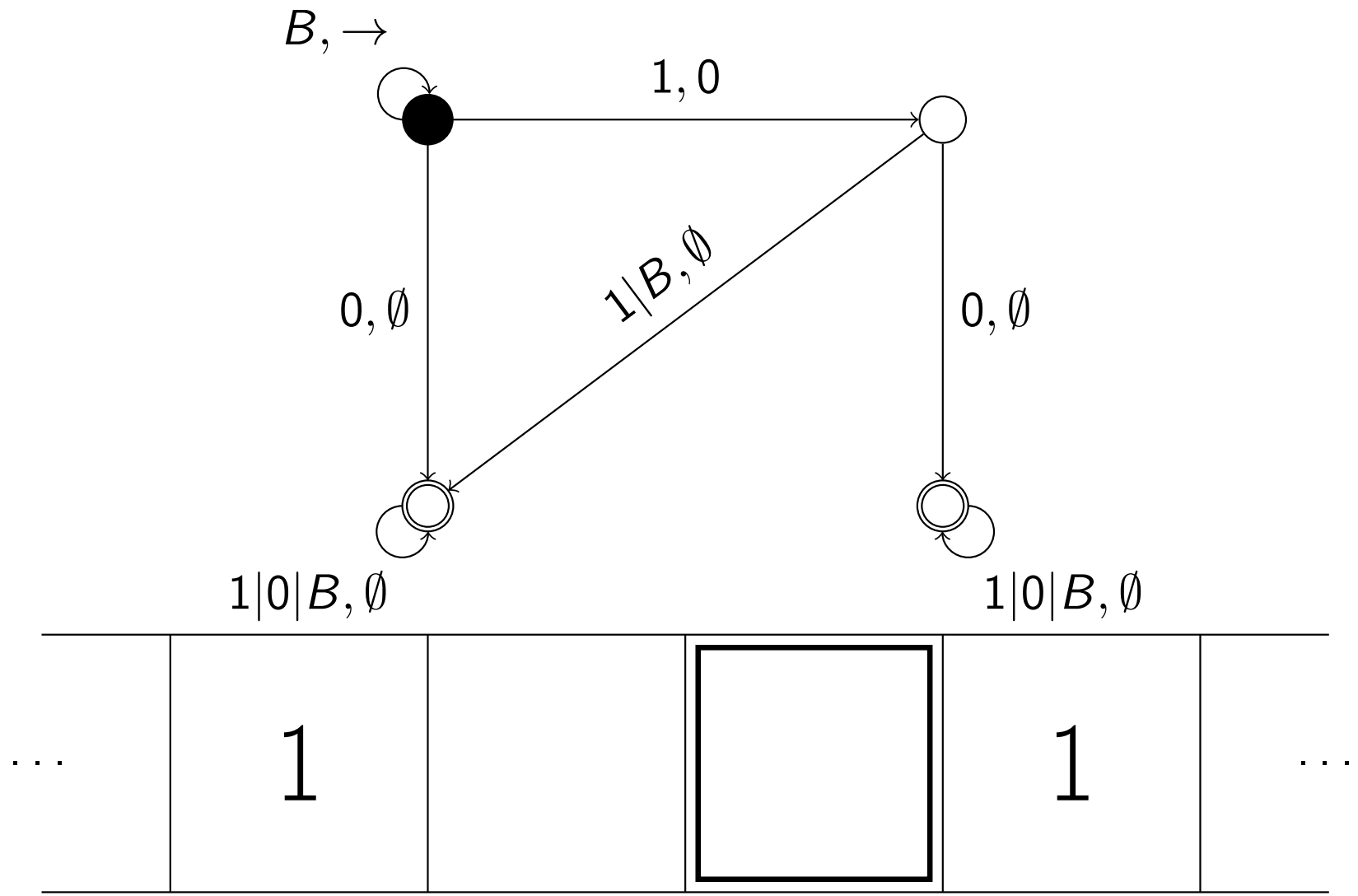
Turing machine - the state graph



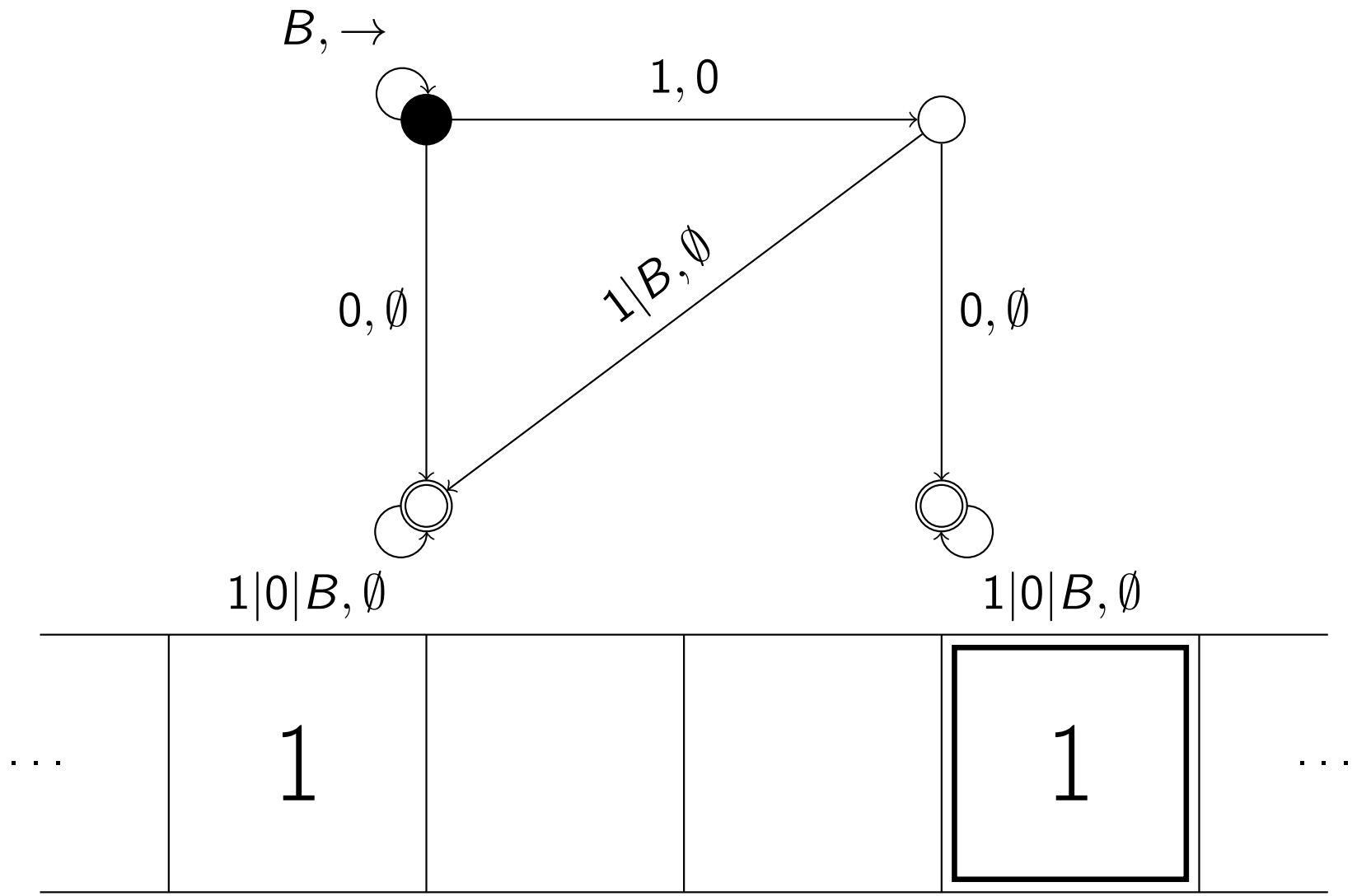
Turing machine - the state register



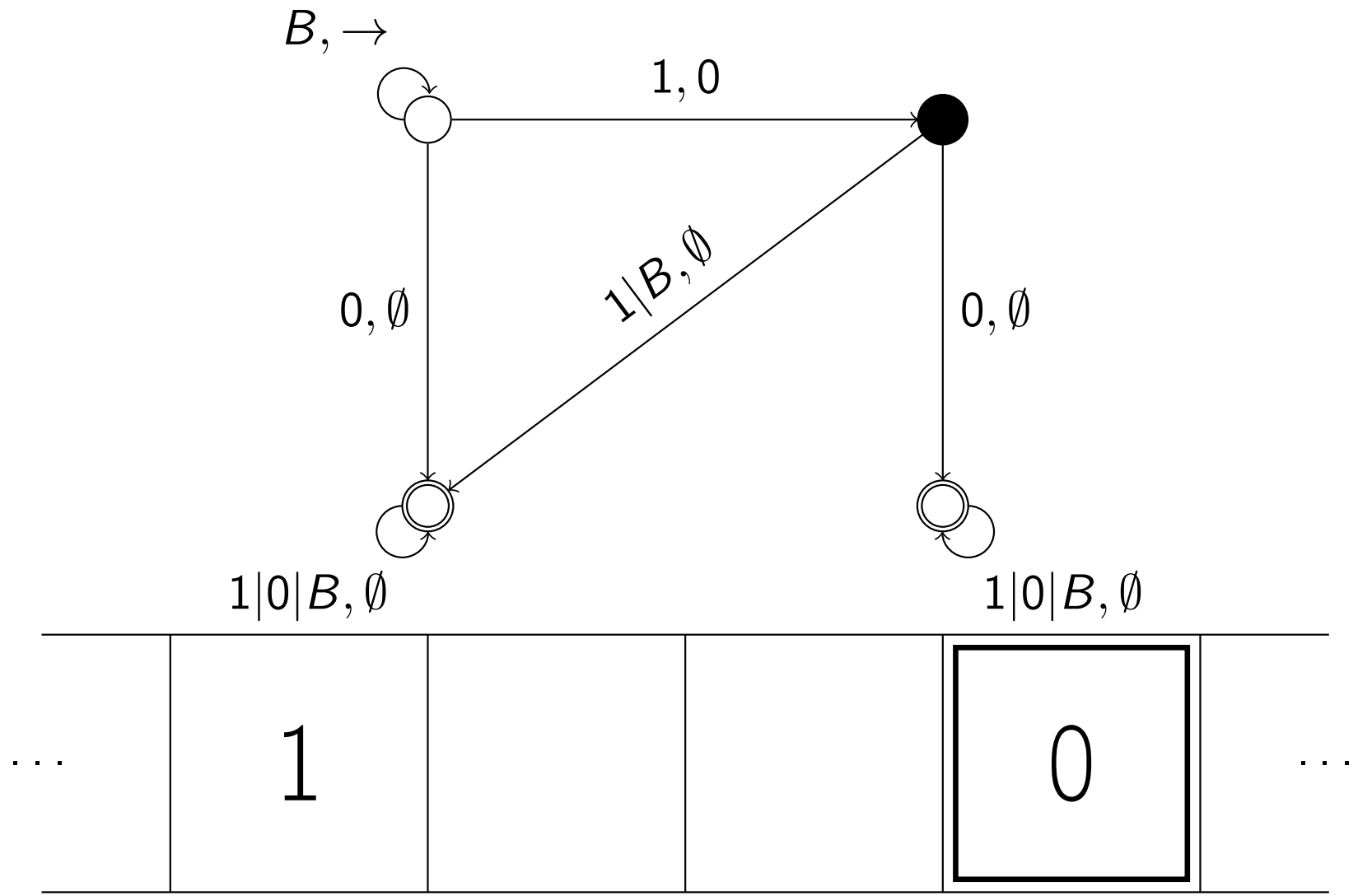
Turing machine - the state register



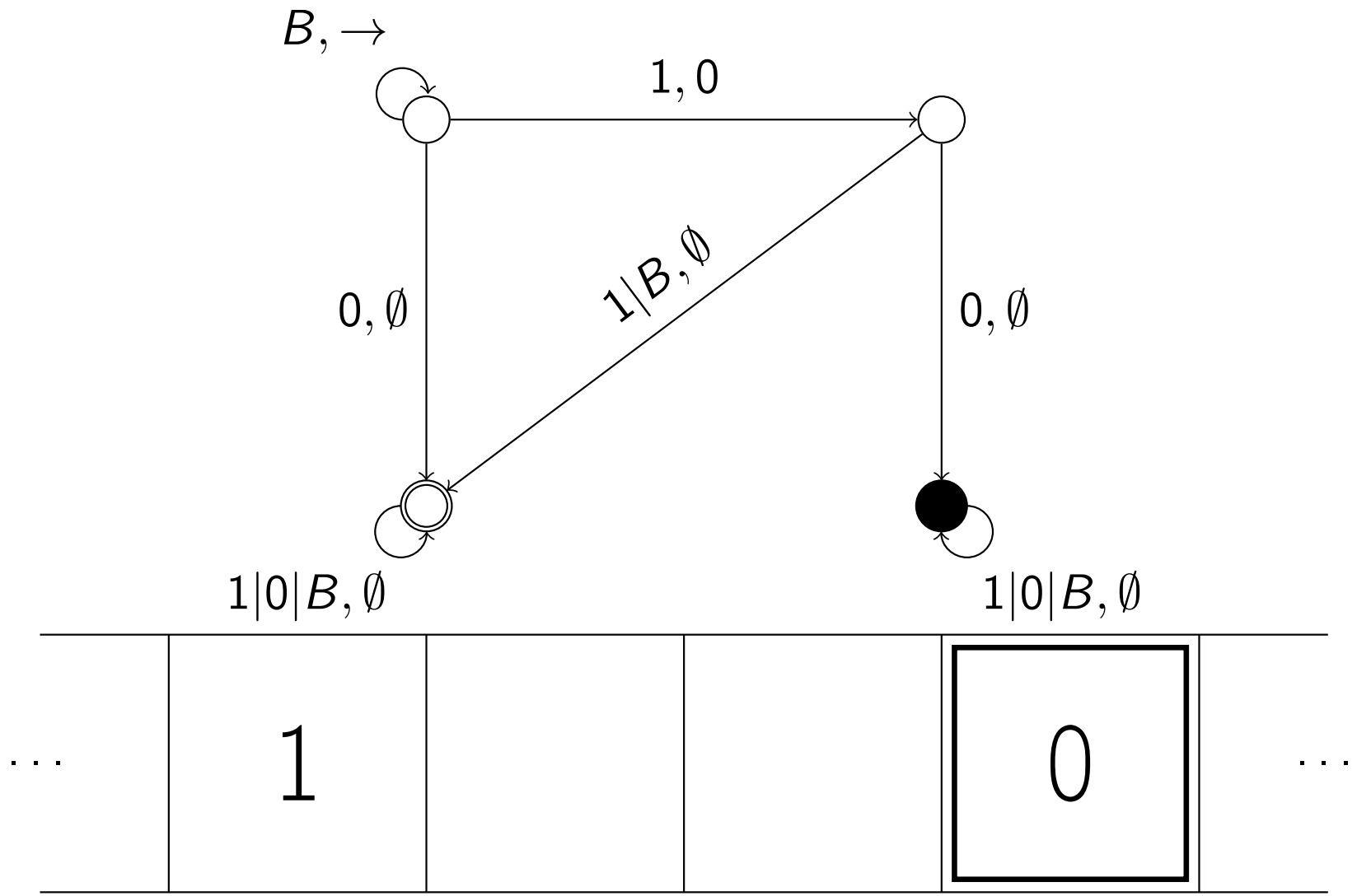
Turing machine - the state register



Turing machine - the state register



Turing machine - the state register



Turing machine

Definition

A Turing machine consists of

- a *tape* $\mathcal{B}$ containint cells numbered from $-\infty$ to $+\infty$
- an *alphabet* $\{0, 1, B\}$ of symbols written on the cells
- a *head* pointing to the cell 0 of the tape
- a finite directed *states graph* $G = (S, A)$ where A is labeled with $\mathcal{P}(\{0, 1, B\}) \times \{0, 1, B, \leftarrow, \rightarrow, \emptyset\}$, a *read symbol* and an *action*
- a *state register* pointing to the *initial state* of S
- a subset $F \subset S$ of *terminal states* of S

Computation of a Turing machine?

We first write a finite word x (containint 1, 0 and B symbols) such that the first symbol is on the cell 0.
At each iteration

1. the head reads the symbol s on the cell c_i it is pointing to;
2. the state register, pointing to the state q, chooses an arc (q, q'), for which the reading symbol is s and the action is a;
3. we execute the action a:
   - if a is 0, 1 or B, the we replace the symbol on the cell c_i by this one,
   - if a is respectively $\leftarrow$ or $\rightarrow$, the head moves respectively for c_i to c_{i-1} or c_{i+1},
   - if a is $\emptyset$, we do nothing;
4. the state register moves from q to q';
5. if $q' \in F$, the machine stops, otherwise it starts again.

Deterministic machine, non deterministic macine

Definition

A Turing machine is said *deterministic* if and only if for each state q and each couple of arcs a_1 and a_2 outgoing from q, the reading symbols of a_1 and a_2 are not the same. Otherwise, we say the machine is *non-deterministic*.

A Turing machine must sometimes make some choices.

Accepting/Refusing a word

Definition

The terminal states F are parted into two sets of the *accepting states* F_Y and the *rejecting states* F_N

Definition

A deterministic Turing machine accepts a word x, if and only if, when we run the machine with this word written on the tape, the machine stops on an accepting state. It rejects it if it stops on a refusing state.

Accepting/Refusing a word

Definition

The terminal states F are parted into two sets of the *accepting states* F_Y and the *rejecting states* F_N

Definition

A non-deterministic Turing machine accepts a word x, if and only if, when we run the machine with this word written on the tape, **there exists a sequence of choices** such that the machine stops on an accepting state. It rejects it if it may stop on a refusing state.

Strongly accepting/rejecting a word

Definition

The terminal states F are parted into two sets of the *accepting states* F_Y and the *rejecting states* F_N

Definition

A non-deterministic Turing machine *strongly* accepts a word x, if and only if, when we run the machine with this word written on the tape, **for every sequence of choices**, the machine stops on an accepting state. It strongly rejects it if it must stop on a refusing state.

This definition is not conventional but convenient.

Other Turing machines

We can equivalently

- add tapes
- add symbols
- separate reading and writing
- merge actions (a finite number of times)
- ...

Not equivalent Turing machines

- Probabilistic Turing machine
- Quantum Turing machine
- Arithmetic machine
- ...