•) $\Sigma = \{a, b\}$

$L_1 = \{ a^n b a^n b a^m \mid n, m \geq 1,\ m \geq n \}$

let $uvwxy = a^n b a^n b a^n$

$= \underbrace{a \dots a}_{1}\, b\, \underbrace{a \dots a}_{1}\, b\, a \dots a$

2 3 4

1) if $k > 1$: $a^{n+(k-1)|vx|} b a^n b a^n \notin L_1$

(same for the 2nd list of as)

careful with $v = \varepsilon$ and $x = \varepsilon$!

2) if $b \notin v \wedge b \notin x$, $k > 1$: $a^{n+\dots} b a^{n+\dots} b a^n \notin L$

if $b \in v \vee b \in x$, $k = 0$: $a^{n-\dots} a^{n-\dots} b a^n \notin L$

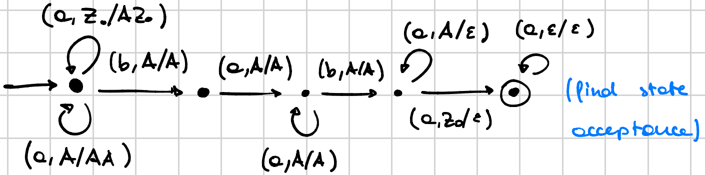
3) if $b \notin v \wedge b \notin x$, $k = 0$: $a^n b a^{n-\dots} b a^{n-\dots} \notin L$

if $b \in v \vee b \notin x$, $k = 0$: $a^n b a^{n-\dots} a^{n-\dots} \notin L$

4) if $k = 0$: $a^n b a^n b a^{n-|vx|} \notin L$

•) $\Sigma = \{a,b\}$

$L_2 = \{ a^n b a^p b a^m \mid n,m,p \geq 1,\ m \geq n \}$



o

$S \rightarrow aSa \mid aFa$

$F \rightarrow bGb \mid Fa$

$G \rightarrow a \mid aC$

State as internal memory

Example : A TM M that "memorizes" the first symbol read and verifies that this does not appear again in the input

$$L(M) = L(01^* + 10^*)$$

Let $M = (Q, \{0,1\}, \{0,1,B\}, \delta, [q_0, B], B, \{[q_1, B]\})$, with $Q = \{q_0, q_1\} \times \{0, 1, B\}$

	0	1	B
$\rightarrow [q_0, B]$	$([q_1, 0], 0, R)$	$([q_1, 1], 1, R)$	
$[q_1, 0]$		$([q_1, 0], 1, R)$	$([q_1, B], B, R)$
$[q_1, 1]$	$([q_1, 1], 0, R)$		$([q_1, B], B, R)$
$\star[q_1, B]$			

Tape with multiple tracks

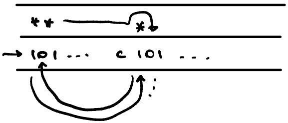
Example : A TM for the language $L = \{wcw \mid w \in \{0,1\}^*\}$

We use a tape track for "marking" those input symbols that we have already tested

$$M = (Q, \Sigma, \Gamma, \delta, [q_1, B], [B, B], \{[q_0, B]\})$$

where

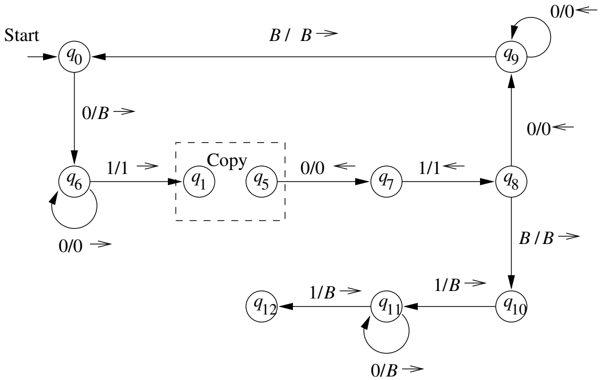
- $Q = \{q_1, q_2, \ldots, q_9\} \times \{0, 1, B\}$
- $\Sigma = \{[B, 0], [B, 1], [B, c]\}$
- $\Gamma = \{B, *\} \times \{0, 1, c, B\}$



See the textbook for the specification of the transition function δ

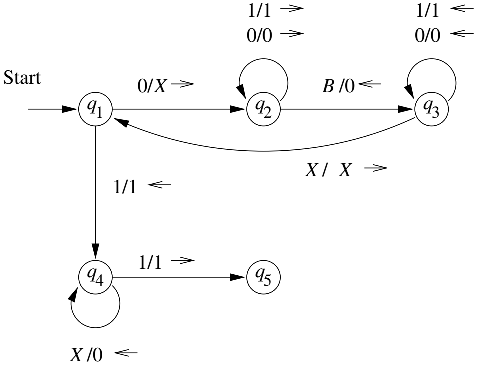
Use of a subroutine

Example : A TM for the computation of the product function $0^m10^n1 \mapsto 0^{m \cdot n}$. We use a *subroutine* "Copy"



Use of a subroutine

The subroutine "Copy" takes ID $0^{m-k}1q_1 0^n 1 0^{(k-1)n}$ to ID $0^{m-k}1q_5 0^n 1 0^{kn}$



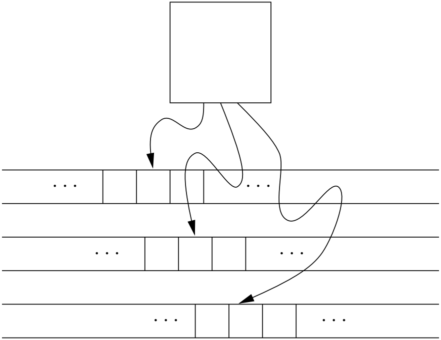
TM extensions

Let us now present some **extensions** of the TM definition

For each extension, we prove that the **computational capacity** is the same as the one of the classic definition of TM

Multi-tape TM

We use a finite number of **independent** tapes for the computation, with the input on the first tape



Multi-tape TM

In a single move the multi-tape TM performs the following actions

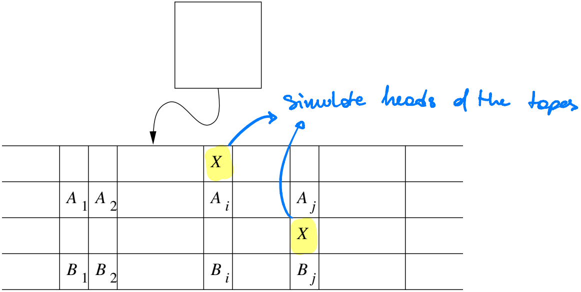
- state update, on the basis of read tape symbols
- for each tape :
  - write a symbol in current cell
  - move the tape head independently of the other heads (L = left, R = right, or S = stay)

Note that the stay option is not available in a TM

Multi-tape TM

Theorem A language accepted by a multi-tape TM M is RE

Proof (sketch) We can simulate M using a TM N with a multi-track tape



Multi-tape TM

We use $2k$ tracks to simulate k tapes : even tracks used for tape content, odd tracks used for tape head position

N visits all k head positions to **simulate** a single move of M

- left to right pass : the number of visited tape heads and the content of the corresponding cells are stored into the state of N
- right to left pass : for each tape head of M, the corresponding action is simulated by N

N updates its state in the same way as M $\square$

Multi-tape TM

Theorem The TM N in the proof of the previous theorem simulates the first n moves of the TM M with k tapes in time $\mathcal{O}(n^2)$

Proof (sketch) After n moves of M, tape head markers in N have **mutual distance** not exceeding $2n$

It follows that any one of the first n moves of M can be simulated by N in a number of moves not exceeding $4n + 2k$, which amounts to $\mathcal{O}(n)$ since k is a constant □