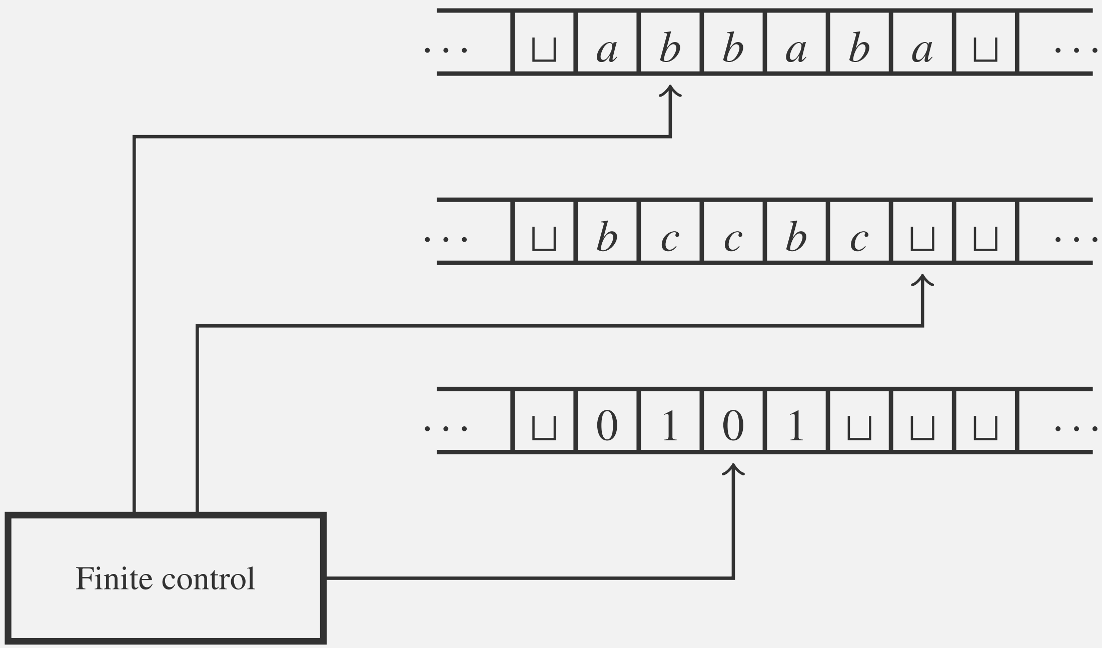
Logic and theory of computation

theory of computation part, 2nd lecture

Multi tape Turing machines

Informal picture



- One step is the following: Reading the k tape symbols and the current state the TM moves to a new state, rewrites the k tape symbols and moves the k tape heads independently.
- Accepting is analogous to 1-tape TM's.
- Running time and time complexity is analogous to 1-tape TM's.

Multi tape Turing machines

Definition

k-tape TM

A k**-tape TM** is a 7-tuple $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r \rangle$ where

- Q is a finite, nonempty set of states,
- $q_0, q_a, q_r \in Q$, q_0 is the starting, q_a is the accepting and q_r is the rejecting state,
- Σ and Γ are the input and the tape alphabets, respectively, where $\Sigma \subseteq \Gamma$ and $\sqcup \in \Gamma \setminus \Sigma$,
- $\delta : (Q \setminus \{q_a, q_r\}) \times \Gamma^k \to Q \times \Gamma^k \times \{L, S, R\}^k$ is the transition function.

Configuration

$(q, u_1, v_1, \ldots, u_k, v_k)$ is called a **configuration** of a k-tape TM, where $q \in Q$ and $u_i, v_i \in \Gamma^*, v_i \neq \varepsilon$ $(1 \leq i \leq k)$.

Multi tape Turing machines

Configurations

Configuration is a finite representation of the machine at a given time. It represents the current state q, the content of the ith tape $u_i v_i$, and the position of the ith head as the first letter of v_i $(1 \le i \le k)$.

Starting configuration

Starting configuration of the word u is $u_i = \varepsilon$ $(1 \le i \le k)$, $v_1 = u\sqcup$, and $v_i = \sqcup$ $(2 \le i \le k)$.

[Why v_1 is defined to be $u\sqcup$ and not u? To avoid $u = \varepsilon$ being another case. The two words represent the same tape content.]

Accepting/rejecting/halting configurations

For a configuration $(q, u_1, v_1, \ldots, u_k, v_k)$ where $q \in Q$ and $u_i, v_i \in \Gamma^*, v_i \neq \varepsilon$ $(1 \le i \le k)$, it is an **accepting configuration** if $q = q_a$, **rejecting configuration**, if $q = q_r$, **halting configuration**, if $q = q_a$ or $q = q_r$.

Multi tape Turing machines

Changing configurations

Defining how a configation yields another one in one step (one step transition relation) is analogous to the way we did it for one-tape TM's. We use the same notation: $\vdash$. We have many (3^k) cases now, so let us consider only an example.

Let k=2 and $\delta(q, a_1, a_2) = (r, b_1, b_2, R, S)$ be a transition of a TM. Then $(q, u_1, a_1v_1, u_2, a_2v_2) \vdash (r, u_1b_1, v_1', u_2, b_2v_2)$, where $v_1' = v_1$, if $v_1 \neq \varepsilon$, otherwise $v_1' = \sqcup$.

Notice, that the heads can move independently of each other.

Definition of multistep transition relation (a configuration yields an other in finitely many steps) is the same as with one-tape TM's. Notation: $\vdash^*$.

Multi tape Turing machines

Recognized language, time complexity

The language recognized by a TM M

$L(M) = \{u \in \Sigma^* \mid (q_0, \varepsilon, u\sqcup, \varepsilon, \sqcup, \ldots, \varepsilon, \sqcup) \vdash^*$
$(q_a, x_1, y_1, \ldots x_k, y_k),\ x_1, y_1 \ldots, x_k, y_k \in \Gamma^*, y_1, \ldots,, y_k \neq \varepsilon\}.$

As in the case of one tape TM's multi tape TM's are accepting those words for which the TM can reach its q_a state.
Languages that can be **recognised** or can be **decided** by a multi tape TM is defined in the same way as in the case of one-tape TM's.

Running time

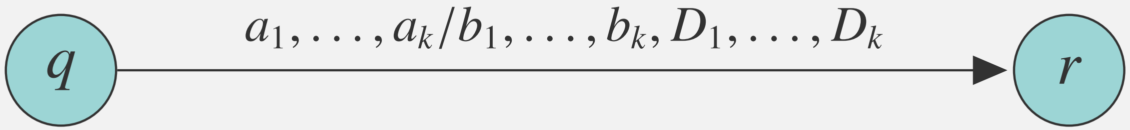
Given a word u, its **running time** on a TM M is the number of computational steps from its starting configuration to a halting configuration.

Time complexity of a multi tape TM is defined the same way as in the case of one tape TM's.

Multi tape Turing machines

An Example

Exercise: Construct a 2-tape TM M having
$L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$.
Transition diagram.



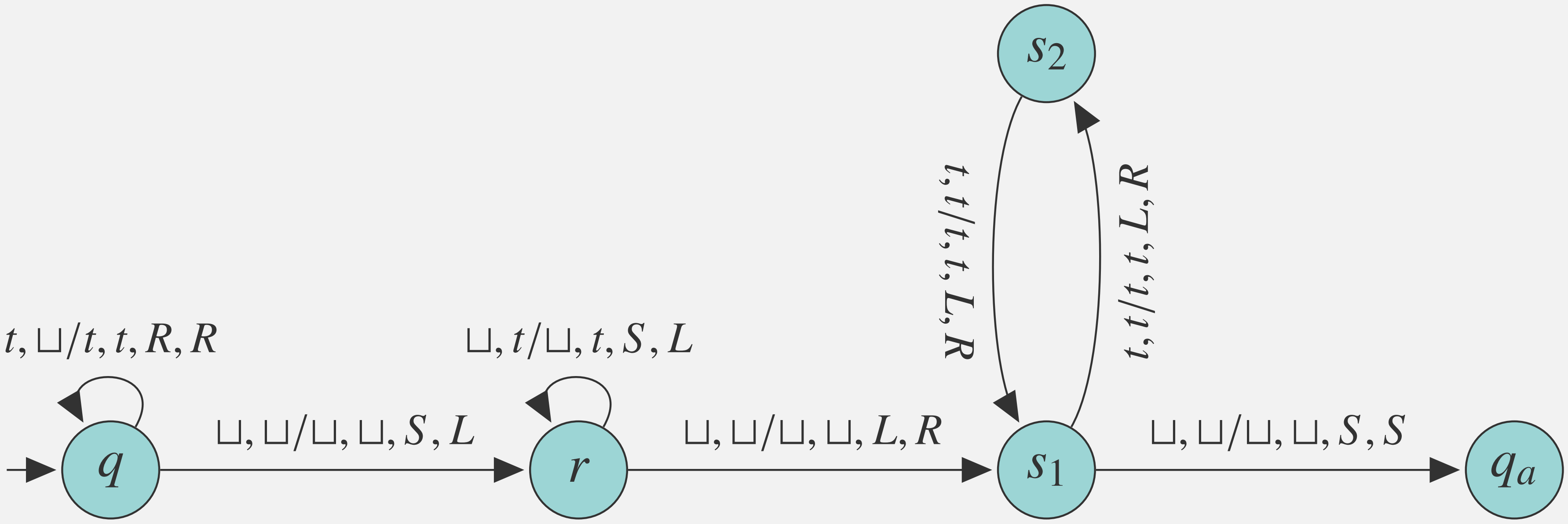
$$\delta(q, a_1, \ldots, a_k) = (r, b_1, \ldots, b_k, D_1, \ldots, D_k)$$
$$(q, r \in Q, a_1, \ldots, a_k, b_1, \ldots, b_k \in \Gamma, D_1, \ldots, D_k \in \{L, S, R\})$$

Multi tape Turing machines

An Example

Exercise: Construct a 2-tape TM M having
$L(M) = \{ww^{-1} \mid w \in \{a, b\}^*\}$.

A **solution**:
(All other transitions are directed to q_r. For all the transitions
$t \in \{a, b\}$ is arbitrary.)



What is the time complexity? It is a $O(n)$ time-bounded TM.

Multi tape Turing machines

Simulating by a single tape

Equivalent TM's

Two TM's are considered to be **equivalent**, if they recognise the same language.

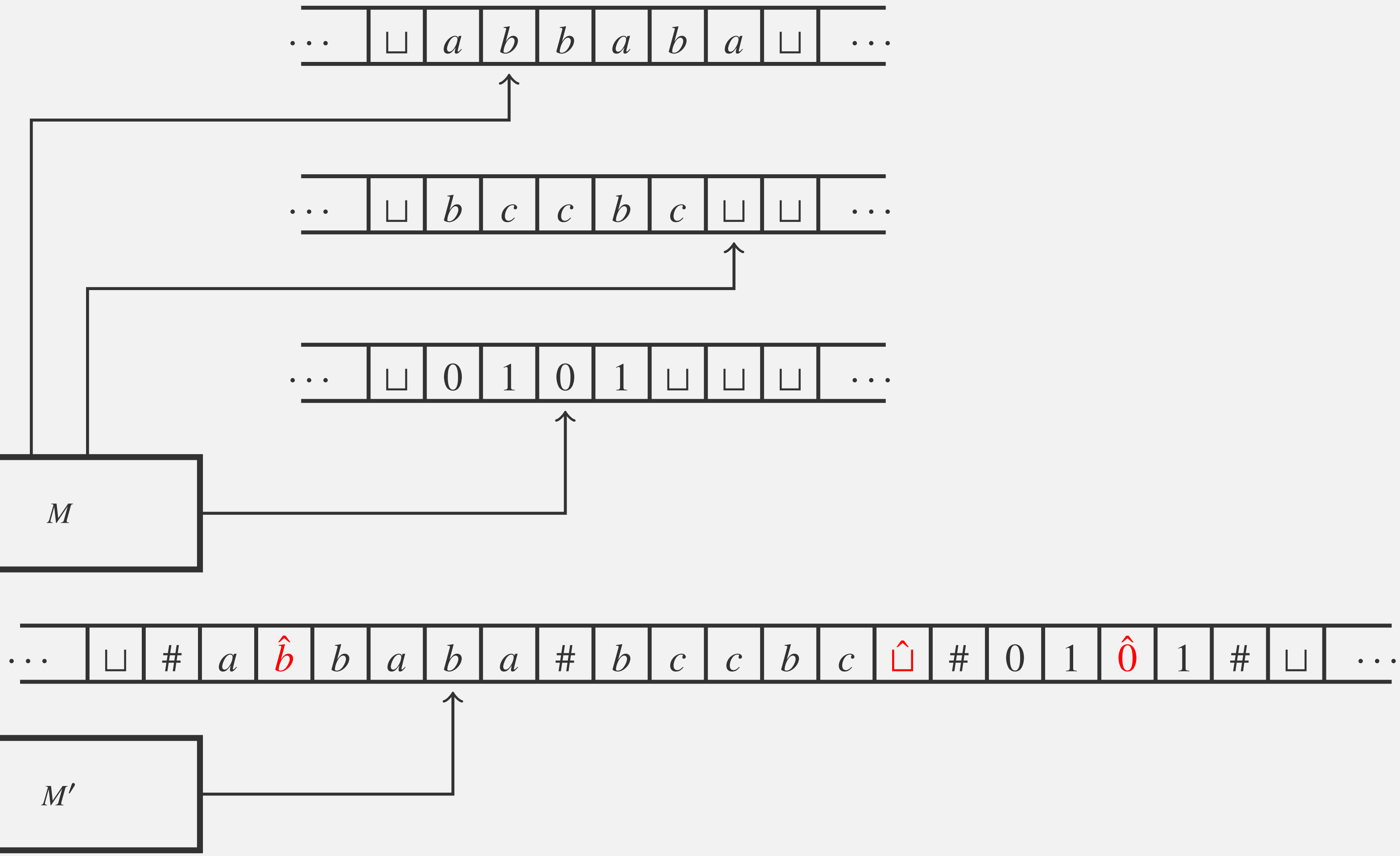
Theorem

For any k-tape TM M there is an equivalent 1-tape TM M'
Furthermore if M has time complexity $O(f(n))$ which is at least linearir then M' has time complexity $O(f(n)^2)$.

Multi tape Turing machines

Simulating by a single tape

Proof (sketch): Main idea of the simulation



Multi tape Turing machines

Simulating by a single tape

Steps of the simulation on input $a_1 \cdots a_n$:

1. Let the starting configuration M' be $q_0'\#\hat{a}_1 a_2 \cdots a_n \#\hat{\sqcup}\# \cdots \hat{\sqcup}\#$
2. M' goes through its tape for the first time counts the #'s and stores symbols denoted by a ˆ in its states.
3. M' goes through its tape for the another time updating it's content according to its transition function.
4. if the length of the content on a tape of M increases M' shifts it's content by a cell.
5. If M reaches its accepting or rejecting state so does M'.
6. Otherwise M' continues with step 2.

Multi tape Turing machines

Simulating by a single tape – time complexity

The following holds for simulating a single step of M:

- Used up space (number of used cells) is an asymptotic upper bound for the number of steps M' takes. (Goes through its content twice, it needs to make space for a $\sqcup$ at most k times which is O(used up space))
- used up space was increased by $O(1)$ cells. (by $\leq k$, in fact)

At the begining M' used $\Theta(n)$ cells, in each step it was increeased by $O(1)$, so $O(n + f(n)O(1)) = O(n + f(n))$ is a common asymptotic upper bound for the used up cells after each step of the simulation.
So it is an asympotic upper bound for the time complexity of a single step, as well.
So altogether M' has a time complexity of $f(n) \cdot O(n + f(n))$. This is $O(f(n)^2)$, if $f(n) = \Omega(n)$.

Nondeterministic TM

definition, condigurations, one step transition relation

A **nondeterministic TM** (NTM) is a 7-tuple
$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r \rangle$ where

- $Q, \Sigma, \Gamma, q_0, q_a, q_r$ is the same as before
- $\delta : (Q \setminus \{q_a, q_r\}) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, S, R\})$

Let C_M denote the set of configurations.

$\vdash \subseteq C_M \times C_M$ yields in one step

Let $uqav$ be a configuration where $a \in \Gamma,\ u, v \in \Gamma^*$.

- If $(r, b, R) \in \delta(q, a)$, then $uqav \vdash ubrv'$, where $v' = v$, if $v \neq \varepsilon$, otherwise $v' = \sqcup$,
- if $(r, b, S) \in \delta(q, a)$, then $uqav \vdash urbv$,
- if $(r, b, L) \in \delta(q, a)$, then $uqav \vdash u'rcbv$, where $c \in \Gamma$ and $u'c = u$, if $u \neq \varepsilon$, otherwise $u' = u$ and $c = \sqcup$.

Nondeterministic TM

multistep transition relation, recognised language

Multistep transition relation is the reflexive, transitive closure of $\vdash$ denoted by $\vdash^*$, i.e.,

$\vdash^* \subseteq C_M \times C_M$ yields in finitely many steps

C yields C' in finitely many steps (denoted by $C \vdash^* C' \Leftrightarrow$

- if $C = C'$ or
- if $\exists\, n > 0 \wedge C_1, C_2, \ldots \in C_n \in C_M$, such that $\forall\, 1 \leq i \leq n-1$ $C_i \vdash C_{i+1}$ holds. Furthermore $C_1 = C$ and $C_n = C'$.

The language recognised by a NTM M

$L(M) = \{u \in \Sigma^* \mid q_0 u \sqcup \vdash^* x q_a y$ for some $x, y \in \Gamma^*, y \neq \varepsilon\}$.

A NTM may have several computations for the same word. It accepts a word if and only if it has at least on computation reaching q_a.

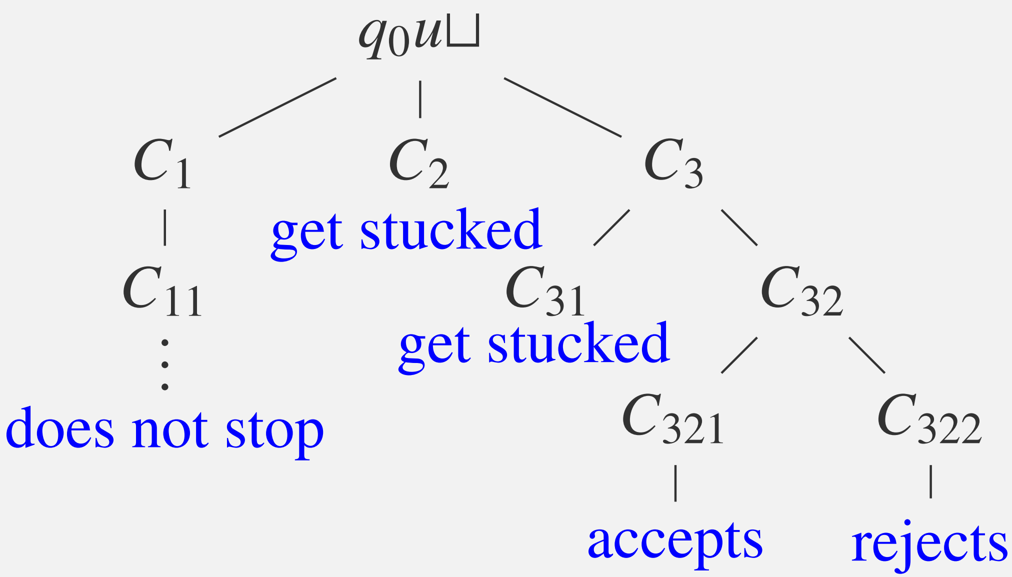
Nondeterministic TM

Nondeterministic configuration tree

nondeterministic configuration tree for $u \in \Sigma^*$

Directed tree, where the nodes are labelled. The root has label $q_0 u \sqcup$. If C is a label of a node, then the node has $|\{C' \mid C \vdash C'\}|$ children with corresponding labels from the set $\{C' \mid C \vdash C'\}$.

Example:



This TM accepts u since $q_0 u \sqcup \vdash C_3 \vdash C_{32} \vdash C_{321}$ is a computation leading to an accepting configuration. Only one such computation is needed.

Nondeterministic TM

decidability by a NTM, time complexity

A NTM M **decides** a language $L \subseteq \Sigma^*$, if it recognises it and for all $u \in \Sigma^*$ the configuration tree has a finite height and all leaves are a halting configuration.

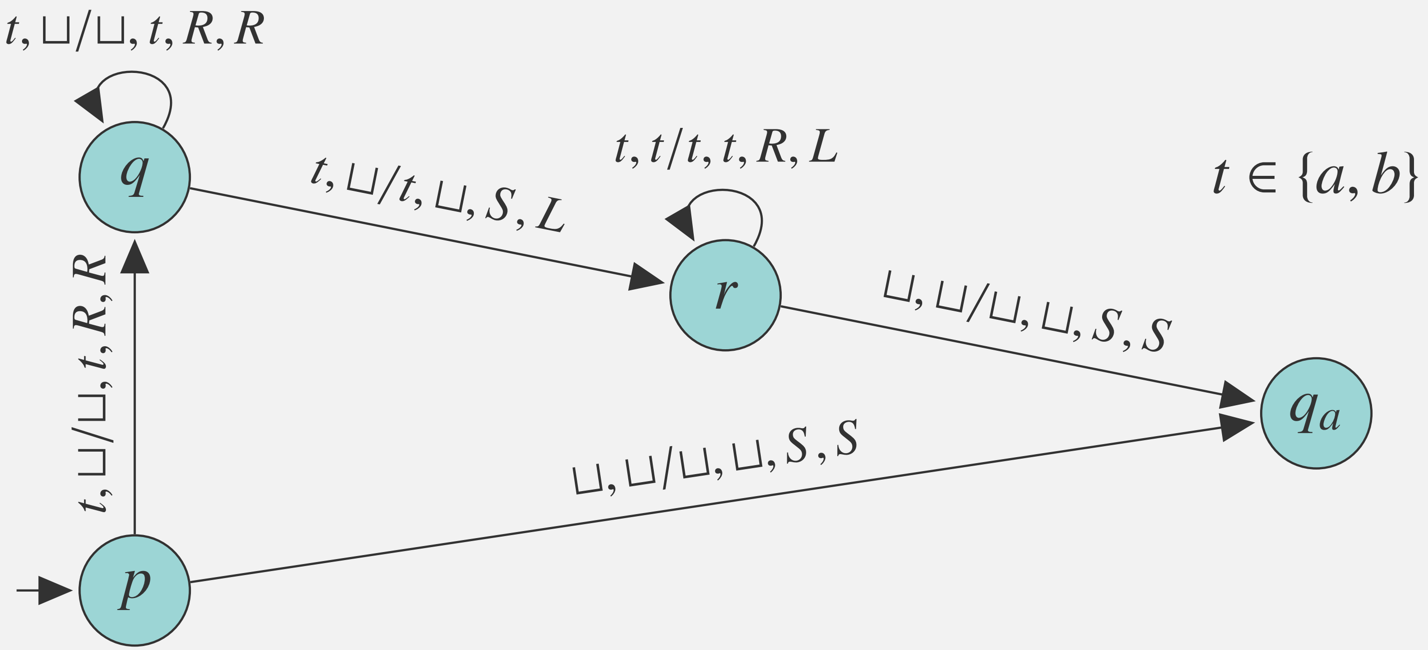
M has **time complexity** $f(n)$, if for all $u \in \Sigma^*$ of length n the height of the configuration tree is at most $f(n)$.

Remark: The definition of multi tape NTM's is analogous the previous definitions.

Nondeterministic TM

Example

Excercise: Construct a NTM M with $L(M) = \{ww^{-1} \mid w \in \{a,b\}^*\}$!



$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (r, \varepsilon, bba, \varepsilon, a) \vdash (q_r, \varepsilon, bba, \varepsilon, a)$

$(p, \varepsilon, abba, \varepsilon, \sqcup) \vdash (q, \varepsilon, bba, a, \sqcup) \vdash (q, \varepsilon, ba, ab, \sqcup) \vdash (r, \varepsilon, ba, a, b) \vdash$
$(r, b, a, \varepsilon, ab) \vdash (r, ba, \sqcup, \varepsilon, \sqcup ab) \vdash (q_a, ba, \sqcup, \varepsilon, \sqcup ab)$

Nondeterministic TM

Simulating NTM's by deterministic TM's

Theorem

For all $M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_a, q_r \rangle$ NTM's of time complexity $f(n)$ there is an equivalent deterministic TM of time complexity $2^{O(f(n))}$.

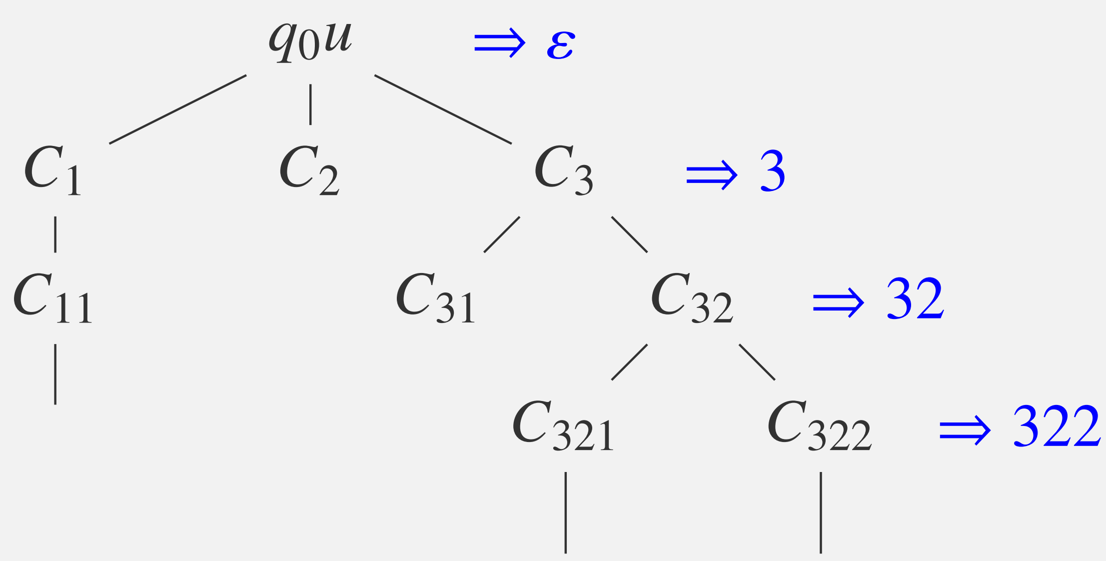
Proof (sketch): Idea M' simulates all computations for an input $u \in \Sigma^*$, by doing a breadth first search on its configuration tree.

- Let d be the following number.
  $d = \max_{(q,a) \in Q \times T} |\delta(q, a)|$.
- Let $T = \{1, 2, \ldots, d\}$ be an alphabet.
- for all $(q, a) \in Q \times \Gamma$ let us fix an order of the set $\delta(q, a)$

Nondeterministic TM

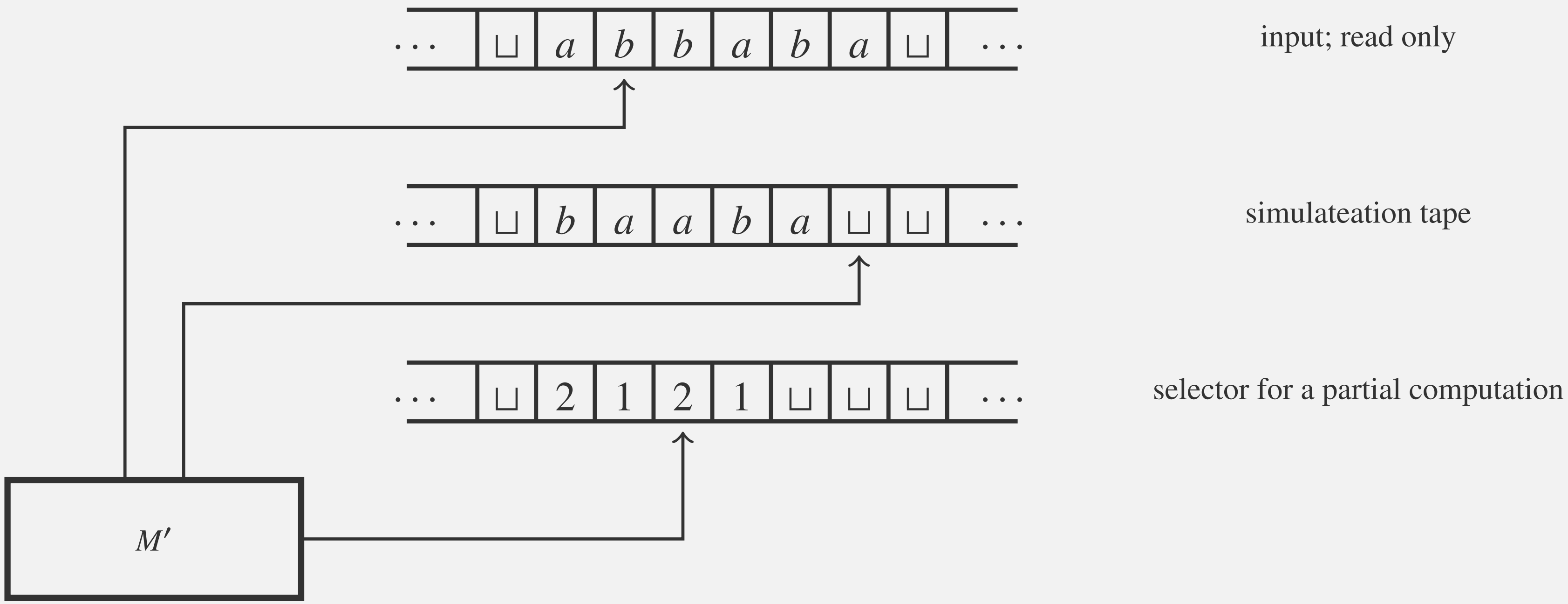
Simulating NTM's by deterministic TM's

For each node of the configuration graph one can associate a unique word over T, the selector of that partial computation.



Nondeterministic TM

Simulating NTM's by deterministic TM's



How does M' work?

Nondeterministic TM

Simulating deterministic TM-pel

- starting configuration of M': input on 1st tape, the other tapes are empty
- WHILE there's no accept
  - copy the content of tape 1 of M' to tape 2
  - WHILE the head of the 3rd tape deas not read $\sqcup$
    - Let k be the current letter on tape 3
    - Let a be the current letter on tape 2
      and q be the current state of M
    - if $\delta(q, a)$ has a kth element, then
      - M' simulates one step of M
      - if this leads to q_a of M, then M' accepts
      - if this leads to q_r of M, then M' breaks this cycle
    - M' moves one cell to the right on the 3rd tape
  - M' deletes the content of tape 2
    and creates the next word on tape 3
    according to shortlex order over T.

Nondeterministic TM

Simulating NTM's by deterministic TM's

- M' goes to it accepting state if and only if M do so, so they are equivalent TM's
- M' needs to examine an exponential function of $f(n)$ many computations ($\leq$ the number of inner nodes of a full d-ary tree of height $f(n)$ which is $O(d^{f(n)})$), so M' has time complexity $2^{O(f(n))}$

Remarks:

- This simulation has exponential time complexity, but it does not follow there's no better.
- But the conjecture is that there's no efficient simulation of a NTM by a deterministic one.