1.4 Context-sensitive and type 0-languages

Kuroda normal form

A type 1 grammar $G = (V, \Sigma, P, S)$ is in Kuroda normal form if

$$P \subseteq V \times (V \cup \Sigma \cup V^2) \cup V^2 \times V^2$$

Proposition: $\forall G$ of type 1 : $\varepsilon \notin L(G) \longrightarrow$

$\exists G'$ in Kuroda normal form with $L(G) = L(G')$

Proof: not here

Idea: A generalization of Chomsky normal form.

Turing Machines

Are the finite automata the last word?

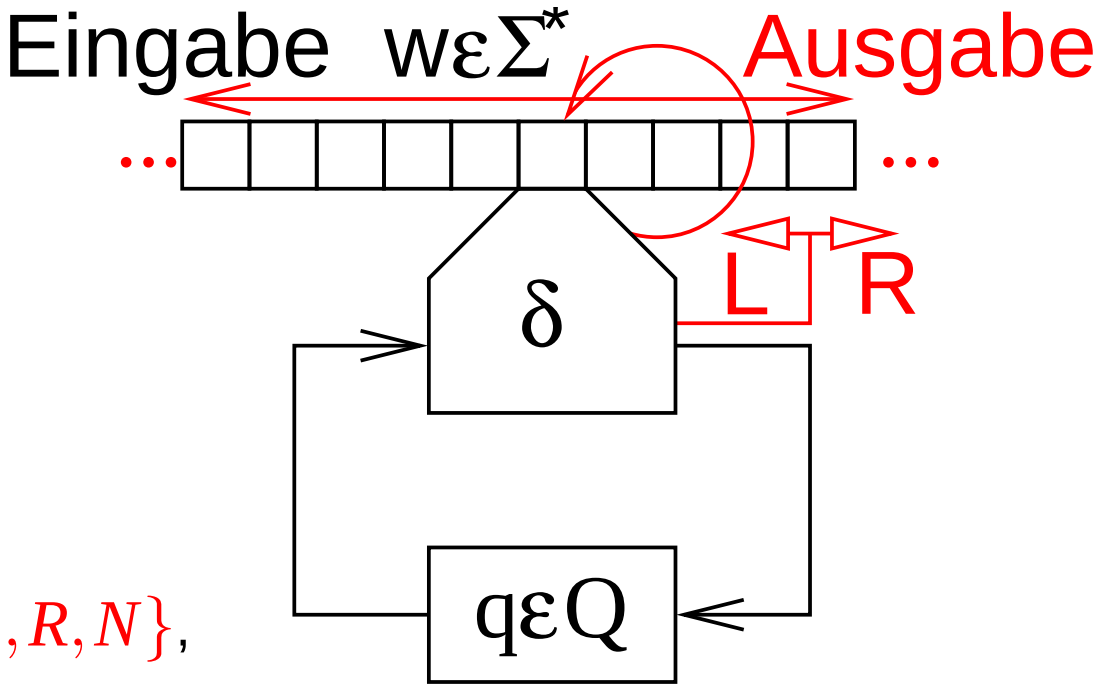
$+$: A digital computer with finite memory is a finite automaton !

$-$: Big memory $\rightsquigarrow$ astronomic many states, very complex automata

$-$: We want a simple machine model for automata which for example accepts the words of the form $a^n b^n c^n$

Deterministic (one-tape)-Turing machines (DTM)

$T = (Q, \Sigma, \Gamma, \delta, s, F)$:

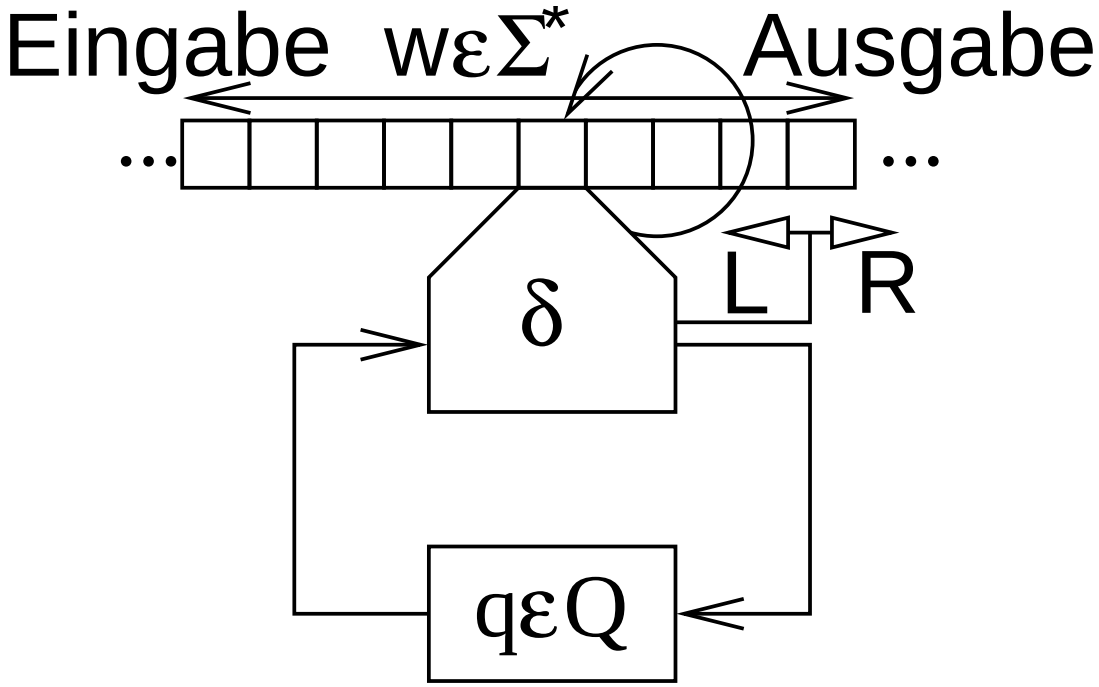


- Q states, Σ input alphabet
- Γ tape alphabet,
  $\sqcup \notin \Sigma$: blank symbol,
  $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$,
  transition function;
- $s \in Q$, initial state
- $F \subseteq Q$, final states

Nondeterministic Turing machines (NTM)

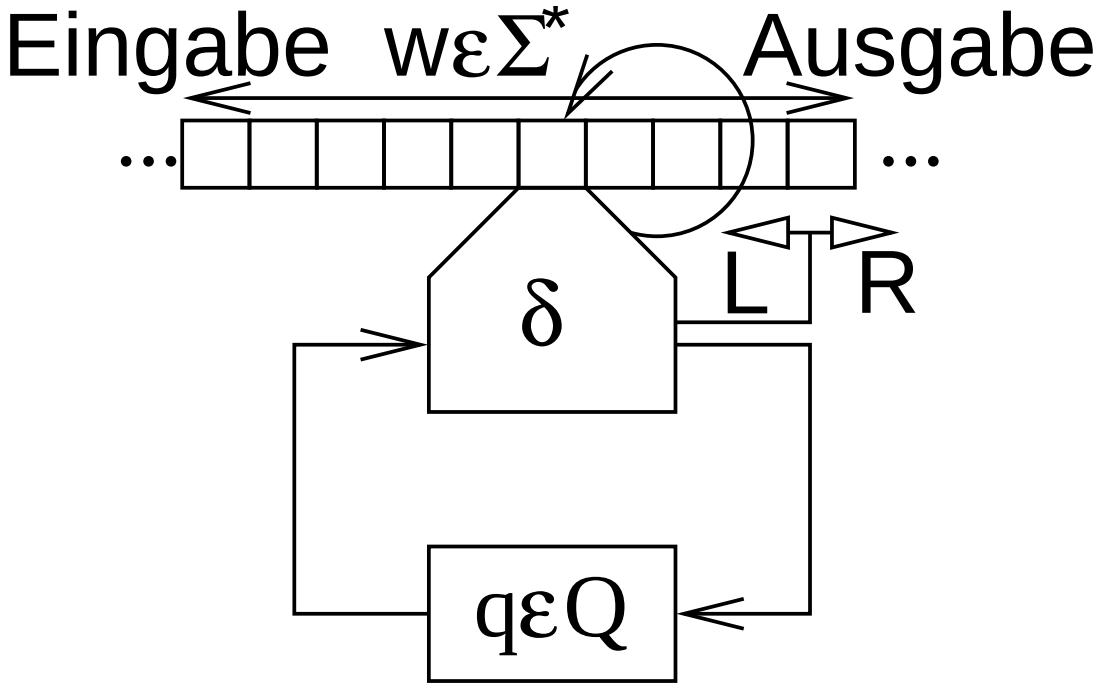
$T = (Q, \Sigma, \Gamma, \delta, s, F)$:

- Q, states
- Σ, input alphabet
- Γ tape alphabet,
  $\sqcup \notin \Sigma$: blank symbol,
  $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R,N\}}$,
  transition function;
- $s \in Q$, initial state
- $F \subseteq Q$, final states



Nondeterministic Turing machines (NTM)

$T = (Q, \Sigma, \Gamma, \delta, s, F)$:



- Q, states
- Σ, input alphabet
- Γ tape alphabet,
  $\sqcup \notin \Sigma$: blank symbol,
  $\Sigma \cup \{\sqcup\} \subseteq \Gamma$
- $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R, N\}$,
  transition relation;
- $s \in Q$, initial state
- $F \subseteq Q$, final states

Way Turing machines?

- Historically the first rudiment
  [Turing 1936]
- Original motivation:
  Reduction of the calculation
  of a human mathematical work
- Minimal extension
  of a finite automata
- Church Thesis:
  All adequate possible
  machine models are equivalent

The origin motivation

analog $\rightsquigarrow$ discrete positions, finite alphabet:

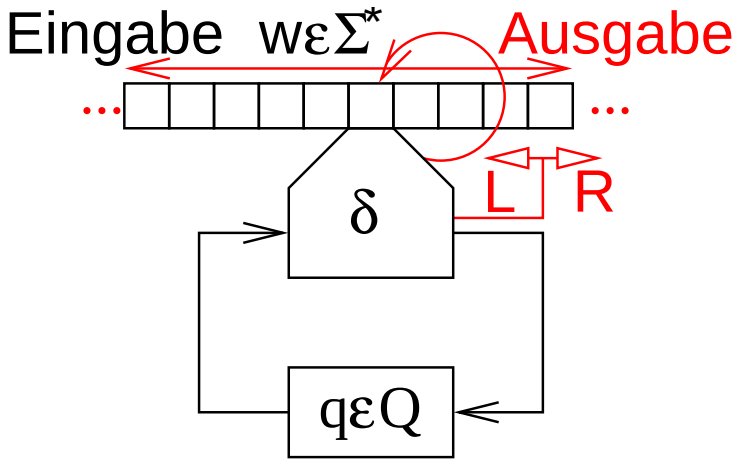
limited precision

papier sheet(2D) $\rightsquigarrow$ tape (1D):

vertical movements$\rightsquigarrow$

many horizontal movements,

lines unmask



The brain follows arithmetical instructions $\rightsquigarrow$ finite automaton

Potentially infinite memory?

+: Unique alternative for huge finite automata

+: The blanks of the start and at the end not memorized

+: When the place is outgoing

more „tapes“ buying

+: When the tape is sold

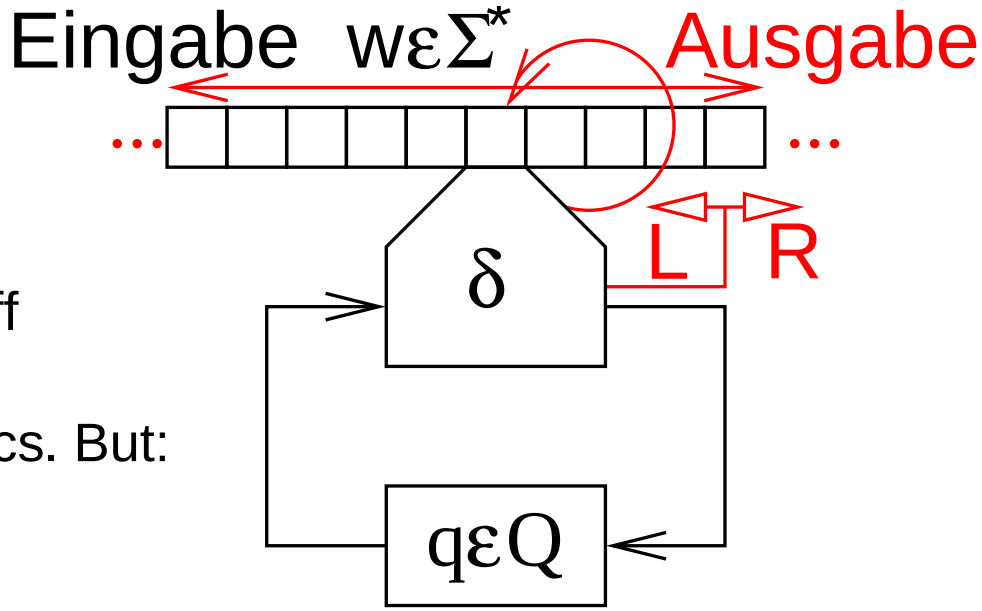
wait for the next technology staff

−: Anytime challenge us the physics. But:

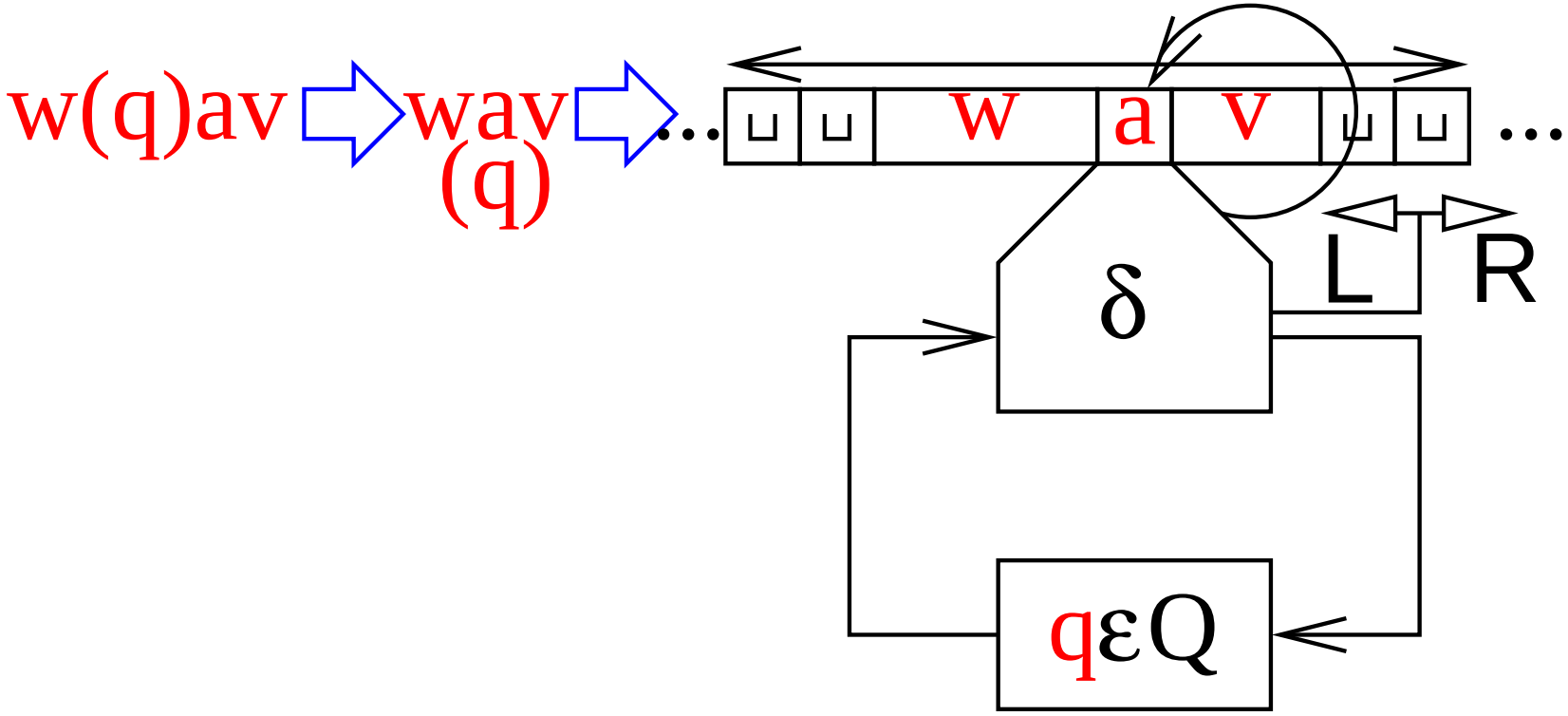
+: Then our sun is going out

+: Till then this is

a useful abstraction



Configuration of a TM



$w, v \in \Gamma^*, a \in \Gamma, q \in Q$

Schöning drops the parentheses out.

Functionality of DTM

$$wa(q)bcv \overset{\delta(q,b)=(q',b',N)}{\vdash} wa(q')b'cv$$

$$wa(q)bcv \overset{\delta(q,b)=(q',b',L)}{\vdash} w(q')ab'cv$$

$$wa(q)bcv \overset{\delta(q,b)=(q',b',R)}{\vdash} wab'(q')cv$$

The transition function δ has three issues:

- New state — like FA
- New tape symbol — overwrites the old symbol in head position
- Moving directions of the head

Functionality of NTM

$$wa(q)bcv \overset{(q',b',N)\in\delta(q,b)}{\vdash} wa(q')b'cv$$

$$wa(q)bcv \overset{(q',b',L)\in\delta(q,b)}{\vdash} w(q')ab'cv$$

$$wa(q)bcv \overset{(q',b',R)\in\delta(q,b)}{\vdash} wab'(q')cv$$

Possible transitions between configurations

When does one DTM halt ?

T halts in configuration $w(q)av$ iff

$\delta(q,a) = (q,a,N)$.

Convention:

$\forall q \in F : \forall a \in \Gamma : \delta(q,a) = (q,a,N)$

(not in Schöning ?)

When does one NTM halt ?

T halts in configuration $w(q)av$ iff

$\delta(q,a) = \{\}$.

Convention:

$\forall q \in F : \forall a \in \Gamma : \delta(q,a) = \{\}$

Graph interpretation

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ defines an

infinite multigraph

Knots: configurations of T.

Edges: admitted configuration transitions from δ.

$w \in L(A) \Leftrightarrow \exists$ path $P = (s)w \rightarrow \cdots \rightarrow u(f)v : f \in F$

Differences DTM versus NTM:

δ appoints configuration transitions versus

δ admits configuration transitions.

Turing machines as acceptors

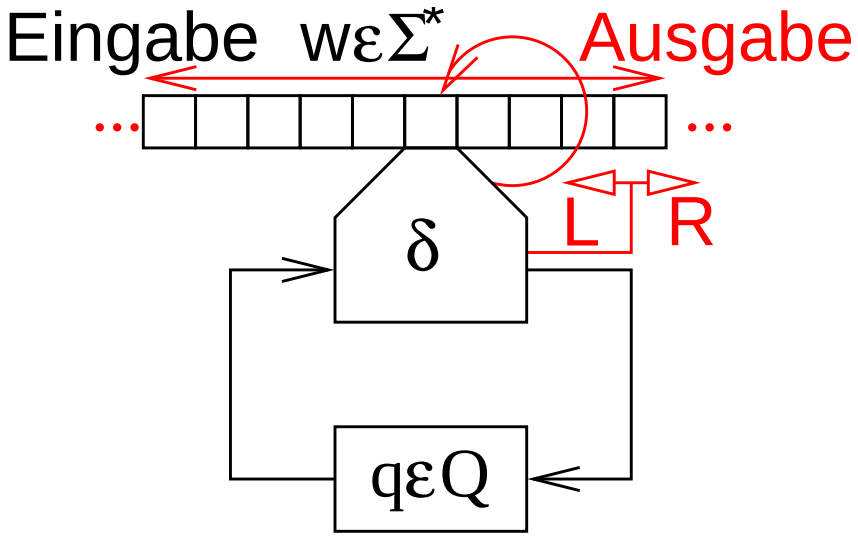
$T = (Q, \Sigma, \Gamma, \delta, s, F)$.

$L(T)$?

T accepts $w \Leftrightarrow$

$\exists \alpha, \beta \in \Gamma^*, f \in F : (s)w \vdash^* \alpha f \beta$

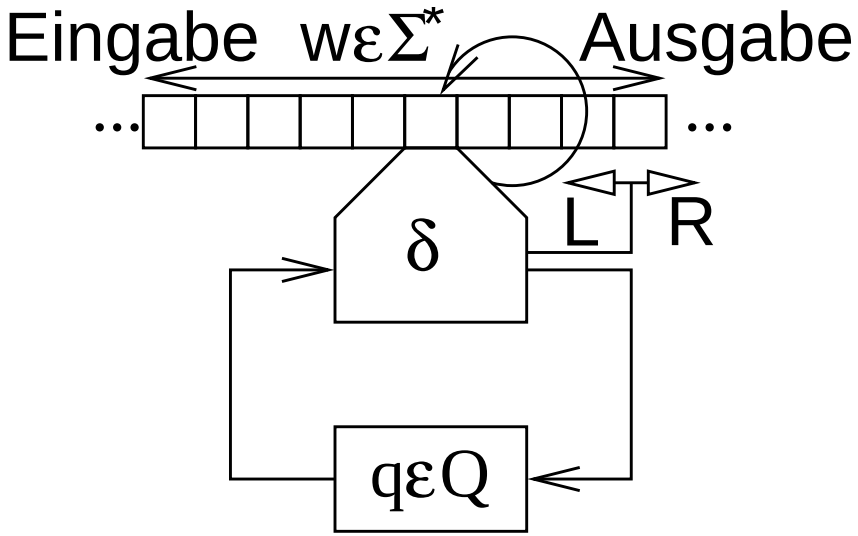
$L(T) := \{w \in \Sigma^* : T \text{ accepts } w\}$.



Graph interpretation

$T = (Q, \Sigma, \Gamma, \delta, s, F)$.

$L(T)$?



Definition:

T accepts $w \in \Sigma^*$ if

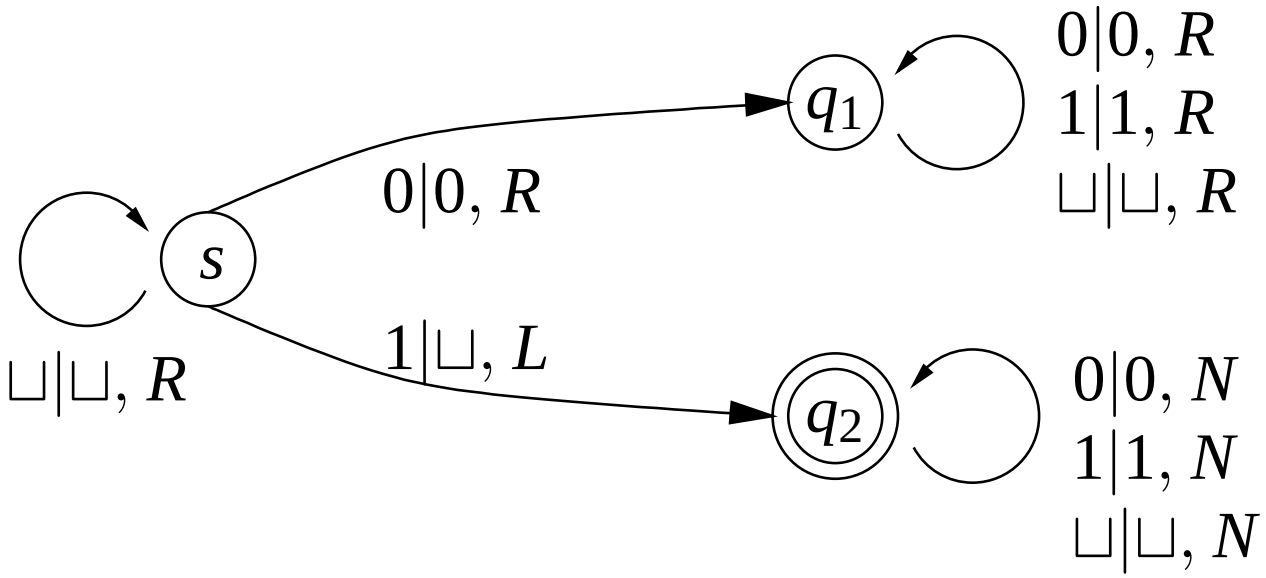
$\exists$ a sequence of (by δ admitted)

configuration transitions

$(s)w \rightarrow \cdots \rightarrow x(f)y$ with $f \in F$.

$L(T) := \{w \in \Sigma^* : T \text{ accepts } w\}$.

Example: acceptors for $L = 1(0,1)^*$

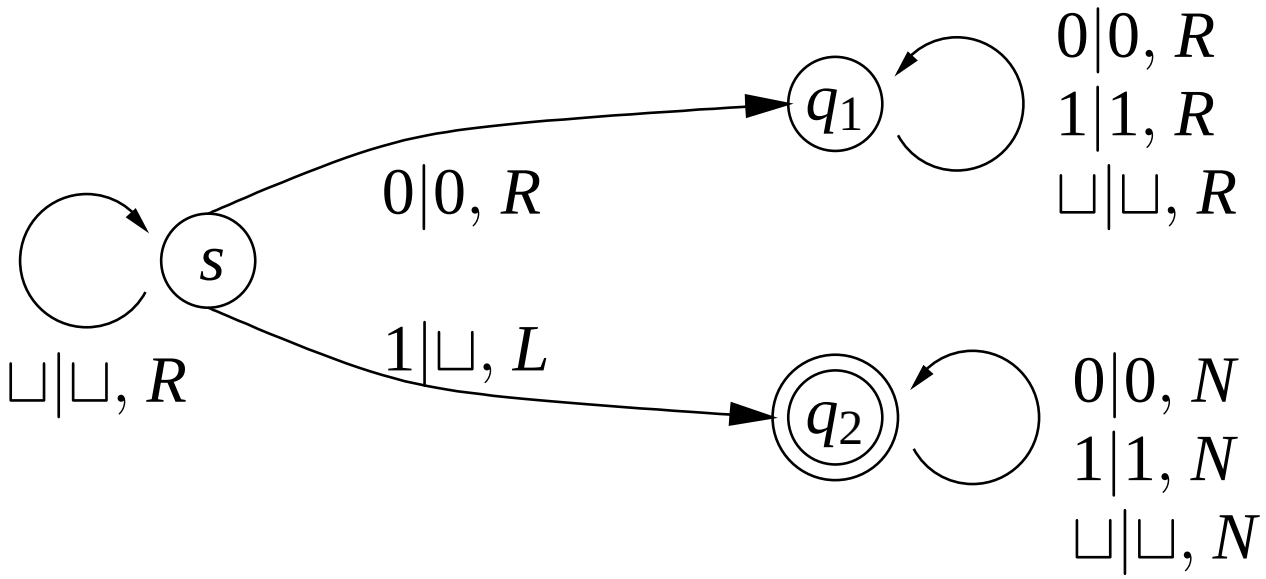


Graphic Notation:

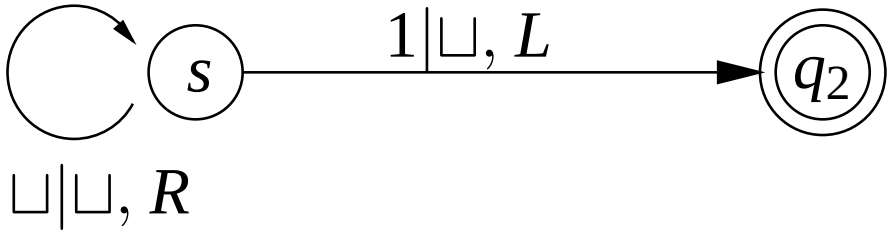
Extension of FA. Extended marking of the edges:

input symbols|output symbols,L,N,R

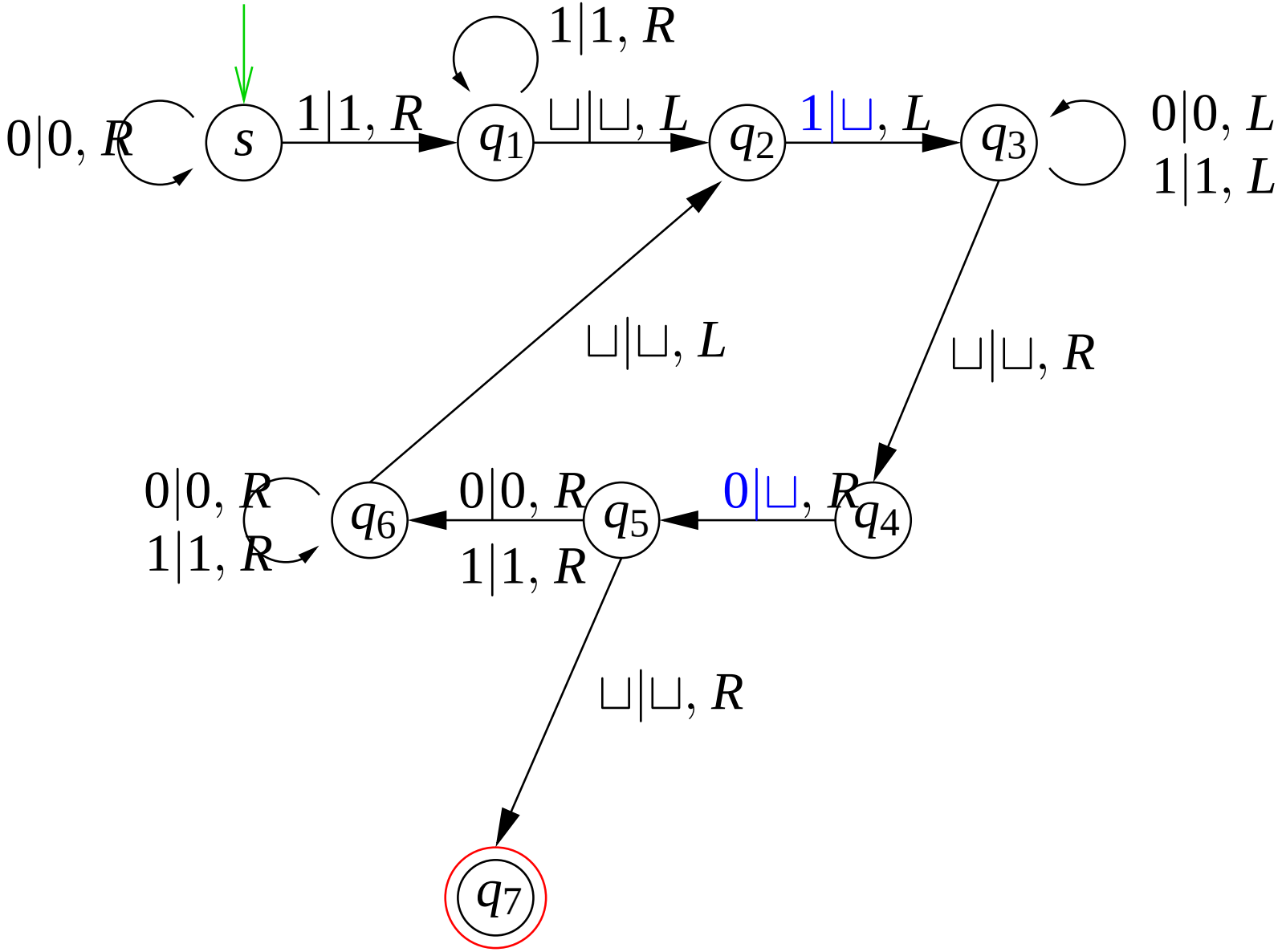
1.5 Integration



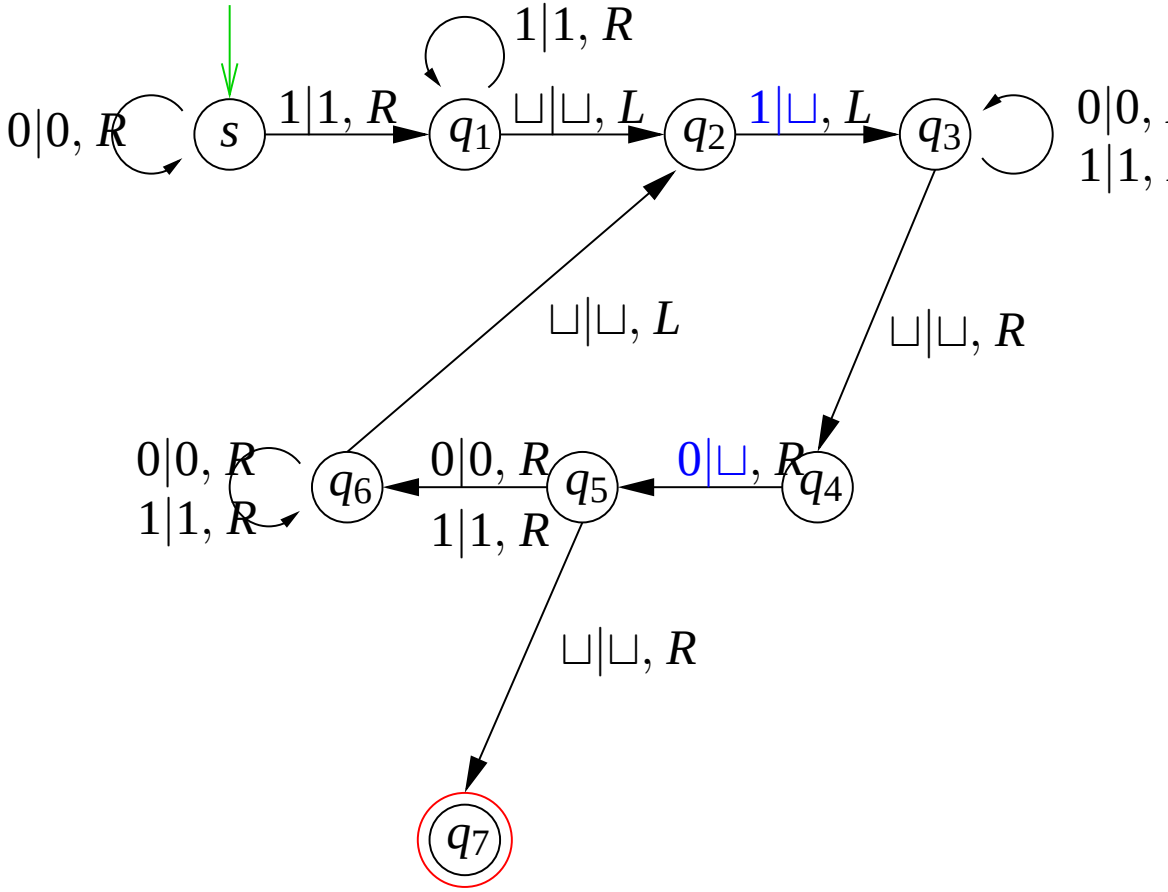
Convention: Error ending transition $\rightsquigarrow$ the TM halts.



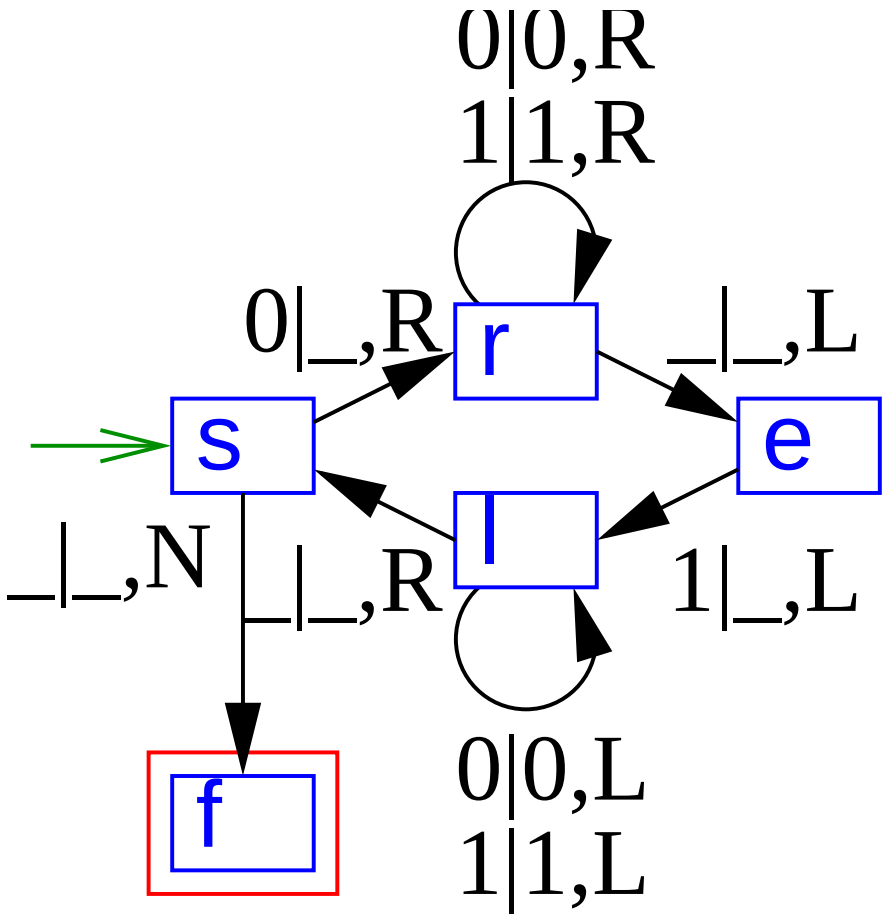
Example: acceptors for $\{0^n 1^n : n \geq 1\}$



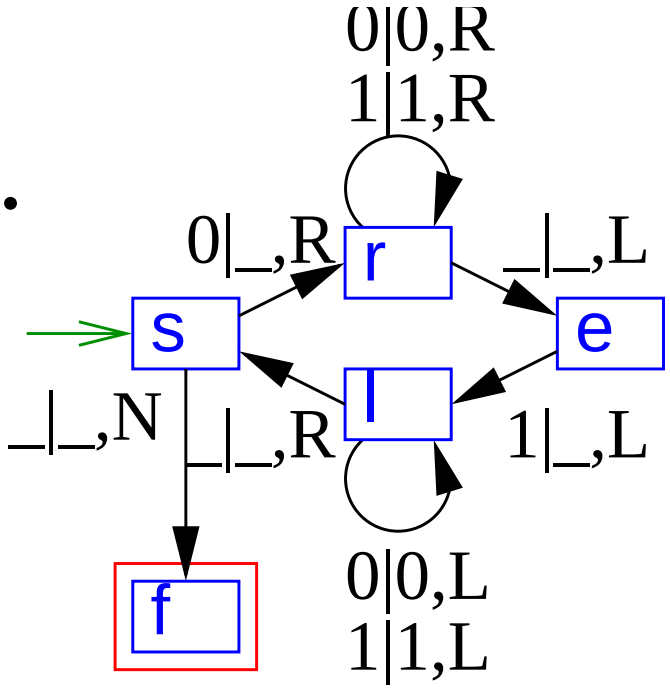
$(s)000111$	$(q_4)00011$	$0(q_2)1$	(q_7)
$0(s)00111$	$\sqcup(q_5)0011$	$(q_3)0\sqcup$	
$00(s)0111$	$0(q_6)011$	$(q_3)\sqcup 0$	
$000(s)111$	$00(q_6)11$	$(q_4)0$	
$0001(q_1)11$	$001(q_6)1$	$\sqcup(q_5)$	
$00011(q_1)1$	$0011(q_6)$		
$000111(q_1)$	$001(q_2)1$		
$00011(q_2)1$	$00(q_3)1\sqcup$		
$0001(q_3)1\sqcup$	$0(q_3)01$		
$0001(q_3)1$	$(q_3)001$		
$000(q_3)11$	$(q_3)\sqcup 001$		
$00(q_3)011$	$(q_4)001$		
$0(q_3)0011$	$\sqcup(q_5)01$		
$(q_3)00011$	$0(q_6)1$		
$(q_3)\sqcup 00011$	$01(q_6)$		



Example: $\{0^n 1^n : n \geq 0\}$.



Example: $\{0^n 1^n : n \geq 0\}$.



Let $k \geq 1$, $w \in \{0,1\}^*$

ε: $(s) \vdash (f)$.

0: $(s)0 \vdash (r) \vdash (e)$ halts.

$1w$: $(s)1w$ halts.

$0w0$: $(s)0w0 \vdash (r)w0 \overset{|w|+1}{\vdash} w0(r) \vdash w(e)0$ halts.

$0w1$: $(s)0w1 \vdash (r)w1 \overset{|w|+1}{\vdash} w1(r) \vdash w(e)1 \overset{|w|+1}{\vdash} (\ell) \sqcup w \vdash (s)w$

$0^n 1^n$: $(s)0^n 1^n \vdash^* (s)0^{n-1}1^{n-1} \vdash^* \cdots \vdash^* (s) \vdash (f)$

$0u1, u \notin \{0^n 1^n : n \geq 0\}$: $(s)0u1 \vdash^* (s)u$. Does not halt in f.

(Induction)

Example Let M be a TM, which stores the first symbol of the input word and stops successfully, if it is not in another place in the word.

$$\delta([s,\sqcup],0) = ([q,0],0,R) \quad \delta([s,\sqcup],1) = ([q,1],1,R)$$

$$\delta([q,0],1) = ([q,0],1,R) \quad \delta([q,1],0) = ([q,1],0,R)$$

$$\delta([q,0],\sqcup) = (f,\sqcup,N) \quad \delta([q,1],\sqcup) = (f,\sqcup,N)$$

Variants of Turing machines

k heads: $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L, R, N\}^k$

k tapes: i.e. one head per a tape

d-dimensional tapes: for example $d = 2$,

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, U, D, N\}$$

probabilistic: additional directions of the reading tape with random bits

LBA:

Linear bounded nondeterministic Turing machines

NTM $T = (Q, \Sigma, \Gamma, \delta, s, F)$ is linear bounded, when

$$\forall a = a_1 \cdots a_n \in \Sigma^+ : (s)a \vdash^* \alpha(q)\beta \longrightarrow |\alpha\beta| \leq n$$

Proposition: $\forall$type 1 language L : $\exists$LBA T : $L(T) = L$

Proof sketch: Let $G = (V, \Sigma, P, S)$-type 1 grammar, $L(G) = L$.

Consider the NTM $T = (Q, \Sigma, (\Sigma \cup V), \delta, s, F)$:

Procedure inL(z) // initial configuration $(s)z$

invariant "tape content"$\stackrel{*}{\Rightarrow} z$

invariant $|\text{tape}| \leq |z|$

while tape$\neq S$ **do**

if $\exists w \to \alpha \in P$: tape $= x\alpha y$ **then** // $2\times$ nondet. choice!

tape:= xwy // contracting rule!

else reject z

accept z

accepting computation $\stackrel{Inv.}{\longrightarrow} \exists$ derivation.

$S \stackrel{*}{\Rightarrow} z \longrightarrow \exists$ corresponding accepting computation .

Subprogram: search for an appropriate left side

- Let G be in Kuroda-normal form $\longrightarrow \left|\text{right letters}\right| \leq 2$
- run to the left margin
- run to the right over the tape:
  - the states stored the tape symbol L left to the head
  - δ could depend of L, actual tape symbol and knowledge of P (finite !) determining if there is a more suitable production
  - Yes? $\longrightarrow$ Do the substitution
  - No? Proceed

Subprogram: replacement for $AB \rightarrow CD$

- One step to the left
- Write C
- One step to the right
- Write D
- Back to the main loop

Subprogram: replacement for $AB \rightarrow C$

Write C; one step to the left

write $\sqcup$

go to the left of the tape

memorize the first tape symbol in the state

repeat

swap the stored and actual tape symbol; one step to the right

until $\sqcup$ overwritten

Proposition: $\forall L : \exists$LBA $T : L(T) = L \longrightarrow L$ is of type 1 language.

Proof (sketch): Let $T = (Q, \Sigma, \Gamma, \delta, s, F)$ be LBA with $L(T) = L$.

Consider a type 1 grammar

$G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.

Idea: TM-configuration $\alpha(q)a\beta \rightsquigarrow$ proposition form $\alpha(q,a)\beta$ plus extra info for the original input.

3 phases of the derivation:

1. generate word from Σ^*.
2. simulate computation of TM.
3. after acceptance regenerate the input word

Phase 1: generate word from Σ^*

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA with $L(T) = L$.

$G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.

$\{S \to S(a,a) : a \in \Sigma\} \subseteq P$

$\{S \to (s,a,a) : a \in \Sigma\} \subseteq P$ end of phase 1.

(and special treatment if $\varepsilon \in L$)

Example: $S \Rightarrow S(c,c) \Rightarrow S(b,b)(c,c) \Rightarrow (s,a,a)(b,b)(c,c)$

corresponds to the initial configuration $(s)abc$

Phase 2: simulate the computation of TM

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA with $L(T) = L$.

$G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.

$$
\begin{aligned}
P := P \cup \{ & (q,a,c) \to (q',a',c) && : (q',a',N) \in \delta(q,a), c \in \Sigma \} \\
\cup \{ & (b,c')(q,a,c) \to (q',b,c')(a',c) && : (q',a',L) \in \delta(q,a), c \in \Sigma \} \\
\cup \{ & (q,a,c)b \to a'(q',b,c) && : (q',a',R) \in \delta(q,a), c \in \Sigma \}
\end{aligned}
$$

Example: $(s,a,a)(b,b)(c,c) \stackrel{*}{\Rightarrow} (x,a)(f,y,b)(z,c)$

correspond to the configuration sequence $(s)abc \vdash \cdots \vdash x(f)yz$

Phase 3: regenerate input word

$T = (Q, \Sigma, \Gamma, \delta, s, F)$ LBA with $L(T) = L$.

$G = (V = \{S\} \cup (\Gamma \times \Sigma) \cup (Q \times \Gamma \times \Sigma), \Sigma, P, S)$.

$\{(f,a,c) \to c : f \in F, a \in \Gamma, c \in \Sigma\} \subseteq P$ Acceptance

$\{(a,b) \to b : a \in \Gamma \wedge b \in \Sigma\} \subseteq P$

Example: $(x,a)(f,y,b)(z,c) \Rightarrow (x,a)b(z,c) \Rightarrow ab(z,c) \Rightarrow abc$

Closure properties of type 1

Closure under

- union $\cup$
- concatenation $\cdot$
- star $*$
- intersection $\cap$
- complement $\overline{\cdot}$

Closure under union of type 0/1

$L(A_1) \cup L(A_2)$ has type 0/1 ?

Two subprograms U_1 for $L(A_1)$ and U_2 for $L(A_2)$.

New program for $U_1 \vee U_2$.

No additional place is necessary.

Exercise: Closure under intersection of type 0/1

Closure under concatenation of type 0/1

$L(A_1) \cdot L(A_2)$ have type 0/1 ?

Two subprograms U_1 for $L(A_1)$ and U_2 for $L(A_2)$.

New programm for $w \in L(A_1) \cdot L(A_2)$?:

Split $w = w_1 w_2$ (nondeterministic).

return $w_1 \in L(A_1) \wedge w_2 \in L(A_2)$

Go without extension of the tape by means of tape marking.

Exercise: Closure under Kleene's hull ($*$) of type 0/1

Closure under complement of type 1

Given: $L = L(G) \subseteq \Sigma^*$, $G = (V, \Sigma, P, S)$.

Idea: find LBA M, which accepts $x \in \Sigma^n$, $x \notin L$.

$a := \left| \left\{ \alpha \in (V \cup \Sigma)^* : |\alpha| \leq n \wedge S \stackrel{*}{\Rightarrow} \alpha \right\} \right|$ // todo

$c := 0$

foreach $\alpha \in (V \cup \Sigma)^* \setminus \{x\}$ with $|\alpha| \leq n$ **do**

  if $S \stackrel{*}{\Rightarrow} \alpha$ **then** $c{+}{+}$

  else continue // nondeterministic fail !

return $c = a$

Compute $\left|\left\{\alpha \in (V \cup \Sigma)^* : |\alpha| \leq n \wedge S \stackrel{*}{\Rightarrow} \alpha\right\}\right|$

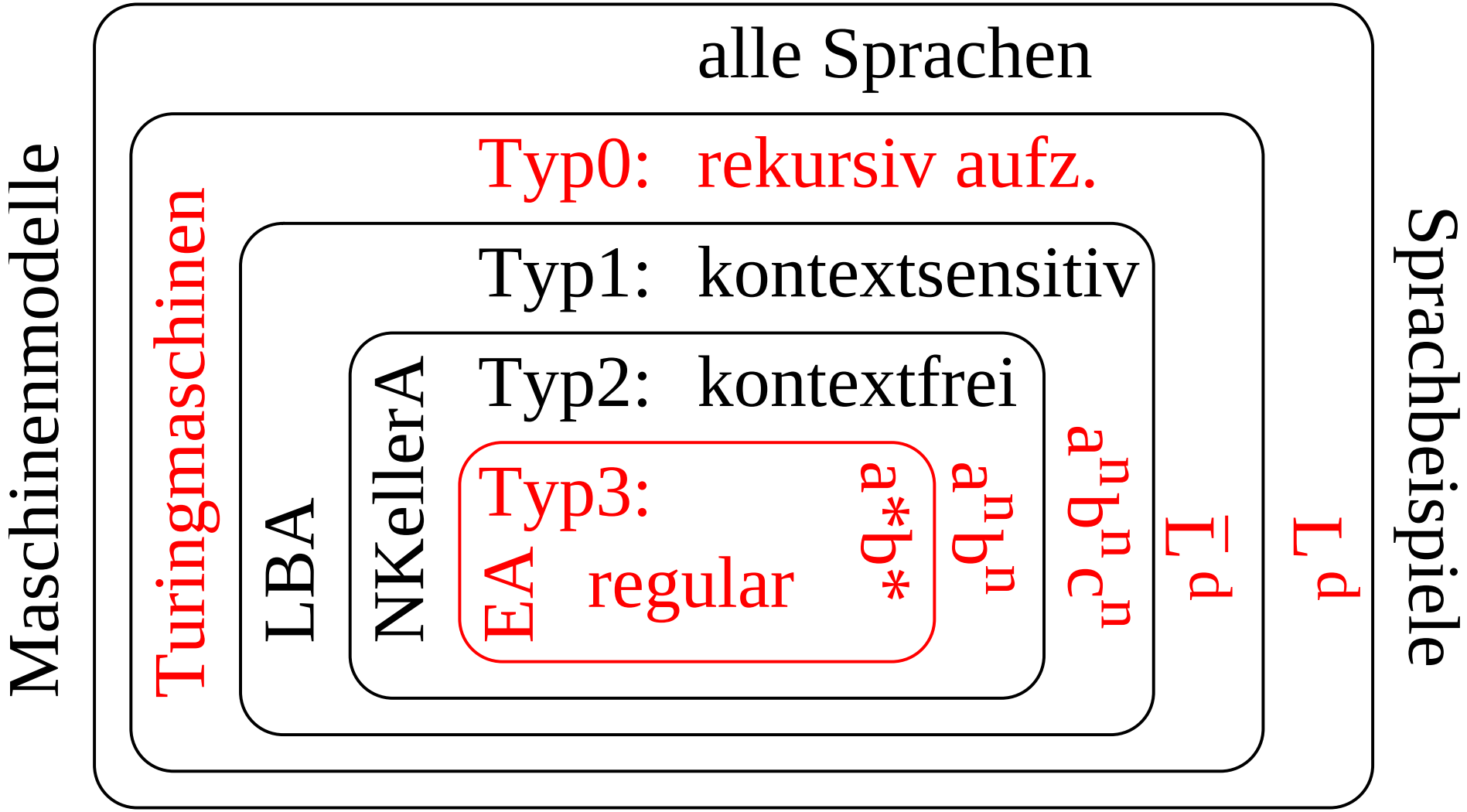
```
a:= 1                                                    // |{S}|
for m := 1 to ∞ do                                       // end of derivation
    b:= 0                                                // counter for new or old words
    foreach w' ∈ (V ∪ Σ)* with |α| ≤ n do
        z:= 0                                            // counter for old words
        foreach w ∈ (V ∪ Σ)* with |α| ≤ n do
            if S ⇒^≤m w then z++                         // nondeterministic
            if w = w' ∨ w ⇒ w' then b++
        if z ≠ a then fail
    if a = b then break loop
    a:= b
```

type 0 languages

Proposition: L accepted by TM $\Leftrightarrow$ L has type 0

Proof: Analogously as in the proof for type 1.

Outline of Chomsky-Hierarchy



type	description
3	right linear or left linear grammar DFA, $\overline{\varepsilon}$NFA, εNFA regular expression
Det. CF	LR(k) grammar DKellerA with final st.
2	context-free grammar (1-sate)NKellerA
1	context-sensitive grammar LBA
0	type 0 grammar Turing machine

type	Nondet.	Deterministic	equivalent?
3	NFA	DFA	yes
2	NKellerA	DKellerA	no
1	LBA	DLBA	???
0	NTM	DTM	yes (spec. comp.)

Closure properties

type	$\cap$	$\cup$	$\bar{\cdot}$	$\cdot$	$*$
3	yes	yes	yes	yes	yes
Det. KF	no	no	yes	no	no
2	no	yes	no	yes	yes
1	yes	yes	yes	yes	yes
0	yes	yes	no	yes	yes

Decidability problem

type	Word-	emtiness-	equivalence	intersection empty
3	yes	yes	yes	yes
Det. CF	yes	yes	yes [97]	no
2	yes	yes	no	no
1	yes	no	no	no
0	no	no	no	no

Complexity of the word problems

type	complexity
3	$\mathcal{O}(n)$
Det. CF	$\mathcal{O}(n)$
2	$\mathcal{O}(n^3)$
1	$\lvert\Sigma\rvert^{\mathcal{O}(n)}$, “NP-hard”$\rightsquigarrow$ complexity theory
0	“computable enum.” $\rightsquigarrow$ computability

Chomsky-Hierarchy: A criticism

2: (Only?) these grammars are “exactly right”

3: “by accident” do the linear productions likewise one finite automaton

0,1: context-sensitive rules are of too low level and too similar to TM to make an interesting model implementation

1: One of the many decisions of the special case?
Why exactly the linear restriction of the place?
There are beneficially generalization of CFGs.