Linear Context-Free Languages and Watson-Crick Automata

Benedek Nagy

Department of Mathematics,
Eastern Mediterranean University
Famagusta, North Cyprus

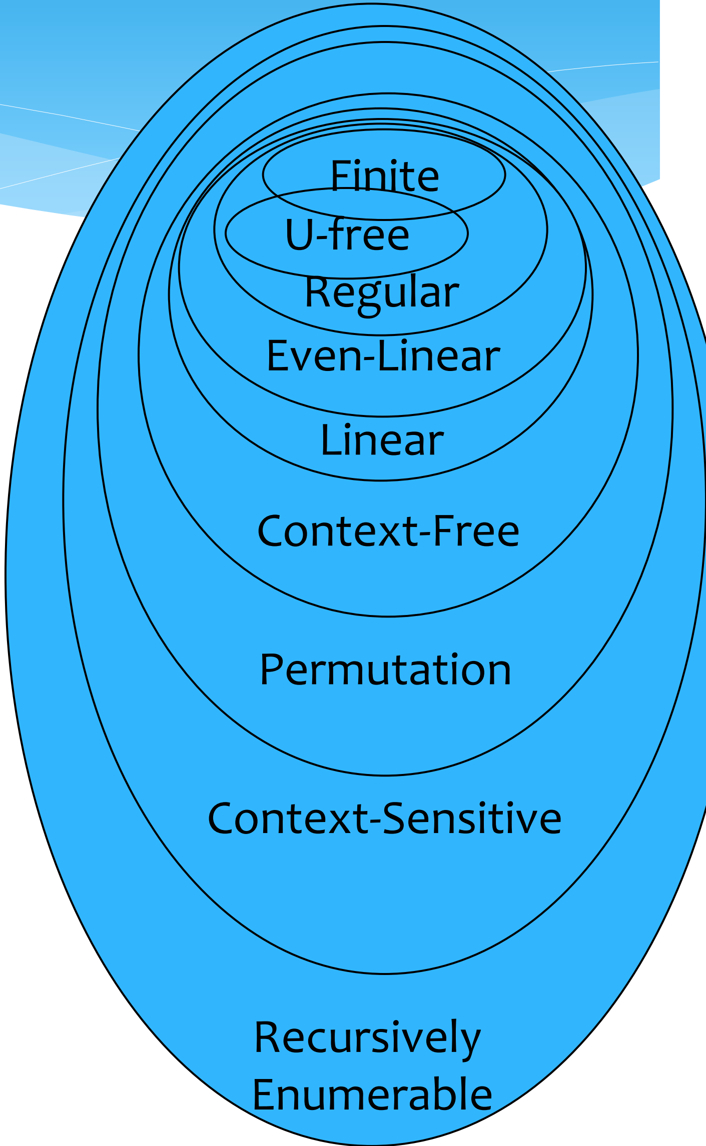
november 2017, Brno

Outline

- Classical computing (Chomsky hierarchy)
- DNA, DNA computing
- Watson – Crick finite automata
- 5’ – 3’ direction
- Sensing
- 5’ – 3’ WK automata and linear languages
- Restricted models
- Langue hierarchy

An extended Chomsky hierarchy

- Finite
- Union-free regular
- Regular
- Even/Fix-rated linear
- Linear
- Context-free
- Permutation
- Context-sensitive
- Recursively enumerable



An extended Chomsky hierarchy

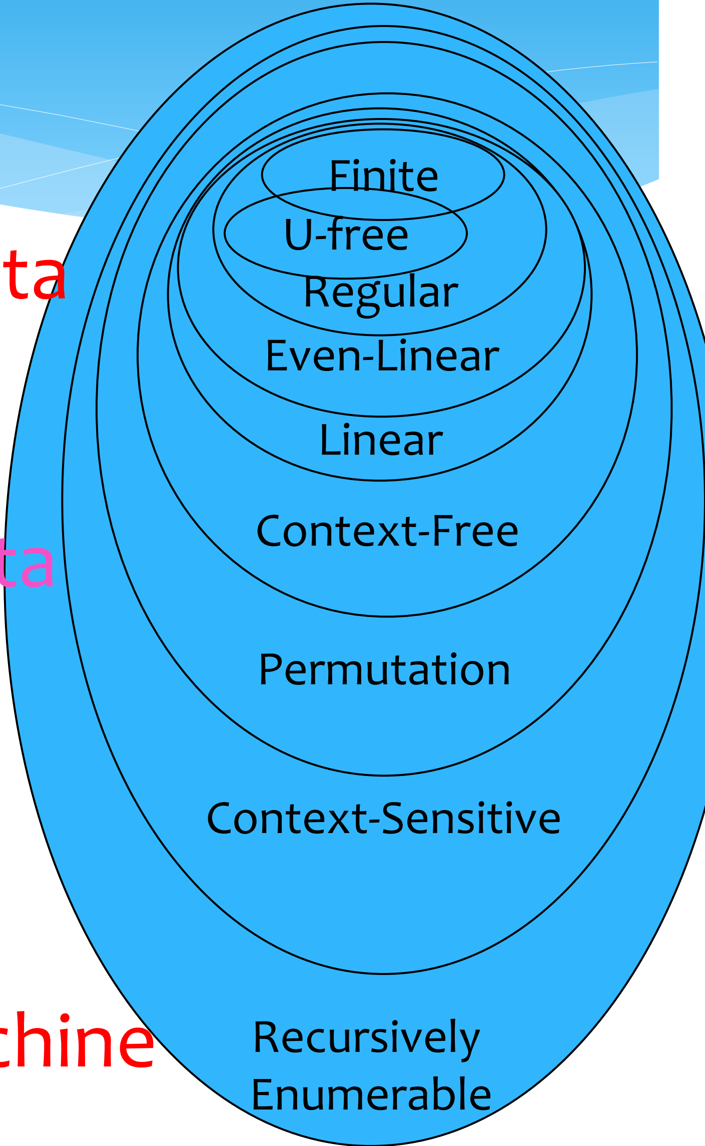
- Finite
- Union-free regular
- Regular
- Even/Fix-rated linear
- Linear
- Context-free
- Permutation
- Context-sensitive
- Recursively enumerable

Finite automata

2-head (WK) finite automata

Pushdown automata

LBA, Turing machine



Finite

U-free

Regular

Even-Linear

Linear

Context-Free

Permutation

Context-Sensitive

Recursively Enumerable

Finite automata

- (Q,s,V,F,d)
  Q: set of states, s: initial state (in Q)
  V: input alphabet (terminal alphabet in grammars)
  F: set of final states (subset of Q)
  d: transition function
- Deterministic: $d:QxV \to Q$
- Non-deterministic: $d:Qx(VU\{\lambda\}) \to 2^Q$
  (ε is also used in the role of the empty word)
- NOTE: allowing to read strings or even ” regular expressions”, still exactly the regular languages are accepted!

$$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$$

$(\{S,A,B\},\{a,b,c\},S,\{S\to aaS, S\to bA, A\to bA, A\to babB, B\to cA, B\to cc\})$

S

S

S

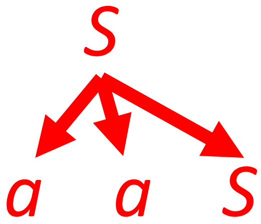
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$

S

S

S

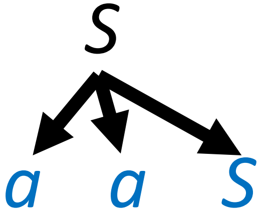
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aaS

$S \Rightarrow aaS$

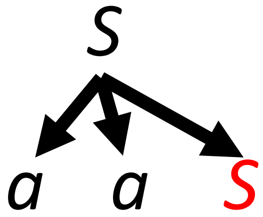
$(\{S,A,B\},\{a,b,c\},S,\{S\to aaS, S\to bA, A\to bA, A\to babB, B\to cA, B\to cc\})$



aaS

$S \Rightarrow aaS$

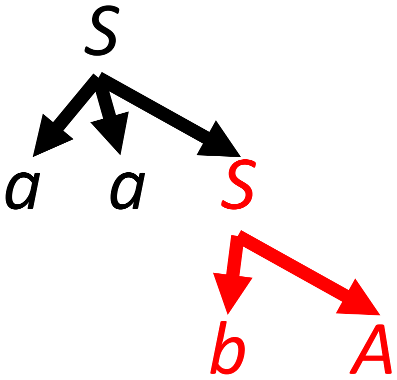
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aaS

$S \Rightarrow aaS$

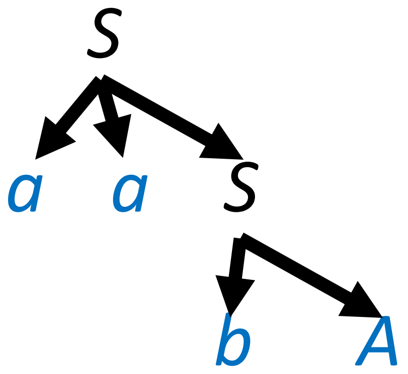
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



$aabA$

$S\Rightarrow aaS\Rightarrow aabA$

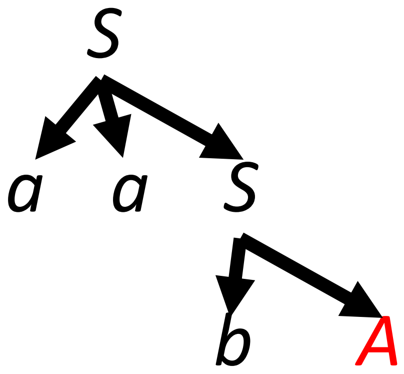
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aabA

$S \Rightarrow aaS \Rightarrow aabA$

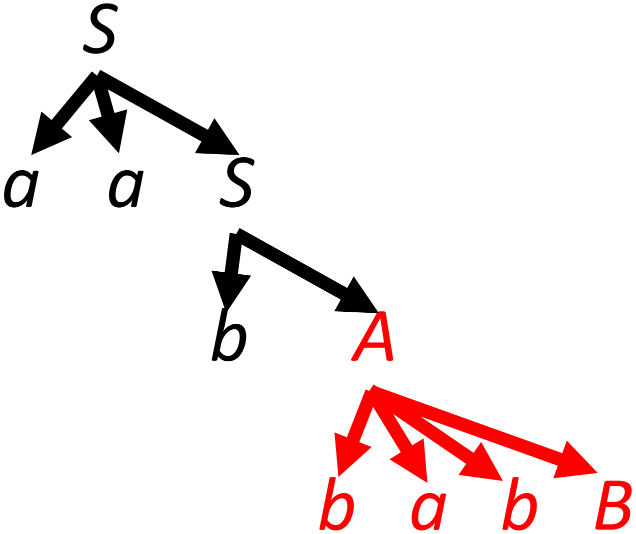
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS, S\rightarrow bA, A\rightarrow bA, A\rightarrow babB, B\rightarrow cA, B\rightarrow cc\})$



$aabA$

$S \Rightarrow aaS \Rightarrow aabA$

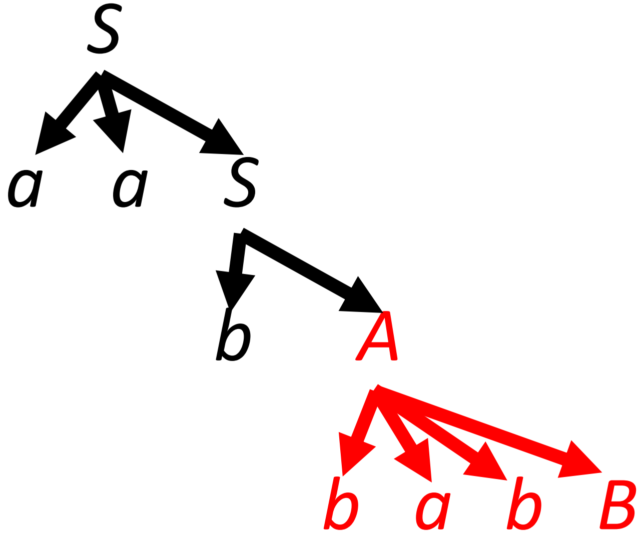
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



$aabA$

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow$

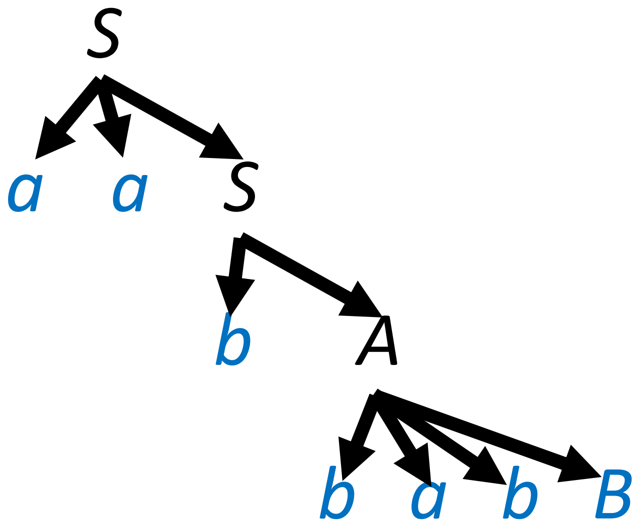
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS, S\rightarrow bA, A\rightarrow bA, A\rightarrow babB, B\rightarrow cA, B\rightarrow cc\})$



$aabbabB$

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB$

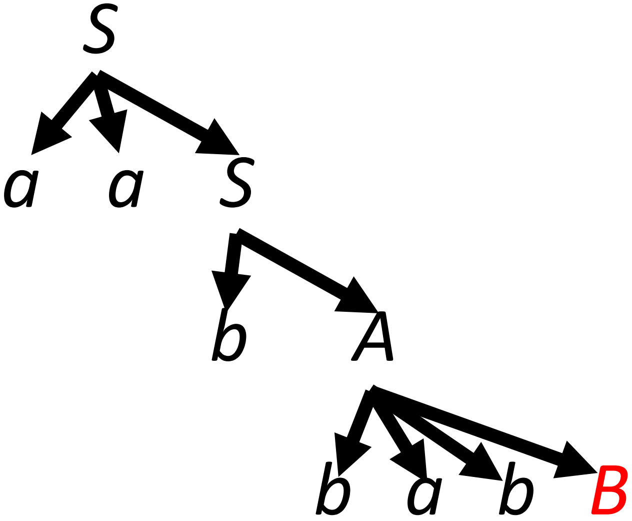
$(\{S,A,B\},\{a,b,c\},S,\{S \rightarrow aaS, S \rightarrow bA, A \rightarrow bA, A \rightarrow babB, B \rightarrow cA, B \rightarrow cc\})$



$aabbabB$

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB$

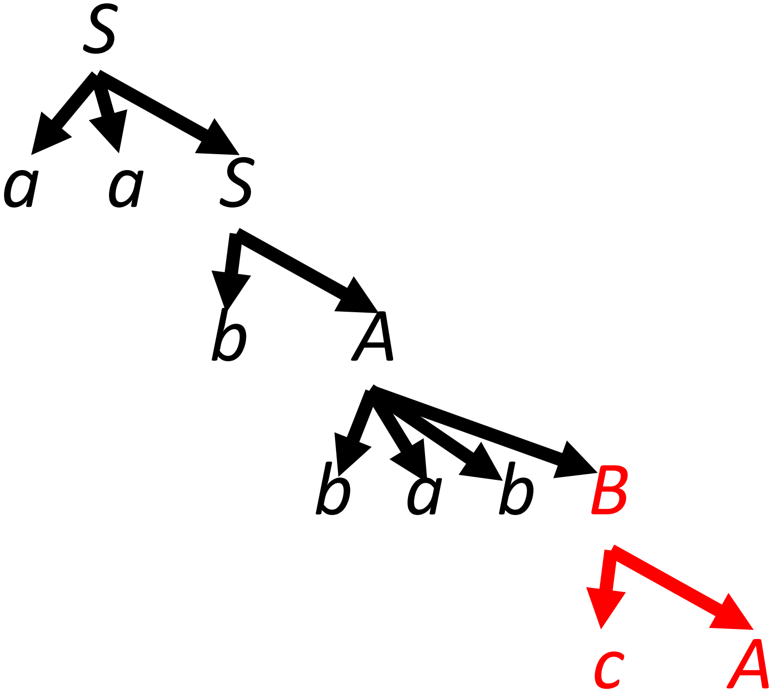
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



$aabbabB$

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB \Rightarrow$

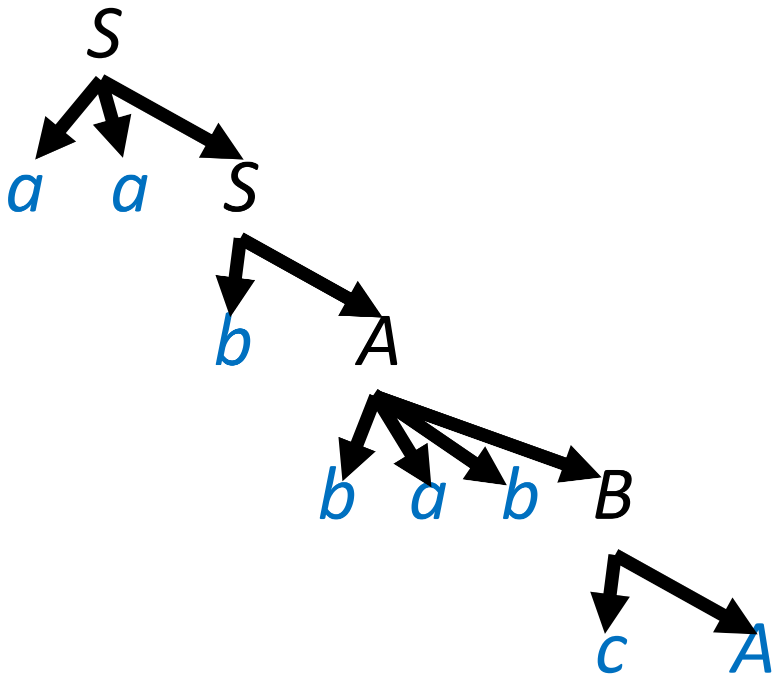
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aabbab cA

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB \Rightarrow aabbabcA$

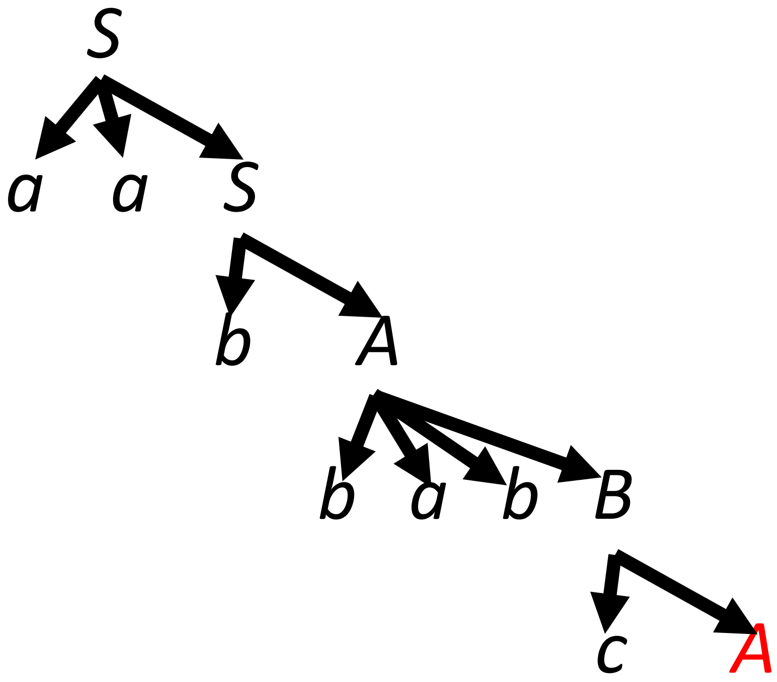
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aabbabcA

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB \Rightarrow aabbabcA$

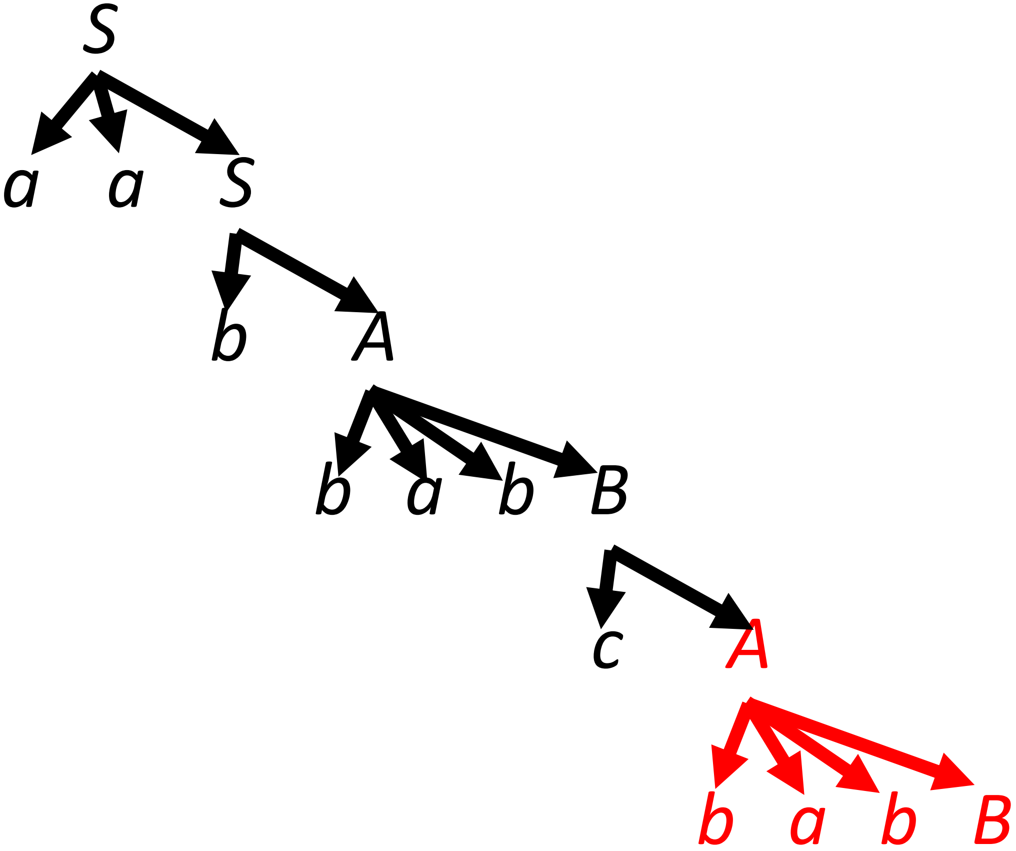
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



$aabbabcA$

$S\Rightarrow aaS\Rightarrow aabA\Rightarrow aabbabB\Rightarrow aabbabcA$

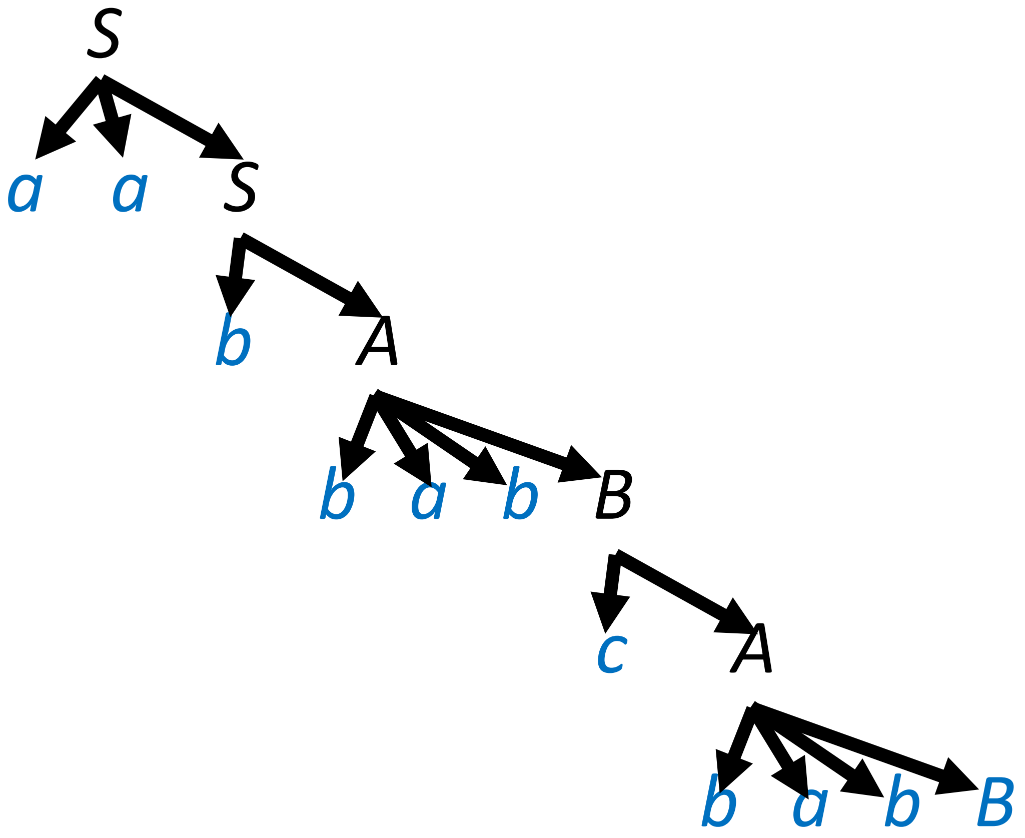
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aabbabcbabB

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB \Rightarrow aabbabcA \Rightarrow aabbabcbabB$

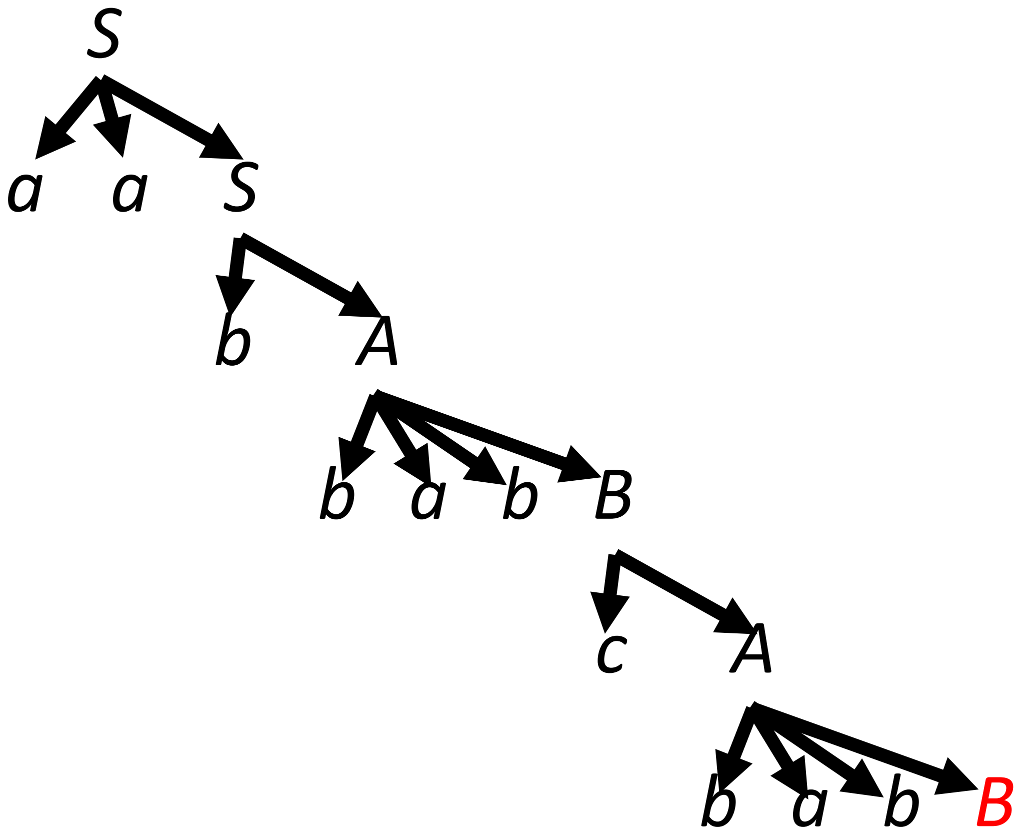
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS, S\rightarrow bA, A\rightarrow bA, A\rightarrow babB, B\rightarrow cA, B\rightarrow cc\})$



aabbabcbabB

$S \Rightarrow aaS \Rightarrow aabA \Rightarrow aabbabB \Rightarrow aabbabcA \Rightarrow aabbabcbabB$

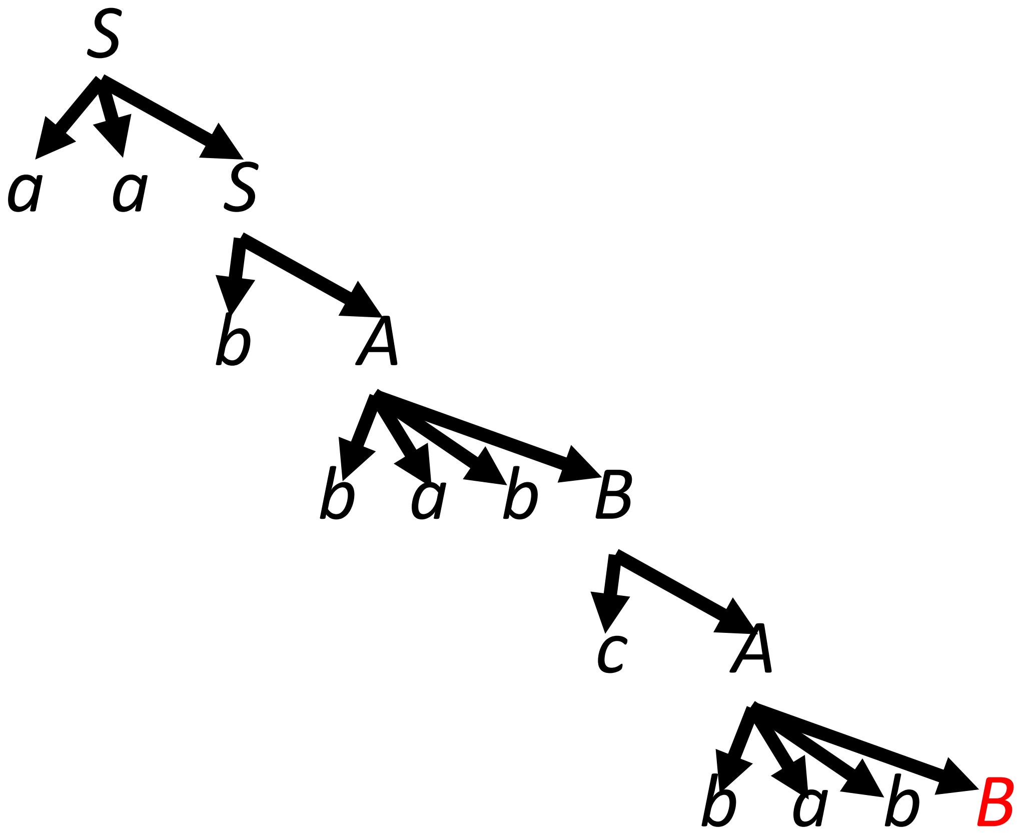
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS, S\rightarrow bA, A\rightarrow bA, A\rightarrow babB, B\rightarrow cA, B\rightarrow cc\})$



$aabbabcbabB$

$S \Rightarrow aaS \Rightarrow^{*} aabbabcA \Rightarrow aabbabcbabB$

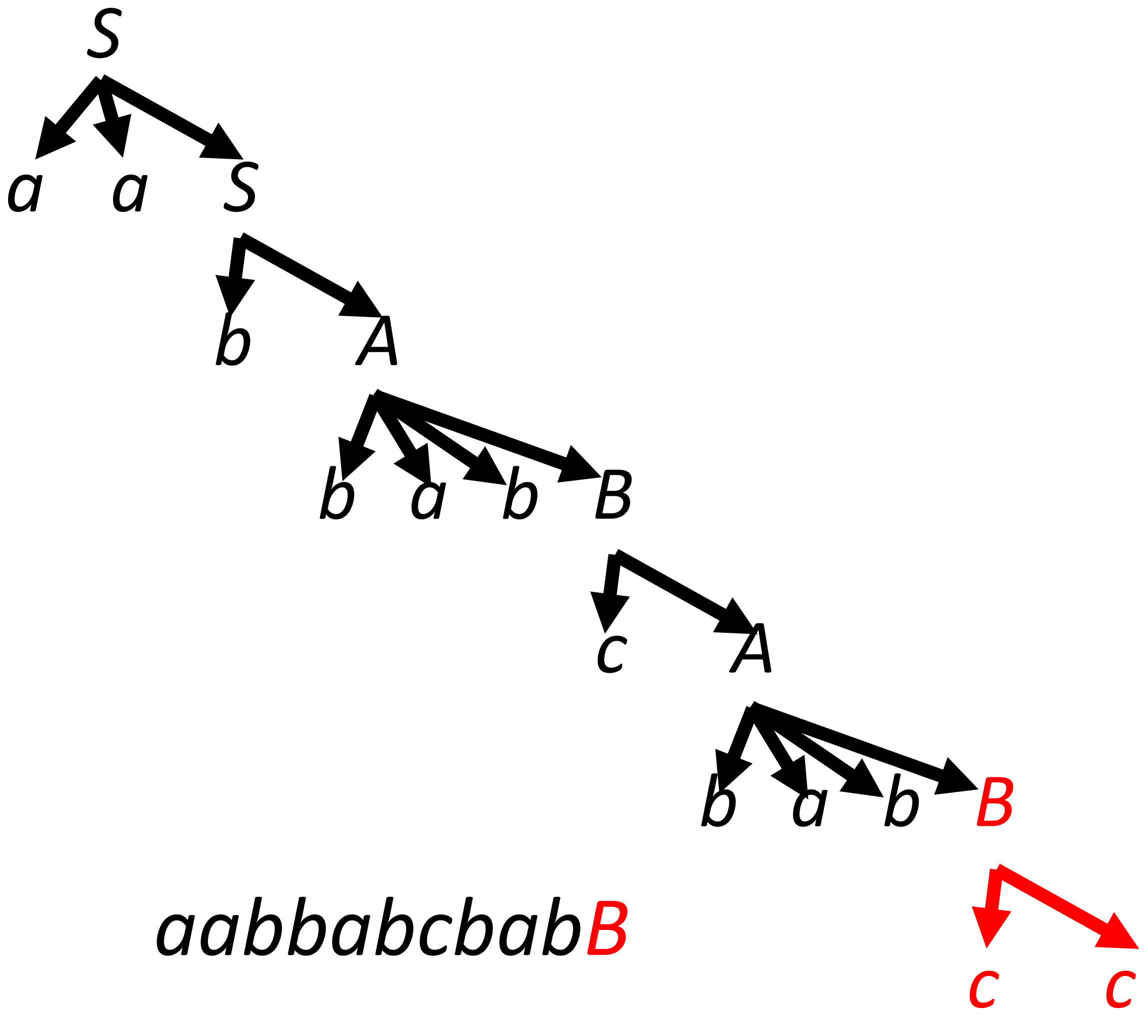
$(\{S,A,B\},\{a,b,c\},S,\{S\rightarrow aaS,S\rightarrow bA,A\rightarrow bA,A\rightarrow babB,B\rightarrow cA,B\rightarrow cc\})$



aabbabcbabB

$S \Rightarrow aaS \Rightarrow^{*} aabbabcA \Rightarrow aabbabcbabB \Rightarrow$

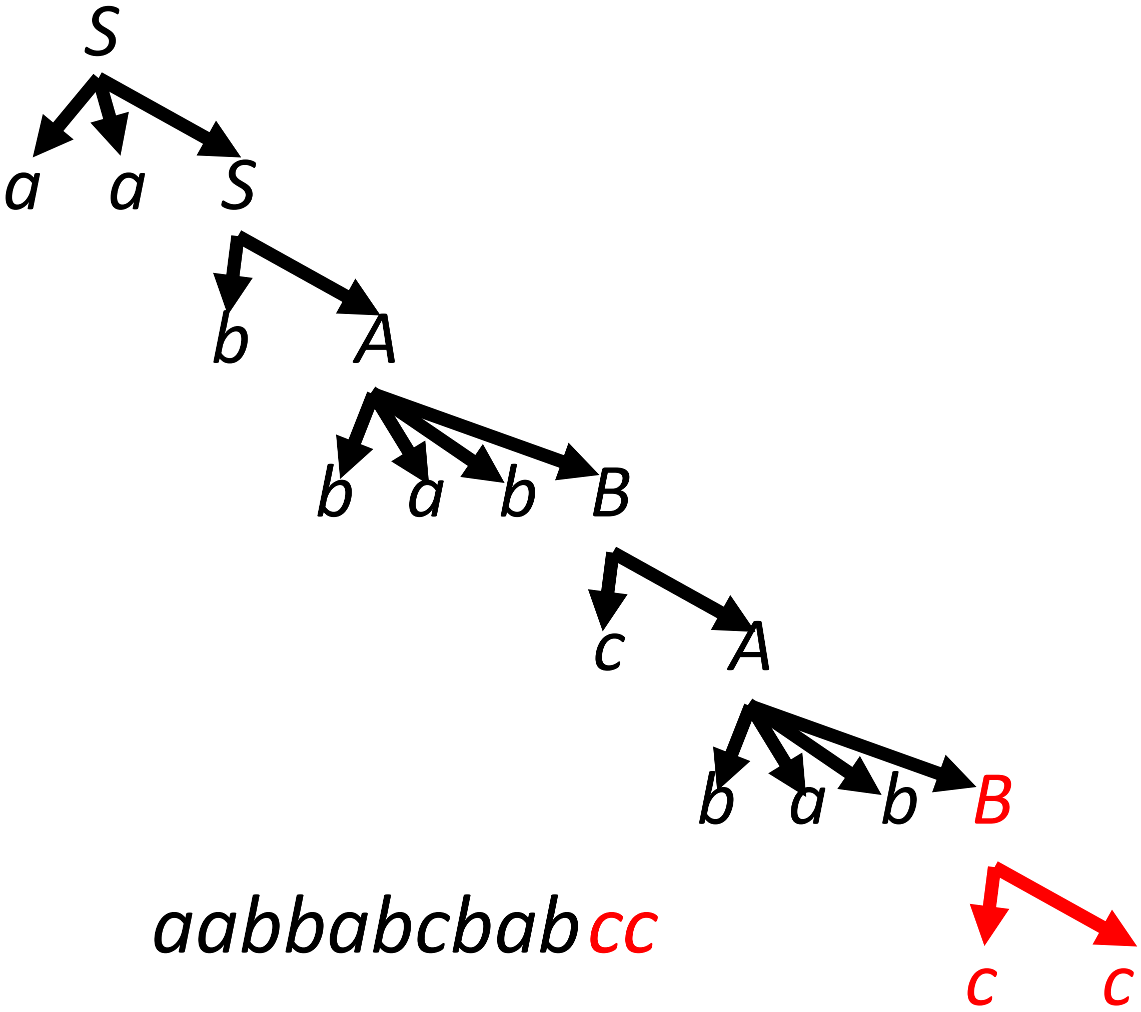
$(\{S,A,B\},\{a,b,c\},S,\{S\to aaS,S\to bA,A\to bA,A\to babB,B\to cA,B\to cc\})$



aabbabcbabB

$S \Rightarrow aaS \Rightarrow^{*} aabbabcA \Rightarrow aabbabcbabB \Rightarrow$

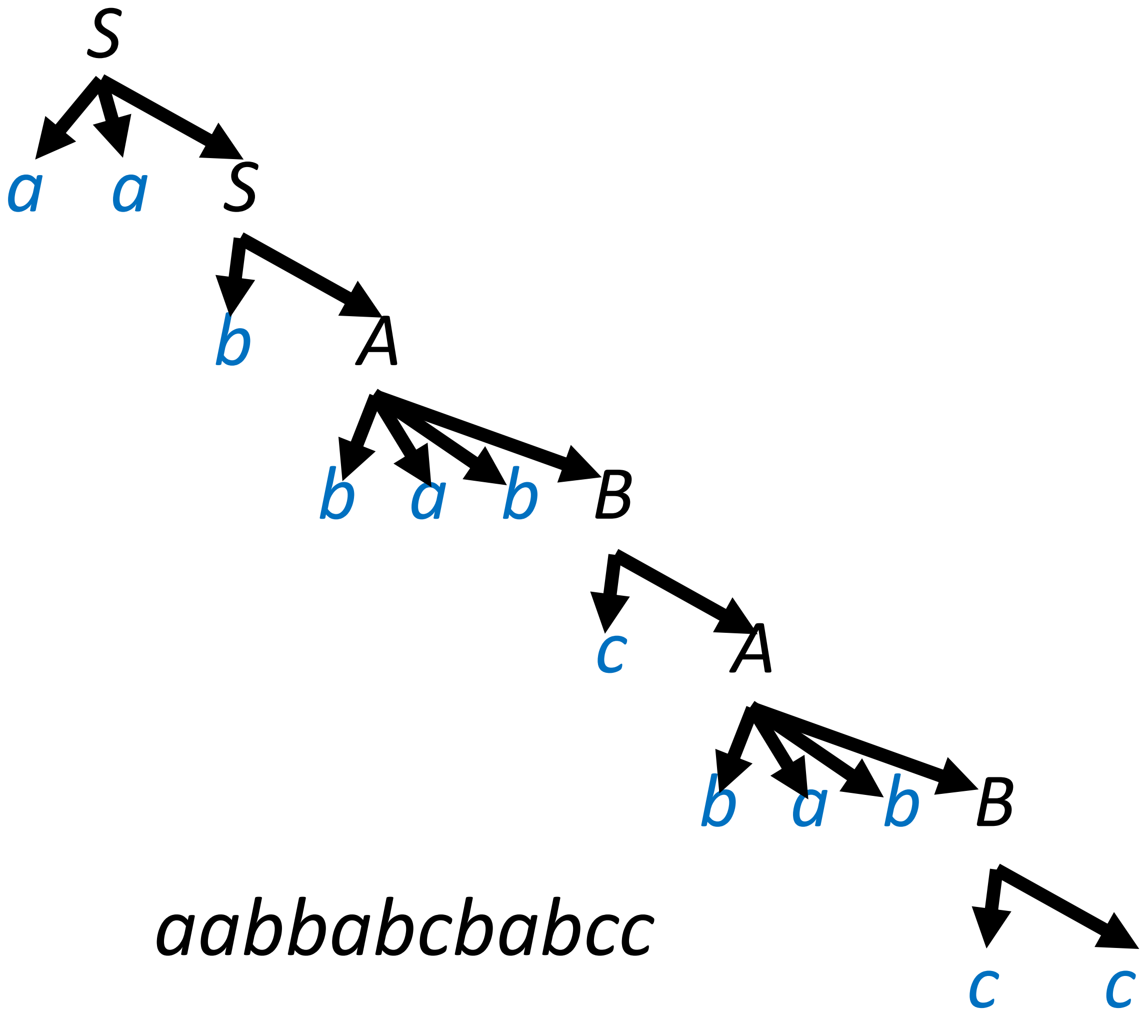
$(\{S,A,B\},\{a,b,c\},S,\{S\to aaS,S\to bA,A\to bA,A\to babB,B\to cA,B\to cc\})$



aabbabcbabcc

$S \Rightarrow aaS \Rightarrow^* aabbabcA \Rightarrow aabbabcbabB \Rightarrow aabbabcbabcc$

$(\{S,A,B\},\{a,b,c\},S,\{S\to aaS,S\to bA,A\to bA,A\to babB,B\to cA,B\to cc\})$



aabbabcbabcc

$S \Rightarrow^*$ *aabbabcb*

Linear languages

- Definition by grammar:

$$A \rightarrow v,\ A \rightarrow vBw$$

- Normal form for the grammar:

$$A \rightarrow aB, A \rightarrow Ba, A \rightarrow a\ (A, B \in N, a \in T)$$

- Even-linear languages (normal form):

$$A \rightarrow aBb, A \rightarrow a.\ A \rightarrow \lambda$$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$

S

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$

S

S

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$

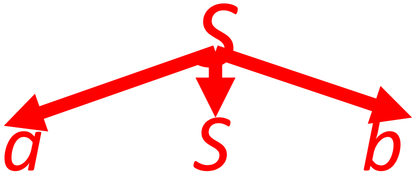
S

S

$$(\{S\}, \{a, b\}, S, \{\underset{S}{S \rightarrow aSb}, S \rightarrow \lambda\})$$

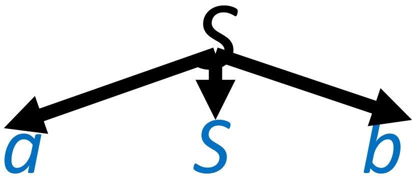
$S \Rightarrow$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



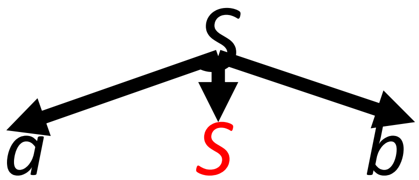
$S \Rightarrow aSb$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



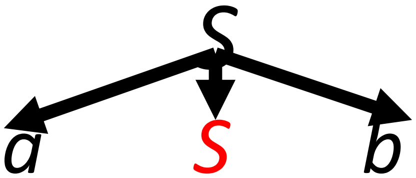
$$S \Rightarrow aSb$$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



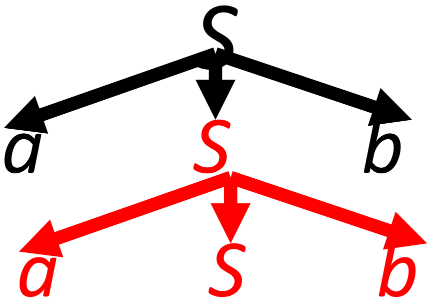
$$S \Rightarrow aSb$$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



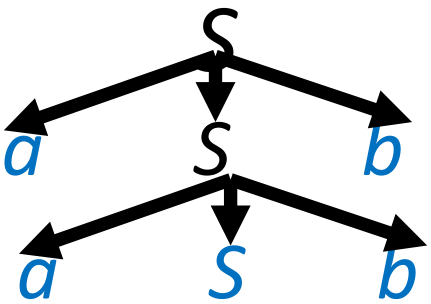
$$S \Rightarrow aSb \Rightarrow$$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



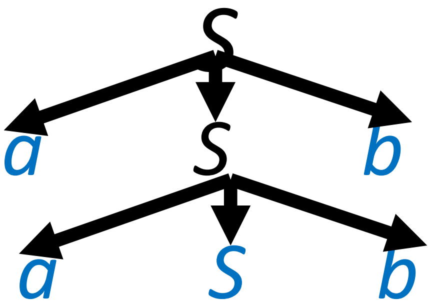
$S \Rightarrow aSb \Rightarrow aaSbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



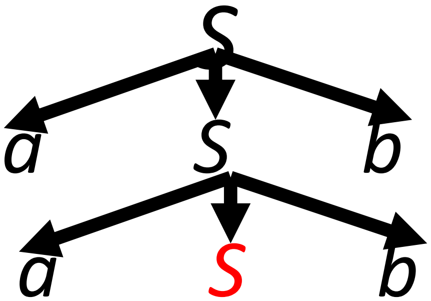
$S \Rightarrow aSb \Rightarrow aaSbb$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



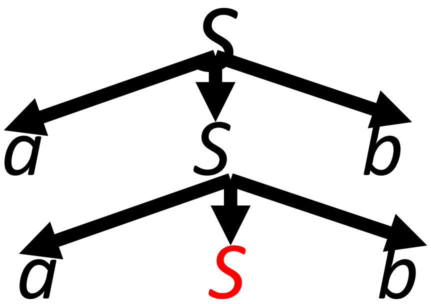
$$S \Rightarrow aSb \Rightarrow aaSbb$$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



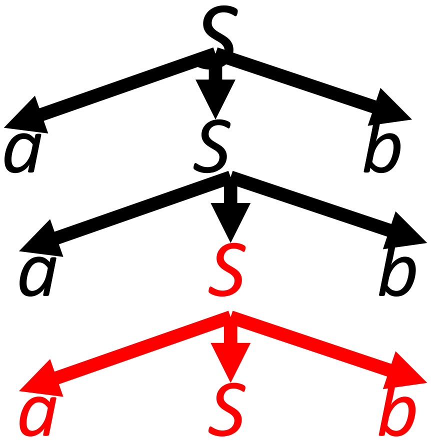
$$S \Rightarrow aSb \Rightarrow aaSbb$$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



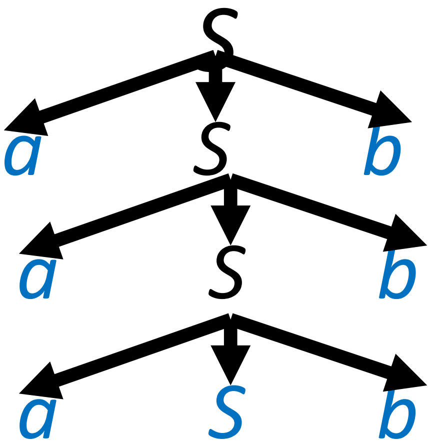
$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow$$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



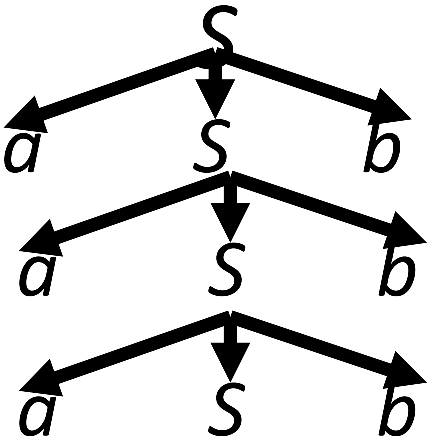
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



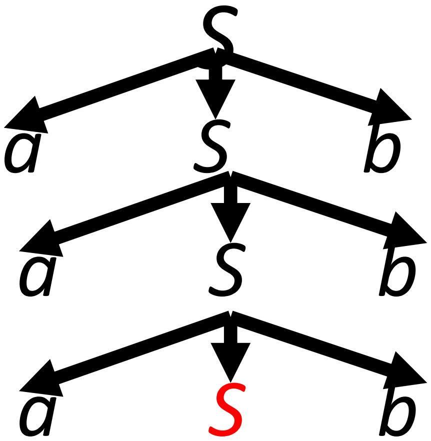
$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



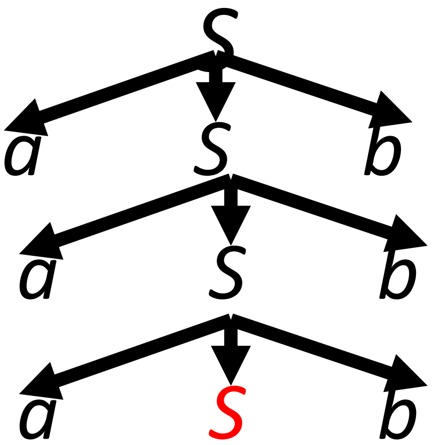
$S \Rightarrow aaaSbbb$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



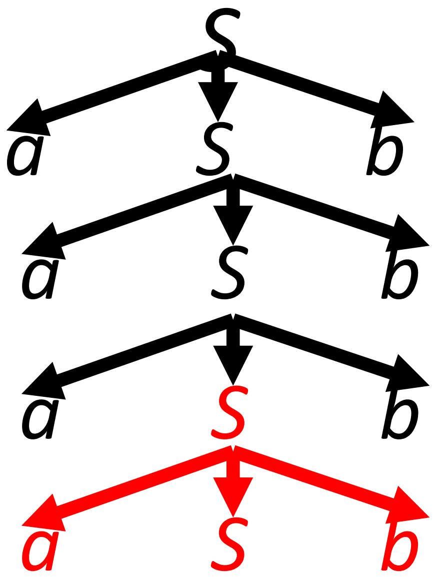
$$S \Rightarrow^* aaaSbbb$$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



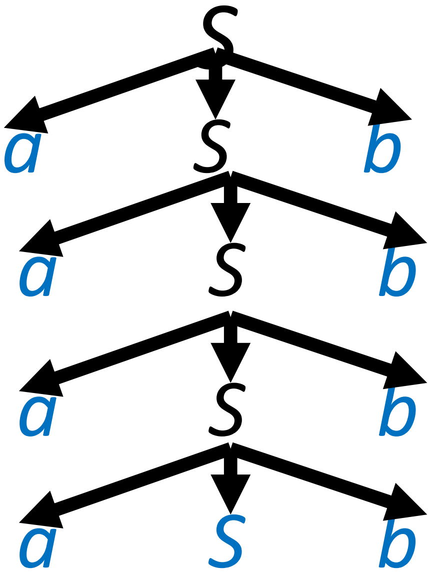
$S \Rightarrow^* aaaSbbb \Rightarrow$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



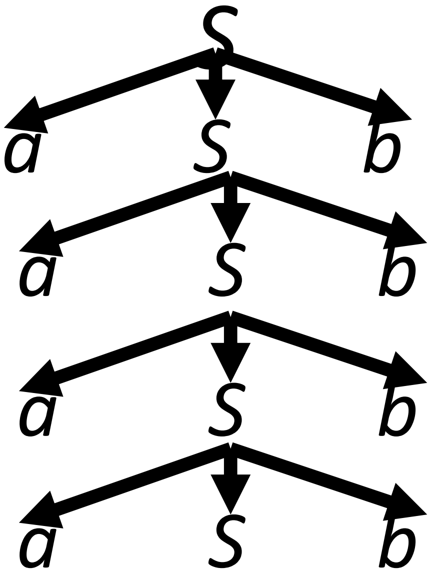
$S \Rightarrow^* aaaSbbb \Rightarrow aaaaSbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



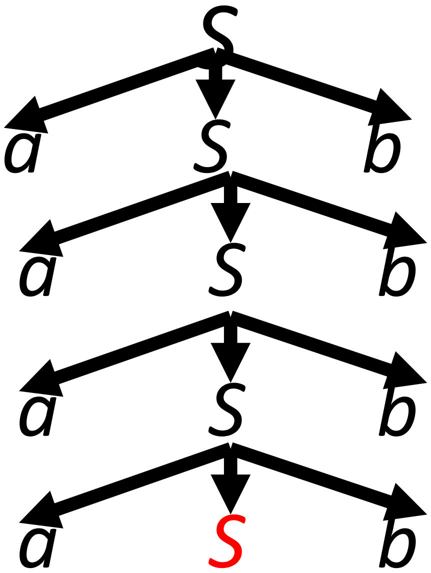
$S \Rightarrow^* aaaSbbb \Rightarrow aaa\textcolor{red}{aSb}bbb$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



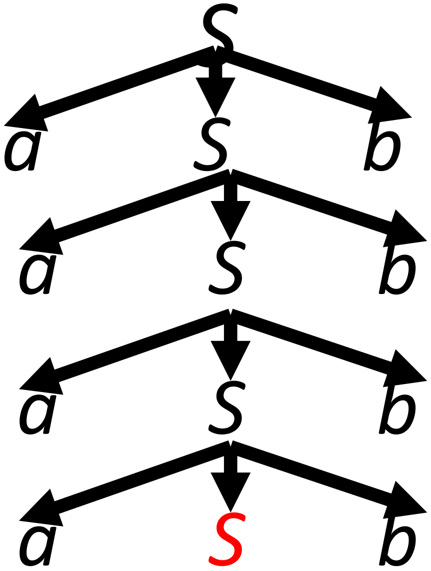
$$S \Rightarrow^{*} aaaaSbbbb$$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



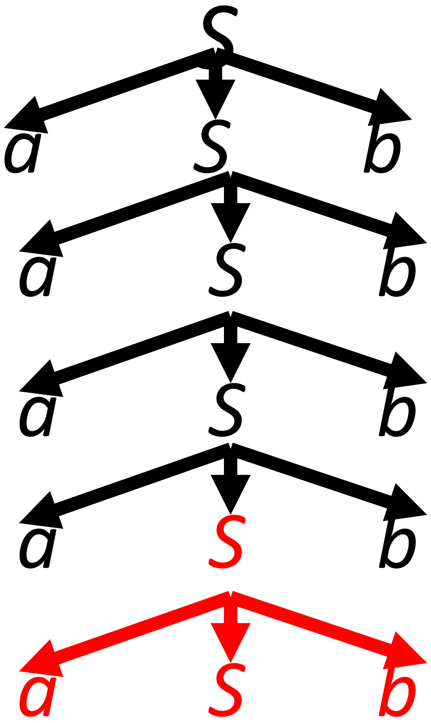
$S \Rightarrow^* aaaaSbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



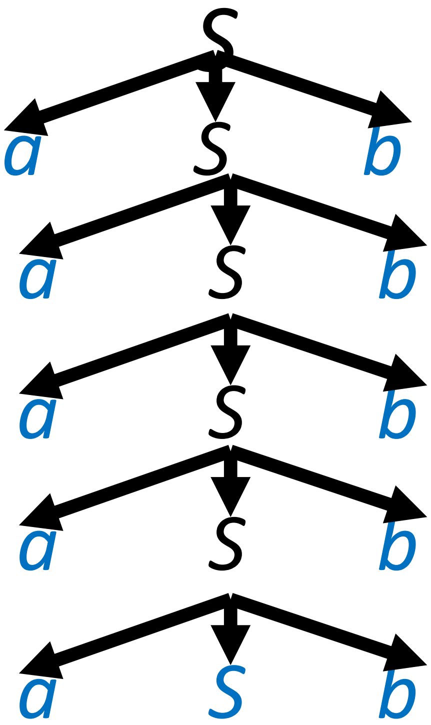
$S \Rightarrow^* aaaaSbbbb \Rightarrow$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



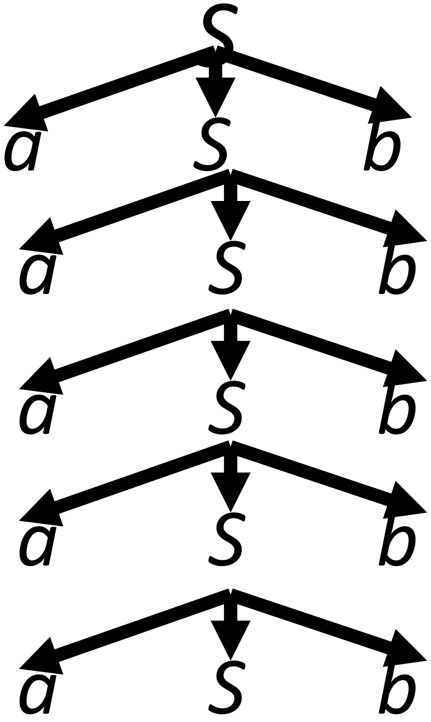
$S \Rightarrow^* aaaaSbbbb \Rightarrow aaaaaSbbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



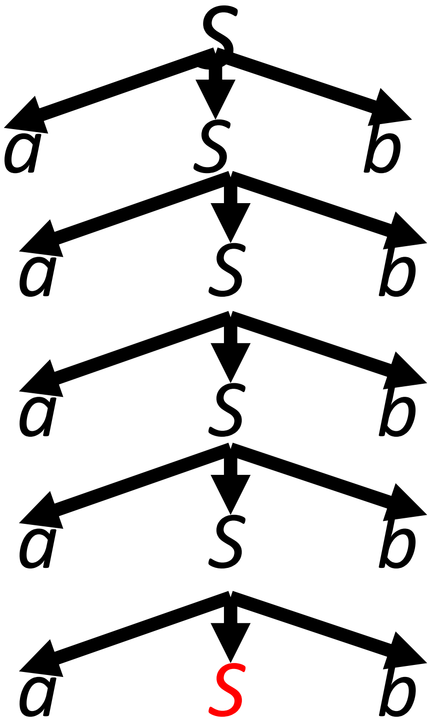
$S \Rightarrow^* aaaaSbbbb \Rightarrow aaaa\color{red}{aSb}\color{black}{bbbb}$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



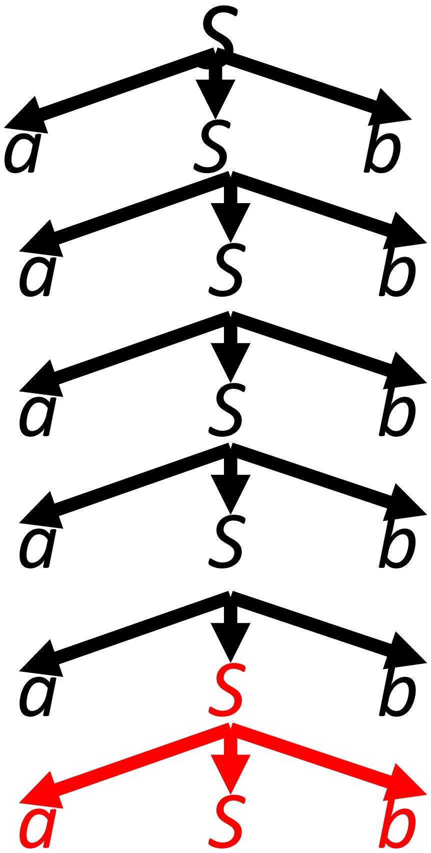
$S \Rightarrow^* aaaaaSbbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



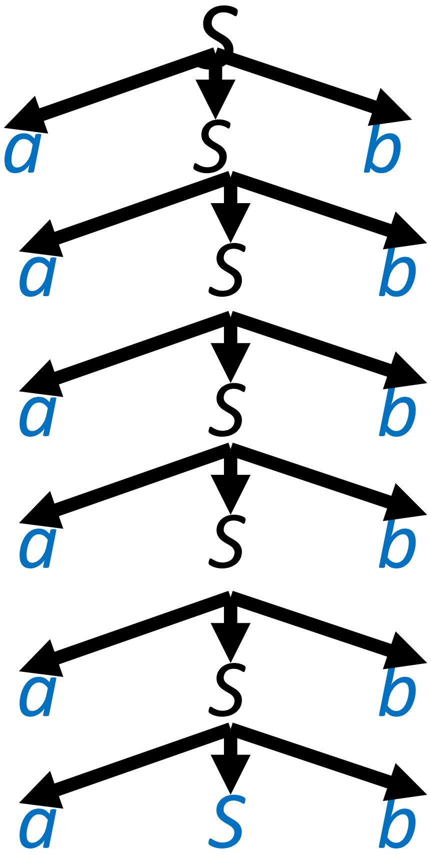
$S \Rightarrow^{*} aaaaaSbbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



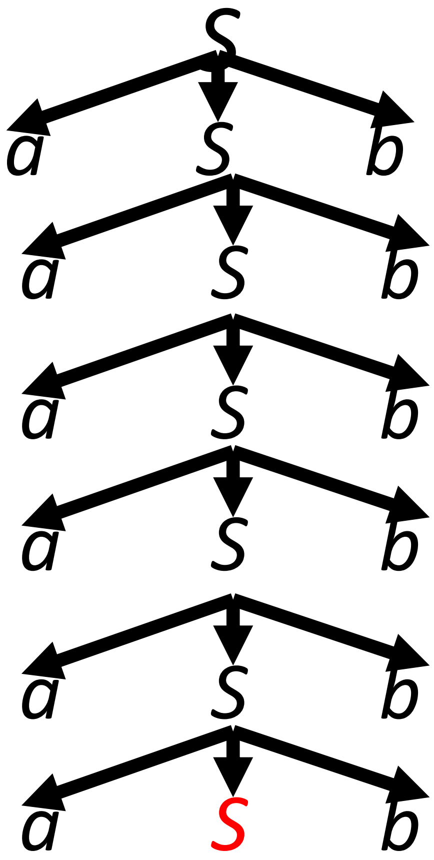
$S \Rightarrow^* aaaaa\ aSb\ bbbbb$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



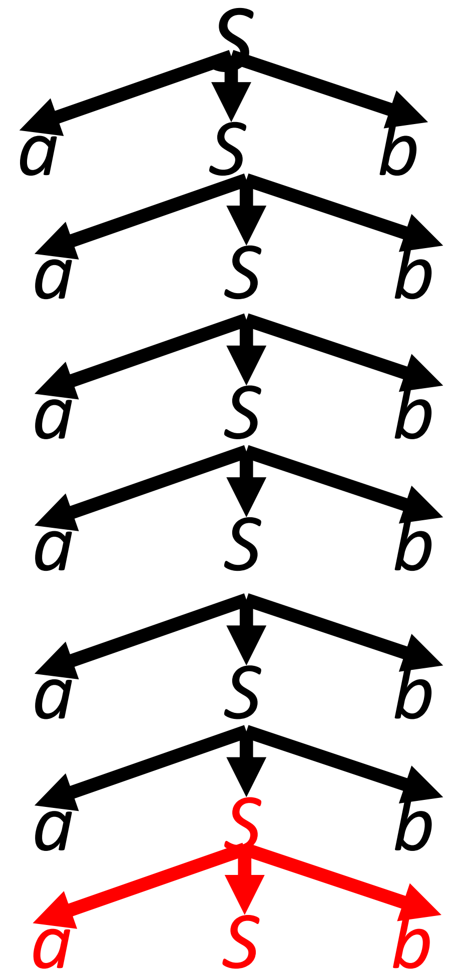
$$S \Rightarrow^* aaaaaaSbbbbbb$$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



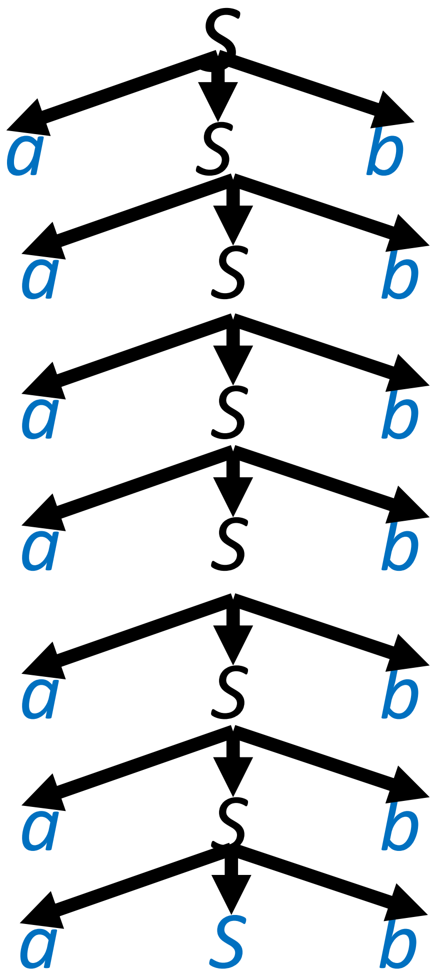
$$S \Rightarrow^* aaaaaaSbbbbbb$$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



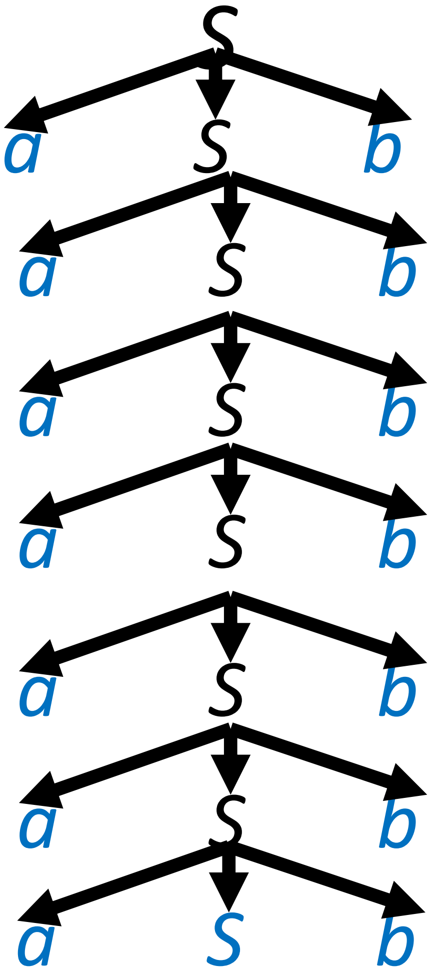
$S \Rightarrow^* aaaaaa\ aSb\ bbbbbb$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



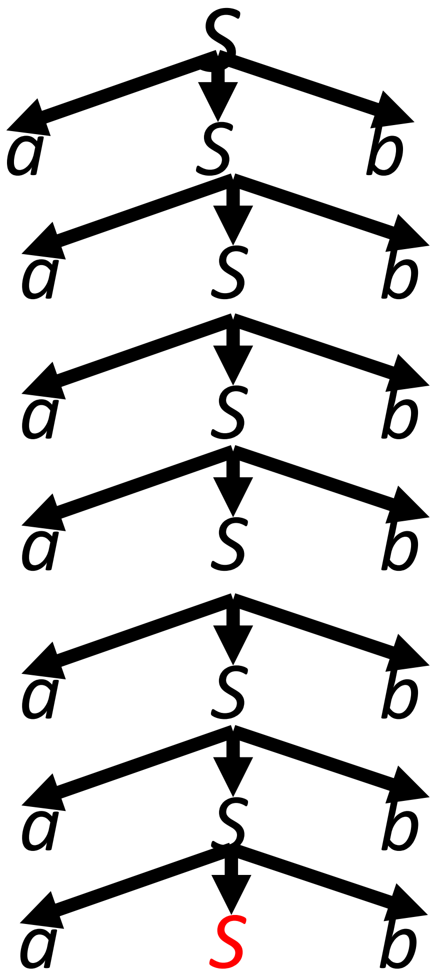
$$S \Rightarrow^{*} aaaaaa\color{red}{aSb}\color{black}{bbbbbb}$$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



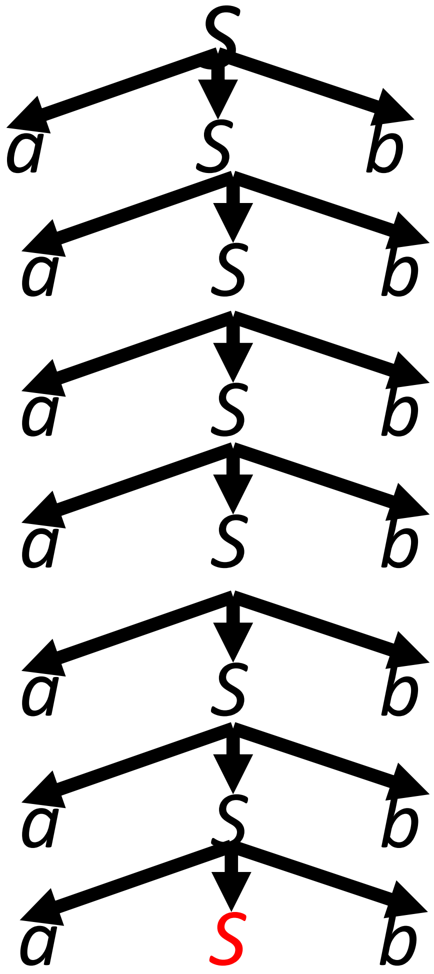
$S \Rightarrow^* aaaaaaaSbbbbbbb$

$$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$$



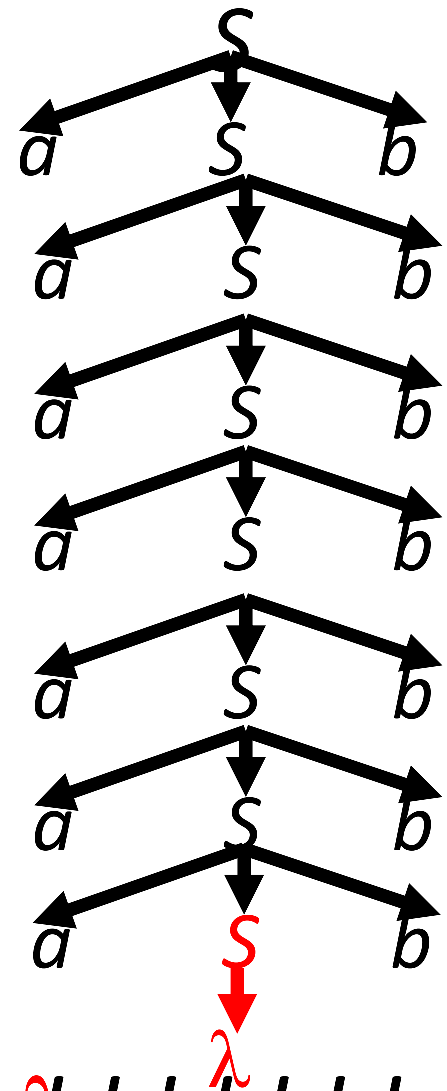
$$S \Rightarrow^{*} aaaaaaaSbbbbbbb$$

$$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$$



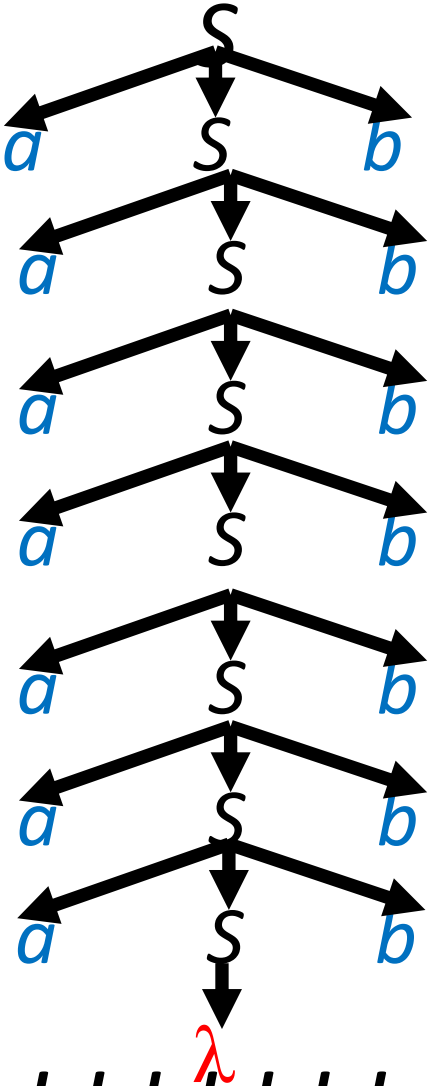
$$S \Rightarrow^{*} aaaaaaaSbbbbbbb$$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



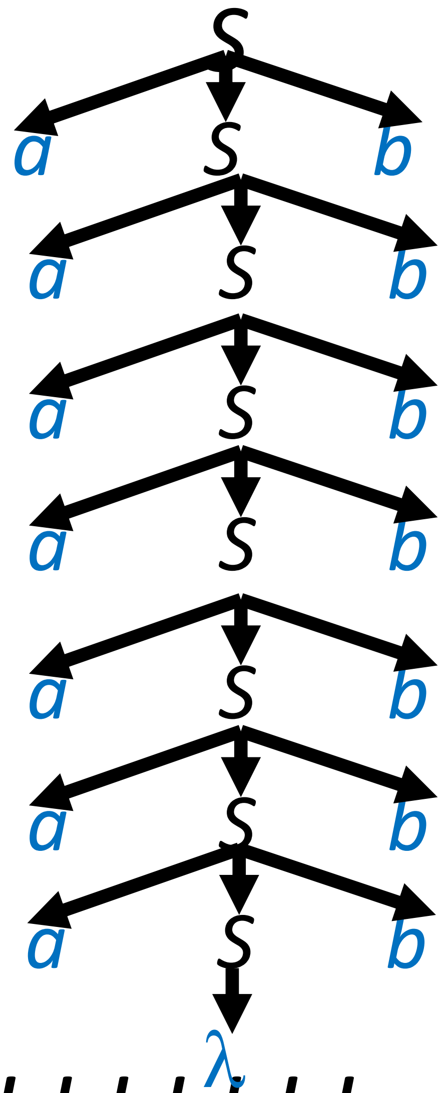
$S \Rightarrow^* aaaaaaa\ \lambda bbbbbbb$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



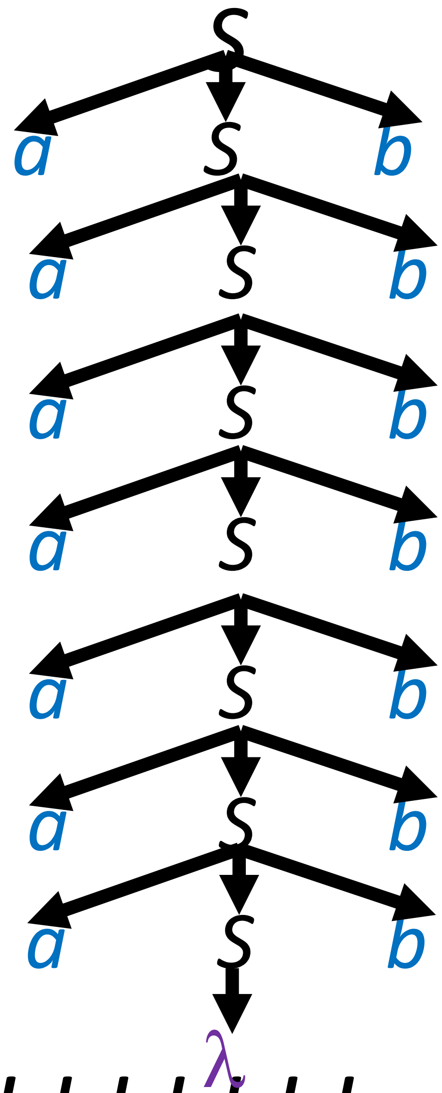
$S \Rightarrow^{*} aaaaaaa\ bbbbbbb$

$(\{S\}, \{a, b\}, S, \{S \to aSb, S \to \lambda\})$



$S \Rightarrow^* aaaaaaabbbbbbb$

$(\{S\}, \{a, b\}, S, \{S \rightarrow aSb, S \rightarrow \lambda\})$



$S \Rightarrow^* aaaaaaabbbbbbb$

DNA computing & automata

- Several new paradigms of computing are based on the structure and natural operations of DNA. They appeared at the end of the last century. In contrast, the theory of finite automata is well developed and intensively used in both theory and practice.
- The Watson-Crick automata, introduced in (Freund et al., 1997), relate to both fields. They are important in the field of (DNA) computing and have important relation to formal language and automata theory as well.

Biological Background and Motivation, the DNA

- Five chemical elements can be found in DNA molecules:

  Hydrogen (1 connection),
  Oxygen (2 connections),
  Nitrogen (3 connections),
  Carbon (4 connections), and
  Phosphorus (5 connections).

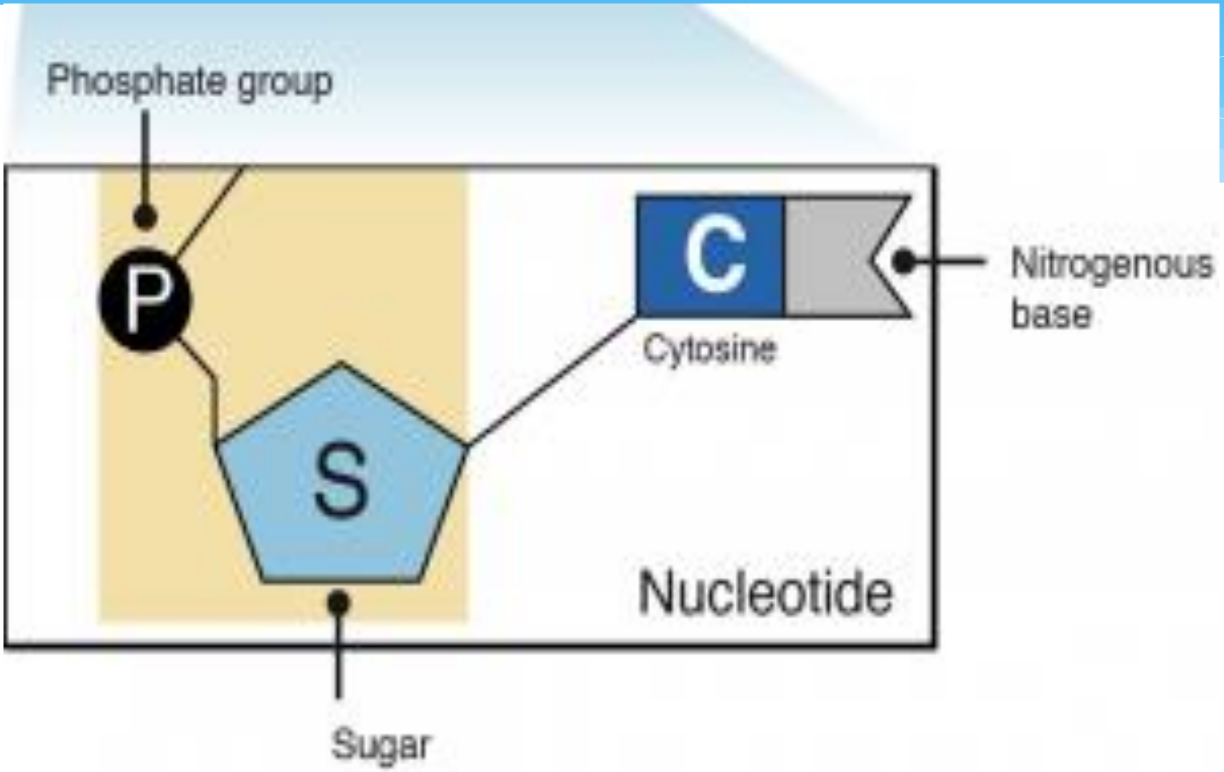
- They can be connected in various ways by covalent bond (one of the strongest chemical bonds).
- There are four possible nukleotide bases:

Adenine(A), Cytosine (C), Guanine (G) and Thymine (T).

the DNA

- The bases are connected to sugar, which is connected to the Phosphate group. Each nucleotide has three components: a base (that can be four types in a DNA molecule), a sugar and a Phosphate.
- The sugar has five Carbon atoms which can be identified as $1'$ through $5'$. The Phosphate is connected to the $5'$ Carbon. Two nucleotides can be bonded through the Phosphate group (a water molecule H_2O appears; this process is catalyzed by the Ligase enzyme in a DNA molecule): the connection goes from the $3'$ Carbon to the Phosphate of the next nucleotide.

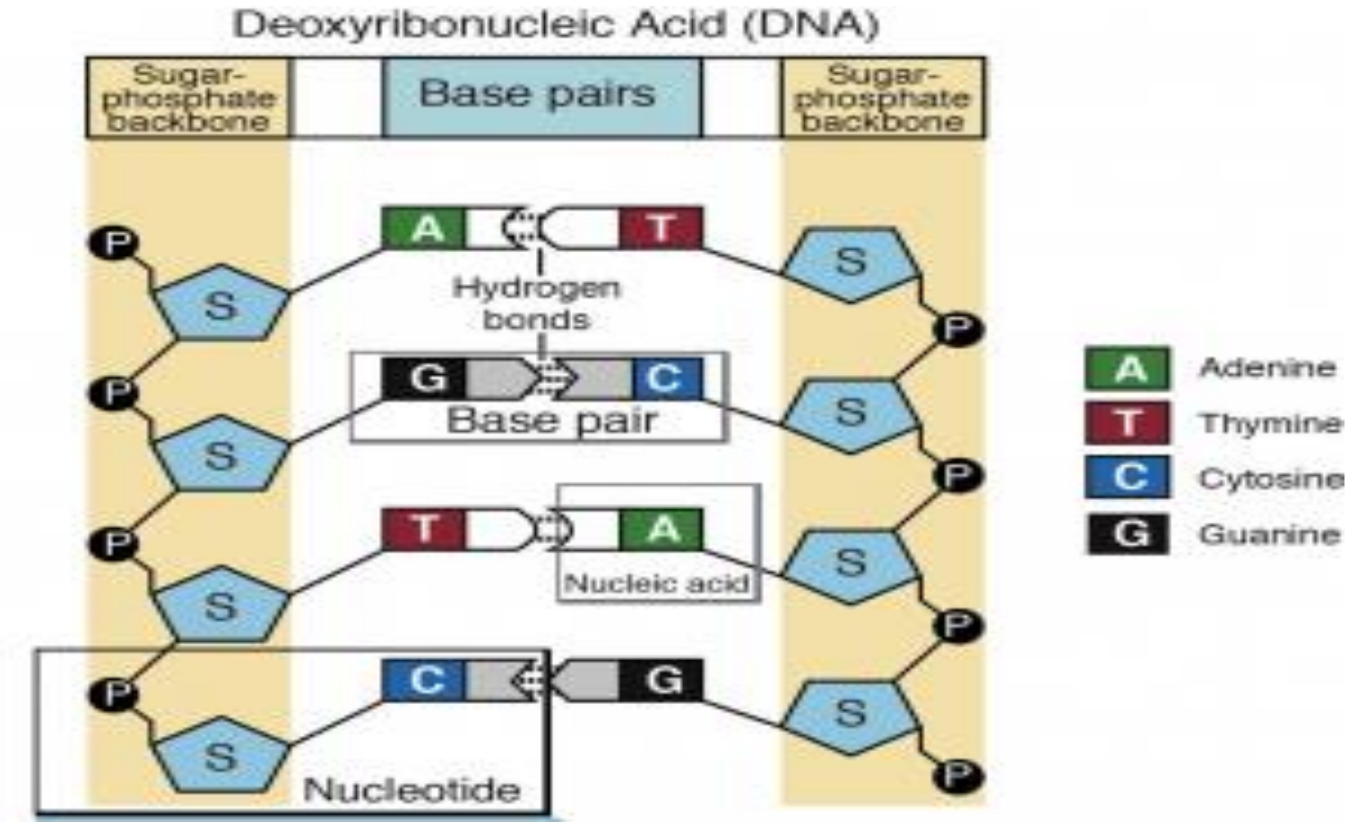
the DNA


Phosphate group
P
C
Nitrogenous base
Cytosine
S
Nucleotide
Sugar

the DNA

- Thus, a DNA strand is a directed sequence of nucleotides and can be interpreted as a string over the alphabet {A,C,G,T}.
- A DNA molecule consists of two strands. Watson and Crick discovered the double helix structure of DNA and the fact that the two strands are elementwise complementary of each other.
- On one end of the strand, there is a 5′ side, and on the other end, a 3′ side of a base.

the DNA


Deoxyribonucleic Acid (DNA)
Sugar-phosphate backbone
Base pairs
Sugar-phosphate backbone
P
S
A
T
Hydrogen bonds
G
C
Base pair
T
A
Nucleic acid
C
G
Nucleotide
A Adenine
T Thymine
C Cytosine
G Guanine

The title

- Watson – Crick
  - Motivation
  - DNA computing
  - finite automata
  - 5’ - 3’ ends of DNA strands

The title

- W K
  - Motivation
  - DNA computing
  - finite automata
  - 5’ - 3’ ends of DNA strands

The title

- W K
  - DNA molecule
  - finite automata
  - 5’ - 3’ ends of DNA strands

finite automata (NFA)

A = (V, Q, q_0, F, δ)

V input alphabet

Q: set of states

T: set of terminals

$q_0 \in$ Q initial state

F $\subseteq$ Q : final (or accepting) states

δ: transition function

Q x $(V \cup \{\lambda\}) \rightarrow 2^Q$

The title

- W K
  - DNA molecule
  - finite automata
  - 5’ - 3’ ends of DNA strands

finite automata (NFA)

A = (V, Q, q_0, F, δ)

V input alphabet

Q: set of states

T: set of terminals

$q_0 \in$ Q initial state

$F \subseteq Q$: final (or accepting) states

δ: transition function

Q x $(V \cup \{\lambda\}) \rightarrow 2^Q$

The title

- DNA molecule
- finite automata
- 5’ - 3’ ends of DNA strands

Linear context-free grammars & languages

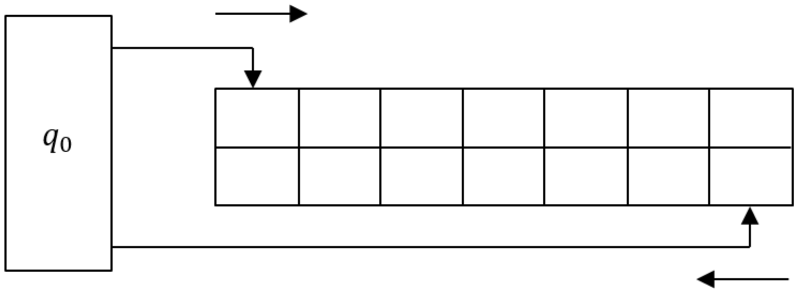
G = (N,T,S,P)
N,T alphabets:
N: set of nonterminals
T: set of terminals
$S \in N$ start or sentence symbol
P: set of poductions,

$A \rightarrow uBv$
$A,B \in N, \quad u,v \in T^*$

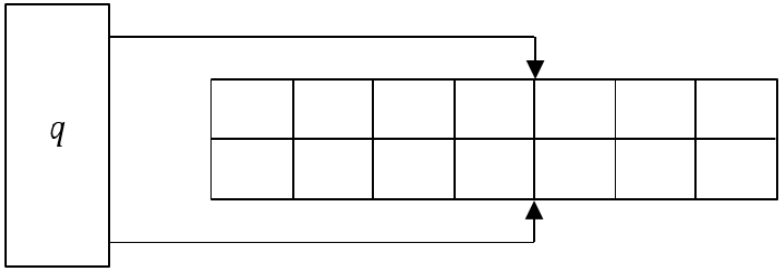
Definitions

The two strands of the DNA molecule have opposite $5' \to 3'$ orientations.

5’-3’ variant of Watson-Crick finite automaton that parses two strands of the Watson-Crick tape in opposite directions.

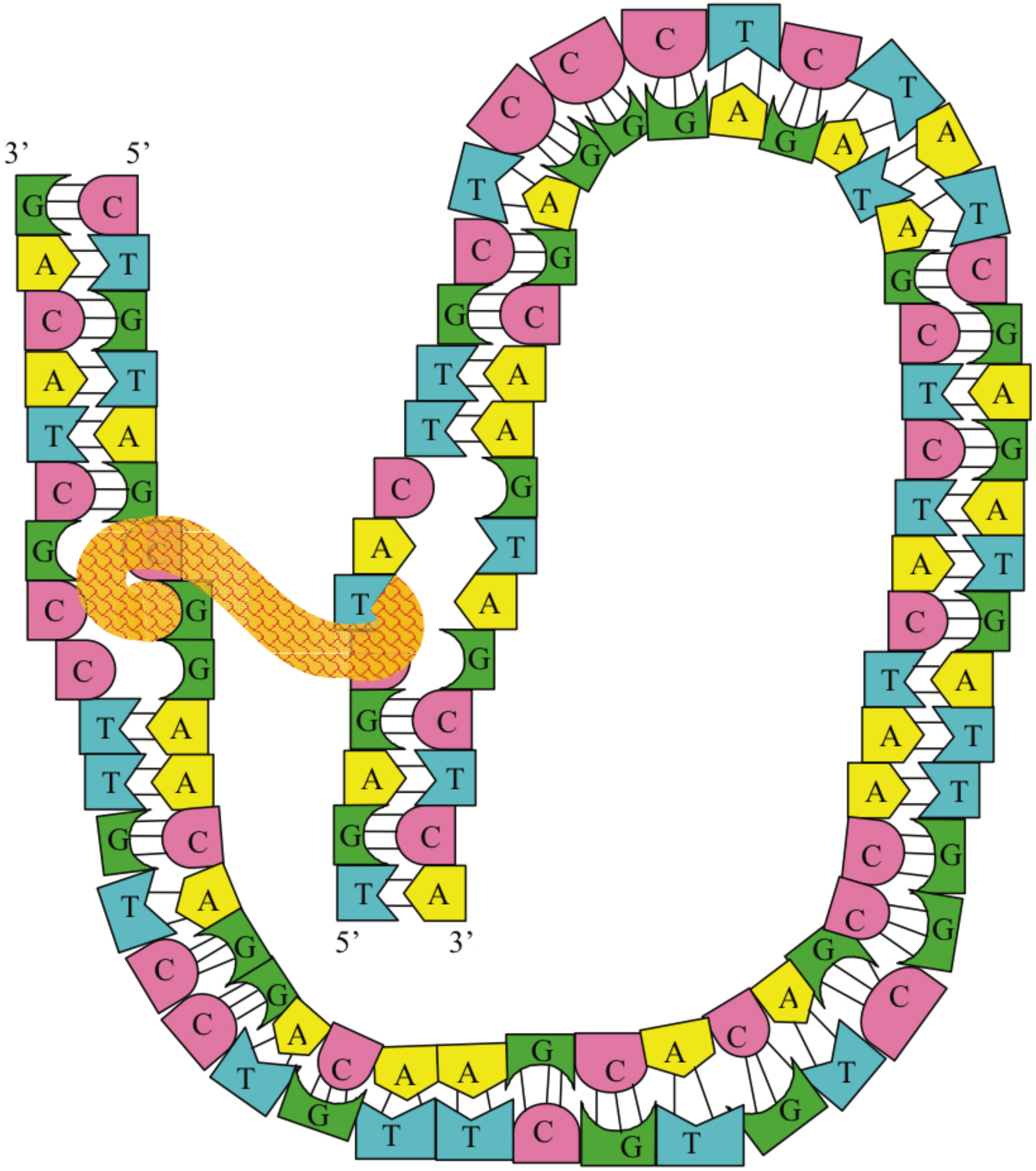


Initial configuration



An accepting configuration
with a final state q

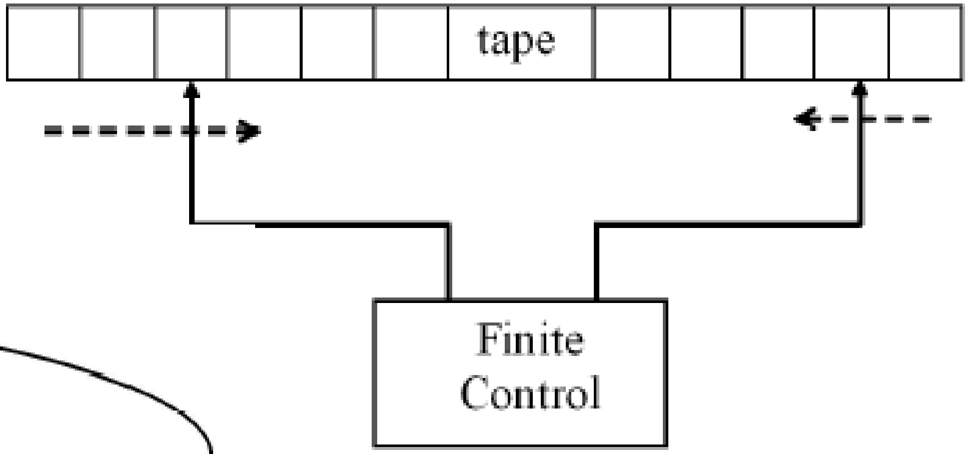
A possible figure of a 5’- 3’ Watson-Crick automaton. The two strands of the DNA molecule is read by an enzyme simultaneously. The first head has already read CTGTAGC and is reading G, while the second head has read TGAGC and is reading T.

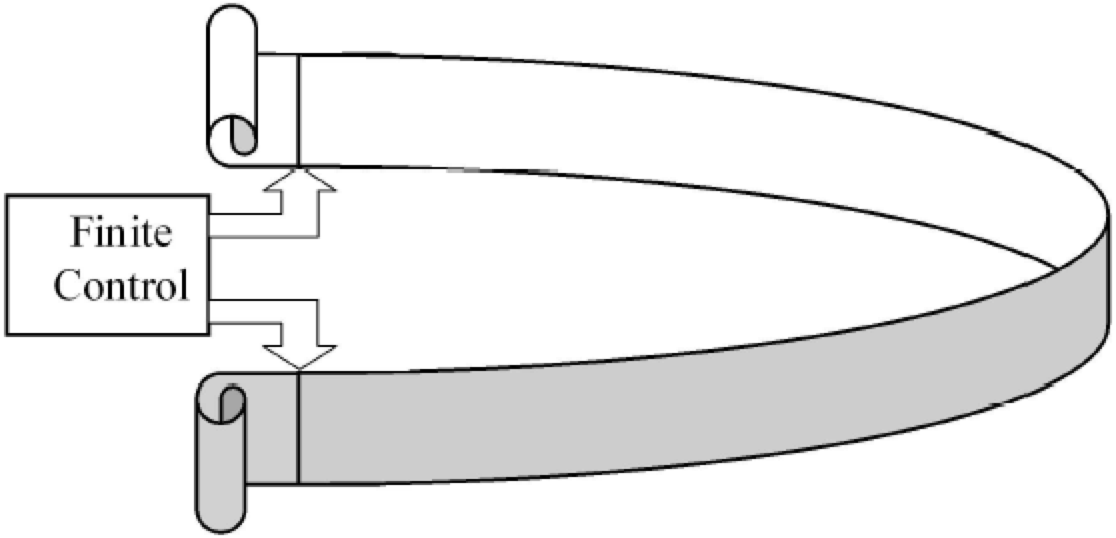


Linear languages – 2-head finite automata

$$\langle Q, s, V, d, F \rangle$$

$$d : Q \times (V \cup \{\varepsilon\}) \times (V \cup \{\varepsilon\}) \to 2^Q$$





Watson - Crick Automata – WK tape

- WK-automata are (finite) automata working on a Watson-Crick tape, that is a double-stranded sequence (or molecule) in which the lengths of the strands are equal and the elements of the strands are pairwise complements of each other:

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix} \begin{bmatrix} a_2 \\ b_2 \end{bmatrix} \cdots \begin{bmatrix} a_n \\ b_n \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \end{bmatrix}$$

with a_i,b_i ∈V and (a_i,b_i)∈ρ (i=1,...,n).

- where $\rho \subseteq V \times V$ is a symmetric relation, the Watson-Crick complementarity.

The $5' \rightarrow 3'$ WK automaton is sensing if the heads sense that they are meeting. Formally, a Watson-Crick automaton is a 6-tuple $M = (V, \rho, Q, q_0, F, \delta)$.

- V is the (input) alphabet,
- $\rho \subseteq V \times V$ denotes a complementarity relation,
- Q represents a finite set of states,
- $q_0 \in Q$ is the initial state,
- $F \subseteq Q$ is the set of final states and
- δ is called transition mapping and it is of the form $\delta: Q \times \binom{V^*}{V^*} \longrightarrow 2^Q$ such that it is non empty only for finitely many triplets $(q, u, v), q \in Q, u, v, \in V^*$.

In sensing $5' \rightarrow 3'$ WK automata every pair of positions in the Watson-Crick tape is read by exactly one of the heads in an accepting computation, and therefore the complementarity relation cannot play importance, instead, we may assume that it is identity relation (that is we can work with usual strings).

Let us define the radius of an automaton by r which shows the maximum length of the substring of the input that can be read by the automaton in a transition.

A configuration of a Watson-Crick automaton is a pair (q, w).

The current state

the part of the input word which has not been read yet.

For $w', x, y \in V^*, q, q' \in Q$, a transition between two configurations can be written as:

$(q, xw'y) \Rightarrow (q', w')$ if and only if $q' \in \delta(q, x, y)$.

For a given w ∈ V*, an accepting computation is a sequence of transitions

$(q_0, w) \Rightarrow^* (q_F, \lambda)$.

The language accepted by a WK automaton M is:

$L(M) = \{w \in V^* | (q_0, w) \Rightarrow^* (q_F, \lambda), q_F \in F\}$.

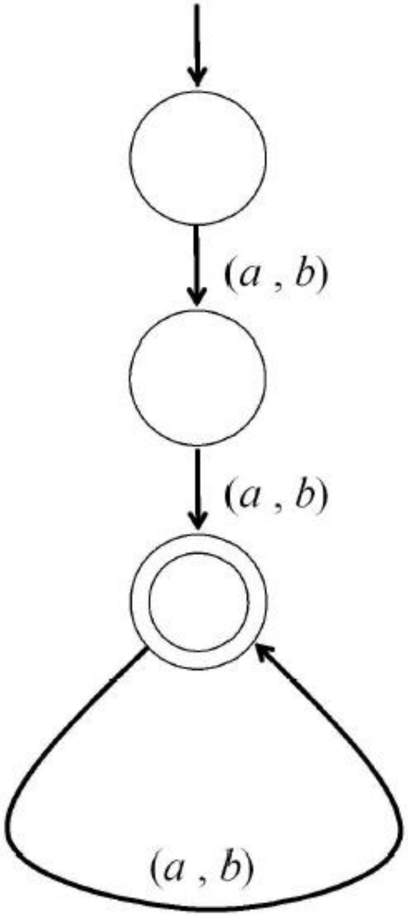
The shortest nonempty word accepted by M is denoted by w_s, if it is uniquely determined or any of them if there are more than one such word(s).

There are some restricted versions of WK automata which can be defined as follows:

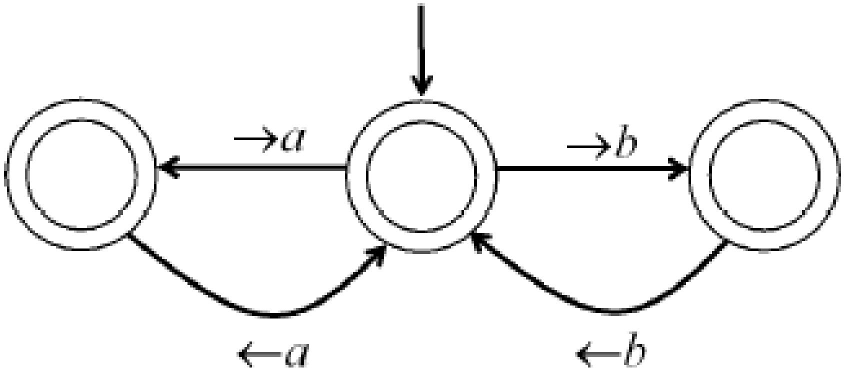
- **N:** stateless: if $Q = F = \{q_0\}$;
- **F:** all-final: if $Q = F$;
- **S:** simple: $\delta: \left(Q \times \left((\lambda, V^*) \cup (V^*, \lambda)\right)\right) \to 2^Q$.
- **1:** 1-limited: $\delta: \left(Q \times \left((\lambda, V) \cup (V, \lambda)\right)\right) \to 2^Q$.

We can define additional version using multiple constrains such as **F1, N1, FS, NS** WK automata.

Examples (two possible notions)



- Palindromes:



- $a^n b^n \ (n > 1)$

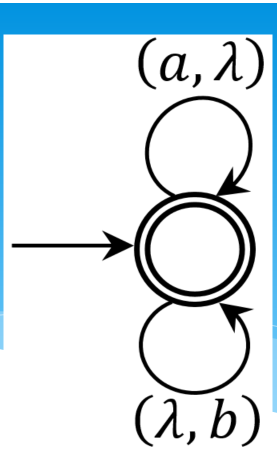


Figure A sensing $5' \rightarrow 3'$ WK automaton of type **N1** accepting the language $\{a^n b^m | n, m \geq 0\}$.

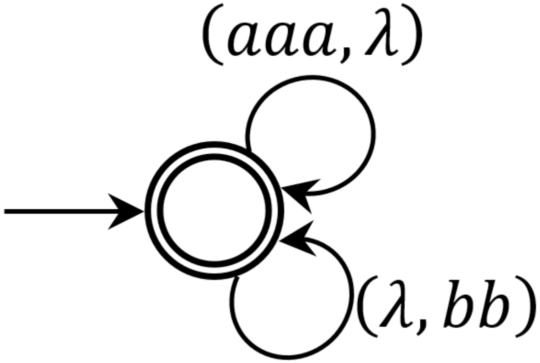


Figure A sensing $5' \rightarrow 3'$ WK automaton of type **NS** accepting the language $\{a^{3n} b^{2m} | n, m \geq 0\}$.

NS type can accept only regular languages!

$' \rightarrow 3'$ WK automaton:

$\rightarrow 33'$ WK automaton:

Example of NS sensing $5' \rightarrow 3'$ WK automaton:

nn a 3n b 2m bb b 2m 2mm b 2m   a 3n b 2m n,m≥0 nn,mm

≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

Language: *a* 3 3*n* 3*n a* 3*n* *b* 2*m* *n*, *m* ≥ 0 .

nn a 3n b 2m bb b 2m 2mm b 2m   a 3n b 2m n,m≥0 nn,mm
≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

Word: *aaaaaaaaabbbb*

$\boldsymbol{q\ 0\ ,}\ aaaaaaaaaaaaaaaaaabbbbbbbb)$

$nn\ \boldsymbol{a\ 3n\ \ b\ 2m\ bb\ b\ 2m\ 2mm\ b\ 2m\ \ \ a\ 3n\ \ b\ 2m\ \ n,m\geq 0\ nn,mm}$

$\geq 0\ \ a\ 3n\ \ b\ 2m\ \ n,m\geq 0\ .$

→33′ WK automaton:

Word: $aaaaaaaaabbbb$

A configuration of a Watson-Crick automaton:

$(\ q\ 0\ 0\ 0\ ,aaaaaaaaabbbb)$

$\boldsymbol{q\ 0\ ,\ aaaaaaaaaaaaaaaaaabbbbbbbb)}$

$\boldsymbol{q\ 0\ ,}\ aaaaaaaaaaaaaaaaaabbbbbbbb)$

$nn\ \boldsymbol{a\ 3n\ \ b\ 2m\ bb\ b\ 2m\ 2mm\ b\ 2m\ \ \ a\ 3n\ \ b\ 2m\ \ n,m\geq 0\ nn,mm}$

$\geq 0\ \ a\ 3n\ \ b\ 2m\ \ n,m\geq 0\ .$

→33′ WK automaton:

A configuration of a Watson-Crick automaton:

$(\ q\ 0\ 0\ 0\ ,aaaaaaaaabbbb)$

A configuration of a Watson-Crick automaton:

$(\ q\ 0\ 0\ 0\ ,aaaaaaaaabbbb)$

$\boldsymbol{q\,0\,,\,aaaaaaaaaaaaaaaaaabbbbbbbb)}$

$\boldsymbol{q\,0\,,}\,aaaaaaaaaaaaaaaaaabbbbbbbb)$

$nn\ \boldsymbol{a\ 3n\ \ b\ 2m\ bb\ b\ 2m\ 2mm\ b\ 2m\ \ \ a\ 3n\ \ b\ 2m\ \ n{,}m{\geq}0\ nn{,}mm}$ $\geq 0\ \ a\ 3n\ \ b\ 2m\ \ n{,}m{\geq}0\ .$

→33′ WK automaton:

By using transition (q_0, aaa, λ): $(q_0, aaaaaaaaabbbb) \Rightarrow (q_0, aaaaaabbbb)$

A configuration of a Watson-Crick automaton:

$(\ q\ 0\ 0\ 0\ , aaaaaaaaabbbb)$

A configuration of a Watson-Crick automaton:

$(\ q\ 0\ 0\ 0\ , aaaaaaaaabbbb)$

***q* 0 , *aaaaaaaaaaaaaaaaaabbbbbbbb*)**

***q* 0 ,** *aaaaaaaaaaaaaaaaaabbbbbbbb)*

nn ***a* 3*n* *b* 2*m* *bb* *b* 2*m* 2*mm* *b* 2*m*   *a* 3*n* *b* 2*m* *n*,*m*≥0 *nn*,*mm***

≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

By using transition (q_0, aaa, λ): $(q_0, aaaaaaaaabbbb) \Rightarrow (q_0, aaaaaabbbb)$

A configuration of a Watson-Crick automaton:

$(q 0 0 0 , aaaaaaaaabbbb)$

(q_0, aaa, λ): $(q_0, aaaaaabbbb) \Rightarrow (q_0, aaabbbb)$

A configuration of a Watson-Crick automaton:

$(q 0 0 0 , aaaaaaaaabbbb)$

q 0 , $aaaaaaaaaaaaaaabbbbbbb$)

q 0 , $aaaaaaaaaaaaaaabbbbbbbb$)

nn **a 3n b 2m bb b 2m 2mm b 2m a 3n b 2m n,m≥0 nn,mm**

≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

By using transition (q_0, aaa, λ): $(q_0, aaaaaaaaabbbb) \Rightarrow (q_0, aaaaaabbbb)$

A configuration of a Watson-Crick automaton:

(q 0 0 0 , $aaaaaaaaabbbb$)

(q_0, aaa, λ): $(q_0, aaaaaabbbb) \Rightarrow (q_0, aaabbbb)$

A configuration of a Watson-Crick automaton:

(q 0 0 0 , $aaaaaaaaabbbb$)

(q_0, λ, bb): $(q_0, aaabbbb) \Rightarrow (q_0, aaabb)$

***q* 0 , *aaaaaaaaaaaaaaaaaabbbbbbbb*)**

***q* 0 ,** *aaaaaaaaaaaaaaaaabbbbbbbbb)*

nn ***a* 3*n* *b* 2*m* *bb* *b* 2*m* 2*mm* *b* 2*m*   *a* 3*n* *b* 2*m* *n*,*m*≥0 *nn*,*mm***

≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

By using transition (q_0, aaa, λ): $(q_0, aaaaaaaaabbbb) \Rightarrow (q_0, aaaaaabbbb)$

A configuration of a Watson-Crick automaton:

(q 0 0 0 , aaaaaaaabbbb)

(q_0, aaa, λ): $(q_0, aaaaaabbbb) \Rightarrow (q_0, aaabbbb)$

A configuration of a Watson-Crick automaton:

(q 0 0 0 , aaaaaaaabbbb)

(q_0, λ, bb): $(q_0, aaabbbb) \Rightarrow (q_0, aaabb)$

(q_0, λ, bb): $(q_0, aaabb) \Rightarrow (q_0, aaa)$

***q* 0 , *aaaaaaaaaaaaaaabbbbbbb*)**

***q* 0 ,** *aaaaaaaaaaaaaaabbbbbbb)*

nn ***a* 3*n* *b* 2*m* *bb* *b* 2*m* 2*mm* *b* 2*m*   *a* 3*n* *b* 2*m* *n*,*m*≥0 *nn*,*mm***

≥0 a 3n b 2m n,m≥0 .

→33′ WK automaton:

By using transition $(q_0, aaa, \lambda): (q_0, aaaaaaaaabbbb) \Rightarrow (q_0, aaaaaabbbb)$

A configuration of a Watson-Crick automaton:

(*q* 0 0 0 , *aaaaaaaaabbbb*)

$(q_0, aaa, \lambda): (q_0, aaaaaabbbb) \Rightarrow (q_0, aaabbbb)$

A configuration of a Watson-Crick automaton:

(*q* 0 0 0 , *aaaaaaaaabbbb*)

$(q_0, \lambda, bb): (q_0, aaabbbb) \Rightarrow (q_0, aaabb)$

$(q_0, \lambda, bb): (q_0, aaabb) \Rightarrow (q_0, aaa)$

$(q_0, aaa, \lambda): (q_0, aaa) \Rightarrow (q_0, \lambda)$

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

- $' \rightarrow 3'$ WK finite automata,
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

Theorem. The following classes of languages coincide:

- $' \rightarrow 3'$ WK finite automata,
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

Theorem. The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- $' \rightarrow 3'$ WK finite automata,
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

$' \rightarrow 3'$WK finite automata,

Theorem. The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- the language class accepted by sensing $5' \rightarrow 3'$ WK finite automata,
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

$' \rightarrow 3'$ *WK automata*,

$' \rightarrow 3'$WK finite automata,

Theorem**.** The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- the language class accepted by **S** sensing $5' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

′→3′ WK automata.

′→3′ WK automata,

*′→3′*WK finite automata,

Theorem**.** The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- the language class accepted by **1** sensing $5' \rightarrow 3'$ WK automata.
- $' \rightarrow 3'$ WK automata,
- $' \rightarrow 3'$ WK automata.

Sensing $5' \rightarrow 3'$ WK automata characterizing the linear languages

′→3′ WK automata.

′→3′ WK automata,

*′→3′*WK finite automata,

Theorem**.** The following classes of languages coincide:

- the class of linear context-free languages defined by linear context-free grammars,
- ′ → 3′ WK automata,
- ′ → 3′ WK automata.

Constructive proofs

- Let G (N,T,S,P) be a linear grammar, then
- Define A (T, id, N $\cup$ { q_f }, S, { q_f }, δ) with
- δ as follows:
  - for each production of P of form $A \to uBv$ with nonterminal B on the right side, let
    $B \in \delta(A,u,v)$ and
  - for each production of P of form $A \to u$ without nonterminal on the right side, let
    $q_f \in \delta(A,u,\lambda)$.
- Sucessful derivations in G coincide to accepting computations in A.

Constructive proofs

- Let A (V, id, Q, q_0, F, δ) be a 5’-3’ WK automaton, then if Q and V are not disjunct rename the names of the states (!)
- define G (Q,V, q_0,P) with
- P as follows:
  - for each transition $q' \in \delta(q,u,v)$ of δ, let $q \rightarrow u\, q'\, v$ in P and
  - moreover, for each $q \in F$ let $q \rightarrow \lambda$ in P.
- Again, sucessful derivations in G coincide to accepting computations in A.

Restricted variants

- Each transition $q' \in \delta(q,u,v)$ of a WK automaton can be broken to 2 consecutive transitions by adding a new (intermediate) state to the automaton:

- $q'' \in \delta(q,u,\lambda)$
- $q' \in \delta(q'',\lambda,v)$
- Further, reading strings of lenght n can be broken to n consecutive transitions reading them letter by letter...

Hierarchy by sensing $5' \rightarrow 3'$ WK automata

- Some restricted variants can only accept sublinear or even subregular language classes….

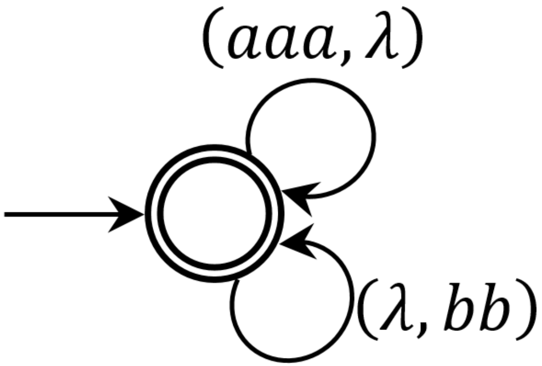
Lemma 1. Let M be an $\boldsymbol{F1}$ sensing $5' \rightarrow 3'$ WK automaton and let the word $w \in V^{+}$ that is in $L(M)$. Let $|w| = k$, then for each l, where $0 \leq l \leq k$, there is at least one word $w_l \in L(M)$ such that $|w_l| = l$.

Proof. According to the definition of $\boldsymbol{F1}$ sensing $5' \rightarrow 3'$ WK automaton, w can be accepted in k steps such that in each step, the automaton can read exactly one letter. Moreover, each state is final, therefore by considering the first l steps of k steps, the word $w_l = w'_l w''_l$ is accepted by M, where w'_l is read by the left head and w''_l is read by the right head during these l steps, respectively.

Remark 1. Since, by definition every **N1** sensing $5' \rightarrow 3'$ WK automaton is **F1** sensing $5' \rightarrow 3'$ WK automaton at the same time, Lemma 1 applies for all **N1** sensing $5' \rightarrow 3'$ WK automata also.

Theorem 2. The class of languages that can be accepted by **N1** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class accepted by **NS** sensing $5' \rightarrow 3'$ WK automata.

Proof. We can prove the theorem by using the language $L = \{a^{3n}b^{2m} | n, m \geq 0\}$. In this language w_s is bb and it can be accepted by these transitions: (bb, λ) or (λ, bb). Although using Lemma 1, w_s cannot be the shortest nonempty accepted word in a language accepted by an **N1** sensing $5' \rightarrow 3'$ WK automaton (see the Figure below).



Theorem 3. The class of languages that can be accepted by **NS** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class accepted by **N** sensing $5' \rightarrow 3'$ WK automata.

Proof. The language $L = \{a^{(2n+m)}b^{(2m+n)} \mid n, m \geq 0\}$ proves the proper inclusion. The w_s of L is aab (or abb) and it can be accepted by one of the following loop transitions: $(aab, \lambda), (\lambda, aab), (abb, \lambda)$ or (λ, abb) by an **NS** sensing $5' \rightarrow 3'$ WK automaton. Each of mentioned transitions can lead to accept different language from the language $\{a^{(2n+m)}b^{(2m+n)} \mid n, m \geq 0\}$. Therefore, the language L cannot be accepted by **NS** sensing $5' \rightarrow 3'$ WK automata.

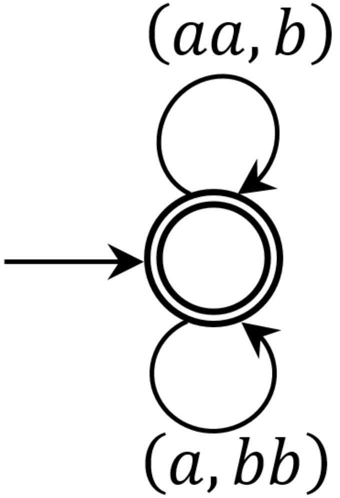


Figure 4: An **N** sensing $5' \rightarrow 3'$ WK automaton accepts the language L.

Theorem 4. The class of languages that can be accepted by **F1** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class accepted by **FS** sensing $5' \rightarrow 3'$ WK automata.

Proof. Consider the language $L = \{(aa)^n(bb)^m | m \leq n \leq m+1, m \geq 0\}$. The word w_s can be aa and by Lemma 1, w_s cannot be the shortest nonempty accepted word for an **F1** sensing $5' \rightarrow 3'$ WK automaton.

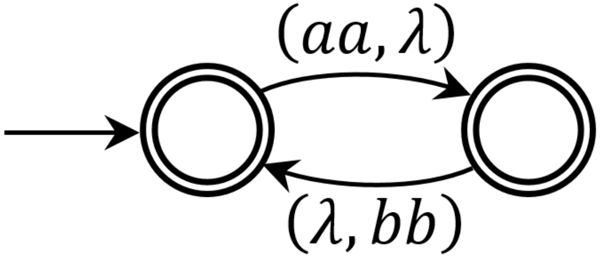


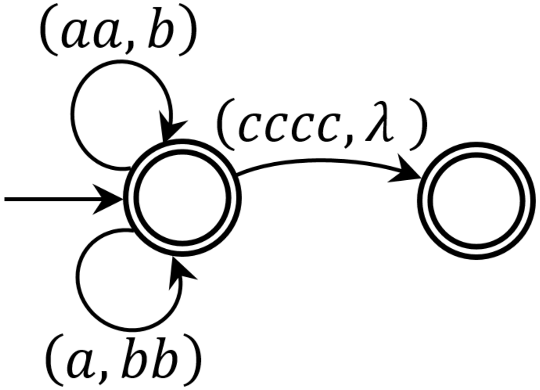
Figure 5: A sensing $5' \rightarrow 3'$ WK automaton of type **FS** accepting the language $\{(aa)^n(bb)^m | m \leq n \leq m+1, m \geq 0\}$.

Theorem 5. The class of languages that can be accepted by **FS** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class accepted by **F** sensing $5' \rightarrow 3'$ WK automata.

Proof. Let us assume, contrary that $L = \{a^{2n+q}c^{4m}b^{2q+n} | n, q \geq 0, m \in \{0,1\}\}$ is accepted by an **FS** sensing $5' \rightarrow 3'$ WK automaton. Let the radius of this automaton be r. Let $w = a^{2n+q}b^{2q+n} \in L$ with $n, q \geq r$ such that $|w| = 3n + 3q > r$. Then the word w cannot be accepted by using only one of the transitions (from the initial state), i.e., $\delta(q_0, a^{2n+q}b^{2q+n}, \lambda)$ or $\delta(q_0, \lambda, a^{2n+q}b^{2q+n})$ is not possible. Therefore, by considering the position of the heads after using any of the transitions from the initial state q_0 in **FS** sensing $5' \rightarrow 3'$ WK automaton, it is clear that either a prefix or a suffix of w with length at most r is accepted by the automaton. But neither a word from a^+, nor from b^+ is in L. This fact contradicts to our assumption, hence L cannot be accepted by any **FS** sensing $5' \rightarrow 3'$ WK automata.

However, L can be accepted by an **F** sensing $5' \rightarrow 3'$ WK automaton (next)

Figure 6: A sensing 5′ → 3′ WK automaton of type **F** accepting the language $\{a^{2n+q}c^{4m}b^{2q+n} | n, q \geq 0, m \in \{0,1\}\}$.



Theorem 6. The language class accepted by **F** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class of sensing $5' \rightarrow 3'$ WK automata.

Proof. A sensing $5' \rightarrow 3'$ WK automaton (without restrictions) accepts the language $L = \{a^n c b^n c | n \geq 1\}$. Now we show that there is no **F** sensing $5' \rightarrow 3'$ WK automaton which accepts L. Assume the contrary that the language L is accepted by an **F** sensing $5' \rightarrow 3'$ WK automaton. Let the radius of the automaton be r. Let $w = a^n c b^n c \in L$ with $m \geq r$. Thus the word w cannot be accepted by applying exactly one transition from the initial state q_0. Now, suppose that there exists $q \in \delta(q_0, w_1, w_2)$ such that w can be accepted by using transition(s) from q. Since in **F** sensing $5' \rightarrow 3'$ WK automaton all states are final, then the concatenation of w_1 and w_2 is accepted, thus, it must be in L (i.e. $w_1 w_2 \in L$). Therefore $w_1 w_2 = a^{m'} c b^{m'} c$ where $2m' + 2 \leq r \leq m$. To expand both blocks a^+ and b^+ to continue the accepting path of w, the left head must be right after the subword $a^{m'}$, and the right head must be right before after the subword $b^{m'}$. However, this is contradicting the fact that the two heads together already read $a^{m'} c b^{m'} c$.

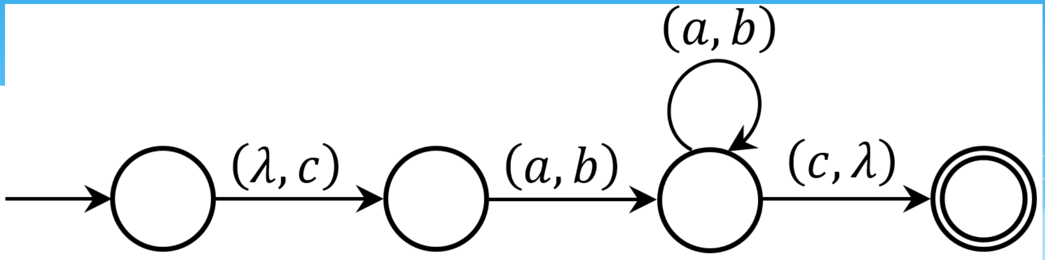


Figure 7: A sensing $5' \rightarrow 3'$ WK automaton accepts the language $L = \{a^n c b^n c | n \geq 1\}$.

Proposition 1. The language $L = \{a^n b^m | n = m \; or \; n = m + 1\}$ can be accepted by **F1** sensing $5' \to 3'$ WK automata, but cannot be accepted by **N1, NS** and **N** sensing $5' \to 3'$ WK automata.

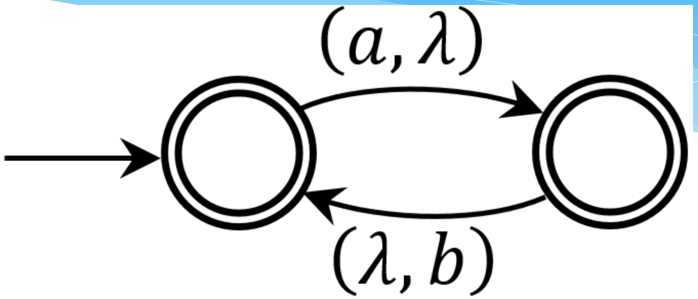


Figure 9: An **F1** sensing $5' \to 3'$ WK automaton accepts the language $L = \{a^n b^m | n = m \; or \; n = m + 1\}$.

Remark 2. The following statements follow from Proposition 1:

- **N1⊂ F1**
- **NS⊂FS**
- **N⊂F**

Incomparability results

Theorem 7. The class of languages that can be accepted by **N** sensing $5' \rightarrow 3'$ WK automata is incomparable with the classes of languages that can accepted by **FS** and **F1** sensing $5' \rightarrow 3'$ WK automata under set theoretic inclusion.

Proof. The language $L = \{ww^R | w \in \{a, b\}^*\}$ can be accepted by an **N** sensing $5' \rightarrow 3'$ WK automaton. Suppose that an **FS** sensing $5' \rightarrow 3'$ WK automaton accepts L. Let the radius of this automaton be r. Let $w_2 = w_1 w_1^R \in L$ with $w_1 = (bbbaaa)^m$ and $m > r$. The word w_2 cannot be accepted by using only one of the transitions from the initial state q_0, i.e., $\delta(q_0, w_1 w_1^R, \lambda)$ or $\delta(q_0, \lambda, w_1 w_1^R)$ is not possible (because the length of w_2). Therefore there exists either $q \in \delta(q_0, w_3 w_3^R, \lambda)$, $w_3 \in V^*$ or $\delta(q_0, \lambda, w_3 w_3^R)$,$w_3 \in V^*$ such that w_2 can be accepted by using transition(s) from q. Since the word $w_3 w_3^R$ should be in the language L (i.e., it is an even palindrome) and the length of bbb and aaa patterns in w_2 is odd, the only even palindrome proper prefix (suffix) of w_2 is bb. Thus $w_3 w_3^R = bb$ must hold.

Without loss of generality, assume that there exists $q \in \delta(q_0, bb, \lambda)$ in the automaton. By continuing the process, we must have at least one of $q' \in \delta(q, w_4, \lambda)$ or $q' \in \delta(q, \lambda, w_4)$ such that $bbw_4 \in L$ and w_4 is either the prefix or the suffix of the remaining unread part of word w_2 , i.e., $ba^3(b^3a^3)^{m-1}(a^3b^3)^m$, with length less than m. Clearly, w_4 cannot be a prefix, and it can be only the suffix bb. Thus, in q' the unprocessed part of the input is $ba^3(b^3a^3)^{m-1}(a^3b^3)^{m-1}a^3b$. Now the automaton must read a prefix or a suffix of this word, let us say w_5 such that $bbw_5bb \in L$, that is w_5 itself is an even palindrome, and its length is at most $r < m$. But such a word does not exist, the length of bbb and aaa patterns in the unread part is odd and their length is more than r. It contradicts to accept the language L by any **FS** sensing $5' \to 3'$ WK automaton.

The language $L = \{a^n b^m | n = m \; or \; n = m + 1\}$ proves the other direction which is accepted by an **F1** sensing $5' \to 3'$ WK automaton as it is shown in Figure 8 .

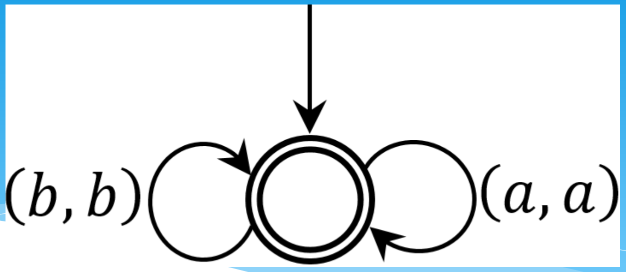


Figure 10: A sensing $5' \rightarrow 3'$ WK automaton of type **N** accepting the language of even palindromes $L = \{ww^R | w \in \{a,b\}^*\}$.

Theorem 8. The class of languages that can be accepted by **NS** sensing $5' \rightarrow 3'$ WK automata is incomparable with the language class accepted by **F1** sensing $5' \rightarrow 3'$ WK automata.

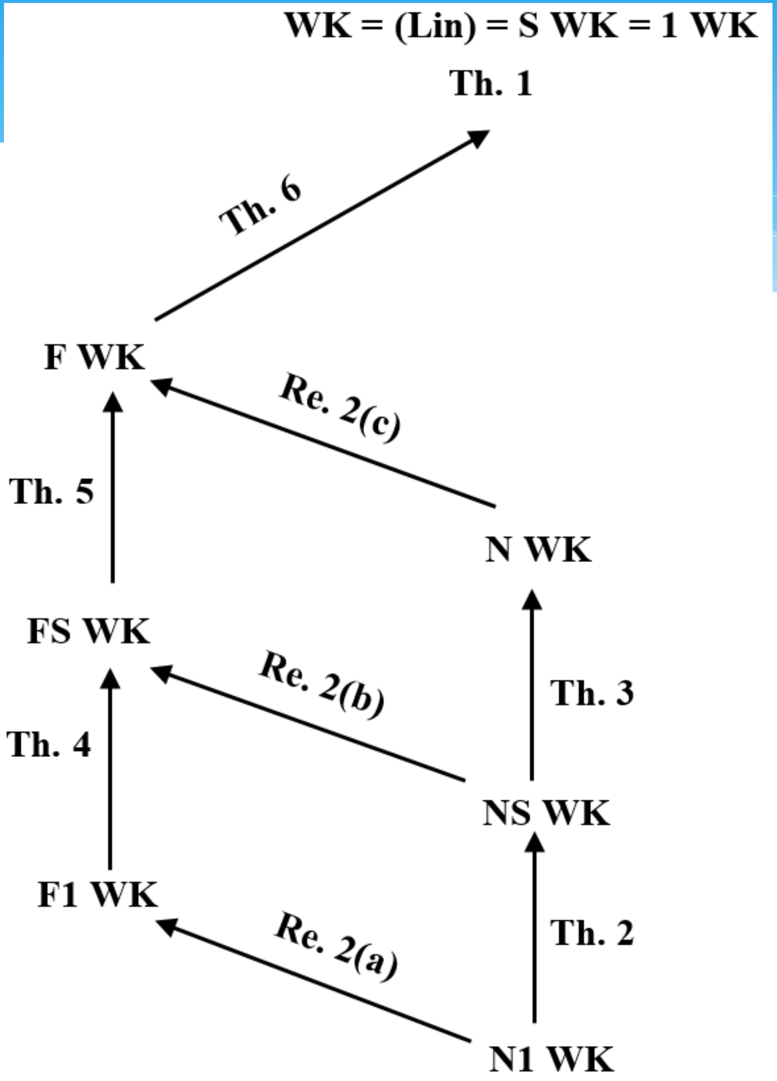
Proof. Consider the language $L = \{a^{3n}b^{2m} | n, m \geq 0\}$. It can be accepted by **NS** sensing $5' \rightarrow 3'$ WK automaton (Figure 3). Although, according to Lemma 1, w_s is bb and it cannot be the shortest nonempty accepted word for an **F1** sensing $5' \rightarrow 3'$ WK automaton. Therefore, this language cannot be accepted by an **F1** sensing $5' \rightarrow 3'$ WK automaton.

Now let us consider the language $L = \{a^n b^m | n = m \; or \; n = m + 1\}$ which can be accepted by an **F1** sensing $5' \rightarrow 3'$ WK automaton (Figure 9). By proposition 1, it is already shown that L cannot be accepted by **N** sensing $5' \rightarrow 3'$ WK automata and obviously by any **NS** sensing $5' \rightarrow 3'$ WK automata, neither.

Table 1: Some specific languages belonging to language classes accepted by various classes of **WK** automata. Reference to figures indicate a specific automaton that accept the given language. × indicates that the language cannot be accepted by the automata type of the specific column. Trivial inclusion are shown, e.g., in the first line **N1**in, e.g., column **F** means that every **N1** automaton is, in fact, also an **F** automaton.

Languages	**N1**	**NS**	**N**	**F1**	**FS**	**F**	**WK**
$\{a^n b^m \mid n, m \geq 0\}$	+	N1	N1	N1	N1	N1	N1
$\{a^{3n} b^{2m} \mid n, m \geq 0\}$	×	+	NS	NS	NS	NS	NS
$\{ww^R \mid w \in \{a, b\}^*\}$	×	×	+	×	×	N	N
$\{a^n b^m \mid n = m\ or\ n = m + 1\}$	×	×	×	+	F1	F1	F1
$\{(aa)^n (bb)^m \mid m \leq n \leq m + 1, m \geq 0\}$	×	×	×	×	+	FS	FS
$\{a^{2n+q} c^{4m} b^{2q+n} \mid n, q \geq 0, m \in \{0,1\}\}$	×	×	×	×	×	+	F
$\{a^n c b^n c \mid n \geq 1\}$	×	×	×	×	×	×	+

Figure 11: Hierarchy of sensing $5' \rightarrow 3'$ WK finite automata languages in a Hasse diagram.



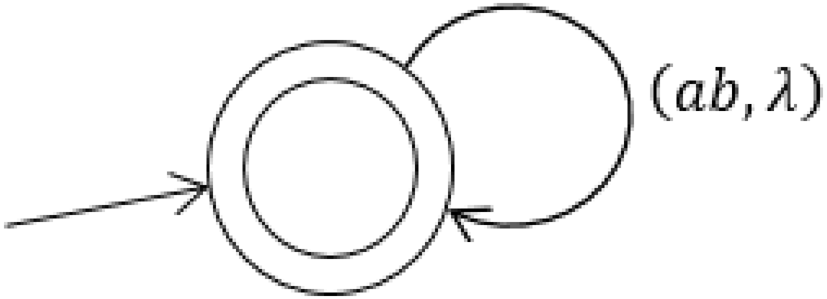
Deterministic sensing $5' \rightarrow 3'$WK automata

Now, we consider the deterministic variants of these automata. If at each possible configuration at most one transition is possible, then a WK automaton is deterministic. We note that for the traditional WK automata reading both stands completely, there are various definitions of determinism (related also to the used complimentary relation), but for our automata there is only one of them applicable.

2detLIN is a proper of subset of LIN.

Theorem. The language class that can be accepted by deterministic ***N*** sensing $5' \rightarrow 3'$ WK automata is properly included in the language class accepted by deterministic ***NS*** sensing $5' \rightarrow 3'$ WK automata.

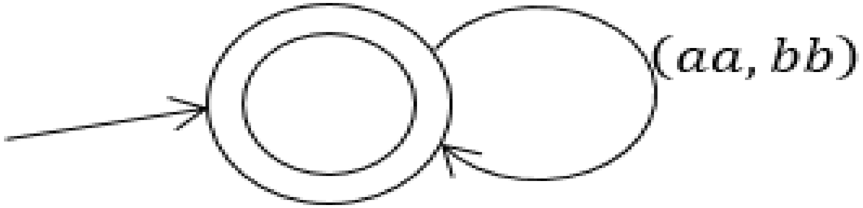
Proof. We can prove it easily by the definition. Let us consider the language $L = \{(ab)^n | n \geq 0\}$. The word w_s is ab and in **NS** sensing $5' \rightarrow 3'$WK automaton it can be accepted by one of the transitions: (λ, ab) or (ab, λ). By Lemma 2, w_s cannot be the shortest nonempty word accepted by an **N1** sensing $5' \rightarrow 3'$ WK automaton. In the other hand, the language L, as shown in Figure 9, can be accepted by an **NS** sensing $5' \rightarrow 3'$WK automaton.



A deterministic sensing $5' \rightarrow 3'$ WK automaton of type **NS** accepting the language $\{(ab)^n | n \geq 0\}$.

Remember: NS contains only regular.

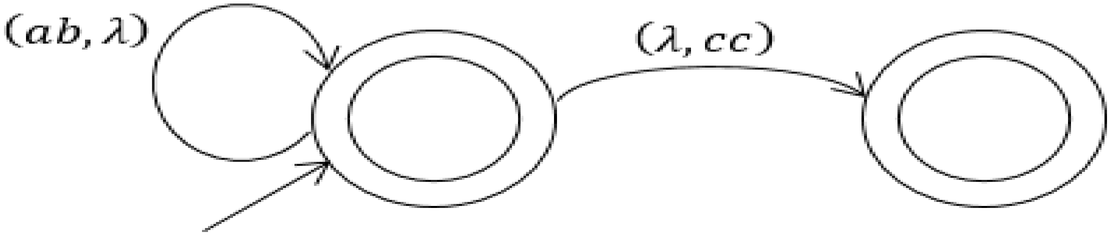
A non regular example for determinsitic no state automaton:



A deterministic sensing $5' \rightarrow 3'$WK automaton of type N accepting the language $\{a^{2n}b^{2n}|n \geq 0\}$.

Theorem. The language class that can be accepted by deterministic $\boldsymbol{F1}$ sensing $5' \rightarrow 3'$WK automata is properly included in the language class accepted by deterministic $\boldsymbol{FS}$ sensing $5' \rightarrow 3'$ WK automata.

Proof. Let us consider the language $L = \{(ab)^n c^{2m} | n \geq 0, m \in \{0, 1\}\}$. The word w_s of this language is ab or cc which can be accepted by $(\lambda, ab), (ab, \lambda), (cc, \lambda)$ or (λ, cc) in **FS** sensing $5' \rightarrow 3'$ WK automaton. According to Lemma 2, w_s cannot be the shortest nonempty accepted word in **F1** sensing $5' \rightarrow 3'$ WK automaton. But L can be accepted by **FS** sensing $5' \rightarrow 3'$WK automata.



A deterministic sensing $5' \rightarrow 3'$ WK automaton of type **FS** accepting the language $\{(ab)^n c^{2m} | n \geq 0, m \in \{0, 1\}\}$

Theorem. The class of languages that can be accepted by deterministic $\boldsymbol{FS}$ sensing $5' \rightarrow 3'$ WK automata is properly included in the language class of deterministic $\boldsymbol{F}$ sensing $5' \rightarrow 3'$ WK automata**.**

Proof. The inclusion is obvious by the definition. Now, we present the language $L = \{a^{2n}c^{5q}b^{2n} | n \geq 0, q \in \{0,1\}\}$. Suppose that this language is accepted by **FS** sensing $5' \rightarrow 3'$ WK automaton. Therefore, w_s is accepted by one of the transitions: $(\lambda, aabb)$ or $(aabb, \lambda)$. If this transition is a loop, then the automaton can accept all the words of $\{(aabb)^n | n \geq 0\}$ which is not a subset of L. Therefore, automaton should has one of transitions from initial state to another state.

In this case, it is impossible for the machine to accept the word(s) which contains $a^{2n}b^{2n}$ where $n \geq 2$ after reading $(\lambda, aabb)$ or $(aabb, \lambda)$ transitions. Indeed, in each transition by considering position of heads, the automaton can only read and accept from one of the blocks a^{2n} or b^{2n}. Since the automaton is deterministic, it is impossible to have another transition from initial state to other states by reading any a and b letters by the first and second heads, respectively. Therefore, L cannot be accepted by any **FS** sensing $5' \rightarrow 3'$ WK automata, but it can be accepted by. **F** sensing $5' \rightarrow 3'$WK automata.

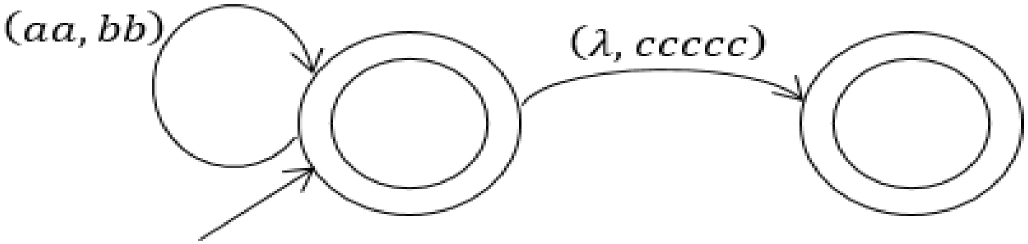


Figure 12: A deterministic sensing $5' \rightarrow 3'$ WK automaton of type **F** accepting the language $\{a^{2n}c^{5q}b^{2n} | n \geq 0, q \in \{0,1\}\}$

Even linear and fix-rated linear languages

- A linear grammar is k-rated linear (for a non-negative rational value of k) if for every production with nonterminal on the right side, e.g., A → uBv the ratio |v| / |u| = k.

- k=0: regular
- k=1: even linear (e.g., palindromes, a^nb^n) both-head stepping 5'-3' WK automata

- Fix-rated linear languages are in 2detLIN.

- Note:
- 2detLIN is incomaparble with detLIN (defined by determinsitic one-turn automata)

- The language of palindromes is in 2detLIN,

- $\{a^n b^n\} \cup \{a^n c b^{2n}\}$ is in detLIN…

Thank you!

The end.