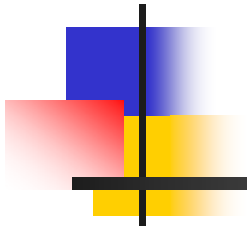


Computational Learning Theory
Extending Patterns with Deductive Inference

Akihiro Yamamoto 山本 章博
http://www.iip.ist.i.kyoto-u.ac.jp/member/akihiro/
akihiro@i.kyoto-u.ac.jp

Contents

- What about a pair of patterns?
- Elementary formal systems
- Transferring FA into EFS
- Transferring CFG into EFS

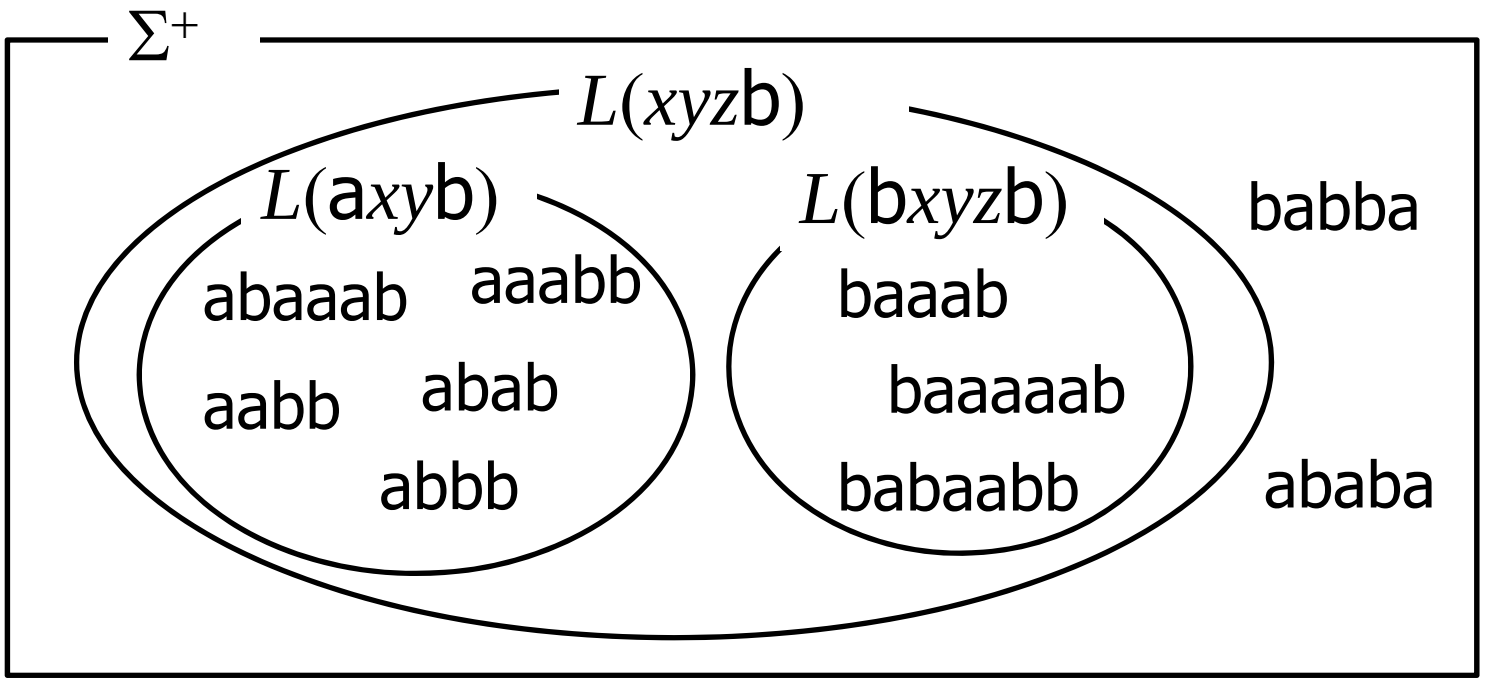
What about a pair of patterns?

Outputting a Pair of Patterns

$C = \{$babaabb, aaab, baaab, aabb, abab, baaaaab, abbb, aaabb, baaab$\}$

abaaab, aabb, abab, abbb, aaabb $\in L($axb$)$

baaab, baaaaab, babaabb $\in L($bxyzb$)$

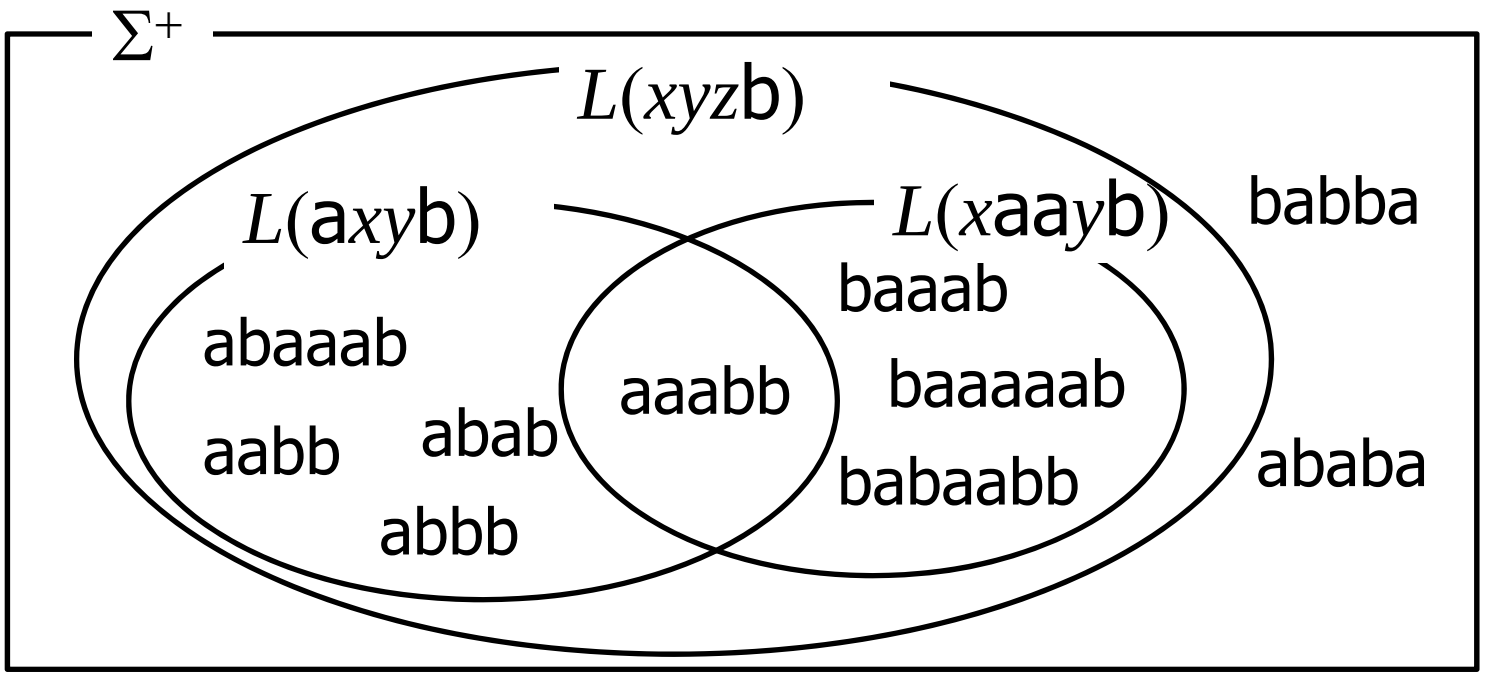


Outputting a Pair of Patterns

$C = \{$babaabb, aaab, baaab, aabb, abab, baaaaab,
abbb, aaabb, baaab$\}$

abaaab, aabb, abab, abbb, aaabb $\in L($axyb$)$

baaab, baaaaab, aaabb, babaabb $\in L($$xaayb)$



Downward Coverset Algorithm

Given a data set C

For each minimally common anti-unifier π of C

For each a pair π_1, π_2 of patterns just beneth π in the Hasse Diagram

if $L(\pi_2) \subset C - L(\pi_1)$ then

make minimally common anti-unifier π_1'of $C \cap L(\pi_1)$

and

minimally common anti-unifier π_2'of $C - L(\pi_1)$

Elementary Formal Systems

Introducing Predicate Symbols

- Introducing a new class of symbols called predicate symbols.
  - Every predicate symbol is interpreted as a language (a set of strings) or a set of tuples $(s_1, s_2,\ldots,s_n)$ of strings.
  - In this course we use symbols p, q, r,… which are respectively interpreted as sets P, Q, R,….
- An atomic formula is a formula of the form

$$\mathsf{p}(\pi_1, \pi_2,\ldots, \pi_n)$$

where $\pi_1, \pi_2,\ldots, \pi_n$ are patterns. If n=1 and $\pi_1 = s$ is ground, p(s) is interpreted as $s \in$ P.

Example

Some examples of atomic formulae are p(axb), q(ax, by), q(x, bxb), p(aabb), q(aa, bb). The last two formulae are ground.

Definite Clause (Rules) and EFS

- An definite clause is a formula of the form

$$p(\pi_1,\ldots,\pi_n) \leftarrow q_1(\tau_{11},\ldots), q_2(\tau_{21},\ldots),\ldots, q_k(\tau_{k1},\ldots)$$

where $\pi_1, \pi_2,\ldots, \tau_{11},\ldots, \tau_{k1},\ldots$ are patterns. The definite clause is interpreted as

"for any substitution θ, if $(\tau_{11}\theta,\ldots)\in \mathrm{Q}_1, (\tau_{21}\theta,\ldots)\in \mathrm{Q}_2,\ldots,$ $q_k(\tau_{k1}\theta,\ldots)\in \mathrm{Q}_k$ then $(\pi_1\theta, \pi_2\theta,\ldots, \pi_n\theta) \in \mathrm{P}$"

  - A clause $p(\pi_1,\ldots,\pi_n)\leftarrow$ which has no conditions is sometimes called a unit clause.

- A finite set of definite clause is called an elementary formal system (EFS). [Smullyan 61]

Examples

Some examples of definite clauses are

$p(a x) \leftarrow r(x)$

$r(b)$

$p(a x b y) \leftarrow r(x), r(y)$

$q(a x, b y) \leftarrow q(x, y)$

…

Raymond Smullyan

From Wikipedia, the free encyclopedia

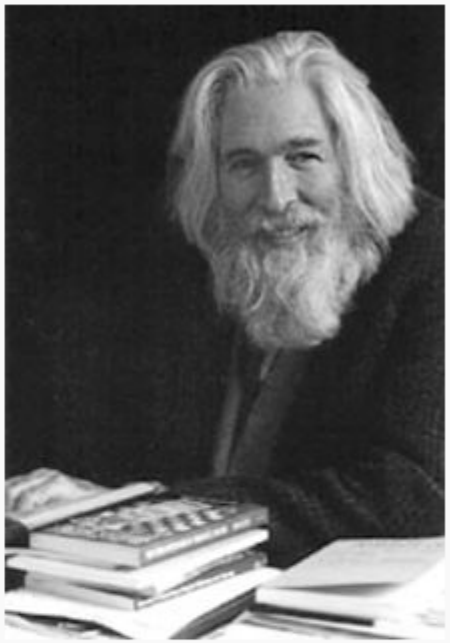
This article includes a list of references, but **its sources remain unclear because it has insufficient inline citations**. Please help to improve this article by introducing more precise citations. *(February 2008)*

Raymond Merrill Smullyan (born May 25, 1919)[1] is an American mathematician, concert pianist, logician, Taoist philosopher, and magician.

Born in Far Rockaway, New York, his first career was stage magic. He then earned a BSc from the University of Chicago in 1955 and his Ph.D. from Princeton University in 1959. He is one of many logicians to have studied under Alonzo Church.[1]

Contents [hide]

Raymond Merrill Smullyan



Born	May 25, 1919 (age 95) Far Rockaway, New York
Occupation	mathematician, logician, philosopher, pianist and magician
Nationality	American

Life [edit]

Inference Rules for Definite Clauses

- A goal clause is a sequence (conjunction) of atomic formulae $p_1(\tau_{11},\ldots),\ldots, p_k(\tau_{k1},\ldots)$
- We use the following two rules

Instantiation

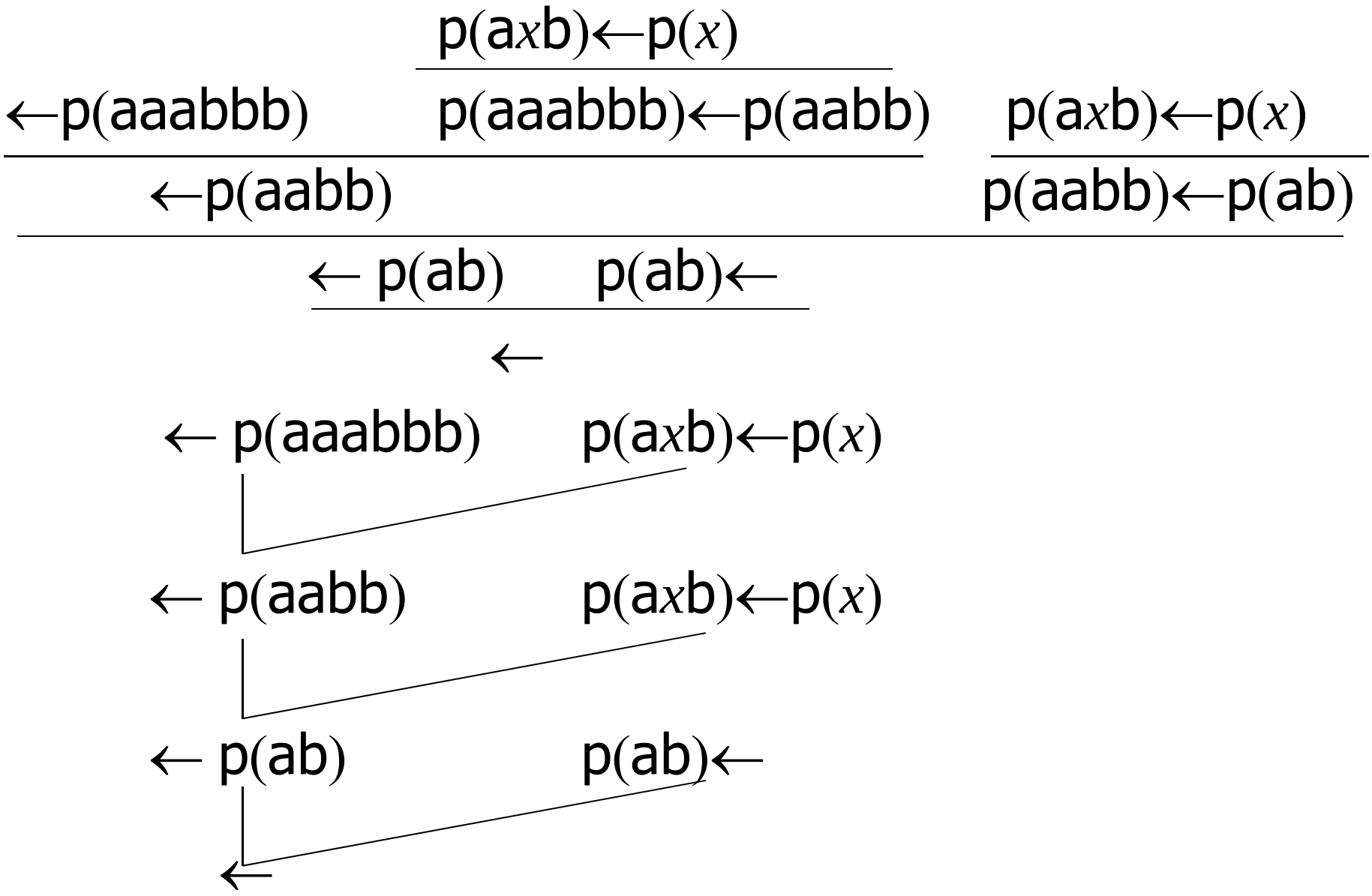
$$\frac{p(\pi_1,\ldots) \leftarrow q_1(\tau_{11},\ldots), q_2(\tau_{21},\ldots),\ldots, q_k(\tau_{k1},\ldots)}{(p(\pi_1,\ldots) \leftarrow q_1(\tau_{11},\ldots), q_2(\tau_{21},\ldots),\ldots, q_k(\tau_{k1},\ldots))\theta}$$

Modus Pones

$$\frac{\leftarrow p_1(\pi_{11},\ldots),\ldots, p_k(\pi_{k1},\ldots) \qquad p_1(\pi_1,\ldots) \leftarrow q_1(\tau_{11},\ldots),\ldots}{\leftarrow q_1(\tau_{11},\ldots),\ldots, p_2(\pi_{21},\ldots),\ldots, p_k(\pi_{k1},\ldots)}$$

- A proof is a continuous application of the inference rules.

Example of Proof (1)



Defining a language by proofs

- A ground atomic formula $p(s_1,\ldots,s_n)$ is provable from an EFS S if

  there is a proof which derives an empty clause from is $\leftarrow p(s_1,\ldots,s_n)$ and S.

- We define a language with a proof from an EFS.

  $P = L(p, S) = \{ s \mid p(s)$ is provable from $S\}$

Example

S : p(axb)←p(x)

p(ab)←

P = L(p, S) ={ab, aabb, aaabbb, aaabbb,…}

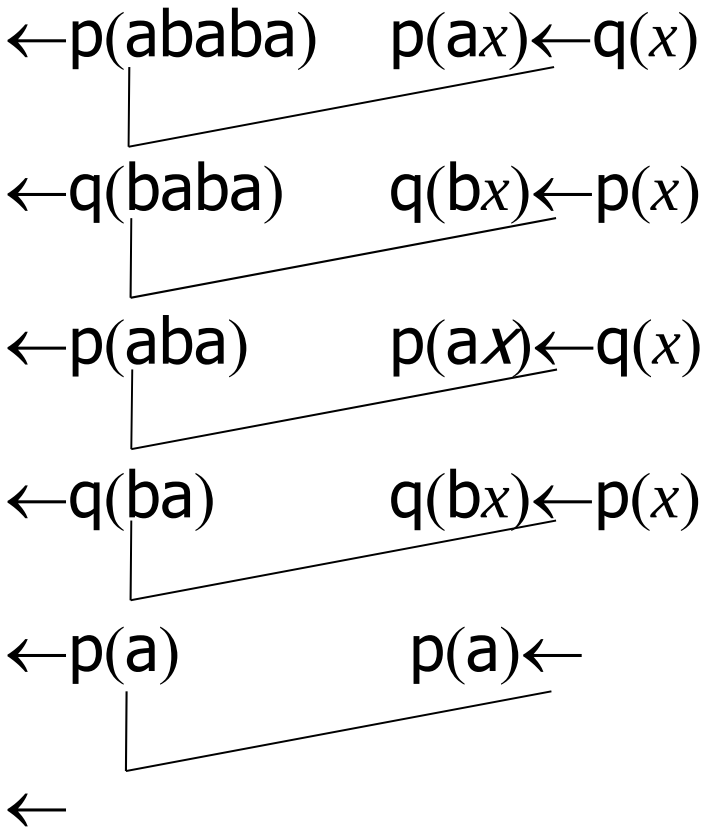
Example of Proof (2)

S: p(ax)←q(x) q(bx)←p(x)

p(a)← q(b)←

L(p, S) ={a, ab, aba, abab,…}

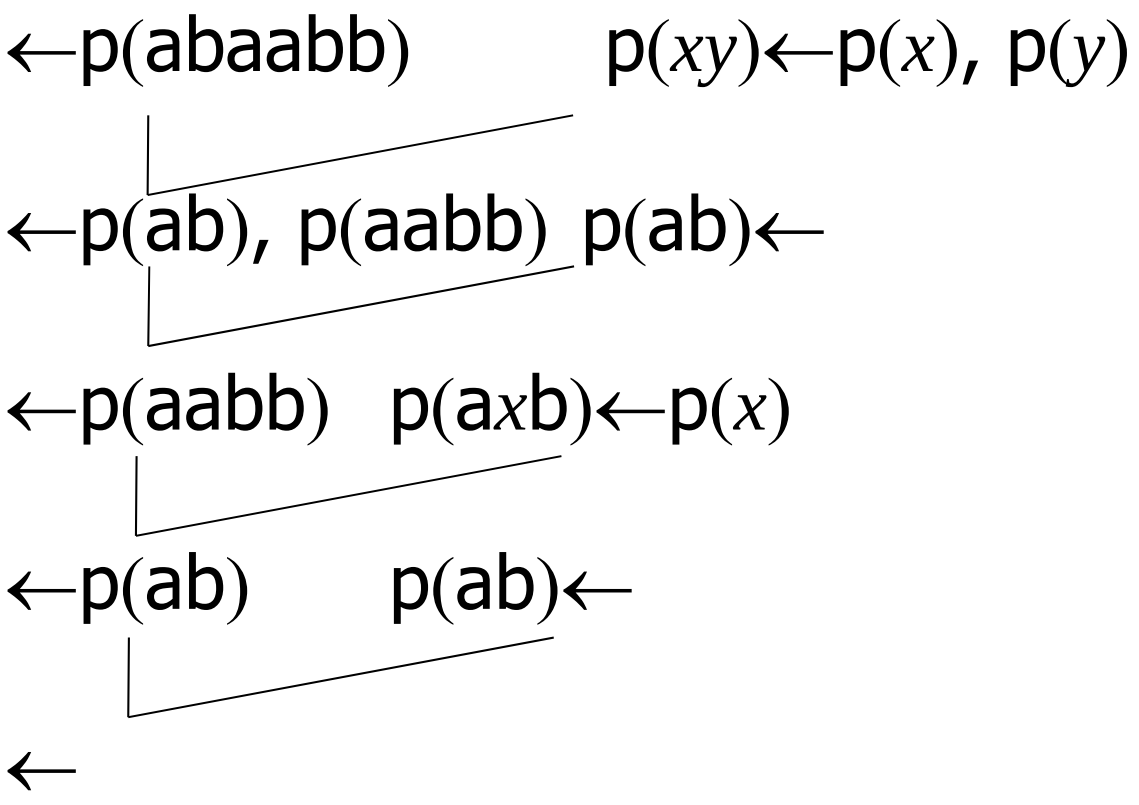
L(q, S) ={b, ba, bab, baba,…}



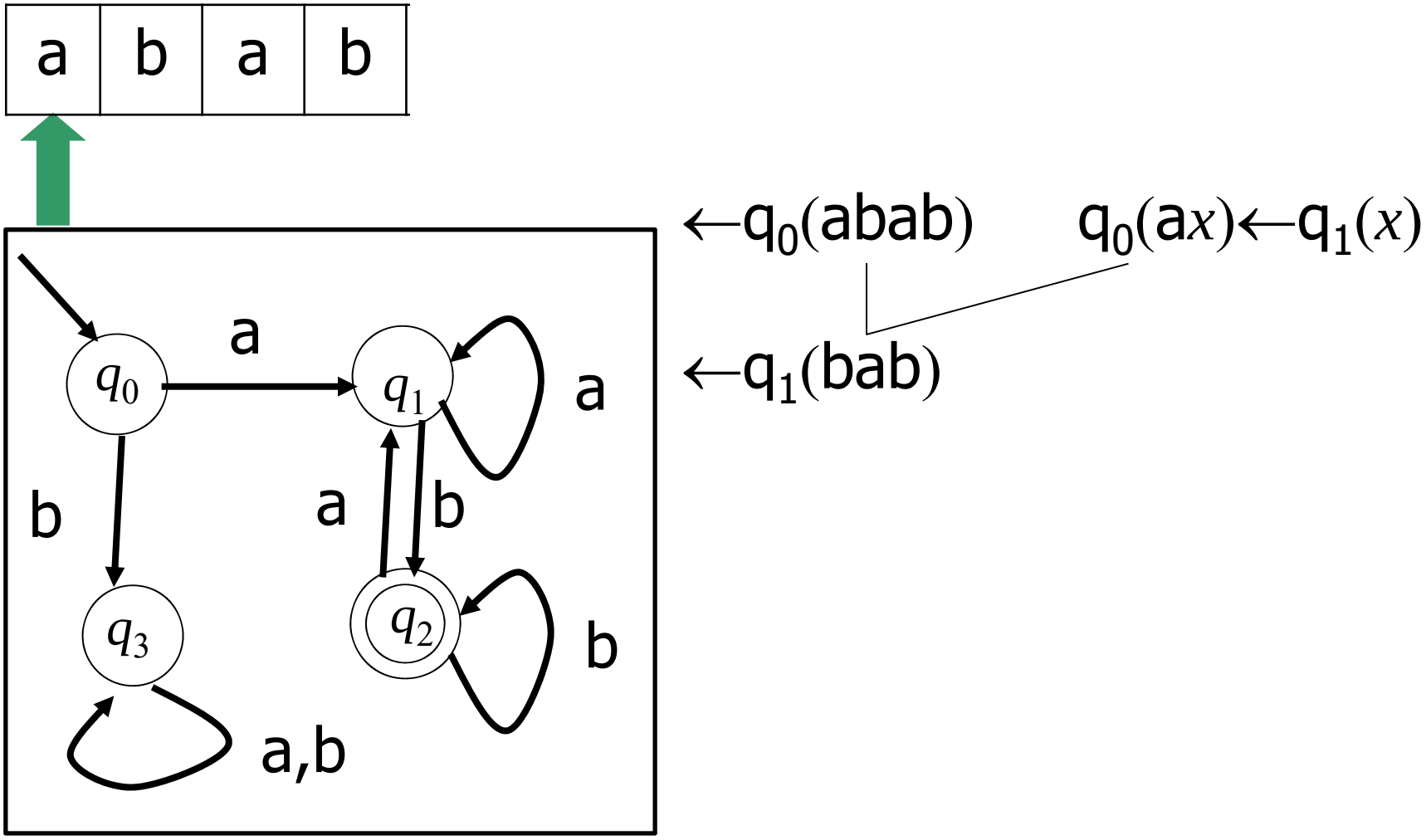
Example of Proof (3)

S: p(axb)←p(x) p(xy)←p(x), p(y) p(ab)←

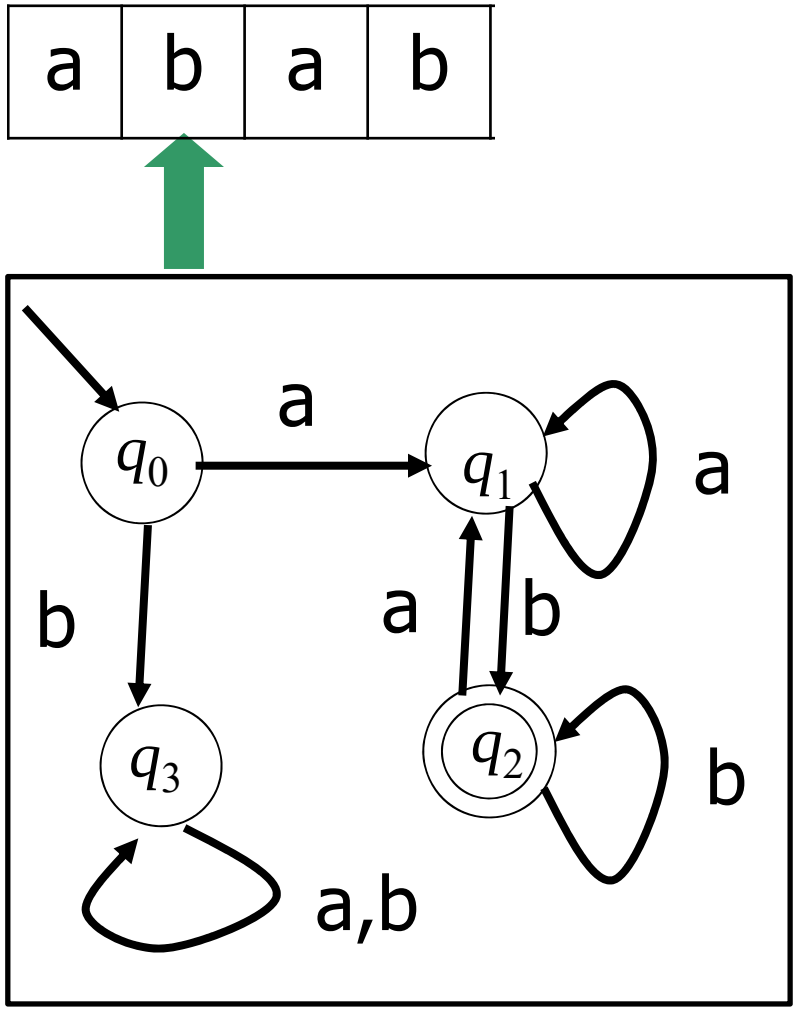
L(p, S) ={ab, aabb, aaabbb, abaabb,…}



States of Languages in FA



States of Languages in FA

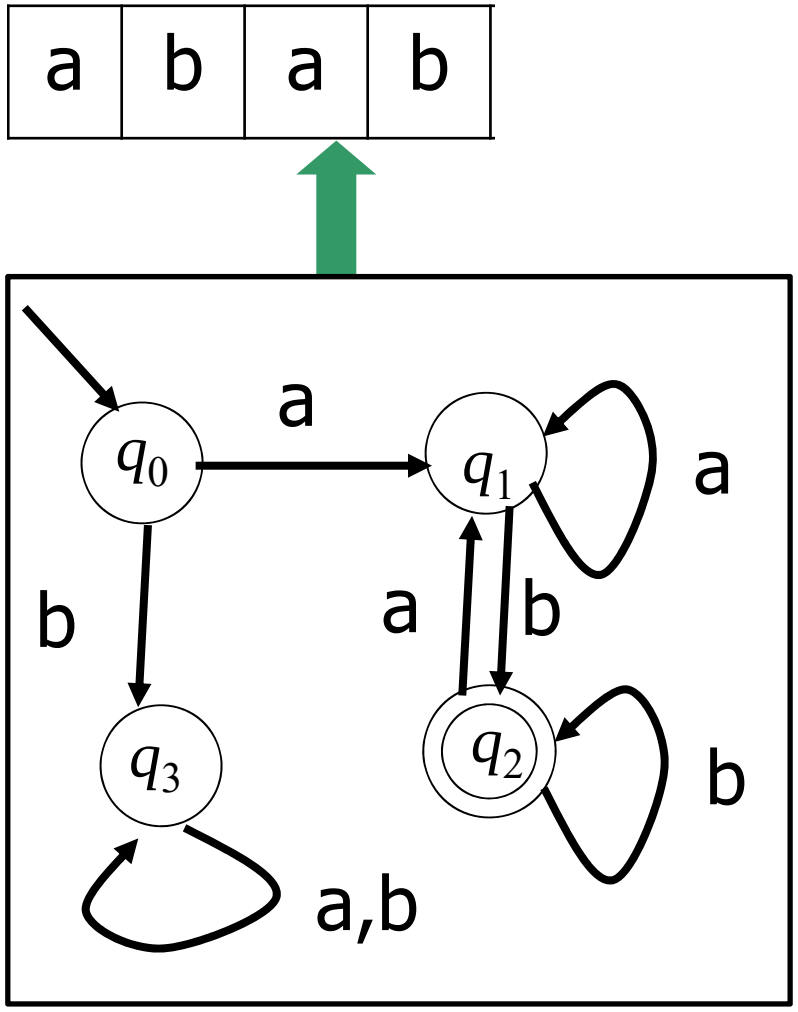


$\leftarrow q_0(abab)$ $\quad q_0(ax) \leftarrow q_1(x)$

$\leftarrow q_1(bab)$ $\quad q_1(bx) \leftarrow q_2(x)$

$\leftarrow q_2(ab)$

States of Languages in FA



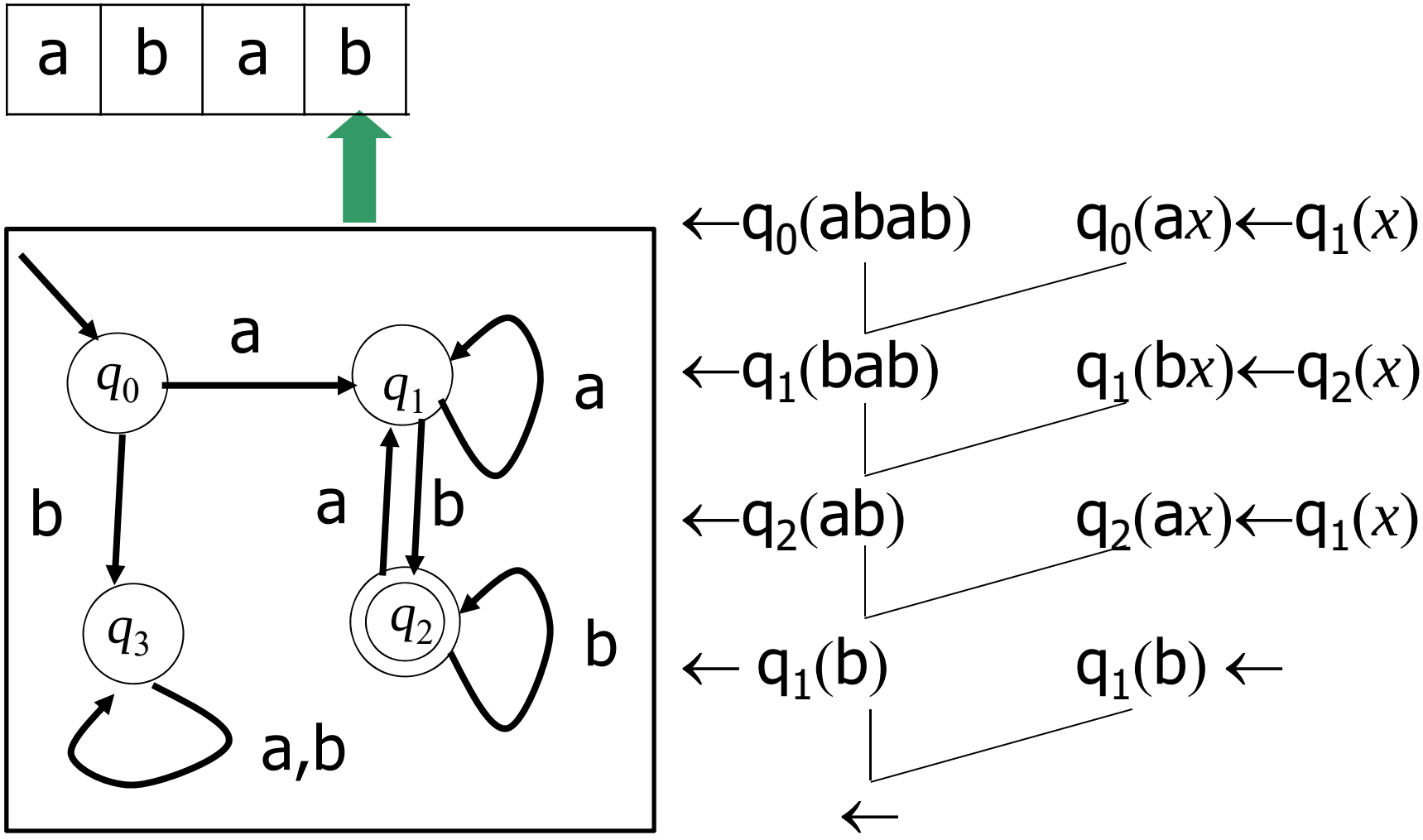
$\leftarrow q_0(abab)$ $\quad q_0(ax) \leftarrow q_1(x)$

$\leftarrow q_1(bab)$ $\quad q_1(bx) \leftarrow q_2(x)$

$\leftarrow q_2(ab)$ $\quad q_2(ax) \leftarrow q_1(x)$

$\leftarrow q_1(b)$

States of Languages in FA



Representation of Finite State Automata

- Mathematically, a finite state automaton is represented in the form $M=(\Sigma, S, \delta, s_0, F)$

  where

  Σ is the alphabet,

  S is a set of states,

  $\delta : S \times \Sigma \rightarrow S$ is a transition function represented as a transition table,

  $s_0 \in S$ is an initial state,

  $F \subset S$ is a set of final states.

	F	a_1	…	a_n
q_0				
…				
q_m				

Transformation from Finite State Automata

- We use the elements in S as a predicate symbols.
- If $\delta(p, c) = q$, we prepare a definite clause

  $p(cx) \leftarrow q(x)$
- If $\delta(p, c) = q$ and $q \in F$ we prepare a definite clause

  $p(c) \leftarrow$

	F	a_1	…	a_n
q_0				
…				
q_m				

Context Free Grammar

- A context free grammar is defined as $G = (N, \Sigma, P, S)$ where

  N is a finite set of non-terminal symbols (non-terminals, syntactic categories).

  Σ is a finite set of characters or terminals.

  P is a set of rules (productions).

  $S \in N$ is an initial state.

- A production is of the form $A \rightarrow \gamma$ where

  A is a non-terminal and γ is a sequence of terminals and non-terminals.

  - Note: In general definition γ can be an empty sequence ε.

    In this course we do not allow γ to be ε.

Derivations

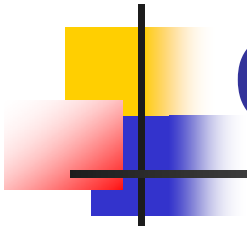
- An application of a production rule $A \to \gamma \in P$ is written as $\alpha A \beta \Rightarrow \alpha \gamma \beta$ where $\alpha, \beta \in (N \cup \Sigma)^*$
  - This means that only one occurrence of A in a string is replaced with γ.

$$S * (S + S) \Rightarrow V * (S + S)$$
$$S * (S + S) \Rightarrow S * (V + S)$$
$$S * (S + S) \not\Rightarrow V * (V + V)$$

- A derivation from α to β is a finite sequence of applications starting with α and ending with β:

$$\alpha = \gamma_1 \Rightarrow \gamma_2 \Rightarrow \ldots \Rightarrow \gamma_n = \beta$$

We write $\alpha \Rightarrow^* \beta$ if $\alpha = \beta$ or there is a derivation from α to β.



Context Free Languages

- The language defined a context free grammar $G = (N, \Sigma, P, S)$ is $L(G) = \{ w \in \Sigma^* \mid \text{there is a derivation } S \Rightarrow^* w \}$.

- A language L is context free if there is a context free grammar G such that $L = L(G)$.

Example For $G = (\{S\}, \{\mathsf{a}, \mathsf{b}\}, \{S \to \mathsf{ab},\ S \to \mathsf{aSb}\}, S)$, $L(G) = \{ \mathsf{a}^n\mathsf{b}^n \mid n \geq 1 \}$.

Example 1

- $\Sigma = \{a, b\}$
- $L = \{ a^n b^n \mid n \geq 1 \} = \{ \underbrace{a...a}_{n \text{ times}} \underbrace{b...b}_{n \text{ times}} \mid n \geq 1 \}$

  $= \{ab, aabb, aaabbb, aaaabbbb,... \}$
- The language is defined with a set of productions:

  $S \rightarrow ab, \quad S \rightarrow aSb$
- Some examples of derivations:

  $S \Rightarrow ab$

  $S \Rightarrow aSb \Rightarrow aabb$

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

  $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaabbbb$
- It is easy to show that there is no FA which accepts L.

Example 2

- Mathematical formulae consisting of x, y, +, *, (, and)

  correct: $x + y$, $y * (x + y)$, $((x * x) + x)$, x , …

  incorrect: $x +$, $(y * x + y$, $( )$, …

  Note: We assume strict use of (and).

  correct: $(x + y)$, (x)

  incorrect: $x * x * x$, $y * x + x$

- The set are defined by a set of rules

$$S \to V \qquad V \to x$$
$$S \to S + S \qquad V \to y$$
$$S \to S * S$$
$$S \to (S)$$

Example 2 (cont.)

- An example of a derivation:

$$S \to V \qquad S \to S * S \qquad V \to x$$
$$S \to S + S \qquad S \to (S) \qquad V \to y$$

$$S \Rightarrow S * S \Rightarrow S * (S) \Rightarrow S * (S + S)$$
$$\Rightarrow V * (S + S) \Rightarrow V * (V + V)$$
$$\Rightarrow y * (V + V) \Rightarrow y * (x + V) \Rightarrow y * (x + y)$$

Transforming CFG to EFS

- For every production rule

$$P \rightarrow w_1 Q_1 w_2 Q_2 \ldots Q_n w_{n+1} \qquad P,Q \in N,\ w_i \in \Sigma^*$$

we define a definite clause

$$\mathsf{p}(w_1 x_1 w_2 x_2 \ldots x_n w_{n+1}) \leftarrow \mathsf{q}_1(x_1), \mathsf{q}_2(x_2), \ldots, \mathsf{q}_n(x_n)$$

Example

$$P \rightarrow \mathsf{a}P\mathsf{b} \quad \Rightarrow \quad \mathsf{p}(\mathsf{a}x\mathsf{b}) \leftarrow \mathsf{p}(x)$$

$$P \rightarrow \mathsf{a}P\mathsf{b} \quad \Rightarrow \quad \mathsf{p}(\mathsf{ab}) \leftarrow$$

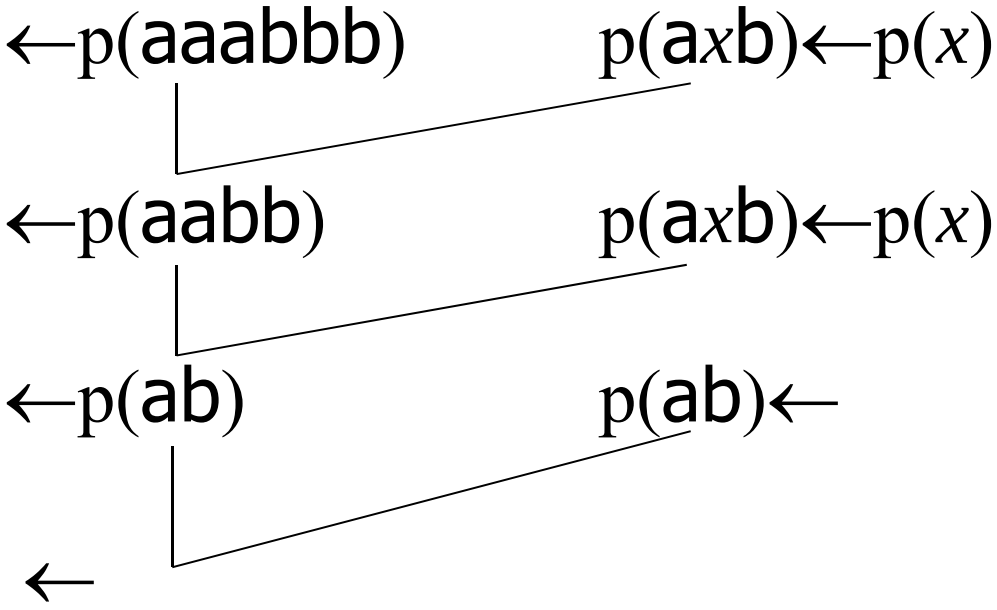
Derivation and Proof

- Productions and derivation

$$P \to \mathrm{ab},\ \ P \to \mathrm{a}P\mathrm{b}$$

$$P \Rightarrow \mathrm{a}P\mathrm{b} \Rightarrow \mathrm{aa}P\mathrm{bb} \Rightarrow \mathrm{aaabbb}$$

- EFS and proof



Notes

- There is a class of EFS which corresponds to the context sensitive grammar.
- There is a class of EFS which corresponds to TM.
- It is not easy to check which class $L(p, S)$ belongs to

$$p(xx) \leftarrow p(x)$$

$$p(\mathsf{a}) \leftarrow$$

References

- Arimura, H., Shinohara, T., and Otsuki, S. : Finding Minimal Generalization for Unions of Pattern Languages and Its application to Inductive Inference from Positive Data, Proc. of STACS'94 (LNCS 775), 649-660, 1994.
- Arikawa, S. Shinohara, T., and Yamamoto, A. : Learning Elementary Formal Systems, Theoretical Computer Science, 95(1), 97-113, 1992.