Computational Learning Theory
Learning from positive presentations

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

Revised version of *learn-patterns*

- ~~Fix an effective enumeration of patterns on $\Sigma \cup X$:~~

  ~~$\pi_1, \pi_2, \ldots,$~~

$k = 1,\ \pi = \pi_1$

for $n = 1$ forever

  receive $e_n = \langle s_n, b_n \rangle$

  while ($0 \leq \exists j \leq n$

    ($e_j = \langle s_j, + \rangle$ and $s_j \notin L(\pi)$) ~~and~~

    ~~($e_j = \langle s_j, - \rangle$ and $s_j \in L(\pi)$)~~

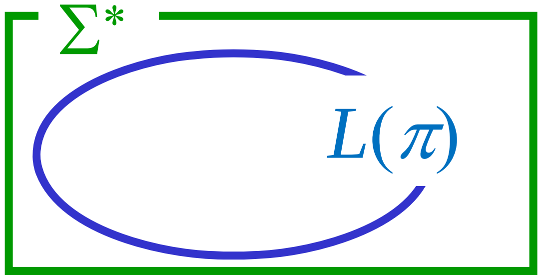
      $\pi = \pi'$ for an appropriate π'; k ++

  output π

Positive Presentations

$e_1, e_2, e_3, \ldots$

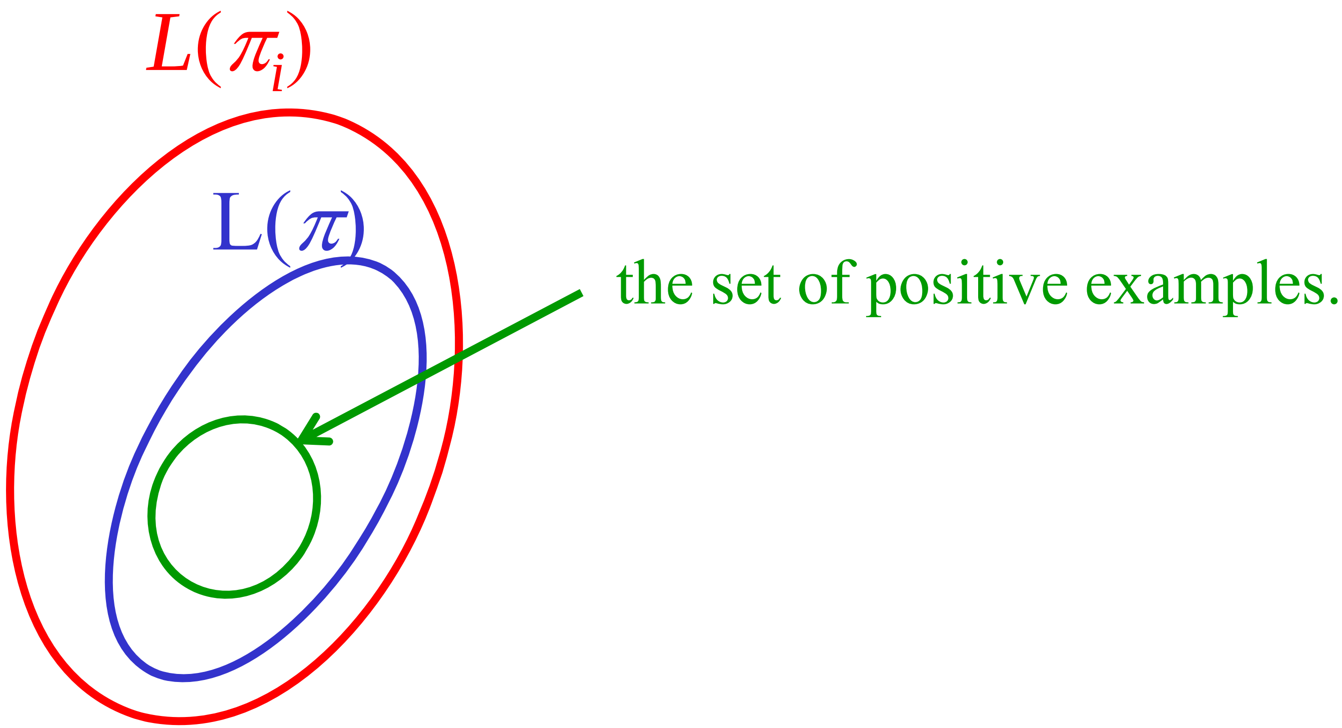
$\pi_1, \pi_2, \pi_3, \ldots$



- A presentation of $L(\pi)$ is a infinite sequence consisting of positive and negative example.
- A presentation σ is positive if σ consists only of positive example $< s, +>$ and any positive example occurs at least once in σ.

Which patterns should be chosen?

- Intuitively, choose a minimal language which contains all of the positive examples at the moment.
  - That is, avoid over-generalization!



Identification in the limit [Gold]

- A learning algorithm *A* EX-identifies a class C of languages in the limit from psoitive presentations if *A* EX-identifies every language in C in the limit from positive presentations.

- A learning algorithm *A* BC-identifies a class C of languages in the limit from positive presentations if *A* BC-identifies every language in C in the limit from positive presentations.

Analysis of Patterns

Analysis of Patterns (1)

Example π = a*xx*bb*y*aa

L(a*xx*bb*y*aa)

={aaabbaaa, aaabbbaa, abbbbaaa, abbbbbaa,
aaaaabbaaa, aaaaabbbaa, aababbbaaa,
aababbbbaa,…, aabaaabaabbbbbababaa,…}

- Using examples as long as π :

  aaabbaaa, aaabbbaa, abbbbaaa, abbbbbaa

θ_1={(x,a), (y,a)} θ_2={(x,a), (y,b)} θ_3={(x,b), (y,a)} θ_4={(x,a), (y,b)}

We can know that the 2nd, 3rd, and 6th positions must be variables.

The variable at the 6th position is different from those at the 2nd and 3rd.

Analysis of Patterns (2)

- Any language $L(\pi')$ containing the four strings must be a superset of $L(\pi)$.

aaabbaaa, aaabbbaa, abbbbaaa, abbbbbaa

$\theta_1=\{(x,\mathrm{a}), (y,\mathrm{a})\}$ $\theta_2=\{(x,\mathrm{a}), (y,\mathrm{b})\}$ $\theta_3=\{(x,\mathrm{b}), (y,\mathrm{a})\}$ $\theta_4=\{(x,\mathrm{a}), (y,\mathrm{b})\}$

  - If π' and π are of same length, π' has more variables than π.
  - If π' is shorter than π, π' has at least one variable with which some substring of π longer than 2 must be replaced.

Characteristic Set of $L(\pi)$

- Let π be a pattern which contains variables $x_1, x_2, ..., x_n$. Consider the following substitutions:

  $\theta_{\mathrm{a}} = \{(x_1, \mathrm{a}), (x_2, \mathrm{a}), ..., (x_n, \mathrm{a})\}$,

  $\theta_{\mathrm{b}} = \{(x_1, \mathrm{b}), (x_2, \mathrm{b}), ..., (x_n, \mathrm{b})\}$,

  $\sigma_1 = \{(x_1, \mathrm{a}), (x_2, \mathrm{b}), ..., (x_n, \mathrm{b})\}$,

  …

  $\sigma_n = \{(x_1, \mathrm{b}), (x_2, \mathrm{b}), ..., (x_n, \mathrm{a})\}$

- The set $\{p\theta_{\mathrm{a}}, p\theta_{\mathrm{b}}, p\sigma_1, ..., p\sigma_n\}$ is a characteristic set of $L(\pi)$.

Anti-Unifcation of Strings

- For a set C of stings of same length

$$
\begin{aligned}
s_1 &= c_{11}\, c_{12} \ldots c_{1i} \ldots c_{1k} \\
s_2 &= c_{21}\, c_{22} \ldots c_{2i} \ldots c_{2k} \\
&\ldots \\
s_n &= c_{n1}\, c_{n2} \ldots c_{nj} \ldots c_{nk}
\end{aligned}
$$

the anti-unification of C is a pattern

$$\pi = \gamma(c_{11}c_{21}\ldots c_{n1})\gamma(c_{12}c_{22}\ldots c_{n2}) \ldots \gamma(c_{1k}c_{2k}\ldots c_{nk})$$

where

$$\gamma(c_1c_2\ldots c_n) = \begin{cases} c & \text{if } c_1 = c_2 = \ldots = c_n = c \\ x_{\iota(c1c2\ldots cn)} & \text{otherwise.} \end{cases}$$

and $\iota(c_1c_2\ldots c_n)$ is the "index" of $c_1c_2\ldots c_n$.

Analysis of Patterns (3)

Lemma 1 For every string s, there are only finite number of pattern languages containing s.

Proof. If $s \in L(\pi)$, then $|s| \geq |\pi|$.

Example The languages containing s = aab are

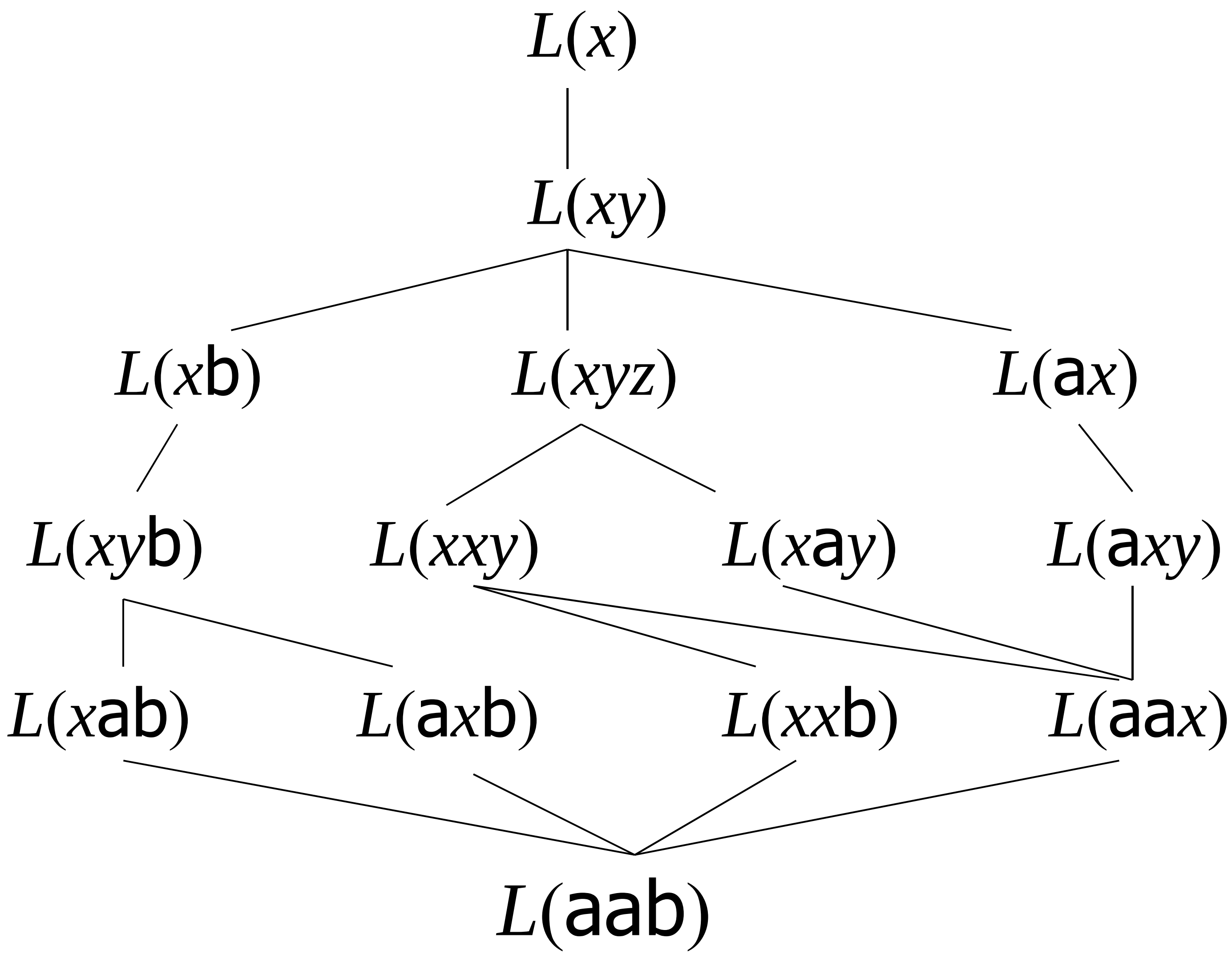
L(aab),

$L(x$ab), $L($axb), L(aa$x$), $L(xx$b), $L(x$b), L(ax), $L(x)$,

$L(xy$b), $L(x$a$y)$, L(axy), $L(xxy)$, L(xy),

$L(xyz)$,

Hasse Diagram

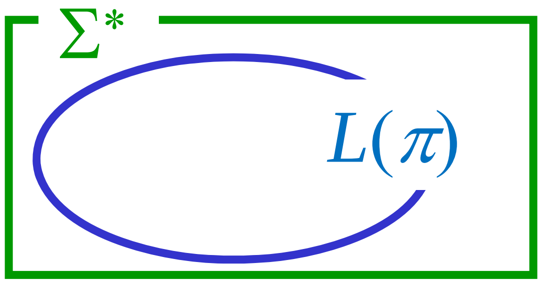


General Theory of Learning from Positive Data with Characteristic Sets

Positive presentations

$e_1, e_2, e_3, \ldots$

$\pi_1, \pi_2, \pi_3, \ldots$



- A presentation of $L(\pi)$ is a infinite sequence consisting of positive and negative example.
- A presentation σ is positive if σ consists only of positive example $< s, +>$ and any positive example occurs at least once in σ.

Identification of patterns

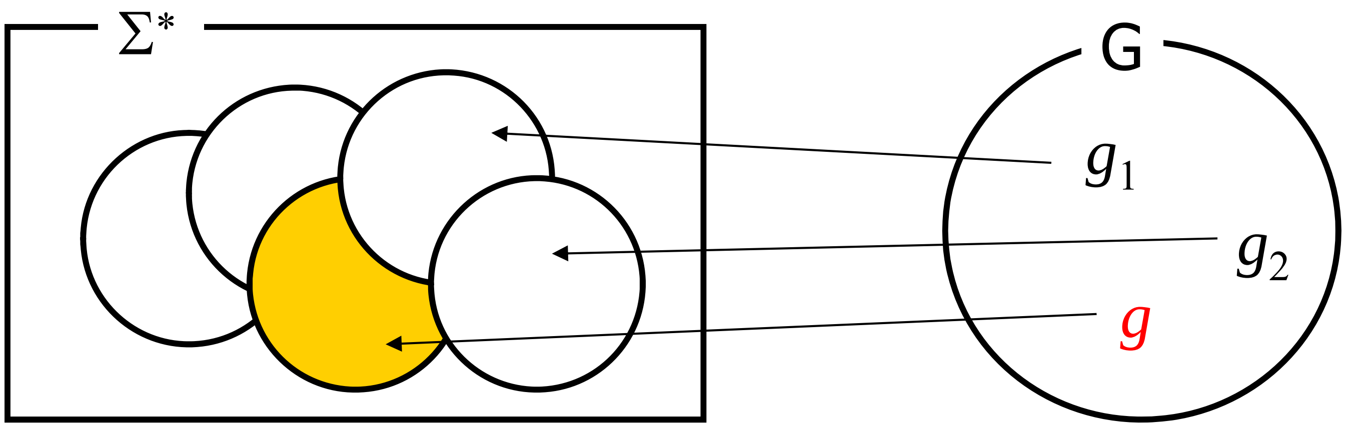
Theorem The revised algorithm of *Learn-pattern* with the minimal language strategy EX-identifies the class of all pattern languages in the limit from positive presentations.

- The minimal language strategy means that when revising conjecture π a pattern generating a minimal language for positive data is chosen as the "appropriate" pattern.

A General Framework of Learning

- A class of formal languages L(G) indexed with G
- G: A set of expressions such that each expression in G represents one language in L(G), and every language in L(G) is represented by at least one expression in G.
  - We assume that There is an algorithm which determines whether or not $w \in L(g)$ for every string $w \in \Sigma^*$ and g.

Examples of G : a set of finite state automata, a set of CFGs, a set of patterns,…



Identification in the limit [Gold]

 $s_1, s_2, s_3, \ldots$ →  $g_1, g_2, g_3, \ldots$

- A learning algorithm A EX-identifies $L(g)$ in the limit from positive presentations if
  for any positive presentation $\sigma = s_1, s_2, s_3, \ldots$ of $L(g)$ and the output sequence $g_1, g_2, g_3, \ldots$ of A, there exists N such that for all $n > N$ $g_n = g'$ and $L(g') = L(g)$
- A learning algorithm A BC-identifies $L(g)$ in the limit from positive presentations if
  for any positive presentation $\sigma = s_1, s_2, s_3, \ldots$ of $L(g)$ and the output sequence $g_1, g_2, g_3, \ldots$ of A, there exists N such that for all $n > N$ ~~$g_n = g'$ and~~ $L(g_n) = L(g)$

GCD and Learning

A class of languages in $\mathbf{N}$:

$$\mathsf{L}(\mathbf{N}) = \{L(m) \mid m \in \mathbf{N}\}$$

$$L(m) = \{0\underbrace{1\ldots1}_{n}0 \mid n \bmod m = 0\}$$

$$L(m) = \{n \in \mathbf{N} \mid n \bmod m = 0\}$$

A class of languages in $\mathbf{Z}$:

$$\mathsf{L}(\mathbf{N}) = \{L(m) \mid m \in \mathbf{N}\}$$

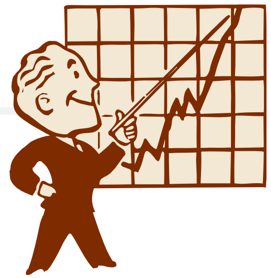
$$L(m) = \{\underbrace{1\ldots1}_{n} \mid n \bmod m = 0\} \cup \{0\underbrace{1\ldots1}_{n} \mid n \bmod m = 0\}$$

$$L(m) = \{n \in \mathbf{Z} \mid |n| \bmod m = 0\}$$

GCD and Learning

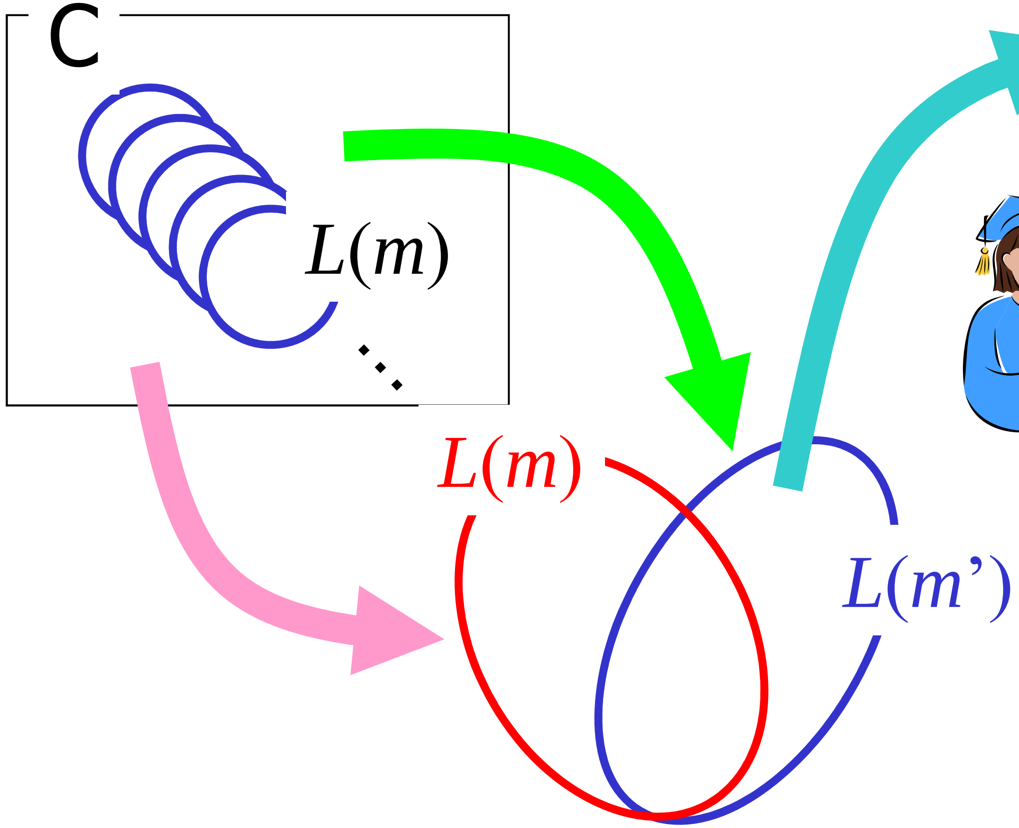
- L($\mathbf{N}$) = $\{L(m) \mid m \in \mathbf{N}\}$

  $L(m) = \{01\ldots10 \mid n \bmod m = 0\}$



Positive presentation

72, 48, 60, …,12,…



Compute the GCD of $s_1, s_2, \ldots, s_k$ with Euclidean Algorithm

Conjecture

72, 24, 12,…,12,…

Proving that L($\mathbf{N}$) is identifiable

- For every $n \in \mathbf{N}$, the characteristic set of $L(m)$ in L($\mathbf{N}$) is $\{ m \}$, that is, $\{ m \} \subseteq L(m')$ implies $L(m) \subseteq L(m')$.

- To see this, assume that $\{ m \} \subseteq L(m')$.
  This is equivalent to $m \in L(m')$ and from the definition of $L(m')$, $m = k' m'$ for some $k' \in \mathbf{N}$ ($\mathbf{Z}$).
  - $L(m) = \{n \in \mathbf{N} \mid n \bmod m = 0\}$ ($\{n \in \mathbf{Z} \mid |n| \bmod m = 0\}$).

  Let n be any element in $L(m)$. Then, from the definition, there exists $k \in \mathbf{N}$ ($\mathbf{Z}$) such that $n = k\, m$. For the k' and k, it holds that $n = k\, k'\, m'$. This means $n \in L(m')$, and therefore $L(m) \subseteq L(m')$.

C2: The Characteristic Set Property

- A subset $C(g)$ of a language of $L(g)$ is a characteristic set of $L(g)$ in $\mathsf{L}(\mathsf{G})$ if

  (1) $C(g)$ is a finite set and

  (2) for every $L(g') \in \mathsf{L}(\mathsf{G})$ $C(g) \subseteq L(g')$ implies

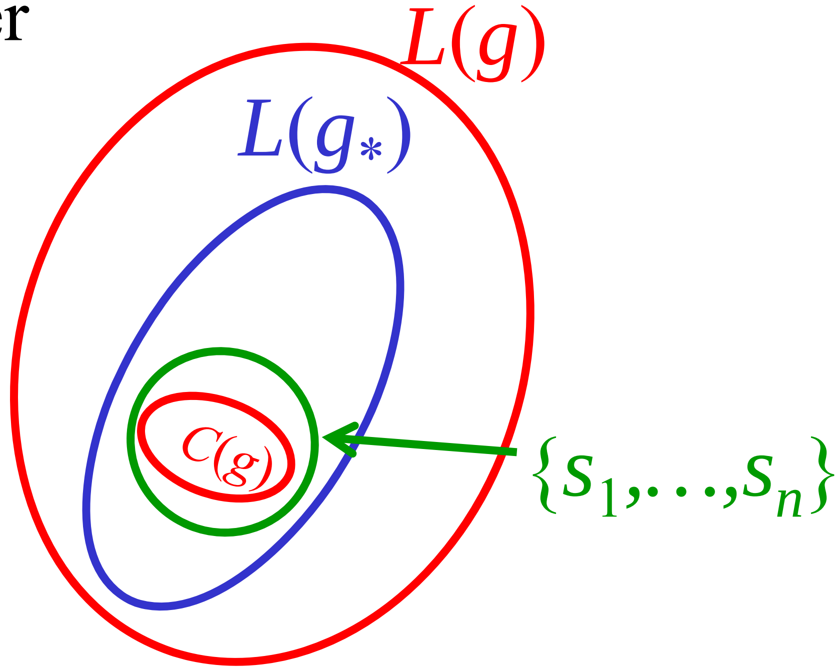
  $L(g) \subseteq L(g')$

Theorem [Kobayashi] A class $\mathsf{L}(\mathsf{G})$ of languages is identifiable in the limit from positive presentation if every language $L(g)$ in $\mathsf{L}(\mathsf{G})$ has a characteristic set $C(g)$ in $\mathsf{L}(\mathsf{G})$.

Which grammar should be chosen?

- Choose g such that $C(g) \subseteq \{s_1, \ldots, s_n\}$
  - The examples are from $L(g_*)$, that is, $\{s_1, \ldots, s_n\} \subseteq L(g_*)$. and therefore $C(g) \subseteq L(g_*)$. From the definition of characteristic sets, this implies $L(g) \subseteq L(g_*)$.

    So over generalization never happens.



EC1: The Finite Tell-tale Property

- A subset $T(g)$ of a language of $L(g)$ is a finite tell-tale of $L(g)$ in **L(G)** if

  (1) $T(g)$ is a finite set and

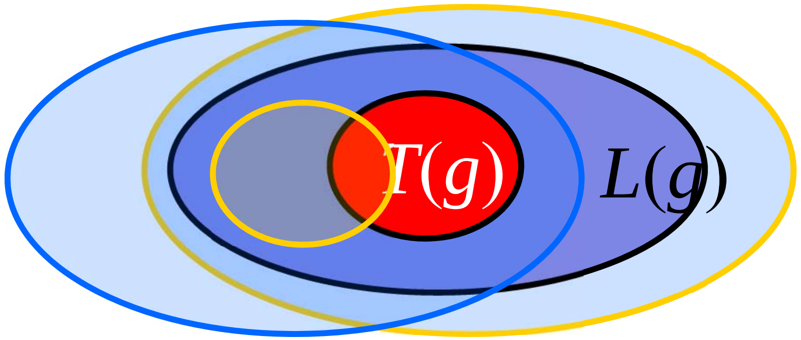
  (2) $T(g) \subseteq L(g') \subsetneq L(g)$ for no $L(g') \in$ **L(G)** other than $L(g)$

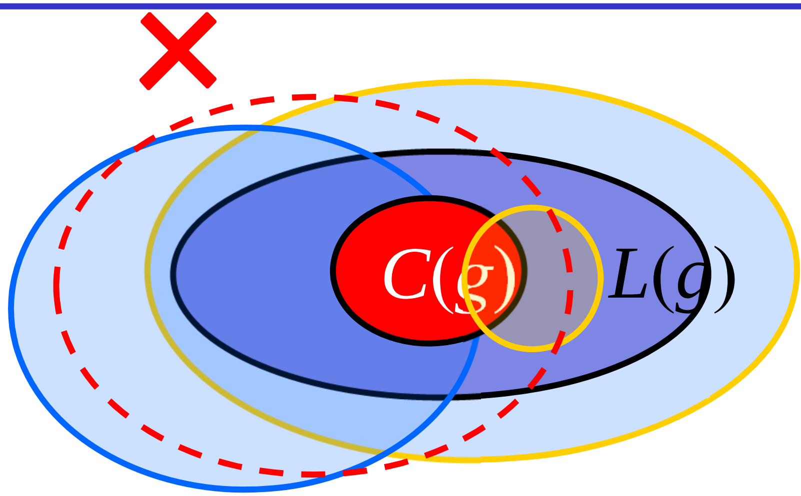
Theorem [Angluin] A class **L(G)** of languages is identifiable in the limit from positive presentation if and only if every language $L(g)$ in **L(G)** has a finite tell-tail $T(g)$ in **L(G)** and there is a procedure which generates elements of $T(g)$ when the grammar g is given as an input.

Tell-tales and Characteristic Sets

Finite Tell-tale $T(g)$ of $L(g)$:

- $T(g) \subseteq L(g)$ (T is a finite set)
- For no $L(g') \in \mathsf{L(G)}$ other than $L(g')$, $T(g) \subseteq L(g') \subsetneq L(g)$





Characteristic set $C(g)$ of $L(g)$:

- $T(g) \subseteq L(g)$ (T is a finite set)
- For every $L(g') \in \mathsf{L(G)}$

$C(g) \subseteq L(g')$ implies $L(g) \subseteq L(g')$

C3:Finite Elasticity

- A class L(G) of languages has the infinite elasticity if there is an infinite sequence of strings $w_0, w_1, w_2, \ldots,$ and an infinite sequence languages in L(G) $L(g_0), L(g_1),$ $L(g_2)$ such that

  $\{w_0, w_1, \ldots, w_{n-1}\} \subseteq L(g_n)$ and $w_n \notin L(g_n)$ for every $n \geq 1$.

  A class L(G) of languages has the finite elasticity if it does not have the infinite elasticity.

Th. [Wright] A class L(G) of languages is identifiable in the limit from positive presentation if L(G) has the finite elasticity.

C4: Finite thickness

- A class L(G) of languages has the finite thickness if for all $w \in \Sigma^*$ there are only a finite number of languages in L(G) which contain w.

Theorem [Angluin] A class L(G) of languages is identifiable in the limit from positive presentation if if L(G) of languages has the finite thickness.

L(**N**) has the Finite Thickness

- From the finite thickness condition:

L(**N**) = $\{L(m) \mid m \in \mathbf{N}\}$ has the finite thickness property.

- From the fact

$$\mathrm{GCD}(e_1, e_2, \ldots, e_k) \geq \mathrm{GCD}(e_1, e_2, \ldots, e_k, e_{k+1})$$

and the following property:

Let $a_1, a_2, \ldots, a_n, \ldots$ be a infinite sequence of natural numbers satisfying that

$$a_n \geq a_{n+1} \text{ for all } n \geq 1.$$

Then there is $N \geq 1$ such that $a_n = a_{n+1}$ for all $n \geq N$.

Relation among the conditions

U : a class of languages

- EC1 (necessary and sufficient) [Angluin]

  $\Uparrow$ $\not\Downarrow$

- C2: [Kobayashi]

  $\Uparrow$ $\not\Downarrow$

- C3: [Wright]

  $\Uparrow$ $\not\Downarrow$

- C4: [Angluin]

A Negative Result

Theorem [Gold] There is no learning algorithm which identifies any regular language from positive data.

- Note that a regular language is a formal language which is accepted by a finite state automaton. It is also represented in a regular expression.

Theorem [Gold] There is no learning algorithm which identifies any regular expression from positive data.

A Negative Result (2)

- We construct a positive presentation σ of L in the following manner.
- Let e_1 be a string in L. Since the set $\{e_1\}$ is also in C and A must identify $\{e_1\}$. So the first N_1 examples of σ are all E_1 , until "A identifies $\{e_1\}$."

$$\exists N_1 \; \forall \; n > N_1 \; h_n = g_1 \text{ and } L(g_1) = \{e_1\}$$

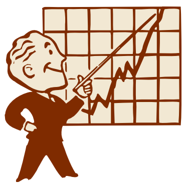
$$\underbrace{e_1, e_1, \ldots, e}_{N_1+1}{}_2, \ldots$$

$$h_1, h_2, h_3, \ldots, g_1, g_1, \ldots$$

A Negative Result (3)

- Let the (N_1+1)-th example be e_2 which is different from e_1 .
- Since C contains $\{e_1, e_2\}$, the learning algorithm A identifies $\{e_1, e_2\}$ in the limit.

$$\exists N_1 \ \forall \ n > N_2 > N_1 \ g_n = g_2 \text{ and } \{e_1, e_2\}$$



$e_1, e_1, \ldots \quad e_2, \ldots, \quad e_3, \ldots$

$\underbrace{e_1, e_1, \ldots}_{N_1+1}$

$\underbrace{e_1, e_1, \ldots \ e_2, \ldots,}_{N_2+1}$

$h_1, h_2, \ldots, g_1, \ldots, g_2, \ldots,$

A Negative Result (4)

- Let the (N_2+1)-th example be e_3 which is different from both of e_1 or e_2.
- Since C contains $\{e_1, e_2, e_3\}$, A identifies $\{e_1, e_2, e_3\}$ in the limit.

$$\exists N_3\ \forall n > N_3 > N_2 > N_1\ h_n = g_3 \text{ and } L(g_3)=\{E_1, E_2, E_3\}$$

- The language $L = \{e_1, e_2, e_3, e_4, \ldots\}$ is a infinite and A cannot identify L.