Polynomial Time Probabilistic Learning of a Subclass of Linear Languages with Queries

Yasuhiro TAJIMA, Yoshiyuki KOTANI
Tokyo Univ. of Agri. & Tech.

This talk…

- Probabilistic learning algorithm of a subclass of linear languages with membership queries

- learning via queries + special examples → Probabilistic learning

Use translation algorithms

representative sample → random examples

equivalence query → random examples

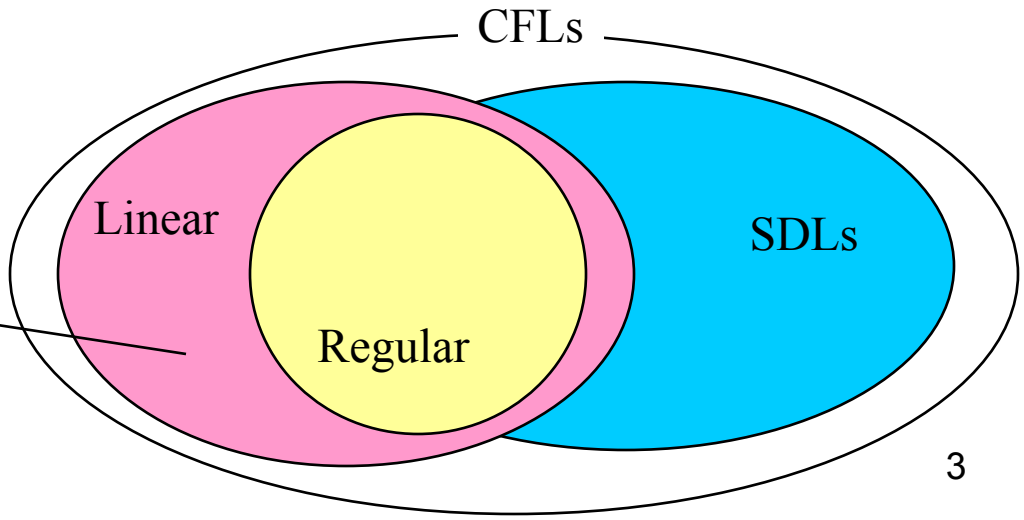
Motivations

A simple deterministic grammar (SDG) has at most one rule $A \to a\beta$ for every pair of $\begin{pmatrix} A \in N \\ a \in \Sigma \end{pmatrix}$

⇒ learning algorithm for SDG from (Tajima et al. 2004)

- membership queries
- representative sample

⇒ for linear languages



Linear grammar

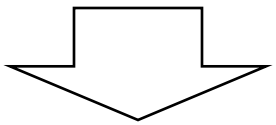
A context-free grammar is a linear grammar
if every rule is of the form

$A \to aBc$ $\quad$ $A \to aB$

$A \to a$ $\quad$ $A \to Bc$

A, B : nonterminal

a, c : terminal



Any linear grammar can be written in RL-linear s.t.
if every rule is of the form

$A \to aB$

$A \to Bc$

$A \to a$

and $\begin{cases} A \to aB, \quad A \to aC \Rightarrow B = C \\ A \to Ba, \quad A \to Ca \Rightarrow B = C \end{cases}$

Strict-deterministic linear grammar

An RL-linear is a Strict-det linear
if, for any pair of rules $A \to u,\ A \to v \quad (|u|, |v| \geq 2)$

$u = aB, \quad v = cD \quad$ or $\quad u = Ba, \quad v = Dc$ for some a,B,c,D

$\Rightarrow$ A has only left linear rules (or right linear rules)

Ex) $L = \{a^i b^i \cup a^i c^i\}$

$S \to aA,$

$A \to Bb,\ A \to Cc,\ A \to b,\ A \to c,$

$B \to aD,\ D \to Bb,\ D \to b,$

$C \to aE,\ E \to Cc,\ E \to c$

Deterministic linear grammar

A linear grammar is deterministic linear (DL)

if every rule is of the form

$$A \to aBu \quad \text{or} \quad A \to \varepsilon$$

and

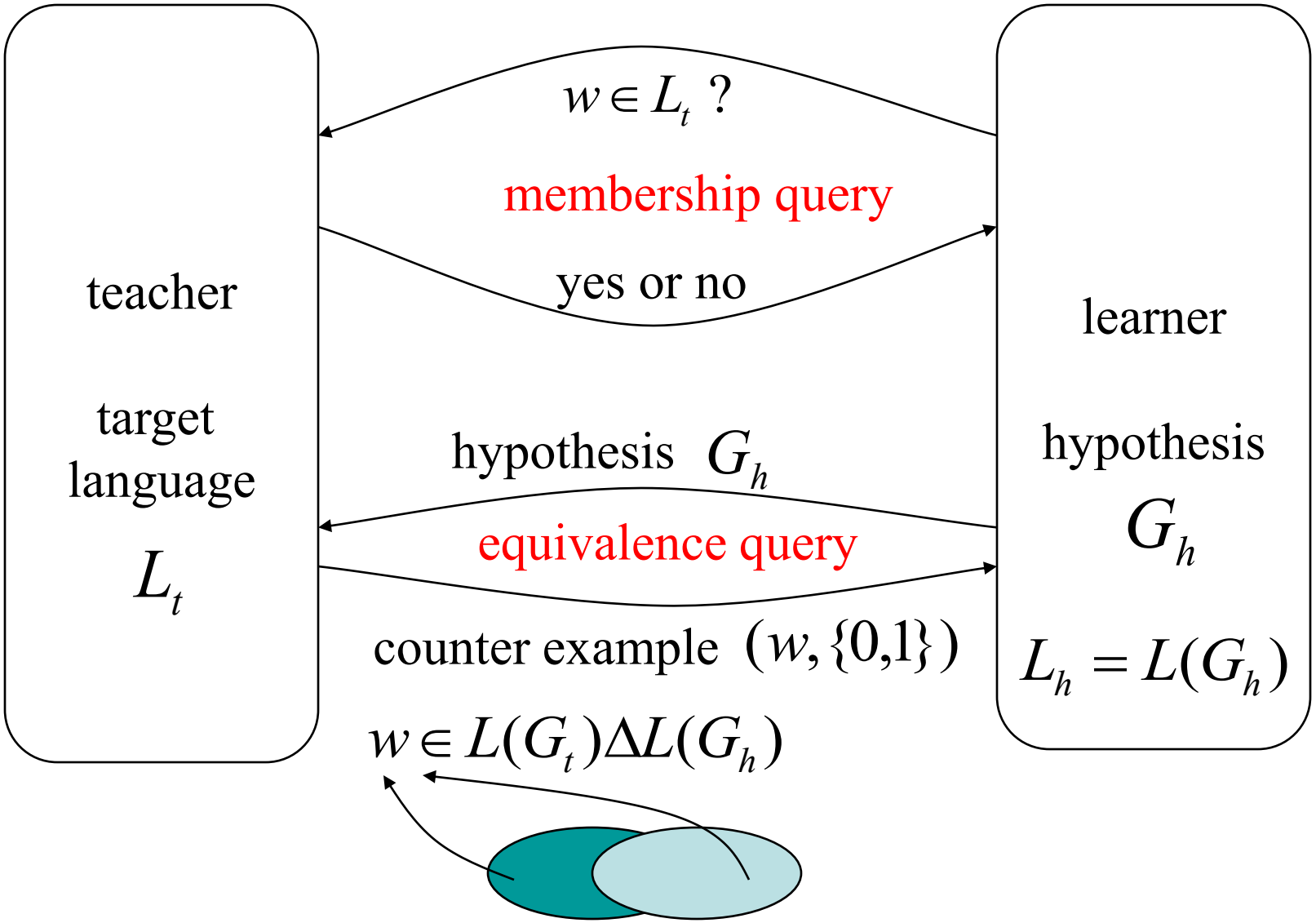
$$A \to aBu, \quad A \to aCv \quad \Rightarrow \quad B = C, u = v$$

$$(u, v \in \Sigma^*, A, B \in N, a \in \Sigma)$$

Theorem(de la Higuera, Oncina 2002) :
DL : identifiable in the limit from polynomial time and data

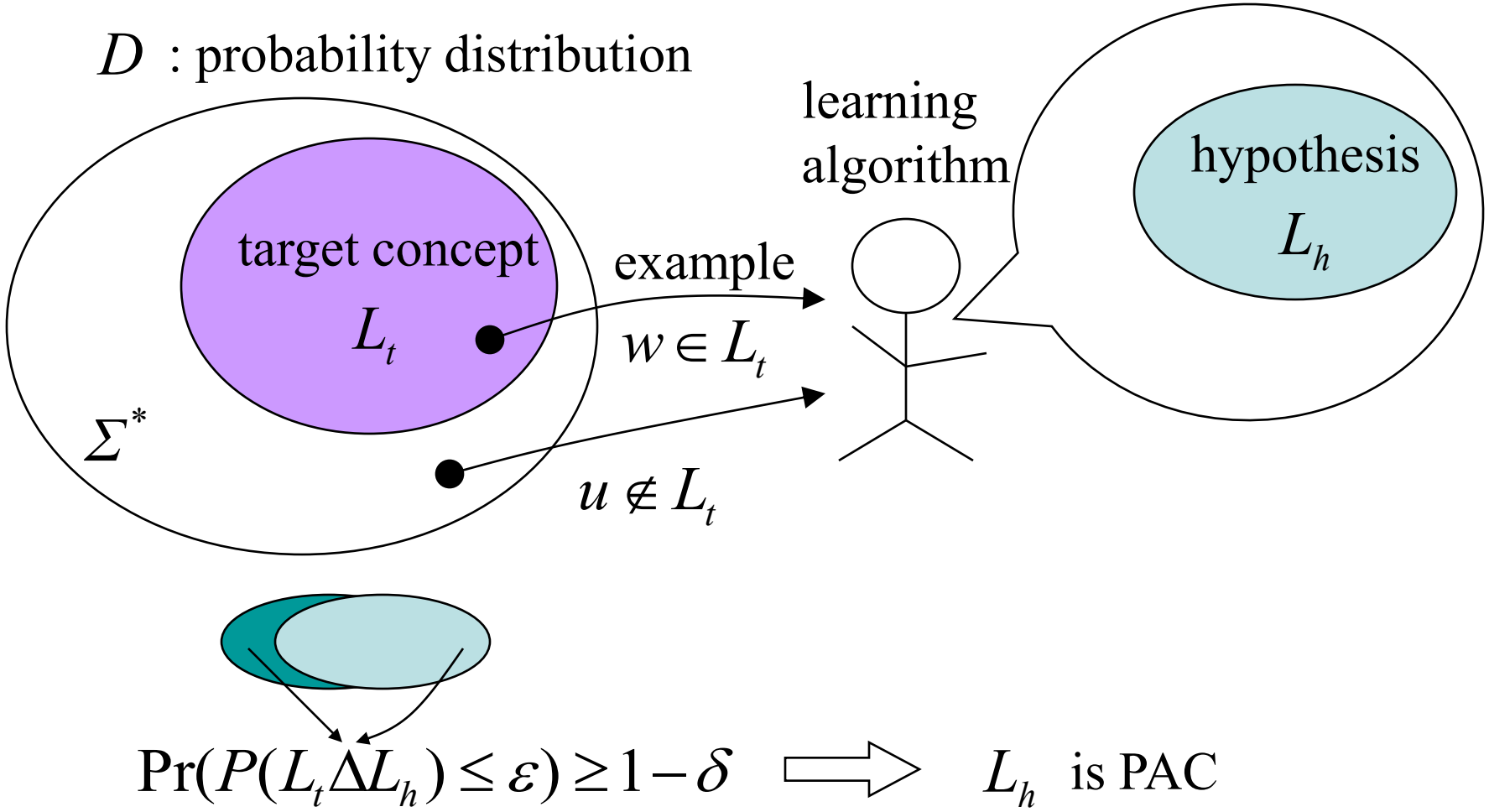
Theorem : $DL \subset \text{strict} - \det$

MAT learning (Angluin1987)



PAC learning (Valiant 1984)

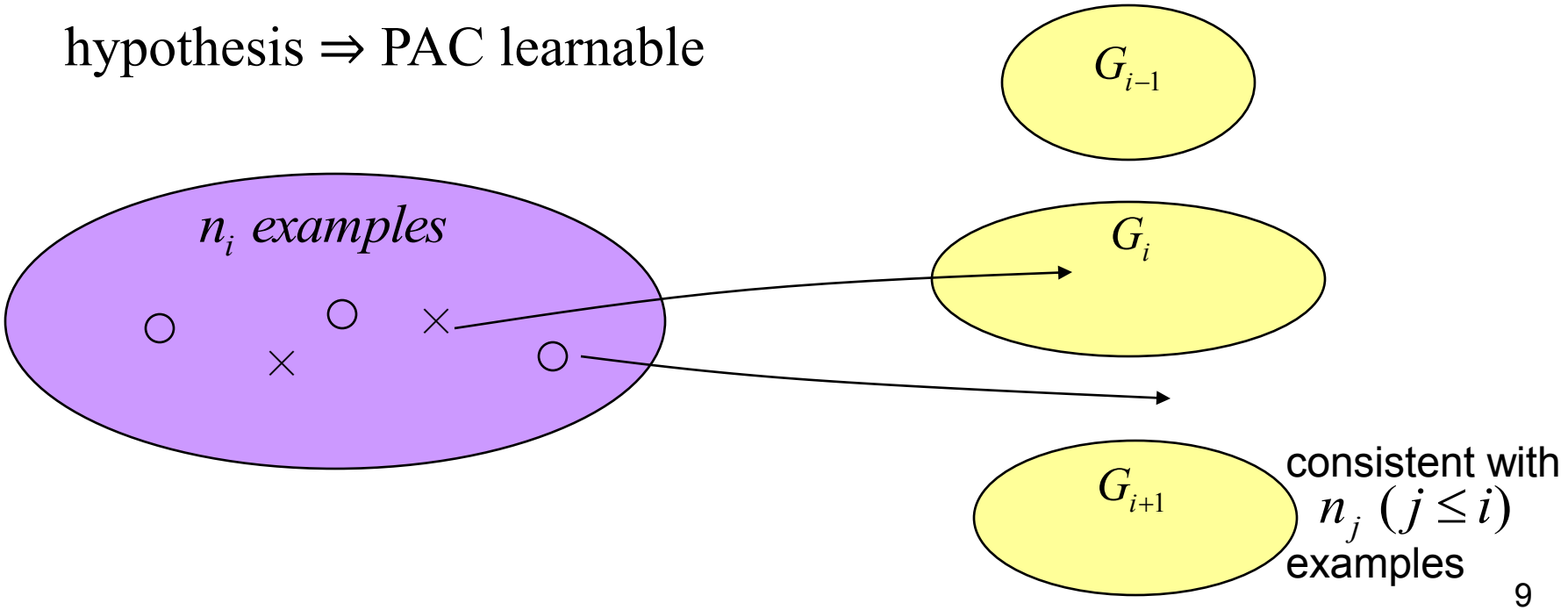
PAC : Probabilistic Approximate Correct



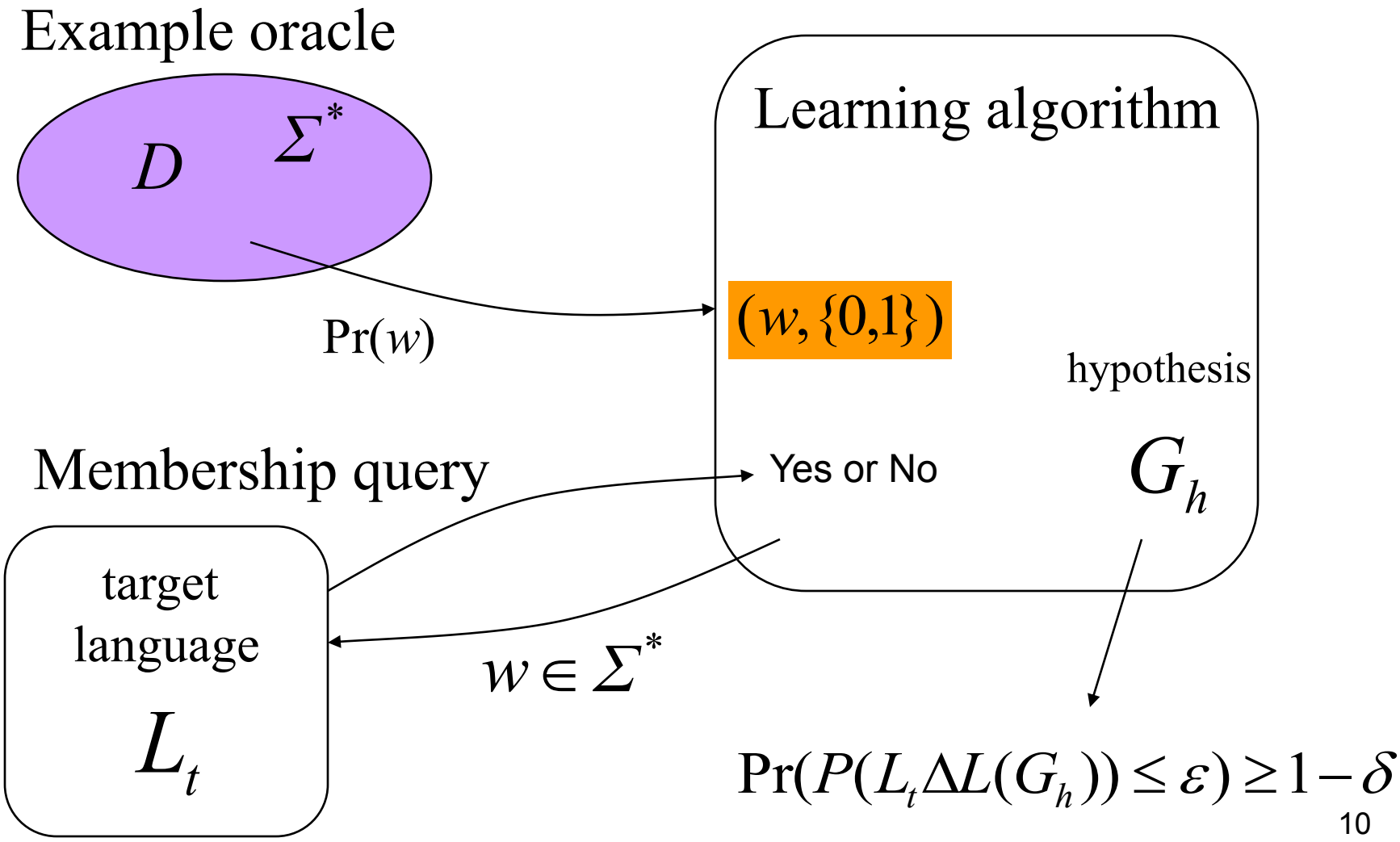
Equivalence query ⇒ PAC learning algorithm (Angluin[1987])

If a hypothesis is consistent with $n_i \geq \frac{1}{\varepsilon}\left(\ln\left(\frac{1}{\delta}\right)+(\ln 2)(i+1)\right)$ *then* PAC examples

If there is a consistent hypothesis ⇒ PAC learnable



Probabilistic learning with queries



Representative sample for a Strict-det

$G = (N, \Sigma, P, S)$: Strict-det

$Q \subseteq L(G)$: representative sample (RS) $\Leftrightarrow$

$\forall (A \to \beta) \in P, \quad \exists w \in Q \quad s.t.$

$$S \overset{*}{\Rightarrow} xA\gamma \Rightarrow x\beta\gamma \overset{*}{\Rightarrow} w$$

for some $x \in \Sigma^*, \gamma \in N^*$

All rules are used to generate Q

Example :

$$G = (N = \{S, A, B, C, D, E\}, \Sigma = \{a, b, c\}, P, S)$$

$$P = \{ S \to aA,$$

$$A \to Bb, \quad A \to Cc, \quad A \to b, \quad A \to c,$$

$$B \to aD, \quad D \to Bb, \quad D \to b,$$

$$C \to aE, \quad E \to Cc, \quad E \to c$$

then

$$Q = \{ ab , ac , aaabbb , aaaccc \}$$

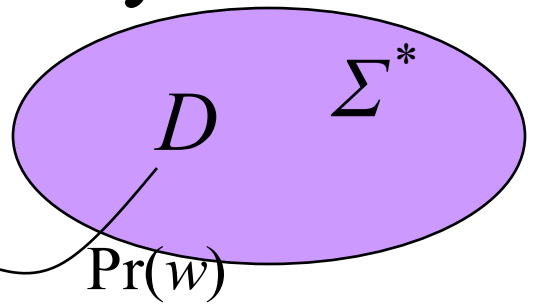
is a representative sample (RS)

Rule occurring probability

G_t : a target grammar

D : a probability distribution on Σ^*

for an example $(w, \{0,1\})$



ε : error parameter

δ : confidential parameter

$|P_t|$: the size of target grammar's rules

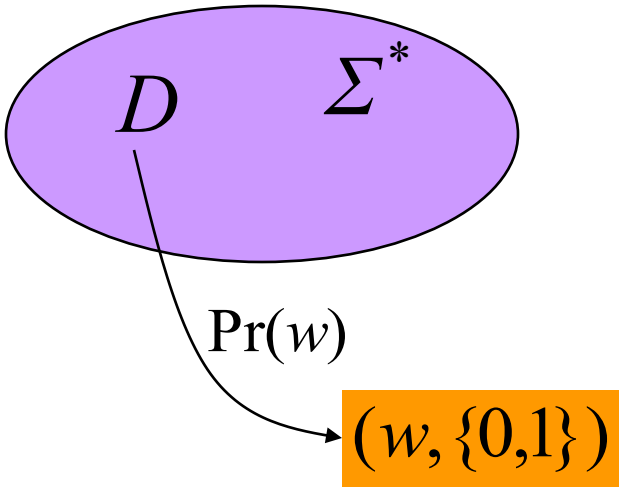
For every rule $A \to \beta$, define

$$Z(A \to \beta) = \{w \in \Sigma^* \mid S \overset{*}{\underset{G_t}{\Rightarrow}} \alpha_1 A \alpha_2 \Rightarrow \alpha_1 \beta \alpha_2 \overset{*}{\underset{G_t}{\Rightarrow}} w,$$

$$for\ some\ \alpha_1, \alpha_2 \in (N \cup \Sigma)^*\}$$

$$\Pr(A \to \beta) = \sum_{w \in Z(A \to \beta)} \Pr(w)$$

is a rule occurring probability s.t. $A \to \beta$ appears in the derivation of an example



$\Pr(A \to \beta)$

is an probability that

- $w \in L(G_t)$
- and $A \to \beta$

is used in the derivation

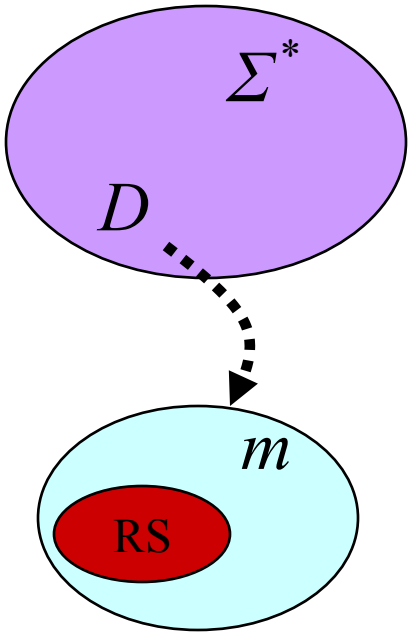
$$S \overset{*}{\underset{G_t}{\Rightarrow}} w$$

Let $d = \min\{\Pr(A \to \beta) \mid A \to \beta \in P_t\}$

Suppose

$$m > \frac{1}{d}\log\left(\frac{|P_t|}{\delta}\right)$$

The set of m-examples contains
a set of RS with the probability $1-\delta$



Proof:

"Any rule doesn't appear in derivations of m-examples" occurs

$$|P_t|(1-d)^m \le |P_t|e^{-dm} < \delta$$

We can conclude that

1. Equivalence query can be replaced by

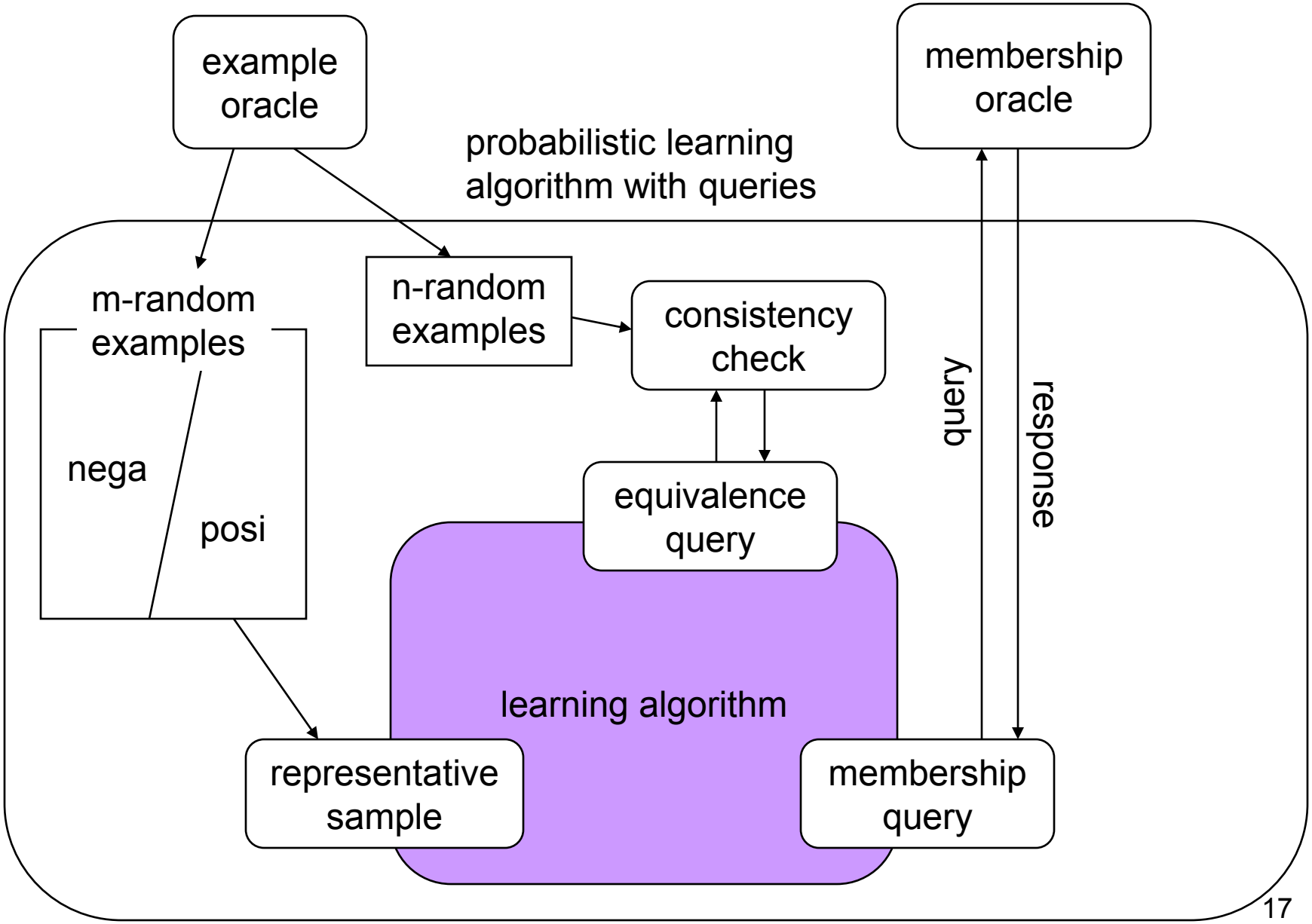
$$n_i \geq \frac{1}{\varepsilon}\left(\ln\left(\frac{1}{\delta}\right)+(\ln 2)(i+1)\right)$$

random examples

2. Representative sample can be replaced by

$$m > \frac{1}{d}\log\left(\frac{|P_t|}{\delta}\right)$$

random examples


example oracle
membership oracle
probabilistic learning algorithm with queries
m-random examples
nega
posi
n-random examples
consistency check
equivalence query
query
response
learning algorithm
representative sample
membership query

Learning algorithm via queries and RS

$M_h = \{(u, v, w) \mid uvw \in RS\}$

$T = \Sigma$

while (finish == 0) begin

make nonterminals N_h from T, M_h

make rules P_h and hypothesis G_h

if (equivalence query for G_h responds "yes")

output G_h , finish = 1

else

update T by the counterexample w

end

Making nonterminals

$$(u,v,w) \overset{T}{=} (x,y,z) \Leftrightarrow MEM(uvw) = MEM(xyz)$$

$$M_h = \{(u,v,w) \mid uvw \in RS\}$$

then

$$N_h = (M_h / \overset{T}{=})$$

$A(u,v,w,\overset{T}{=})$: a nonterminal

= an equivalence class contains (u,v,w)

Making rules

Make all rules as follows except for not consistent with query results

$$P_{CFG} = \{\ A(u, avb, w, \overset{T}{=}) \rightarrow aA(u, vb, w, \overset{T}{=}),$$

$$A(u, avb, w, \overset{T}{=}) \rightarrow A(u, av, w, \overset{T}{=})b,$$

$$A(u, a, w, \overset{T}{=}) \rightarrow a \quad \}$$

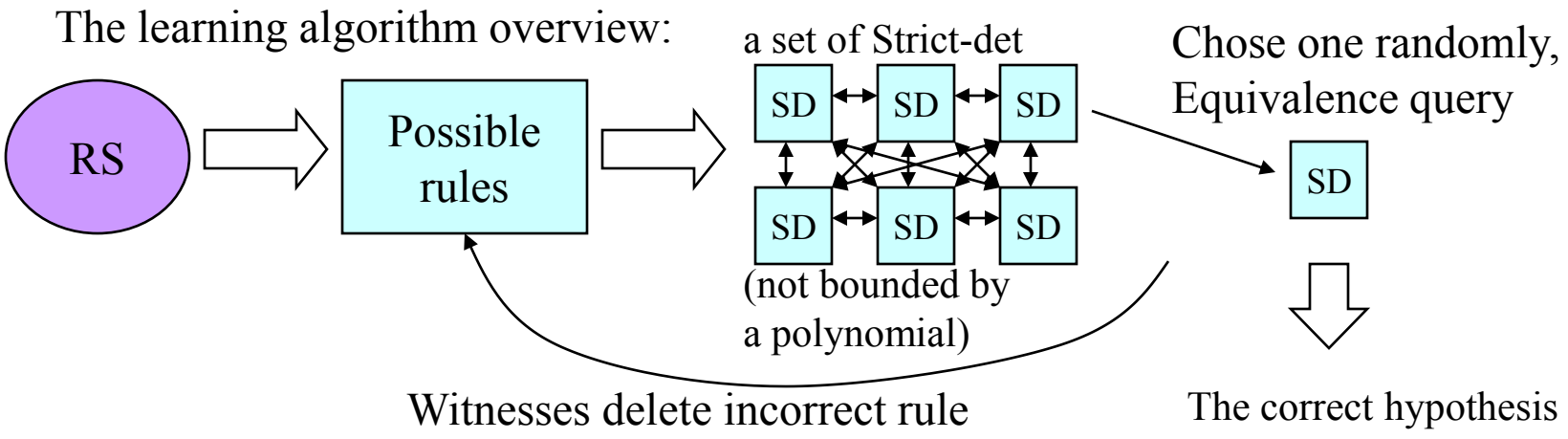
Select a hypothesis randomly $P_h \subseteq P_{CFG}$

$$G_h = (N_h, \Sigma, P_h, A(\varepsilon, w, \varepsilon, \overset{T}{=}))$$

Exact learning of strict-det

- Strict-det is polynomial time exact learnable via
  - membership queries, and
  - a representative samples (RS)

c.f. [Angluin(1980)] for regular sets



Conclusions

- Strict-det linear language can be probabilistic learnable with queries in polynomial time

Future works

- Identification from polynomial time and data (teachability)
- RS → Correction queries

Theorem

Strict-det linear languages are

polynomial time probabilistic learnable with membership queries

Simple Deterministic Languages

- Context-free grammar(CFG) $G = (N, \Sigma, P, S)$
  in 2-standard *Greibach normal form* is
  Simple Deterministic Grammar (SDG) iff
  $A \to a\beta \qquad (\beta \in N^*, |\beta| \leq 2)$
  is unique for every $A \in N$and $a \in \Sigma$

- Simple Deterministic Language (SDL) is the generated language by a SDG

Representative sample for an SDG

$G = (N, \Sigma, P, S)$: SDG

$Q \subseteq L(G)$: representative sample (RS) $\Leftrightarrow$

$\forall (A \to a\beta) \in P, \quad \exists w \in Q \quad s.t.$

$$S \overset{*}{\Rightarrow} xA\gamma \Rightarrow xa\beta\gamma \overset{*}{\Rightarrow} w$$

for some $x \in \Sigma^*, \gamma \in N^*$

All rules are used to generate Q

Example :

$$G = (N = \{S, A, B, C\}, \Sigma = \{a, b, c\}, P, S)$$

$$P = \{ S \to aA,\ S \to cC,$$

$$A \to aAB,\ A \to b,$$

$$B \to b,\ C \to cCB,\ C \to b \quad \}$$

then

$$Q = \{ ab\ ,\ aabb\ ,\ ccbb\ \}$$

is a representative sample (RS)

PAC learning

(Valiant1984)

Target language : L_t

Hypothesis language : $L(G_h)$

Probability distribution : P on Σ^*

A PAC learning algorithm outputs G_h such that

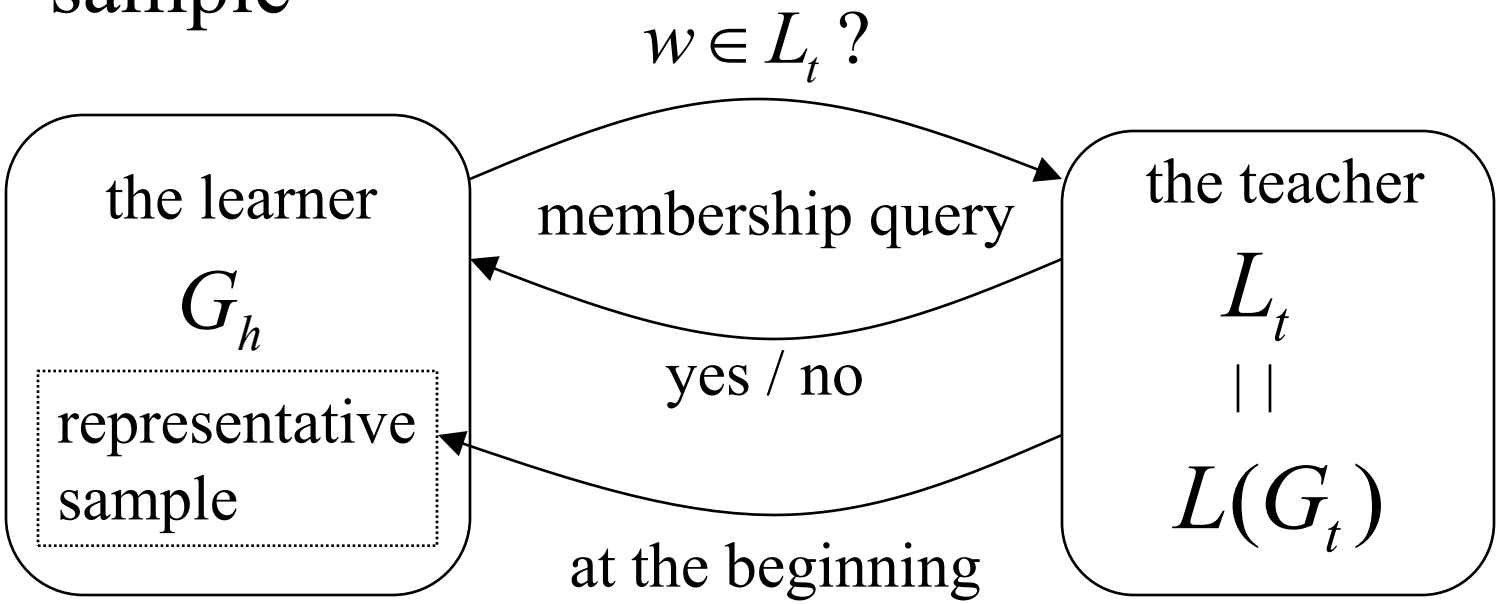
$$\Pr(P(L_t \Delta L(G_h)) \leq \varepsilon) \geq 1 - \delta$$

where $P(L_t \Delta L(G_h)) = \sum_{w \in L_t \Delta L(G_t)} P(w)$

Query learning of SDLs

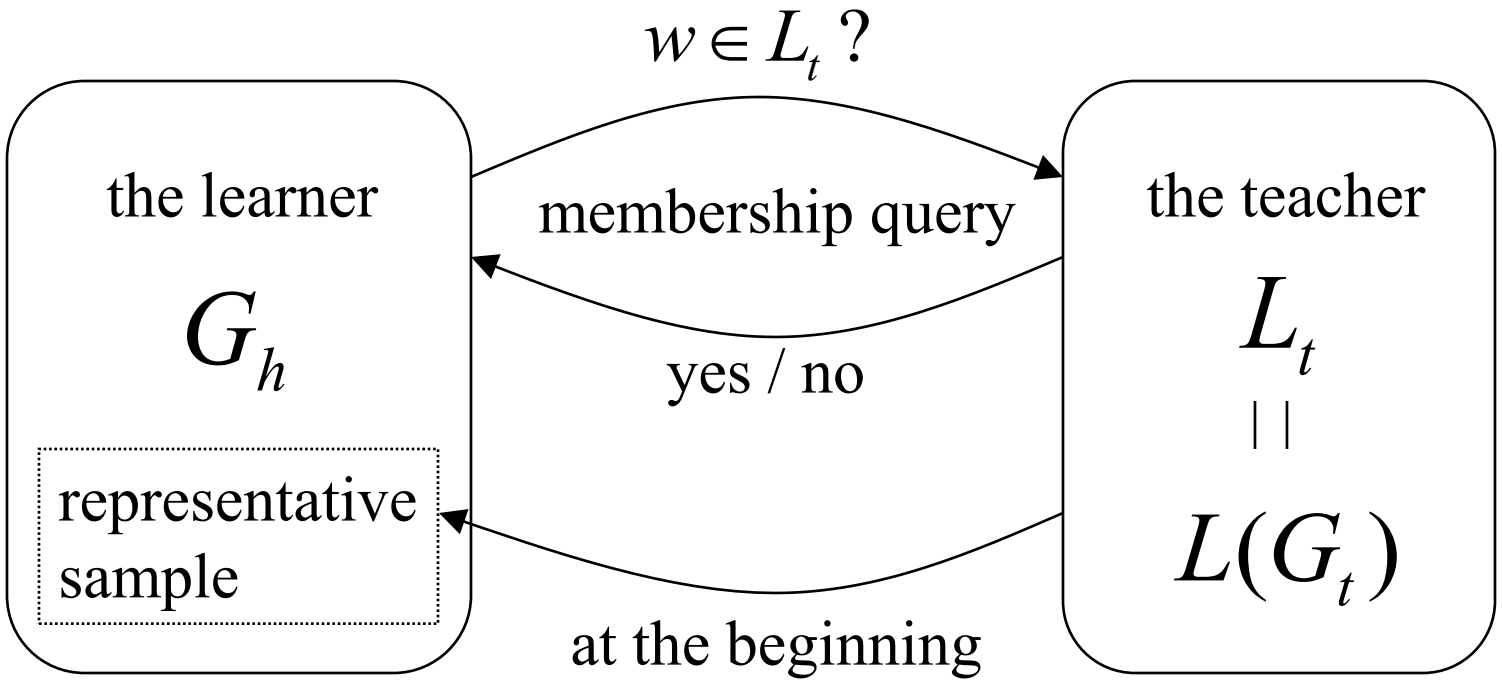
(Tajima2000)

- SDLs are polynomial time learnable via membership queries and a representative sample



representative sample : a special finite subset of L_t

Learning model



representative sample : a special finite subset of L_t