On Computability and Learnability of the Pumping Lemma Function

Dariusz Kalociński

University of Warsaw, Poland

March 11, 2014
8^{th} International Conference, LATA 2014

Structure

- what is the pumping lemma function?
- how complex is it?
  - computable?
  - learnable?
- exact placement of the function in the arithmetical hierarchy

Structure

- what is the pumping lemma function?
- how complex is it?
  - computable?
  - learnable?
- exact placement of the function in the arithmetical hierarchy
- on the way: we get a ,,natural" Π^0_2-complete problem
- final remarks

Pumping Lemma (for Regular Languages)

For regular $L \subseteq \Sigma^*$ $(\exists\, c > 0\,)$

Pumping Lemma (for Regular Languages)

For regular $L \subseteq \Sigma^*$ $(\exists\, c > 0\,)$

$(\forall\, \omega \in L, |\omega| \geq c)\,(\exists\, \alpha\beta\gamma\,)$:

- $\alpha\beta\gamma = \omega$
- $|\alpha\beta| \leq c$
- $\beta \neq \varepsilon$
- $(\forall\, i \in \mathbb{N})\, \alpha\beta^i\gamma \in L$

Pumping Lemma (for Regular Languages)

For regular $L \subseteq \Sigma^*$ $(\exists\, c > 0\,)$

$(\forall\, \omega \in L, |\omega| \geq c)\,(\exists\, \alpha\beta\gamma\,)$:

- $\alpha\beta\gamma = \omega$
- $|\alpha\beta| \leq c$
- $\beta \neq \varepsilon$
- $(\forall\, i \in \mathbb{N})\, \alpha\beta^i\gamma \in L$

- $\phi(L, c)$ - formula in yellow box
- $\phi(L, c)$ means: for given L, c is the witness for $\exists\, c$
- c satisfying $\phi(L, c)$ is called a pumping constant for L

Problem

Input: arbitrary L

Output: the least pumping constant for L (if exists)

- we focus on r.e. languages
- W_e = the domain of the e^{th} algorithm
- L is r.e. $\Leftrightarrow \exists\, e\, (L = W_e)$
- $R(e, c) \Leftrightarrow_{df}$ c is a pumping constant for W_e

Pumping Lemma Function

$$f(e) = \begin{cases} \text{the least } c \text{ st. } R(e, c) & \text{if } \exists c R(e, c) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Questions

$$R(e,c) \Leftrightarrow_{df} c \text{ is a pumping constant for } W_e$$

$$f(e) = \begin{cases} \text{the least } c \text{ st. } R(e,c) & \text{if } \exists c R(e,c) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$Graph(f)$ = the graph of $f = \{(x,y) : f(x) = y\}$

How complex are f and R?

- is f computable?
- is $\overline{Graph(f)}$ r.e.?
- is f algorithmically learnable?
  - if not, how strong oracle we need to make f learnable?
- how exactly does $Graph(f)$ fit in arithmetical hierarchy?
- how exactly does R fit in arithmetical hierarchy?

Is f computable?

We need

- $\text{EMPTY} = \{e \in \mathbb{N} : W_e = \emptyset\}$
- EMPTY is Π^0_1-complete
- $\leq_{rec}$ - reducibility via recursive function
- $R(e, c) \Leftrightarrow_{df} c$ is a pumping constant for W_e

Lemmas

- $\text{EMPTY} \leq_{rec} R$
- If $R(e, c)$ then $(\forall\, d > c)\, R(e, d)$.

Theorem

f is not computable

Proof.

Suppose the contrary. Then R is Σ^0_1. Let $A \in \Pi^0_1$.
$A \leq_{rec} \text{EMPTY} \leq_{rec} R \in \Sigma^0_1$. Hence, $\Pi^0_1 \subseteq \Sigma^0_1$. ↯ □

Is $\overline{Graph(f)}$ r.e.?

We need

- $\text{EMPTY} = \{e \in \mathbb{N} : W_e = \emptyset\}$
- EMPTY is Π^0_1-complete
- $\leq_{rec}$ - reducibility via recursive function
- $R(e, c) \Leftrightarrow_{df}$ c is a pumping constant for W_e

Lemmas

- $\overline{Graph(f)} \in \Sigma^0_1 \Rightarrow \overline{R} \in \Sigma^0_1$
- $\overline{\text{EMPTY}} \leq_{rec} R$

Theorem

$\overline{Graph(f)}$ is not r.e.

Proof.

Suppose the contrary. By lemma $\overline{R} \in \Sigma^0_1$. Since $\overline{\text{EMPTY}} \leq_{rec} R$, then $\text{EMPTY} \leq_{rec} \overline{R}$. Hence, $\Pi^0_1 \subseteq \Sigma^0_1$. ↯ □

Learnability

Definition

$f : \mathbb{N}^k \to \mathbb{N}$ (possibly partial) is learnable if there is a total computable function $g_t(\overline{x})$ st. for all $\overline{x} \in \mathbb{N}^k$:

$$lim_{t\to\infty} g_t(\overline{x}) = f(\overline{x}) \ , \qquad (1)$$

which means that one of the two conditions hold:

- neither $f(\overline{x})$ nor $lim_{t\to\infty} g_t(\overline{x})$ exist
- both $f(\overline{x})$ and $lim_{t\to\infty} g_t(\overline{x})$ exist and are equal

Example

$f(x) = 5$

$$\begin{array}{cccccccccc} 0 & 1 & 2 & 1 & & 7 & 5 & 5 & 5 \\ \| & \| & \| & \| & & \| & \| & \| & \| \\ g_0(x) & g_1(x) & g_2(x) & g_3(x) & \dots & g_{1487}(x) & g_{1488}(x) & g_{1489}(x) & \dots \end{array}$$

Is f learnable?

We need

- $\text{TOT} = \{e : W_e = \Sigma^*\}$
- TOT is Π^0_2-complete
- Gold's lemma: f is learnable $\Leftrightarrow$ $Graph(f) \in \Sigma^0_2$
- $R(e, c) \Leftrightarrow_{df}$ c is a pumping constant for W_e

Lemma

$\text{TOT} \leq_{rec} R$

Theorem

f is not learnable

Proof.

Suppose the contrary. Then $Graph(f) \in \Sigma^0_2$. We have:
$R(x, y) \Leftrightarrow \exists c((x, c) \in Graph(f) \wedge c \leq y) \Leftrightarrow \exists(\exists\forall \ldots \wedge \ldots)$. So $R \in \Sigma^0_2$. But by lemma $\text{TOT} \leq_{rec} R$. Hence, $\text{TOT} \in \Sigma^0_2$. $\lightning$ $\square$

How complex oracle does make f learnable?

We need

- HALT = the halting problem $= \{(e, x) : x \in W_e\}$
- $\leq_{bl}$ - bounded lexicographical order on words
- Gold's lemma: f is learnable $\Leftrightarrow Graph(f) \in \Sigma^0_2$

Theorem

f is learnable in HALT*.*

Proof.

$R(e, x) \Leftrightarrow$

$$(\forall\, \omega)\,\{\overbrace{[\omega \in W_e \,\wedge\, \ldots]}^{\text{rec. in HALT}} \Rightarrow (\exists\, \alpha, \beta, \gamma \leq_{bl} \omega\,)[\overbrace{\ldots}^{\text{rec.}} \wedge (\forall\, i)\; \overbrace{\alpha\beta^i\gamma \in W_e}^{\text{rec. in HALT}})]\}$$

$R(e, x) \Leftrightarrow \forall[\ldots \Rightarrow \forall \ldots]$, so $R \in \Pi^0_1$ in HALT.

$$(e, x) \in Graph(f) \Leftrightarrow \underbrace{R(e, x)}_{\Pi^0_1 \text{ in HALT}} \wedge \underbrace{(\forall\, y < x)\, \neg R(e, y)}_{\Sigma^0_1 \text{ in HALT}}$$

Hence, $Graph(f) \in \Sigma^0_2$ in HALT and f is learnable in HALT. □

How complex is R?

We need

- HALT = the halting problem = $\{(e, x) : x \in W_e\}$
- TOT = $\{e : W_e = \mathbb{N}\}$
- TOT is Π^0_2-complete
- $R(e, c) \Leftrightarrow_{df}$ c is a pumping constant for W_e

Lemma

TOT $\leq_{rec}$ R

Theorem

R is Π^0_2-complete

Proof.

R is Π^0_2-hard, since TOT $\leq_{rec}$ R
$x \in W_e \Leftrightarrow \exists\, c\; T(e, x, c)$, T - Kleene predicate
$R(e, x) \Leftrightarrow \forall[\exists \ldots \Rightarrow \exists^{\leq_{bl}\omega}(\ldots \wedge \forall\exists \ldots)]$ Hence, $R \in \Pi^0_2$. □

f - exact place in arithmetical hierarchy

Lemmas

- $Graph(f) \in \Delta_3^0$ (see paper)
- $Graph(f) \notin \Sigma_2^0$ (proved)
- $R(e, c) \Leftrightarrow_{df} c$ is a pumping constant for W_e
- R is Π_2^0-complete (proved)

Theorem

$Graph(f) \in \Delta_3^0 - (\Sigma_2^0 \cup \Pi_2^0)$

Proof.

We show $Graph(f) \notin \Pi_2^0$. Suppose the contrary.
Now show $R \leq_T Graph(f)$. Algorithm with oracle $Graph(f)$ that computes χ_R: on input (e, x) output YES iff $(e, y) \in Graph(f)$ holds for some $y \leq x$. Hence, $\overline{R} \leq_T Graph(f)$.
Since R is Π_2^0-complete, $\overline{R}$ is Σ_2^0-complete. Let $A \in \Sigma_2^0$. We have $A \leq_T \overline{R} \leq_T Graph(f)$. Then $\Sigma_2^0 \subseteq \Pi_2^0$. $\lightning$ □

Final remarks

- what about other input representations?
  - CFGs: f learnable
  - oracle for characteristic function
    - f learnable
    - use in language identification?
  - time bounded Turing machines
    - f learnable

Thanks for your attention!