PROJECT: CONSISTENCY

ALGORITHMIC LEARNING THEORY, SUMMER 2014

1. Introduction

A learner that conjectures a grammar for a language that is not consistent with what he or she already knows (from the text) to be in the language would be a foolish learner indeed. As an extreme example, imagine a learner that has seen the following text: $1, 2, 4, 6, \ldots$. It would be foolish for him to offer as a guess, "The set of all even numbers." Learners that do not do such silly things are called *consistent learners*. In this project, you'll investigate how requiring that a learner be consistent affects the classes of languages that they can learn.

2. Computability theory: Lemmas and exercises

Lemma 1. *Let $h(j, k)$ be a total, computable function of two variables. For each $j \in \mathbb{N}$, define $f_j(k) = h(j, k)$. Then there is a computable set $S \subseteq \mathbb{N}$ such that for all j, f_j is not the charactaristic function of S.*

Exercise 1. *Let $L_i = \{\langle i, x\rangle \mid x \in W_i\}$. Find a computable function h so that for each i, $h(i)$ gives an index for L_i.*

3. Learning Theory I: Identification, lemmas and exercises

Lemma 2. *For each $i \in \mathbb{N}$, define $L_i = \{\langle i, x\rangle \mid x \in W_i\}$, and let $\mathcal{L} = \{L_i\}_{i\in\mathbb{N}}$. Then $\mathcal{L}$ is identifiable. Moreover, $\mathcal{L}$ is identifiable by a computable learner.*

4. Learning Theory II: Limitations

Definition 1. *Learning function $\varphi \in \mathcal{F}$ is called* consistent *if for all $\sigma \in SEQ$, $rng(\sigma) \subseteq W_{\varphi(\sigma)}$.*

Proposition 1. $[\mathcal{F}^{con}] = [\mathcal{F}]$.

Proposition 2. *Let φ be a consistent computable learning function. If φ identifies $\mathcal{L}$, then $\mathcal{L}$ contains only recursive languages (i.e., $\mathcal{L} \subseteq RE_{REC}$).*

Proposition 3. *There is a set of computable languages $\mathcal{L} \subseteq RE_{REC}$ that is identifiable by a recursive learner, but not by a consistent recursive learner.*