Examen de premiere session de novembre 2019
X3IT040 (Statistical and symbolic language modeling)
All documents allowed. There is an appendix!
2 hours.

1 Even linear grammars (6 points)

An even linear grammar is a grammar $G = \langle \Sigma, V, P, S \rangle$ where all rules are of the form $T \rightarrow u \in \Sigma^*$ or $T \rightarrow uT'v$ with $|u| = |v|$. A language is an *even linear language* if there exists an even linear grammar which generates it exactly. For example, the palindrome language is even linear.

We remember that $|u|$ denotes the length of u.

1. This is an even linear grammar : $T \rightarrow abTab + a$. What is the language generated? Why is it even linear?
   | ← au
2. Build a derivation for string *ababaabab*.
3. Build the derivation tree of T to *ababaabab*
4. We consider the following normal form: a grammar $G = \langle \Sigma, V, P, S \rangle$ is in *normal even linear form* if all rules are in one of the following forms
   - $T \rightarrow uT'v$ with $|u| = |v| = 1$.
   - $T \rightarrow a$ with $a \in \Sigma$
   - $T \rightarrow \epsilon$

   Give some arguments as to why the following theorem holds: Let L be an even linear language. Then there exists an even linear grammar in normal linear form which generates exactly L.
5. We decide to encode an even linear grammar in the following way: $T \rightarrow aT'b$ becomes $T \rightarrow \frac{a}{b}T'$. Notice that $\frac{a}{b}$ is now considered as a symbol. Show that this new grammar is a right regular grammar (see appendix).
6. How is string *ababaabab* now encoded with the new alphabet?
7. Does this suggest a way of using algorithm RPNI and therefore being able to identify even linear grammar in the limit?
8. The following theorem is non trivial: If L is a regular language, it is even linear. But the converse is not true. Furthermore, not all context-free languages are even linear. Can you suggest a context-free language which is not even linear?

2 A concrete example (6 points)

Consider the DFA represented in Figure 1

abaa abbbb.

1. Will RPNI find the right answer when learning from $S+ = \{a, b, bbb\}$, $S- = \{bb, bab, bbab\}$?
2. Just add one single positive example so that the algorithm returns the correct answer.
3. Just add one single negative example so that the algorithm returns the correct answer.

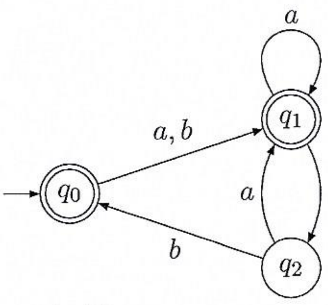

Figure 1: The target automaton

3 RPNI Identification in the limit (3 points)

RPNI identifies in the limit because of the existence of a characteristic sample. Remember that a characteristic sample is a sample of labelled strings such that, whenever this sample is included in the learning sample, the algorithm converges and identifies correctly the target. Give a short proof of this fact.

4 And now for something completely different: On learning pattern languages from queries (5 points)

Let $\Sigma = \{a, b\}$.

We consider the class of pattern languages. These are language defined by patterns that consist of strings over $\Sigma \cup X$ where X is a countably infinite set of variable symbols. A pattern thus is a non empty finite string over $\Sigma \cup X$.

Given a pattern π, $L(\pi)$=$\{w \in \Sigma^* : \exists u_1, \ldots, u_k \in \Sigma^* \wedge w = \pi[u_i/x_i]\}$. Strings are in this way obtained by substituting **non-empty** strings of constant symbols for the variable symbols in π : $\pi[u_i/x_i]\}$ is the pattern/string π in which each occurrence of variable x_i is replaced with sub-string u_i.

Example: for $\pi = aax_1x_2bx_1$ $aababb \in L(\pi)$, $aababbbbbab \in L(\pi)$ but $aababa \notin L(\pi)$ and $aaaba \notin L(\pi)$.

We add to the set of queries studied during the lectures the following restricted superset queries: a pattern π is presented to the oracle who answers YES if $L(\pi)$ is a superset of the target pattern language denoted π^* (if $L(\pi) \supseteq L(\pi^*)$) and NO if not.

a) Propose an algorithm that exactly identifies the class of languages defined by patterns of size n that uses restricted superset queries and runs in time polynomial in n.

b) What is the number of queries your algorithm has to make in order to reach identification?

5 Appendices

The palindrome language over alphabet Σ contains exactly all strings $u_0 \ldots u_n$ with $\forall i \leq n, u_i = u_{n-i}$

A *right regular grammar* (also called right linear grammar) is a formal grammar $\langle \Sigma, V, P, S \rangle$ such that all the production rules in P are of one of the following forms:

- $A \rightarrow a$, where A is a non-terminal in V and a is a terminal in Σ,
- $A \rightarrow aB$, where A and B are non-terminals in V and a is in Σ,
- $A \rightarrow \epsilon$, where A is in V and ϵ denotes the empty string, i.e. the string of length 0.