*Computational Linguistics Group*
*Department of Computer Science*
*University of Haifa*

*בלשנות חישובית*
*החוג למדעי המחשב*
*אוניברסיטת חיפה*

# Challenge 3: Context-free grammars

Context-free grammars are inherently inefficient to process: in the general case, recognition time is $O(n^3)$. For on-line applications such as speech processing this is unacceptable. In contrast, the linear recognition time of finite-state automata makes them ideal for on-line processing, but their expressivity is limited.

A possible compromise is the following: given a context-free grammar $G$, approximate it with a FSA $A$ such that $L(A) \supseteq L(G)$. At run-time, when an input string $w$ is given, it is first run through $A$. If $w \notin L(A)$ then clearly $w \notin L(G)$, and hence it is rejected. Only if $w \in L(A)$ is it submitted to $G$ for parsing. This is likely to reduce the average recognition time.

## 1

Propose an algorithm for approximating a CFG $G$ with an FSA $A$ such that:

- $L(A) \supseteq L(G)$
- $L(A)$ is as small as possible.

Ideally, your algorithm should produce an FSA $A$ such that $L(A) = L(G)$ when $L(G)$ is obviously regular, e.g., when $G$ is a right-linear grammar.

## 2

Implement the algorithm. The format of the input should be:

```
list of terminal symbols
list of non-terminal symbols (beginning with the start symbol)
list of rules, each of the format:
   LHS -> RHS1 ... RHSk
```

The format of the output should be XFST-compatible.

## 3

Test your implementation on the following grammar:

$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$
$S \rightarrow VP$
$VP \rightarrow Verb$
$VP \rightarrow Verb\ NP$
$NP \rightarrow Det\ Nominal$
$NP \rightarrow Proper\text{-}Noun$
$Nominal \rightarrow Noun$
$Nominal \rightarrow Noun\ Nominal$
$Nominal \rightarrow Nominal\ PP$
$PP \rightarrow Prep\ NP$

$Det \rightarrow that \mid this \mid a \mid the$
$Noun \rightarrow book \mid flight \mid meal$
$Verb \rightarrow book \mid include \mid includes$
$Prep \rightarrow from \mid to \mid on$
$Proper\text{-}Noun \rightarrow JFK \mid LA \mid TWA$
$Aux \rightarrow does$

## 4

Use XFST to test the resulting automata. Try to recognize the following strings:

this flight includes a meal

the flight from LA includes a meal

does the flight include a meal

book that flight

the flight from JFK to LA on TWA includes a meal

the flight does include a meal

the flight does not include a meal

the flight

## 5

Test your implementation on the same grammar, to which the following rules are added:

$NP \rightarrow Det\ Nominal\ Rel\ VP \quad Rel \rightarrow which \mid that$

## 6

Use XFST to test the resulting automata. Try to recognize the following strings:

the flight that includes a meal

the flight from LA that includes a meal

the flight which includes a meal includes a meal