Finite Automata Theory and Formal Languages

TMV027/DIT321– LP4 2014

Lecture 13
Ana Bove

May 12th 2014

Overview of today's lecture:

- Decision properties for CFL.

Recap: Context-Free Grammars

- Regular languages are also context-free;
- Chomsky hierarchy;
- Simplification of grammars:
  - Elimination of ϵ-productions;
  - Elimination of unit productions;
  - Elimination of useless symbols:
    - Elimination of non-generating symbols;
    - Elimination of non-reachable symbols;
- Chomsky normal forms;
- Pumping lemma for context-free languages.

Decision Properties of Context-Free Languages

Very little can be answered when it comes to CFL.

The major tests we can answer are whether:

- The language is empty;

  (See the algorithm that tests for generating symbols in slide 6 lecture 12: if $\mathcal{L}$ is a CFL given by a grammar with start variable S, then $\mathcal{L}$ is empty if S is not generating.)

- A certain string belong to the language.

Testing Membership in a Context-Free Language

Checking if $w \in \mathcal{L}(G)$, where $|w| = n$, by trying all productions may be exponential on n.

An efficient way to check for membership in a CFL is based on the idea of *dynamic programming*.

(Method for solving complex problems by breaking them down into simpler problems, applicable mainly to problems where many of their subproblems are really the same; not to be confused with the *divide and conquer* strategy.)

The algorithm is called the *CYK algorithm* after the 3 people who independently discovered the idea: Cock, Younger and Kasami.

It is a $O(n^3)$ algorithm.

The CYK Algorithm

Let $G = (V, T, \mathcal{R}, S)$ be a CFG in CNF and $w = a_1 a_2 \ldots a_n \in T^*$.

Does $w \in \mathcal{L}(G)$?

In the CYK algorithm we fill a table

V_{1n}					
$V_{1(n-1)}$	V_{2n}				
$\vdots$	$\vdots$				
V_{12}	V_{23}	V_{34}	$\ldots$	$V_{(n-1)n}$	
V_{11}	V_{22}	V_{33}	$\ldots$	$V_{(n-1)(n-1)}$	V_{nn}
a_1	a_2	a_3	$\ldots$	a_{n-1}	a_n

where $V_{ij} \subseteq V$ is the set of A's such that $A \Rightarrow^* a_i a_{i+1} \ldots a_j$.

We want to know if $S \in V_{1n}$, hence $S \Rightarrow^* a_1 a_2 \ldots a_n$.

CYK Algorithm: Observations

- Each row corresponds to the substrings of a certain length:
  - bottom row is length 1,
  - second from bottom is length 2,
  - ...
  - top row is length n;
- We work row by row upwards and compute the V_{ij}'s;
- In the bottom row we have $i = j$, that is, ways of generating the string a_i;
- V_{ij} is the set of variables generating $a_i a_{i+1} \ldots a_j$ of length $j - i + 1$ (hence, V_{ij} is in row $j - i + 1$);
- In the rows below that of V_{ij} we have all ways to generate shorter strings, including all prefixes and suffixes of $a_i a_{i+1} \ldots a_j$.

CYK Algorithm: Table Filling

Remember we work with a CFG in CNF.

We compute V_{ij} as follows:

Base case: First row in the table. Here $i = j$.
Then $V_{ii} = \{A \mid A \rightarrow a_i \in \mathcal{R}\}$.

Induction step: To compute V_{ij} for $i < j$ we have all V_{pq}'s in rows below.

The length of the string is at least 2, so $A \Rightarrow^* a_i a_{i+1} \ldots a_j$ starts with $A \Rightarrow BC$ such that $B \Rightarrow^* a_i a_{i+1} \ldots a_k$ and $C \Rightarrow^* a_{k+1} \ldots a_j$ for some k.

So $A \in V_{ij}$ if $\exists k, i \leqslant k < j$ such that

- $B \in V_{ik}$ and $C \in V_{(k+1)j}$;
- $A \rightarrow BC \in \mathcal{R}$.

We need to look at
$(V_{ii}, V_{(i+1)j}), (V_{i(i+1)}, V_{(i+2)j}), \ldots, (V_{i(j-1)}, V_{jj})$.

CYK Algorithm: Example

Consider the grammar given by the rules

$$S \rightarrow AB \mid BA \qquad A \rightarrow AS \mid a \qquad B \rightarrow BS \mid b$$

Does *abba* belong to the language generated by the grammar?

We fill the corresponding table:

$\{S\}$			
$\emptyset$	$\{B\}$		
$\{S\}$	$\emptyset$	$\{S\}$	
$\{A\}$	$\{B\}$	$\{B\}$	$\{A\}$
a	b	b	a

$S \in V_{14}$ then $S \Rightarrow^* abba$.

CYK Algorithm: Example

Consider the grammar given by the rules

$$S \to XY \qquad X \to XA \mid a \mid b$$
$$Y \to AY \mid a \qquad A \to a$$

Does *babaa* belong to the language generated by the grammar?

We fill the corresponding table:

$\emptyset$				
$\emptyset$	$\emptyset$			
$\emptyset$	$\emptyset$	$\{S, X\}$		
$\{S, X\}$	$\emptyset$	$\{S, X\}$	$\{S, X, Y\}$	
$\{X\}$	$\{A, X, Y\}$	$\{X\}$	$\{A, X, Y\}$	$\{A, X, Y\}$
b	a	b	a	a

$S \notin V_{15}$ then $S \not\Rightarrow^* babaa$.

Overview of Next Lecture

Sections 7.3, and bit of 6 and of 8:

- Closure properties for CFL;
- Push-down automata;
- Turing machines.