جامعة
الملك سعود
King Saud University

# Parallel Processing

Project 2 Phase1

CYK Algorithm and Context-Free Grammar

**Students:**

Al-Anoud Al-Subaie
Raghad Al-Suhaibani
Ftoon Al-Rubayea

# Table of Contents:

# 1.Background :

## 1.1 CONTEXT FREE GRAMMAR (CFG):

### 1.1.2 Grammar definition:

In general, formal language uses an alphabet (set of letters), which can be used to create words (finite, orderedseries of letters). Set of words is infinite, however, a language uses only finite subset of them. Formal grammaris a description of the language formulated as set of rules. The rules describes the way of forming the words offormal language from a language alphabet symbols.

### 1.1.3 Chomsky normal form:

Every context-free grammar can be expressed in a Chomsky normal form. This form of grammar requires allproduction rules to be in a form of either:

•A→a(non-terminal symbol is mapped to the terminal symbol) .

•A→BC(non-terminal symbol is mapped to the pair of non-terminal symbols).

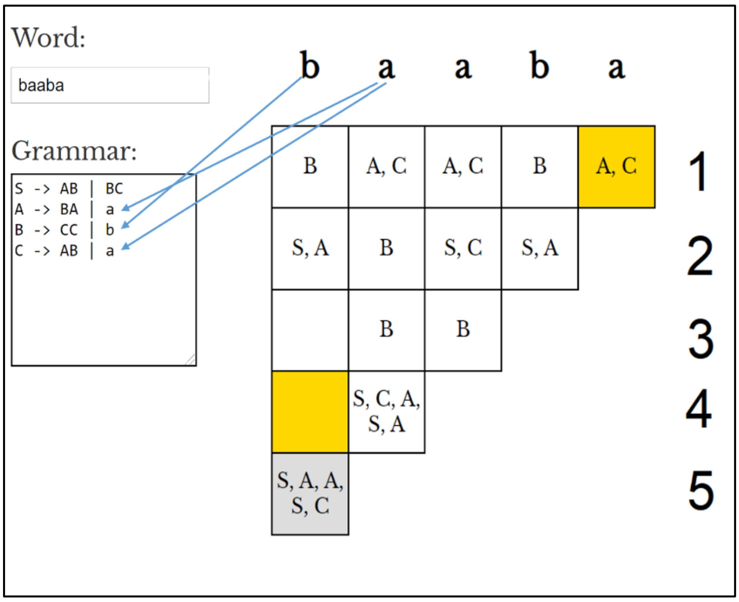
Algorithms that will be further described and implemented require the input grammar to be in Chomsky normalform. This is the form which is used within GCS system to describe classifiers, therefore, it is not a limitationwithin context of this thesis. However, general instances of this problem (with grammars which are not in CNFform) demand converting them to CNF .

## 1.2 CYK Algorithm :

Syntax analysis is a process of analyzing input textdata, given as a series of tokens letters in order to determine its structure, according to the given formalgrammar. Output of the parsing process is answer whether computing string belongs to the language describedby the grammar or not. Additional output is a data structure which represents the parsed text syntax tree.

Basic versionof algorithm gives answer whether the word belongs to the language, or not.Process of parsing cannot be proceeded efficiently on every kind of grammar. Effective algorithms of parsingrequire context-free grammars, which have some bounds and restrictions in describing languages (i. e. lack ofcontext sensitivity, mentioned above). Therefore, not every language can be expressed by this kind of grammar.There are two approaches of parsing process : top-down and bottom-up. First approach (top-downparses giventext starting from the left, attempting to apply the starting production rule first and then descended rules. Secondone (bottom-upapplies the production rules for basic tokens, and replaces them to the following non-terminalsymbols, with the goal in gaining the start symbol at the end.

### The first character:



Word:

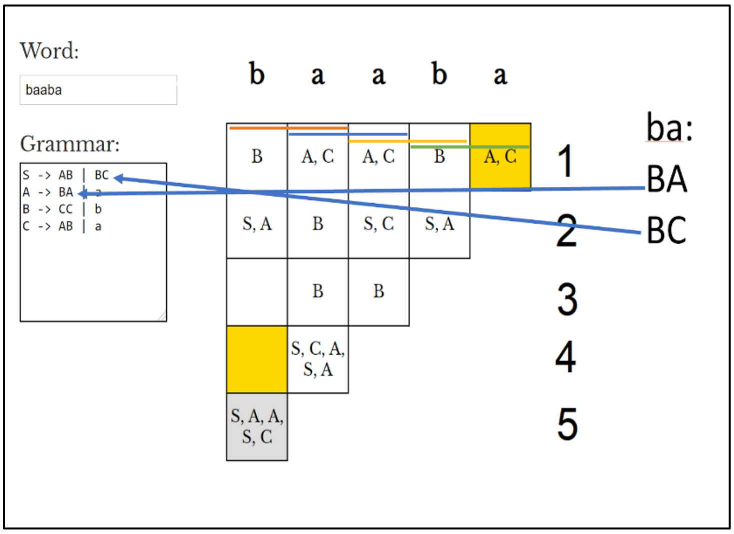
baaba

Grammar:

```
S -> AB | BC
A -> BA | a
B -> CC | b
C -> AB | a
```

| b | a | a | b | a | |
|---|---|---|---|---|---|
| B | A, C | A, C | B | A, C | 1 |
| S, A | B | S, C | S, A | | 2 |
| | B | B | | | 3 |
| | S, C, A,<br>S, A | | | | 4 |
| S, A, A,<br>S, C | | | | | 5 |

### The second character:



Word:

baaba

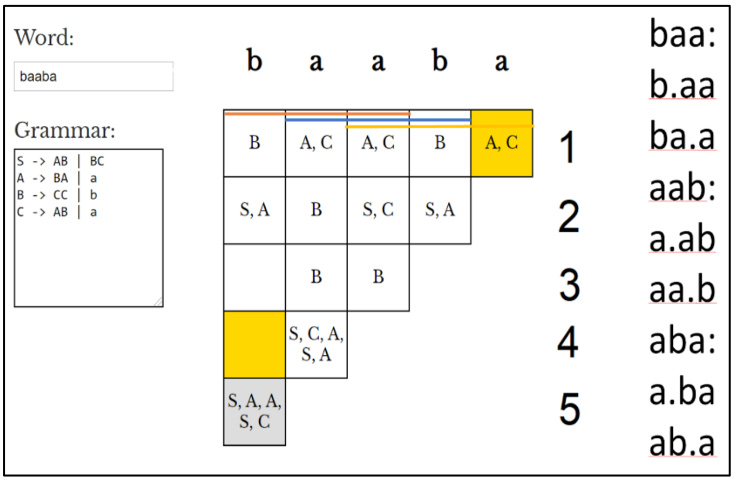
Grammar:

```
S -> AB | BC
A -> BA | a
B -> CC | b
C -> AB | a
```

| b | a | a | b | a | |
|---|---|---|---|---|---|
| B | A, C | A, C | B | A, C | 1 |
| S, A | B | S, C | S, A | | 2 |
| | B | B | | | 3 |
| | S, C, A,<br>S, A | | | | 4 |
| S, A, A,<br>S, C | | | | | 5 |

ba:
BA
BC

### The third character:



Word:

baaba

Grammar:

```
S -> AB | BC
A -> BA | a
B -> CC | b
C -> AB | a
```

| b | a | a | b | a | |
|---|---|---|---|---|---|
| B | A, C | A, C | B | A, C | 1 |
| S, A | B | S, C | S, A | | 2 |
| | B | B | | | 3 |
| | S, C, A,<br>S, A | | | | 4 |
| S, A, A,<br>S, C | | | | | 5 |

baa:
b.aa
ba.a
aab:
a.ab
aa.b
aba:
a.ba
ab.a

## 2.Problem Definition :

**P1 .** Write a multicore program that uses CYK algorithm [13] to parse a string of symbols. The inputs are a context-free grammar Gin Chomsky Normal Form and a string of symbols. At the end, the program should print YES if the string of symbols can be derived by the rules of the grammar and NO otherwise. Write a sequential program (no OpenMP directives at all) as well. Compare the running time of the sequential program with the running time of the multicore-program and compute the speedup for different grammars and different string lengths .

## 3.Results :

The first things we will write code for multicore program and Sequential that using CYK algorithm . The program outputs if the entered string belong to the given grammar (Yes/No), then based on the implementation of the program ( sequential, multi-core) the speedup is calculated and tested with varying the staring lengths.