



Wrocław University  
of Science and Technology

**Faculty of Computer Science and Management**

Field of study: Computer Science

Specialty: —

Engineering Thesis

## **FORMAL GRAMMAR PRODUCTION RULE PARSING TOOL**

Karol Belina

keywords:

Parser combinators, context-free grammars,  
Extended Backus-Naur Form

short summary:

Supervisor	dr inż. Zdzisław Sławski	.....	.....
	Title/degree/name and surname	grade	signature

### The final evaluation of the thesis

Head of the examination commission	.....	.....	.....
	Title/degree/name and surname	grade	signature

*For the purposes of archival thesis qualified to:\**

*a) category A (perpetual files)*

*b) category BE 50 (subject to expertise after 50 years)*

*\* delete as appropriate*

stamp of the faculty

Wrocław 2020



## **Abstract**



# Contents

<b>Introduction</b>	<b>1</b>
<b>1 Problem analysis</b>	<b>3</b>
1.1 Description . . . . .	3
1.2 Motivation . . . . .	3
1.3 Goal . . . . .	3
1.4 Scope . . . . .	3
<b>2 Analysis of similar solutions</b>	<b>5</b>
<b>3 Theoretical preliminaries</b>	<b>7</b>
3.1 Context-free grammars . . . . .	7
3.2 Parser combinators . . . . .	7
<b>4 Design</b>	<b>9</b>
4.1 Functional requirements . . . . .	9
4.2 Non-functional requirements . . . . .	9
4.3 The architecture . . . . .	9
4.4 Interface prototype . . . . .	9
<b>5 Implementation</b>	<b>11</b>
5.1 Used technologies . . . . .	11
<b>6 Testing</b>	<b>13</b>
6.1 Automated tests . . . . .	13
<b>Summary</b>	<b>15</b>
<b>Bibliography</b>	<b>17</b>
<b>List of Figures</b>	<b>19</b>
<b>List of Tables</b>	<b>21</b>
<b>List of Listings</b>	<b>23</b>
<b>A Modified specification</b>	<b>25</b>



## **Introduction**





## **1. Problem analysis**

### **1.1. Description**

### **1.2. Motivation**

### **1.3. Goal**

### **1.4. Scope**



## **2. Analysis of similar solutions**



### **3. Theoretical preliminaries**

#### **3.1. Context-free grammars**

#### **3.2. Parser combinators**



## **4. Design**

### **4.1. Functional requirements**

### **4.2. Non-functional requirements**

### **4.3. The architecture**

### **4.4. Interface prototype**





## **5. Implementation**

### **5.1. Used technologies**



## **6. Testing**

### **6.1. Automated tests**



## **Summary**



## **Bibliography**

- [1] *Information technology, syntactic metalanguage, extended BNF*. ISO/IEC, 1996.





## **List of Figures**



## **List of Tables**



## List of Listings

A.1	Modified version of the EBNF language specification defined in [1]	. . . .	25
-----	--	---------	----



## A. Modified specification

```
1 character
2   = ? any Unicode non-control character ?;
3 letter
4   = ? any Unicode alphabetic character ?;
5 digit
6   = ? any Unicode numeric character ?;
7 whitespace
8   = ? any Unicode whitespace character ?;
9 comment
10  = '(*', {comment | character}, '*)';
11 gap
12  = (whitespace | comment), {whitespace}, {{comment}, {whitespace}};
13 identifier
14  = letter, {{whitespace}, letter | digit};
15 factor
16  = [[gap], digit, {{whitespace}, digit}, [gap], '*'],
17    [gap], [(identifier
18      | ('[' | '(/', alternative, (']' | '/)')
19      | ('{' | '(:', alternative, ('}' | ':)')
20      | '(', alternative, ')')
21      | '"', character - '"', {character - '"'}, '"'
22      | "'", character - "'", {character - "'"}, "'"
23      | '?', {{whitespace}, character - '?', '?'), [gap]]];
24 term
25   = factor,
26     ['- ', ? a factor that could be replaced
27       by a factor containing no identifiers ?];
28 sequence
29   = term, {' ', term};
30 alternative
31   = sequence, {'|', sequence};
32 production
33   = [gap], identifier, [gap], '=', alternative, (';' | '.'), [gap];
34 grammar
35   = production, {production};
```

Listing A.1: Modified version of the EBNF language specification defined in [1]