AKADEMIA GÓRNICZO-HUTNICZA

Wydział Informatyki, Elektroniki i Telekomunikacji

Translator języka dokuWIKI do HTML

Maciej Jadach - mac.jadach@gmail.com

1. Specyfikacja gramatyki języka w notacji ANTL4

Plik ze specyfikacją znajduje się w repozytorium pod podaną ścieżką: src/main/dokuWiki.g4 Plik zawiera zarówno tokeny i reguły lexera jak i reguły parsera.

Nazwy tokenów i reguł lexera według zasad definiowania gramatyki dla tego narzędzia zaczynają się od wielkiej litery, natomiast nazwy reguł parsera zaczynają się od małej litery.

Dodatkowo wykorzystano funkcjonalność ANTL4, dzięki której bezpośrednio w definicji gramatyki możemy dodać obiekty klas z języka Java i operować na nich bezpośrednio przy definicji reguł. W przypadku tego projektu jest to dodanie *StringBuilder'a* jako "parser member".

Definicja gramatyki:

```
grammar docuWiki;


@parser::members {
 protected StringBuilder htmlBuilder = new StringBuilder();
}

SPACE: ' ' | '\t' ;
CHAR
   :         '!' | '"' | '#' | '$' | '%' | '&'
   |         '*' | '+' | ',' | '-' | '.' | '/'
   |         ':' | ';' | '?' | '@' | '^' | '_' | '`'
   |    '0'..'9' | 'A'..'Z'  | 'a'..'z'
   |         '(' | ')' | '~'
   |        '\'' | '<' |  '=' | '[' | ']' | '|'
   ;

HEADLINE_1: '=';
HEADLINE_2: '==';
HEADLINE_3: '===';
HEADLINE_4: '====';
BOLD_MARK: '**';
ITALIC_MARK : '//';
UNDERLINE_MARK: '__';
NEWLINE_MARK:  '\\';
LINK_OPEN: '[[';
LINK_END: ']]';
LIST_MARK: '* ';
NUMBERED_LIST_MARK: '- ';
QUOTE: '>';
PIPE: ' | ';
```

```
CODE_START: '<code>';
CODE_END: '</code>';


SUBSCRIPT_START: '<sub>';
SUBSCRIPT_END: '</sub>';

SUPERSCRIPT_START: '<sup>';
SUPERSCRIPT_END: '</sup>';

DELETED_START: '<del>';
DELETED_END: '</del>';

MEDIA_START: '{{';
MEDIA_END: '}}';

URL_START: 'http://';
//ANY: .?;
NL: '\r'?'\n';
WS  : [ \t\r\n]+ -> skip ;

paragraph: (CHAR | SPACE)+ ;
url: URL_START paragraph;
headline
   : HEADLINE_1 CHAR HEADLINE_1 {htmlBuilder.append("<h1>" + $CHAR.text + "</h1>");}
   | HEADLINE_2 paragraph HEADLINE_2 {htmlBuilder.append("<h2>" + $paragraph.text +
"</h2>");}
   | HEADLINE_3 paragraph HEADLINE_3 {htmlBuilder.append("<h3>" + $paragraph.text +
"</h3>");}
   | HEADLINE_4 paragraph HEADLINE_4 {htmlBuilder.append("<h4>" + $paragraph.text +
"</h4>");}
   ;

bold : BOLD_MARK paragraph BOLD_MARK {htmlBuilder.append("<b>" + $paragraph.text +
"</b>");} ;
italic : ITALIC_MARK paragraph ITALIC_MARK {htmlBuilder.append("<i>" + $paragraph.text +
"</i>");} ;
underline : UNDERLINE_MARK paragraph UNDERLINE_MARK {htmlBuilder.append("<u>" +
$paragraph.text + "</u>");} ;
subscript : SUBSCRIPT_START paragraph SUBSCRIPT_END {htmlBuilder.append("<sub>" +
$paragraph.text + "</sub>");} ;
superscript : SUPERSCRIPT_START paragraph SUPERSCRIPT_END {htmlBuilder.append("<sup>" +
$paragraph.text + "</sup>");} ;
deleted : DELETED_START paragraph DELETED_END {htmlBuilder.append("<strike>" +
$paragraph.text + "</strike>");} ;
externalLink: LINK_OPEN url PIPE paragraph LINK_END {htmlBuilder.append("<a href='" +
$url.text + "'>" + $paragraph.text + "</a>");} ;
media: MEDIA_START url MEDIA_END {htmlBuilder.append("<img src='" + $url.text + "'>");}
;
quote: QUOTE paragraph {htmlBuilder.append("<blockquote>" + $paragraph.text +
"</blockquote>");};
code: CODE_START paragraph CODE_END {htmlBuilder.append("<code>" + $paragraph.text +
```

```
"</code>");};
newline: NEWLINE_MARK {htmlBuilder.append("</br>");} ;


elements
   : bold elements
   | italic elements
   | newline elements
   | paragraph elements
   | underline elements
   | subscript elements
   | superscript elements
   | deleted elements
   | externalLink elements
   | media elements
   | quote elements
   | code elements
   | headline elements
   | newline elements
//    | .+
   ;

body: elements+? ;
```

2. Zrealizowane translacje języka dokuWIKI:

Zrealizowano następujące funkcjonalności języka dokuWiki:

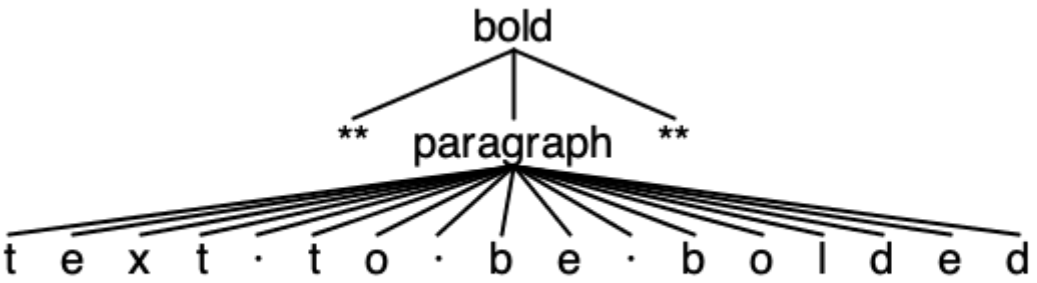
- Pogrubienie tekstu (** TEXT **)
- Kursywa tekstu (//TEXT//)
- Podkreślenie tekstu (__TEXT__)
- Nagłówki 1-4 (=== HEADING3 ===)
- Indeks dolny (_{TEXT}
- Indeks górny (^{TEXT})
- Przekreślenie tekstu (~~TEXT~~)
- Link zewnętrzny ([[URL | NAME]])
- Media (Obrazek) ({{URL}})
- Cytat (>TEXT)
- Blok kodu (<code>TEXT</code>)

Przykładowe wyniki translacji:

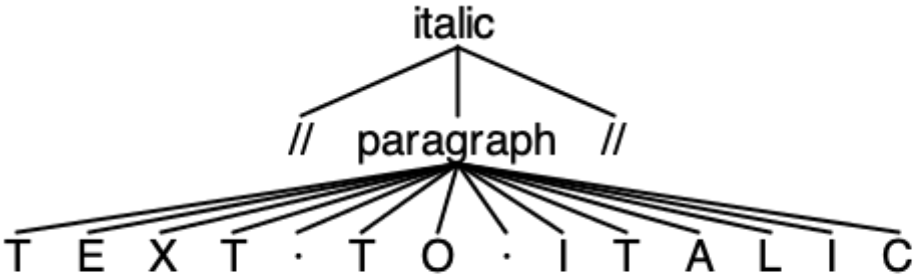
```
---------------HEADLINE3---------------
INPUT : ===Headline 3===
OUTPUT: <h3>Headline 3</h3>

---------------HEADLINE4---------------
INPUT : ====Headline 4====
OUTPUT: <h4>Headline 4</h4>

---------------BOLD---------------
INPUT : **text to be bold**
OUTPUT: <b>text to be bold</b>

---------------ITALIC---------------
INPUT : //text to be italic//
OUTPUT: <i>text to be italic</i>

---------------UNDERLINE---------------
INPUT : __text to be underlined__
OUTPUT: <u>text to be underlined</u>

---------------DELETED---------------
INPUT : <del> test to be deleted </del>
OUTPUT: <strike> test to be deleted </strike>

---------------SUBSCRIPT---------------
INPUT : <sub> text to be subscribed </sub>
OUTPUT: <sub> text to be subscribed </sub>

---------------SUPERSCRIPT---------------
INPUT : <sup> text to be subscribed </sup>
OUTPUT: <sup> text to be subscribed </sup>

---------------EXTERNALLINK---------------
INPUT : [[http://external.link | name ]]
OUTPUT: <a href='http://external.link'>name </a>

---------------MEDIA---------------
INPUT : {{http://image.source.png}}
OUTPUT: <img src='http://image.source.png'>

---------------QUOTE---------------
INPUT : >QUOTE
OUTPUT: <blockquote>QUOTE</blockquote>
```

3. Przykładowe rozkłady drzewa AST

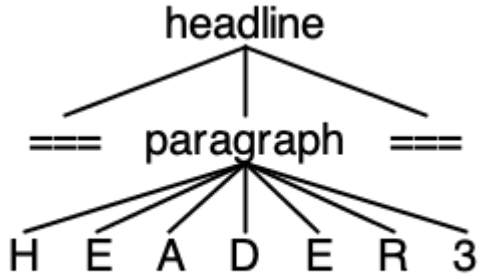
- Pogrubienie tekstu (**TEXT**)



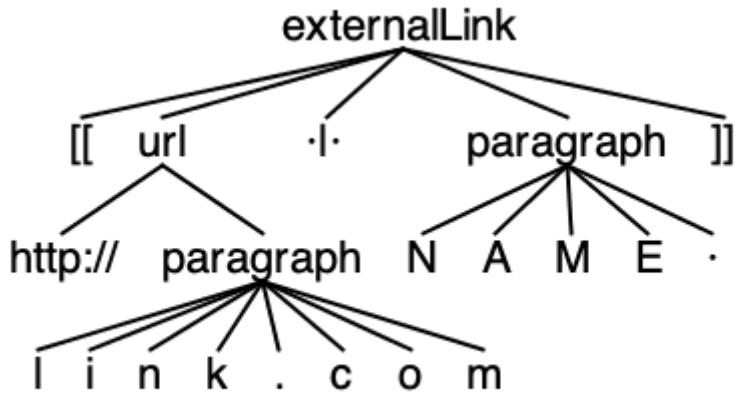
- Kursywa tekstu (//ITALIC//)



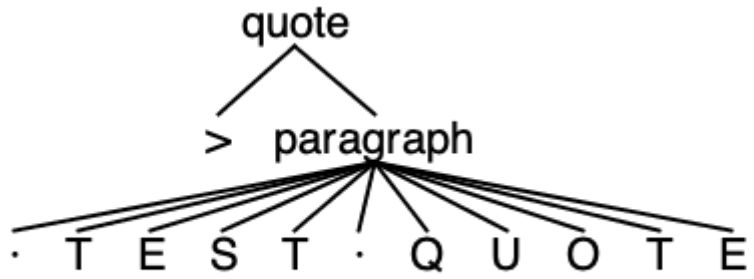
- Nagłowek 3 (===HEADER3===)



- Link zewnętrzny [[url | name]]



- Cytat (>QUOTE)



3. Uzasadnienie wyboru ANTLR4 jako generatora parserów:

Powody:

- Twórca projektu ma doświadczenie w programowaniu w języku Java, więc naturalnym wyborem było wybranie ANTLR4 zamiast PLY (Python)
- Łatwo dostępna, obszerna dokumentacja
- Duża liczba przykładowych gramatyk w internecie
- Dodatkowy mechanizm gramatyki w postaci “parser member” ułatwiający konstruowanie translatora

4. Napotkane problemy i ich rozwiązania

Podstawowym problemem związanym z realizacją projektu była kompletna nieznajomość tematu dotyczącego pisania gramatyk dla generatora parserów. Próg wejścia do technologii generator parserów, w tym przypadku ANTLR4 był bardzo wysoki. Wszelkie źródła, które można było znaleźć w internecie nie pomagały w tym początkowo. Były one bardzo przydatne i można było się z nich wiele nauczyć lecz dopiero na etapie dalszej edukacji. “Getting started” do generatora parserów jakim jest ANTLR4 w bardzo znikomym stopniu tłumaczył podstawową logikę na której on bazuje. Pierwsze kroki oraz pierwsze definicje gramatyki języka były robione metodą prób i błędów, do momentu zrozumienia mechanizmu działania generatora. Umożliwiło to po pewnym czasie korzystanie już z bardziej zaawansowanych źródeł w internecie jak również rozumienie przykładowych gramatyk.

Kolejnym z problemów podczas tworzenia translatora był wybór wędrowania po Abstract Syntax Tree, będącego wynikiem działania parsera. Początkowo do wyboru były dwie opcje, wykorzystanie wzorców Listener lub/oraz Visitor. Użycie każdego z nich wiązało się z projektowaniem oraz pisaniem skomplikowanej logiki, związanej z odwiedzanymi odnogami drzewa AST, co spowolniłoby pracę lub w ostatecznym przypadku uniemożliwiło realizację projektu. Z pomocą przyszedł mechanizm ANTLR, który nazywa się “parser members”. Umożliwia on bezpośrednie dodanie do gramatyki obiektów z wybranego języka w tym przypadku z Javy. Było to idealne i szybkie rozwiązanie na problemy projektu związane z translacją dokuWiki do HTML’a. Korzystając z w/w możliwości dodawania dowolnych obiektów, dodano obiekt typu *StringBuilder* i korzystając z jego metod *append(String s)* zbudowano logikę translacji.

5. Bilbiografia i linkografia

- B. W. Kernighan, D. M. Ritchie, "Język ANSI C. Programowanie.", Wydanie II, 2010r, ISBN: 978-83-246-2578-9
- https://github.com/antlr/antlr4/blob/master/doc/index.md
- https://www.antlr.org
- https://tomassetti.me/antlr-mega-tutorial/