## Grzegorz Bylina

## Tomasz Gajda

## Elpidiusz Wszołek

Translator Dokuwiki ⇒ HTML

Projekt z przedmiotu Języki Formalne i Kompilatory

# Wstęp

Celem projektu jest napisanie aplikacji przetwarzającej tekst sformatowany w składni Dokuwiki na tekst formatowany znacznikami HTML.

Oba języki są językami znacznikowymi, a więc wszystkie informacje oprócz czystego tekstu wyrażone są za pomocą wplecionych w niego znaczników. Składnia obu języków jest dość przejrzysta i niezbyt skomplikowana. Dokuwiki wydaje się być bardziej czytelna i zrozumiała szczególnie dla użytkowników nie przywykłych do czytania i analizy kodu w innych językach programowania, przy zachowaniu dużych możliwości formatowania tekstu oferowanych przez HTML.

Mimo pokrywających się zastosowań obu języków nie wszystkie możliwości Dokuwiki mają bezpośrednie odwzorowanie w HTML i odwrotnie. Przykładowo część możliwości Dokuwiki wspierana jest przez arkusz CSS, więc bez jego użycia nie da się uzyskać zadowalającego wyniku. HTML również posiada funkcjonalności, których wykorzystania nie przewidzieli twórcy Dokuwiki.

# Gramatyka (w rozszerzonej notacji BNF)

CYFRA: 0|1|2|3|4|5|6|7|8|9

LITERA : ( A|B|C|D | …) | ( a|b|c|d |…)

NUMER : {CYFRA}

ALNUM: CYFRA | LITERA

WORD: { ALNUM | OGONEK }

ENTER : "\r"| "\n"| "\r\n"

NOWALINIA : "\\"

INNY\_ZNAK: "~" | "`" | "!" | "@" | "#" | "$" | "%" | "&" | "(" | ")" | "\_" | "+" | "=" | "{" | "}" | "[" | "]" | ";" | ":" | "'" | "\"" | "\\" | "<" | ">" | "," | "." | "/" | "?" | "\*" | "-"

WCIĘCIE : SPACE SPACE

AKAPIT : ENTER ENTER

LISTAN : WCIĘCIE "\*"

LISTAP : WCIĘCIE "-"

BOLD : "\*\*"

ITALICS : "//"

UNDERLINE : "\_\_"

NO\_FORMAT : "%%"

MONO : "''"

SPACE : " " | "\t"

LINIA : "---" {"-"} {[SPACE]} ENTER

H1 : "======" {["="]}

H2 : "====="

H3 : "===="

H4 : "==="

H5 : "=="

HC : "==" {"="} ENTER

DOMENA : { ({ ALNUM} ".") } WORD

URL : ("www." | "http://") DOMENA ( "/" {[ WORD ]} {("." < WORD >)}

EMAIL : "<" EMAIL\_NAZWA "@" DOMENA

EMAIL\_NAZWA : { ({ ALNUM } ".") } { ALNUM }

WYMIARY : "?" NUMER "x" NUMER

EMOTKA: "8-)" | "8-O" | ":-(" | ":-)" | "=)" | ":-/" | ":-\\" | ":-?" | ":-D" | ":-P" | ":-O" | ":-X" | ":-|" | ";-)" | "^\_^" | ":?:" | ":!:" | "LOL" | "FIXME" | "DELETEME"

SYMBOL : "->" | "<-" | "<->" | "=>" | "<=" | "<=>" | "--" | "---" | "(c)" | "(tm)" | "(r)"

HEADER : "^"

KOLUMNA : "|"

VERT\_SPAN : ":::" zrobić albo wyrzucić stąd I nie wspominać

LINKO : "[["

LINKC : "]]"

OBRO : "{{"

OBRC : "}}"

DELO : "<del>"

DELC : "</del>"

SUPO : "<sup>"

SUPC : "</sup>"

SUBO : "<sub>"

SUBC : "</sub>"

CODEO : "<code>"

CODEC : "</code>"

QUOTE : ENTER {(">")}

FILEO : "<file>"

FILEC : "</file>"

NOWIKIO : "<nowiki>"

NOWIKIC : "</nowiki>"

FOOTNOTEO : "((" zrobić albo wyrzucić i zapomnieć

FOOTNOTEC : "))" zrobić albo wyrzucić i zapomnieć

SFORMATOWANY : FOOTNOTES | EMAIL | BOLDED | CZYSTYTEKST | ITALICS | UNDERLINED | MONOSPACED | NOFORMAT | NOWIKI | SUB | SUP | DEL | LINK | CODE | FILE

BOLDED : BOLD { SFORMATOWANY } BOLD

ITALICS : UNDERLINE { SFORMATOWANY } UNDERLINE

MONOSPACED : MONO { SFORMATOWANY } MONO

NOFORMAT : NO\_FORMAT { ANYTHING } NO\_FORMAT

NOWIKI : NO\_WIKIO { ANYTHING } NO\_WIKIC

CODE : CODEO { ANYTHING } CODEC

FILE : FILEO { ANYTHING } FILEC

SUB : SUBO { SFORMATOWANY } SUBC

SUP : SUPO { SFORMATOWANY } SUPC

DEL : DELO { SFORMATOWANY } DELC

HEAD1 : H1 { ANYTHING } HC

HEAD2 : H2 { ANYTHING } HC

HEAD3 : H3 { ANYTHING } HC

HEAD4 : H4 { ANYTHING } HC

HEAD5 : H5 { ANYTHING } HC

QUOTELINIA : QUOTE { SFORMATOWANY } [ ENTER ]

LISTAU : [WCIĘCIE] { LISTAELEMENT }

LISTAELEMENT : ( (LISTAN | LISTAP) { [SFORMATOWANY] } ENTER ) | (LISTA)

TABWIERSZ : ENTER { KOMORKA } { [(KOLUMNA | HEADER)] }

KOMORKA : (KOLUMNA | HEADER) { SFORMATOWANY } { [KOLUMNA] }

FOOTNOTES : FOOTNOTEO CZYSTYTEKST FOOTNOTEC

OBRAZEK : OBRO (URL | DOMENA | CZYSTYTEKST) OBRC

LINK : URL | (LINKO (URL | DOMENA | CZYSTYTEKST) ( [ KOLUMNA CZYSTYTEKST ] ) )

SLOWO : WORD | NUMER

CZYSTYTEKST : { ( SLOWO | INNY\_ZNAK | SPACE ) }

# Javacc

Do budowy translatora użyliśmy narzędzia *javacc* w wersji 1.5. Jest to generator analizatorów składniowych i parserów przeznaczony dla języka Java. Dostępny jest na prawach licencji BSD.

Operuje na gramatyce w rozszerzonej notacji BNF. Jego możliwości ograniczone są do generowania parserów zstępujących, a więc gramatyk typu LL(k). Istnieją narzędzia o dużo szerszych możliwościach, lecz do naszych zastosowań możliwości *javacc* są aż nadto wystarczające.

Początkowo podjęliśmy też pewne próby z generatorem skanerów *jflex*, lecz składnia i sposób komunikacji z zewnętrznymi klasami *javacc* okazał się dla nas bardziej intuicyjny. Dodatkowo *javacc* jest „podwójnym” narzędziem – generatorem analizatorów składniowych i parserów w jednym, więc nie trzeba martwić się komunikacją między tymi modułami.

# Skaner

Realizacja skanera za pomocą *javacc* nie przysporzyła większych trudności (z wyjątkiem konieczności zapoznania się ze składnią przyjmowaną przez generator). Niezbędne było także lepsze zapoznanie się ze składnią samej dokuwiki a także przypadkami „trudnych” lub „dziwnych” układów znaczników, których efekt przetwarzania przez silnik dokuwiki badaliśmy na działającym systemie. W tym celu zainstalowaliśmy dokuwiki na koncie studenckim na serwerach agh.

Założyliśmy, że skaner (i docelowo parser) będzie pokrywał funkcjonalnością składnię opisaną w oficjalnym dokumencie dokuwiki pod adresem <https://www.dokuwiki.org/wiki:syntax>.

Skaner można testować bezpośrednio w konsoli, wpisując wejściowy teskt ręcznie lub przekierowując strumień z pliku. Wykonaliśmy też prosty interfejs graficzny w javie, w którym szybciej można sprawdzić działanie krótkich przykładów.

Na etapach testu skanera moduł parsera został ograniczony do akceptowania i wyświetlania każdego otrzymanego tokenu, bez sprawdzania poprawności składni i wykonywania jakichkolwiek dodatkowych operacji.

## Przykłady działania skanera:

**Podstawowe formatowanie:**

DokuWiki wspiera \*\*pogrubienie\*\*, //pochylenie//, \_\_podkreślenie\_\_ and ''stałą szerokość'' tekstów.

Można je też \*\*\_\_//''połączyć''//\_\_\*\*.

WORD SPACJA WORD SPACJA BOLD WORD BOLD Superznak SPACJA ITALICS WORD ITALICS Superznak SPACJA PODKRESLENIE WORD PODKRESLENIE SPACJA WORD SPACJA MONO WORD SPACJA WORD MONO SPACJA WORD Superznak ENTER

WORD SPACJA WORD SPACJA WORD SPACJA BOLD PODKRESLENIE ITALICS MONO WORD MONO ITALICS PODKRESLENIE BOLD Superznak

**Rozpoznanie linków:**

http://www.google.com albo [www.google.com](http://www.google.com)

WORD SPACJA WORD SPACJA Superznak SPACJA URL SPACJA WORD SPACJA URL

**Listy:**

\*lista

\*drugi element

\*drugi poziom

\*dalej drugi

\*kolejny pierwszy

WORD Superznak ENTER

LISTAn WORD ENTER

LISTAn WORD SPACJA WORD ENTER

2X\_SPACJA LISTAn WORD SPACJA WORD ENTER

2X\_SPACJA LISTAn WORD SPACJA WORD ENTER

LISTAn WORD SPACJA WORD ENTER

Wymienione przykłady wydają się dawać poprawny rezultat. Skaner jednak w niektórych przypadkach (szczególnie ze znakami specjalnymi) daje rezultaty odbiegające od ideału. Te nieścisłości interpretacji zostają rozwiązane później, na etapie parsowania.

# Parser