

# Zadanie 1 z listy 6

Benjamin Jurczok - 244760

20 stycznia 2020

## Treść zadania:

Rozważmy gramatykę:

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow E \text{ or } T|T \\ T &\rightarrow T \text{ and } F|F \\ F &\rightarrow \text{not } F|(E)|\text{true}|\text{false} \end{aligned}$$

Pokazać że gramatyka generuje formuły boolowskie ze stałymi true and false, oraz odpowiedzieć na pytanie czy gramatyka jest jednoznaczna.

## Rozwiązanie:

### Generowanie formuł boolowskich:

Twierdzenie udowodnię za pomocą dowodu indukcyjnego po słowie długości  $n$  wygenerowanego przez gramatykę.

1. Dla  $n=1$  otrzymujemy *true* lub *false*, a więc jest f. boolowską
2. Dla słów długości  $< n$  założymy że także otrzymamy formy boolowskie, wtedy wystarczy pokazać że dla  $n$  także je otrzymamy.
  - Gdy słowo kończy się ')', więc została użyta produkcja  $F \rightarrow (E)$ . Słowo z pochodzące z kolejnej produkcji (od E) będzie więc długość  $n-2$  i więc z założenia indukcyjnego będzie formułą boolowską
  - Słowo kończy się *true* albo *false* i użyta była produkcja  $E \rightarrow E \text{ or } T|T$  albo  $T \rightarrow T \text{ and } F|F$ . Słowo generowane przez E lub T jest długości  $n-2$ , więc założenia indukcyjnego słowa wygenerowane przez E i T są formułą boolowską, zatem słowo o długości  $n$  też będzie.

### Jednoznaczność gramatyki:

Jednoznaczność udowodnię konstruując tablicę parsera LL(1).

Na początku eliminuję lewostronną rekursję z gramatyki:

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow TE' \\ E' &\rightarrow \text{or } TE'|\epsilon \\ T &\rightarrow FT' \\ E' &\rightarrow \text{and } FT'|\epsilon \\ F &\rightarrow \text{not } F|(E)|\text{true}|\text{false} \end{aligned}$$

Wyznaczam FIRST and FOLLOW dla gramatyki:

$$FIRST(E) = \{not, (, true, false\}$$

$$FIRST(T) = \{not, (, true, false\}$$

$$FIRST(F) = \{not, (, true, false\}$$

$$FIRST(E') = \{or, \epsilon\}$$

$$FIRST(T') = \{and, \epsilon\}$$

$$FOLLOW(E) = \{\$, )\}$$

$$FOLLOW(E') = \{\$, )\}$$

$$FOLLOW(T) = \{\$, ), or\}$$

$$FOLLOW(T') = \{\$, ), or\}$$

$$FOLLOW(F) = \{\$, ), and, or\}$$

Konstruuje tablicę parsera:

	<b>true</b>	<b>false</b>	<b>not</b>	<b>and</b>	<b>or</b>	<b>(</b>	<b>)</b>	<b>\$</b>
E	TE'	TE'	TE'			TE'		
E'					or TE'		$\epsilon$	$\epsilon$
T	FT'	FT'	FT'			FT'		
T'				and FT'	$\epsilon$		$\epsilon$	$\epsilon$
F	true	false	not F			(E)		

W tablicy nie ma konfliktów - gramatyka jest jednoznaczna.