Syntax

CMPSC 461
Programming Language Concepts
Penn State University
Fall 2016

1^{st} Midterm

Sep. 30 (Friday) 6:30PM – 7:45PM

010 Sparks Building

HW3 due next Wednesday NOON (**no late submission**)

1^{st} Mideterm covers materials in HW1 to HW3

Next Wednesday: review of assignments and practice problems

Next Friday: evening exam (no lecture)

Regular Expressions to Scanner

RE → NFSA → DFSA → Scanner

Specification

Implementation

All strings with n 0's followed by n 1's, $n \geq 1$?

Not a regular language!

Context-Free Grammar (CFG)

A CFG consists of

- A set of *terminals* T (basic alphabet)
- A set of *non-terminals* N
- A *start symbol* S (a non-terminal)
- A set of *production rules*

$$A ::= \omega$$

nonterminal

string of terminal and nonterminal

Context-Free Grammar (CFG)

Terminals

- Building block of CFG
- E.g., letters, digits, or tokens from scanner

Non-terminals

- Symbols defined by production rules
- E.g., if-stmt = IF LPAREN expr RPAREN stmt

terminal

CFG Example

A pure λ-term is defined inductively as follows:

- Any variable x is a λ-term
- If e is a λ-term, so is $\lambda x.\, e$ (abstraction)
- If e_1, e_2 are λ-terms, so is $e_1 e_2$ (application)

$$
\begin{aligned}
\text{term} ::= &\ \text{var} \\
&|\ \text{term term} \\
&|\ \boldsymbol{\lambda}\ \text{var}\ \boldsymbol{.}\ \text{term}
\end{aligned}
$$

terminals

CFG Example

term ::= var
| term term
| λ var **.** term
var ::= letter var
| letter
letter ::= **a** | **b** | ... | **Z**

In RE: var → (a | b | ... | Z)$^{+}$

CFG Example

$$
\begin{aligned}
\text{expr} &::= \ \text{id} \mid \text{number} \mid - \ \text{expr} \\
&\quad \mid (\text{expr}) \mid \text{expr op expr} \\
\text{op} &::= + \mid - \mid * \mid /
\end{aligned}
$$

...

Terminals Can be Tokens

$$\text{expr} ::= \text{id} \mid \text{number} \mid - \text{expr}$$
$$\mid (\text{expr}) \mid \text{expr op expr}$$

where id number op are tokens recognized by the scanner

Build CFG

All strings with n 0's followed by n 1's, $n \geq 1$?

$$S ::= 0S1 \mid 01$$

Build CFG

All strings with n 0's and n 1's, $n \geq 0$?

$$S ::= 0S1S \mid 1S0S \mid \epsilon$$

Context-Free Language

$L(G)$: the language (set of strings) defined by G

Definition:

$L(G)$ is the set of all terminal strings derivable from the start symbol of G

Derivation

A sequence from start symbol, each step a non-terminal is replace by RHS of production

A derivation of “xyz”

$\text{var} \Rightarrow \text{letter var}$

$\quad \Rightarrow \text{letter letter var} \Rightarrow \text{letter letter letter}$

$\quad \Rightarrow \text{x letter letter} \Rightarrow \text{x y letter} \Rightarrow \text{x y z}$

A multi-step reduction $\text{var} \Rightarrow^{*} \text{xyz}$

RE and CFG

CFG is strictly more expressive than RE

1. For any RE R, there is CFG C, such that $L(R) = L(C)$
2. Exists CFG C, such that for all RE R, $L(R) \neq L(C)$

1. Proof by induction.
2. A language with n 0's followed by n 1's, $n \geq 1$

$$S \rightarrow 0S1 \mid 01$$

RE	CFG
ϵ	$S ::= \epsilon$
a	$S ::=$ a
R_1 R_2	$S ::= R_1\ R_2$
R_1 | R_2	$S ::= R_1 \mid R_2$
R^*	$S ::= S\ R \mid \epsilon$

where R_1, R_2 and R are the equivalent nonterminal for R_1,R_2 and R in RE respectively