A well-parenthesized string is a string with the same number of ('s as)'s which has the property that every prefix of the string has at least as many ('s as)'s.

1. Write down all well-parenthesized strings of length six or less over the alphabet $\Sigma = \{ \, (\, , \,) \, \}$.

2. Let $L = \{ w \in \{ \, (\, , \,) \, \}^* : w \text{ is a well-parenthesized string} \}$. Prove that L is not regular.

Announcements:

Assignment 3 has been posted.

If you do the 10 point bonus question, make sure you submit your code on connex.

Midterm: Wednesday June 27, in class.

Old midterms are available from the course web page.

Assignment #1 and #2 solutions are available in the connex resources.

Alan Turing's 100th Birthday Celebration

A DAY OF FREE EVENTS in ECS 660

Friday, June 22nd, 2012

No class Friday June 22.
Please attend Turing Celebration Events instead

9:00-9:30	The Life of Alan Turing - **Daniel German**
9:30-11:00	Turing's Seminal 1936 Paper - **John Ellis**
11:00-12:00	The history of the Entscheidungsproblem, from Llull to Turing and beyond - **Bruce Kapron**
12:00 – 1:30	Pizza Lunch, Turing Art, Music & Wumpus World
1:30-2:30	The Turing Test and Searle's "Chinese Room" - **Maarten van Emden**
2:30-3:30	The Enigma machine and Turing's Cryptoanalysis - **Venkatesh Srinivasan**
3:30-4:00	Birthday cake, coffee and tea with special guest **Olive Bailey** - worked with Turing at Bletchley Park
4:00-6:00	**"Breaking the Code"** - Acclaimed film about Turing's life – ECS 104

www.csc.uvic.ca/Events/turing100.htm

Sponsors:

PIMS

University of Victoria Computer Science

Outline: Chapter 3: Context-free grammars

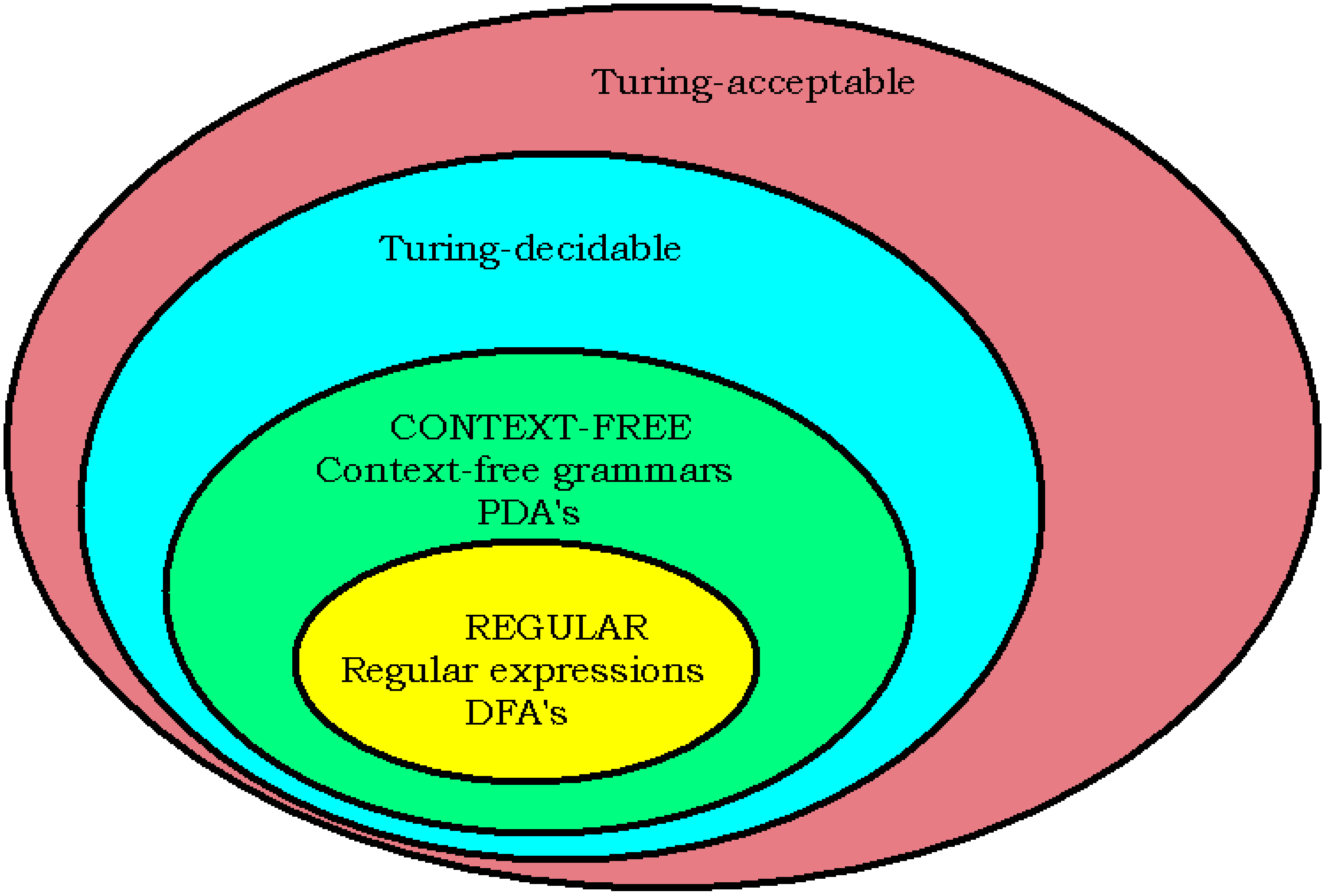
Context-free Grammars (CFG's)- used to specify valid syntax for programming languages, critical for compiling.

Pushdown Automata (PDA's)- machine model corresponding to context-free grammars, DFA with one stack.

All regular languages are context-free but not all context-free languages are regular.

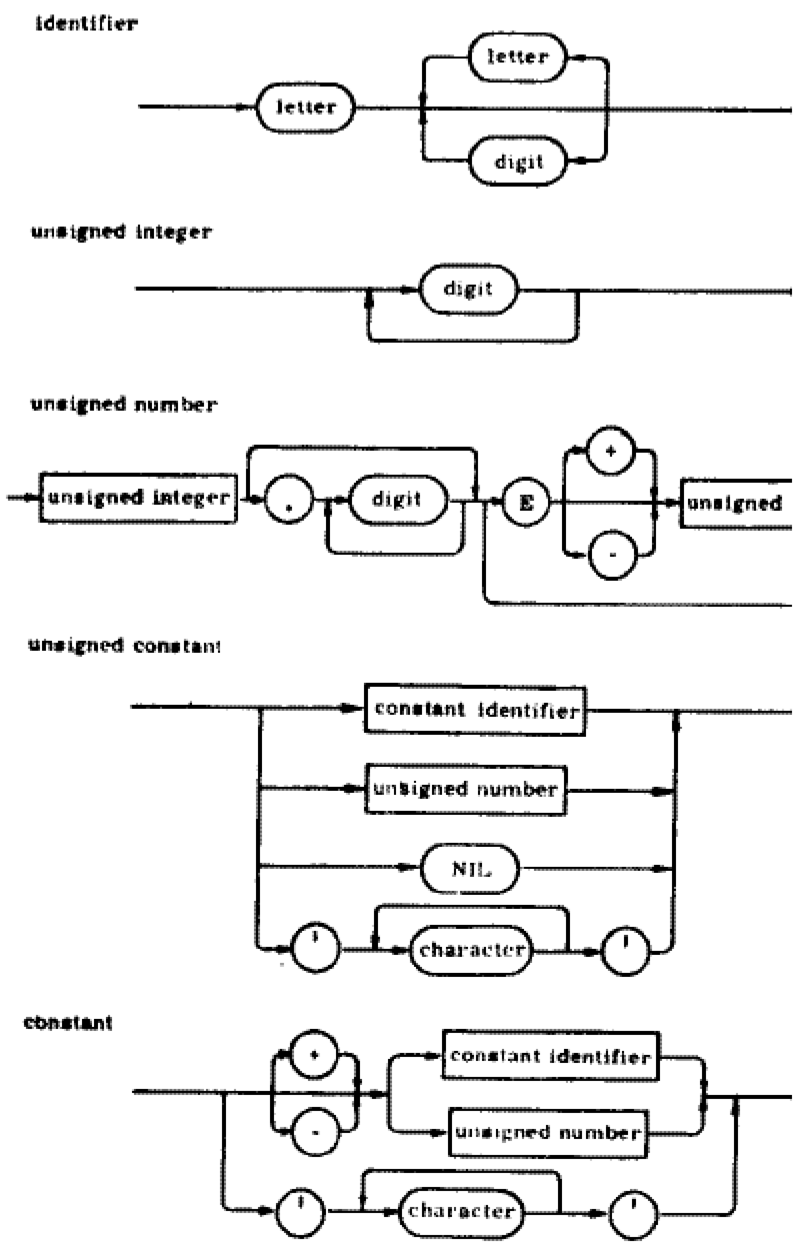
We will study closure properties, equivalence between languages specified by CFG's and PDA's, a pumping theorem and algorithmic questions.

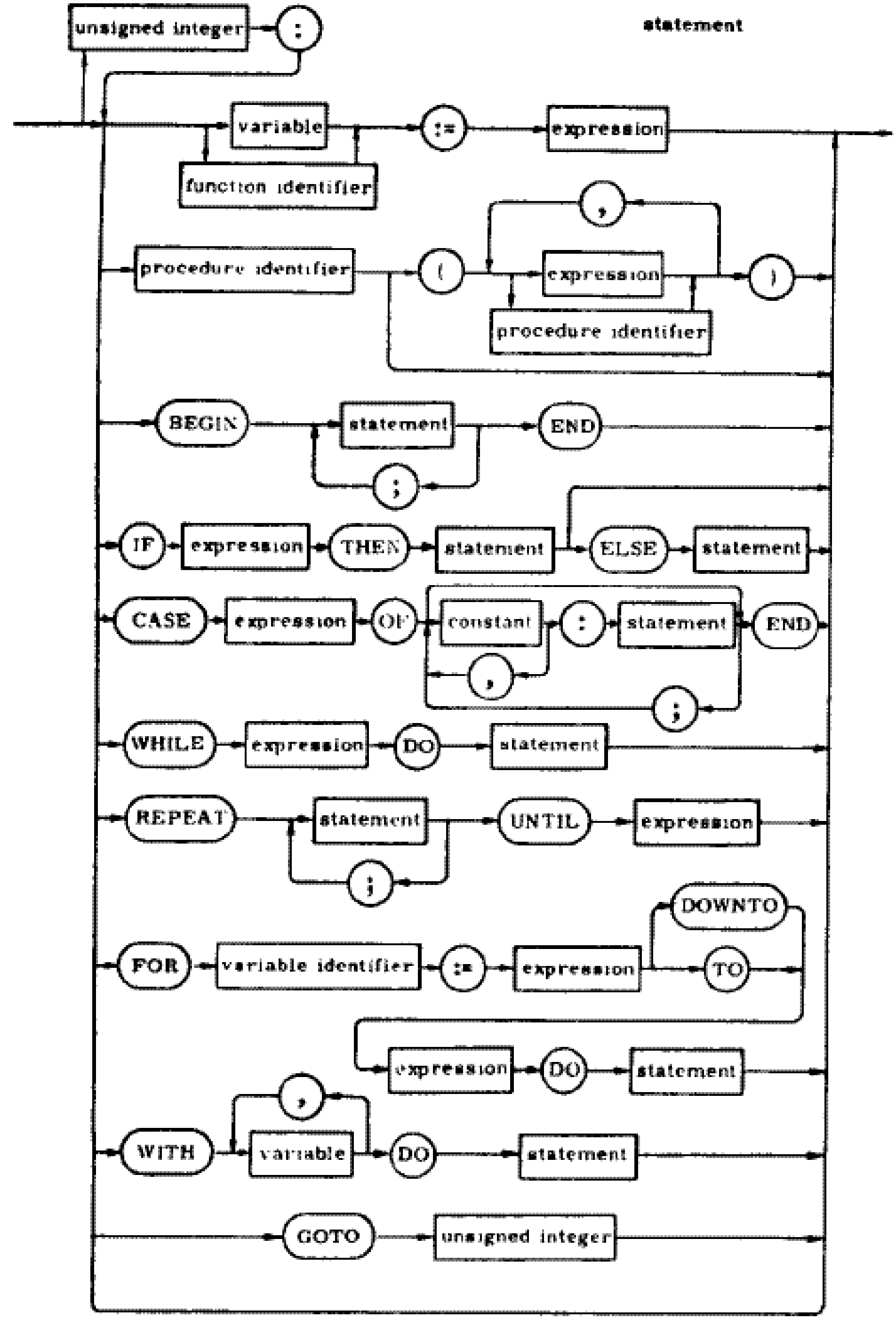
Classes of Languages



PASCAL

Part of the CFG for Pascal





Example 1:

$L = \{ a^n cc\, b^n : n \geq 0 \}$

Example 2:

$L = (aa \cup aba)^* \, bb \, (bb \cup bab)^*$

Example 3:

$L = a\, a^* \, b\, b^* \, c\, c^*$

Notation

a, b, c	lower case	terminals
A, B, S	upper case	non-terminals
e	often reserved for empty string	

Strings in the generated language consist of terminals only.

$\rightarrow$ Used to describe rules.

$\Rightarrow$ Used for derivations.

$\Rightarrow^*$ Derives in zero or more steps.

Example: $L = \{ w w^R : w \in \{a, b\}^* \}$

Rules: Start symbol S

$S \to aSa$

$S \to bSb$

$S \to \varepsilon$

A derivation:

$S \Rightarrow bSb \Rightarrow bbSbb \Rightarrow bbaSabb \Rightarrow$ bbaabb

Shorthand:

$S \overset{*}{\Rightarrow} bbaabb$

A context-free grammar G is a quadruple

(V, Σ, R, S) where

V is an alphabet,

Σ (the set of terminals) is a subset of V,

R (the set of rules) is a finite subset of

$(V- \Sigma) \times V^*$, and

S (the start symbol) is an element of $(V-\Sigma)$.

Elements of $(V- \Sigma)$ are called non-terminals.

If (A, u) is in R, we write $A \to u$

(A can be replaced by u).

If $u, v \in V^*$, then $u \Rightarrow v$ (u derives v) if and only if there are strings x, y and z in V^*

and a non-terminal A

such that $u = x A y$, $v = x z y$, and $A \to z$ is a rule of the grammar.

$L(G) = \{ w \in \Sigma^* \; : \; S \Rightarrow^* w$, S is the start symbol$\}$

Language L is context-free if it is L(G) for some context-free grammar G.

Prove the following language is context-free by designing a context-free grammar which generates it:

L= {w in $\{a,b\}^*$: the number of a's is even and the number of b's is even}

Context-free grammars and regular languages.

More examples of context-free languages.

All regular languages are context-free and a sub-class of context-free languages (those with regular context-free grammars) are regular.

Theorem:

Not all context-free languages are regular.

Proof:

$\{a^n b^n \; : \; n \geq 0\}$ is context-free but not regular.

Context-free grammar:

Start symbol S.

$S \rightarrow a\, S\, b$

$S \rightarrow \varepsilon$

Definition: A regular context-free grammar is a context-free grammar where each rule has its righthand side equal to an element of

Σ^* $(\{\varepsilon\} \cup (V - \Sigma))$

[0 or more terminals] then [at most one non-terminal]

Which rules below are not in the correct form to correspond to a regular context-free grammar?

1. $S \rightarrow A\ B$
2. $S \rightarrow B\ b$
3. $S \rightarrow a\ A$
4. $S \rightarrow aaaA$
5. $B \rightarrow b$
6. $B \rightarrow \varepsilon$
7. $S \rightarrow A\ S\ B$
8. $A \rightarrow a$
9. $A \rightarrow \varepsilon$
10. $A \rightarrow aaa\ bb$

[0 or more terminals] then [at most one non-terminal]

Theorem: If L is regular, then L is context-free.

Proof: A context-free grammar can be constructed from a DFA for L.

Definition: A regular context-free grammar is a context-free grammar where each rule has its righthand side equal to an element of

$$\Sigma^* \qquad\qquad (\{\varepsilon\} \cup (V - \Sigma))$$

[0 or more terminals] then [at most one non-terminal]

Our proof constructs a regular context-free grammar.

Create a NDFA which accepts the language generated by this context-free grammar. Start symbol: S

$S \rightarrow aa\ S$

$S \rightarrow \varepsilon$

$S \rightarrow M$

$M \rightarrow bbb$

$M \rightarrow ab\ M$

$M \rightarrow b\ S$

$M \rightarrow E$

$E \rightarrow aa$