

# CSE 105—Theory of Computability

Lecture 7—April 25, 2006  
Context-Free Grammars

Instructor: Neil Rhodes

# Proving a grammar generates a language

Prove all strings generated by the grammar are in the language

Prove all strings in the language are generated by the grammar

# Example

Language =  $\{w \in \{a,b\}^* \mid w \text{ has equal numbers of } a\text{'s and } b\text{'s}\}$

Grammar:  $S \rightarrow SS \mid aSb \mid bSa \mid \epsilon$

Prove all strings generated by the grammar are in the language

- By induction on the length of the derivation of  $e$  in  $G$

# Example

Language =  $\{w \in \{a,b\}^* \mid w \text{ has equal numbers of } a\text{'s and } b\text{'s}\}$

Grammar:  $S \rightarrow SS \mid aSb \mid bSa \mid \epsilon$

Prove all strings in the language are generated by the grammar

- By induction on the length of  $e \in L$

# Grammars for Programming Languages

Don't use entire power of CFG

Instead, use what's called LR(k) subset

- L means “read Left-to-right”
- R means “Rightmost derivation”
- k is number of symbols that must be looked ahead before deciding how to construct parse tree (usually is 1)

# Modifications of CFG

## BNF (Backus-Naur Form)

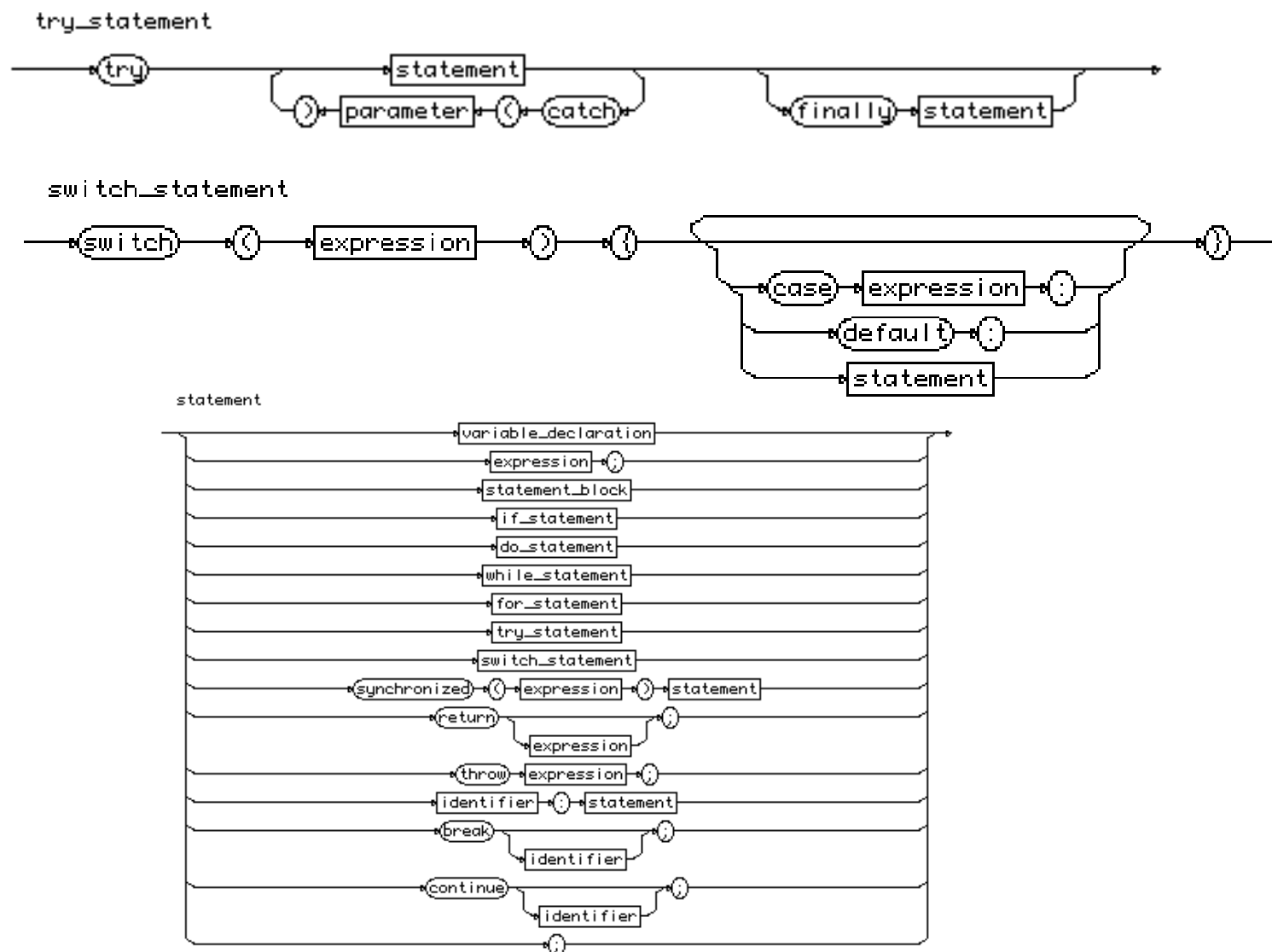
- Many variations
- | for or
- Optional parts (inside [])
- Kleene star (inside {})
- Quoted-characters or bold for non-terminals

```
try_statement
    ::=
    try statement
    { catch "(" parameter ")" statement }
    [ finally statement ]
```

```
switch_statement
    ::=
    switch "(" expression ")" "{"
    { ( case expression ":" )
    | ( default ":" )
    | statement }
    "}"
```

# Syntax diagrams

Use pictures to show syntax



# Chomsky Normal Form (CNF)

Every production looks like:

- $A \rightarrow BC$ 
  - S never appears on RHS
- $A \rightarrow x$ 
  - Can have  $S \rightarrow \epsilon$

Every CFG can be rewritten in CNF

Why do we care?

- Easier to build PushDown Automata (PDA) for constrained CNF than for any arbitrary CFG
- Gives an algorithm to tell whether a CFG generates a particular string
  
- Gives an algorithm to tell whether a CFG generates the empty language

# How to convert to CNF

4 things we need to do:

## Example conversion to CNF

$S \rightarrow XSX \mid aY$

$X \rightarrow Y \mid S$

$Y \rightarrow b \mid \epsilon$