Chapter 3

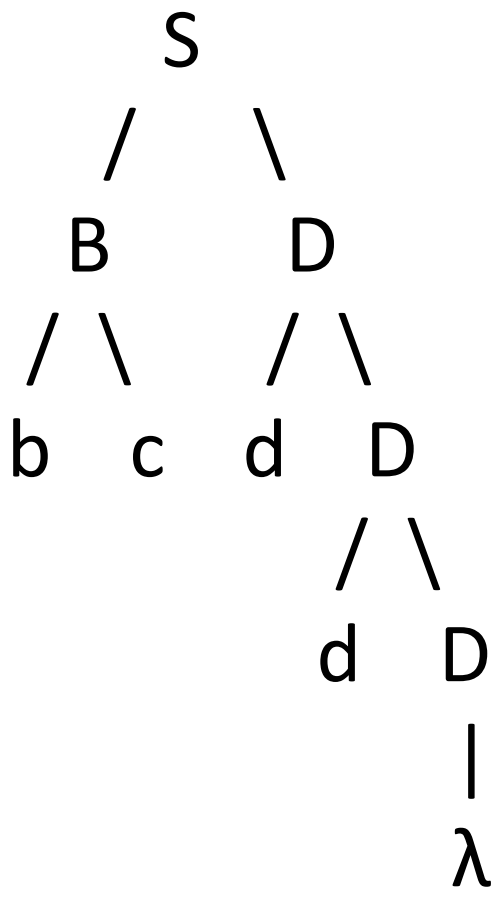
Context-free Grammars Part 2

Parse trees

S -> BD
B -> bc
D -> dD
D -> λ

S => BD => bcD => bcdD => bcddD
=> bcdd

Parse for preceding derivation



Leftmost derivation

S => BD => bcD => bcdD => bcdd

Rightmost derivation

S => BD => BdD => Bdd => bcdd

Substitution

S -> BC
B -> bB
B -> d
C -> c

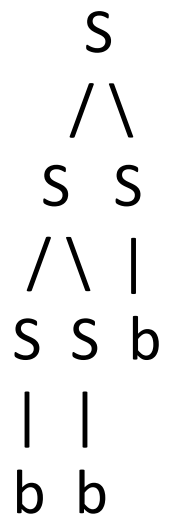
S -> bBC
S -> dC
B -> bB
B -> d
C -> c

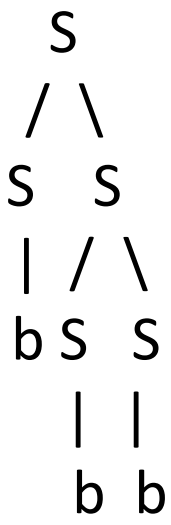
Ambiguous grammar

At least one string for which more that one derivation tree exists

S -> SS

S -> b





Determining nullable nonterminals

Check nonterminals that are in lambda productions.

Check nonterminals with rightside completely checked.

Continue checking until no more checking possible

Nullable nonterminal

```
          xx
1)  S → AB

2)  S → aCS
    x     x
3)  A → aA
    x
4)  A → λ
    x     x
5)  B → bB
    x
6)  B → λ

7)  C → c
```

(a)

```
    x     xx
1)  S → AB
    x       x
2)  S → aCS
    x     x
3)  A → aA
    x
4)  A → λ
    x     x
5)  B → bB
    x
6)  B → λ

7)  C → c
```

(b)

Eliminate lambda productions

S -> dBC

B -> bB

B -> λ

C -> cC

C -> λ

Add:

S -> dB, S -> dC, S -> d

B -> b

C -> c

Then delete lambda productions

Eliminate unit productions

S -> bB

B -> C (unit production)

C -> cc

Add

S -> bC

Then delete unit production

Useless nonterminal

Unreachable: unreachable from start symbol

Dead: cannot generate a terminal string

Eliminating useless nonterminals

S -> e

S -> BD

B -> b

D -> dD

Eliminate dead first (D is dead). Get

S -> e

B -> b

Eliminate unreachable second. Get

S -> e

Adding null string to a language

S -> bS

S -> c

Add new start symbol (S') and

S' -> S

S' -> λ

S' -> S