# CS 536 Announcements for Wednesday, February 28, 2024

**Midterm 1, Thursday, February 29, 7:30 – 9 pm, S429 Chemistry**

**Last Time**

- review for Midterm 1

**Today**

- approaches to parsing
- bottom-up parsing
- CFG transformations
  - removing useless non-terminals
  - Chomsky normal form (CNF)
- CYK algorithm

**Next Time**

- wrap up CYK
- classes of grammars
- top-down parsing

## Parsing: two approaches

**Top-down / "goal driven"**

- start at start nonterminal
- grow parse tree downward until entire sequence is matched

**Bottom-up / "data driven"**

- start with terminals (sequence)
- generate ever larger subtrees until get to single tree whose root is the start nonterminal

**Example:**

**CFG**: expr → expr + term | term

term → term * ID | ID

**Derive**: ID + ID

# Cocke – Younger – Kasami (CYK) algorithm

- Works bottom-up
- Time complexity : $O(n^3)$
- Requires grammar to be in Chomsky Normal Form

**Chomsky Normal Form (CNF)**

- all rules must be in one of two forms
  - $x \rightarrow \mathbf{T}$
  - $x \rightarrow a\ b$
- only rule allowed to derive epsilon is the start symbol *s*

**Why CNF is helpful?**

- nonterminals in pairs

- nonterminals (except start) can't derive epsilon

**CYK : Dynamic Programming**

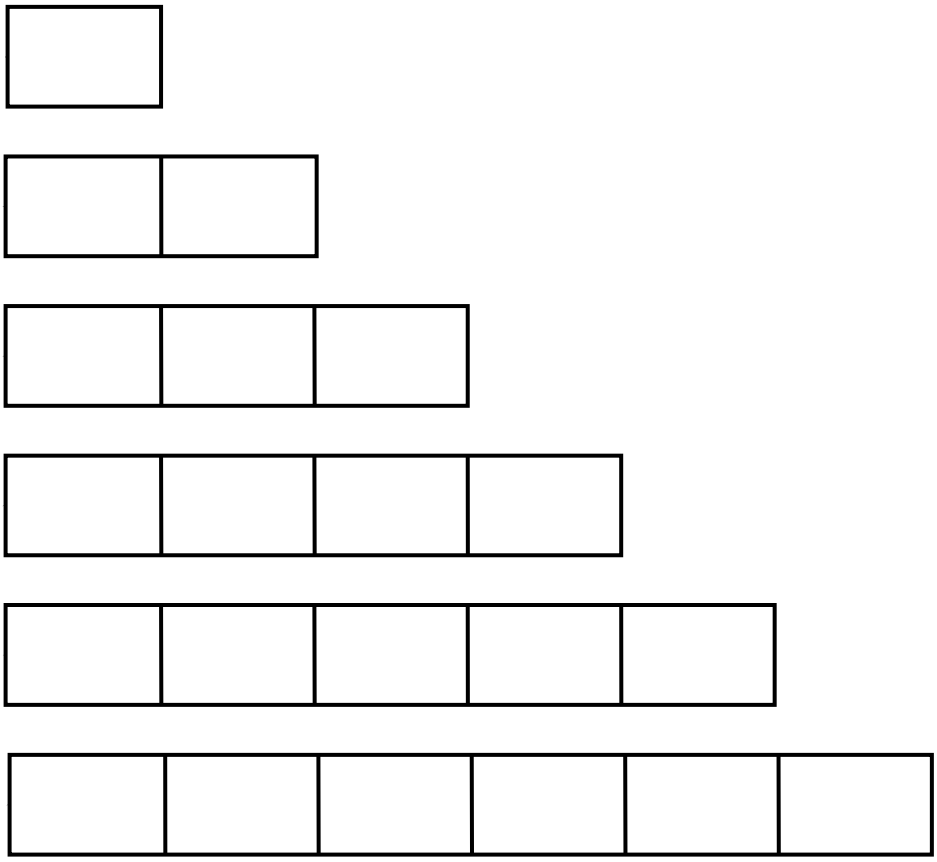
$x \rightarrow \mathbf{T}$

$x \rightarrow a\ b$

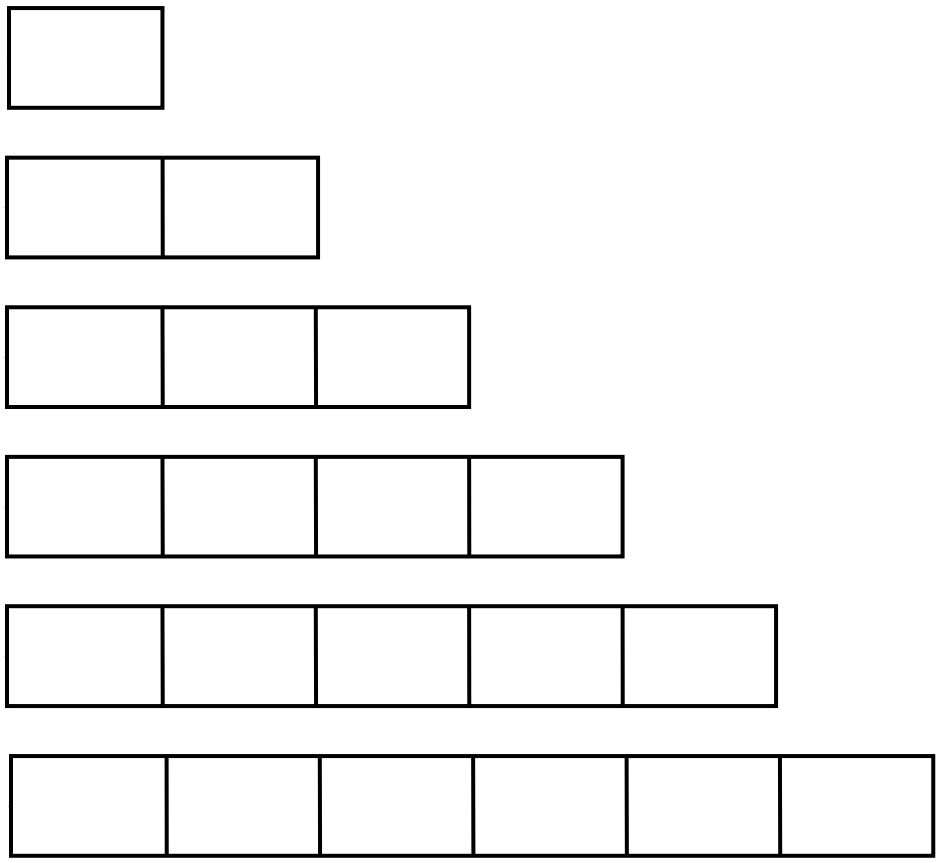
# Running CYK

Track every viable subtree from leaf to root.

All subspans for a sequence (string) with 6 terminals



# CYK Example



*f* → *i w*

*f* → *i y*

*w* → *l x*

*x* → *n r*

*y* → *l r*

*n* → **ID**

*n* → *i z*

*z* → *c n*

*i* → **ID**

*l* → **(**

*r* → **)**

*c* → **,**

# Eliminating useless nonterminals

**Avoid unnecessary work** – remove ***useless*** rules

1. If a nonterminal cannot derive a sequence of terminal symbols, then it is ***useless***
2. If a nonterminal cannot be derived from the start symbol, then it is ***useless***

**Nonterminals that cannot derive a sequence of terminal symbols**

```
mark all terminal symbols
repeat
     if all symbols on the RHS of a production are marked
         mark the LHS nonterminal
until no more nonterminals can be marked
```

Example

*s* → *x* | *y*
*x* → **( )**
*y* → **(** *y y* **)**

**Nonterminals that cannot be derived from the start symbol**

```
mark the start symbol
repeat
     if the LHS of a production is marked
         mark all RHS nonterminals
until no more nonterminals can be marked
```

Example

*s* → *a b*
*a* → **+** | **–** | ε
*b* → **digit** | *b* **digit**
*c* → **.** *b*

# Chomsky Normal Form

**Four steps**

- eliminate epsilon productions
- eliminate unit productions
- fix productions with terminal on RHS
- fix productions with > 2 nonterminals on RHS

**Eliminate (most) epsilon productions**

If nonterminal *a* immediately derives epsilon

Example 1

*f* → **ID ( *a* )**

*a* → ε

*a* → *n*

*n* → **ID**

*n* → **ID , *n***

Example 2

*x* → *a* **X** *a* **Y** *a*

*a* → ε

*a* → **Z**

# Chomsky Normal Form (cont.)

**Eliminate unit productions**

Productions of the form *a* → *b* are called ***unit productions***

Example

- *f* → **ID ( *a* )**
- *f* → **ID ( )**
- *a* → *n*
- *n* → **ID**
- *n* → **ID ,** *n*

# Chomsky Normal Form (cont.)

**Fix RHS nonterminals**

For productions with terminals and something else on the RHS

Example

- *f* → **ID (** *n* **)**
- *f* → **ID ( )**
- *n* → **ID**
- *n* → **ID ,** *n*

For productions with > 2 nonterminals on the RHS

Example