Applications of Regular Closure

The intersection of

a context-free language and

a regular language

is a context-free language

Regular Closure

L_1 context free

L_2 regular

$$\left.\begin{array}{l} L_1 \\ \\ L_2 \end{array}\right\} \Rightarrow L_1 \cap L_2$$

context-free

Linz 6th, section 8.2, example 8.7, page 227

L={a^n b^n | 0≤n, n≠100}

is context free

An Application of Regular Closure

Prove that: $L = \{a^n b^n : \ \ n \neq 100\}$

is context-free

We know: $\{a^n b^n\}$

is context-free

We also know:

$$L_1 = \{a^{100}b^{100}\}$$ is regular

$$\Downarrow$$

$$\overline{L_1} = \{(a+b)^*\} - \{a^{100}b^{100}\}$$ is regular

$\{a^n b^n\}$ context-free

$\overline{L_1} = \{(a+b)^*\} - \{a^{100} b^{100}\}$ regular

(regular closure) $\{a^n b^n\} \cap \overline{L_1}$

is context-free

$$\{a^n b^n\} \cap \overline{L_1}$$

$$= \{a^n b^n : \ n \neq 100\} = L$$ is context-free

Linz 6th, section 8.2, example 8.8, page 227

$L=\{w \mid \#_a(w) = \#_b(w) = \#_c(w)\}$

is not context free

Another Application of Regular Closure

Prove that: $L = \{w: \ n_a = n_b = n_c\}$

is **not** context-free

If $L = \{w: \;\; n_a = n_b = n_c\}$ is context-free

(regular closure)

Then $L \cap \{a^*b^*c^*\} = \{a^n b^n c^n\}$

context-free regular context-free

Impossible!!!

Therefore, L is **not** context free

Decidable Properties
of
Context-Free Languages

Membership Question:

for context-free grammar G

find if string $w \in L(G)$

Membership Algorithms: Parsers

- Exhaustive search parser
- **CYK** parsing algorithm

Empty Language Question:

for context-free grammar G

find if $L(G) = \emptyset$

Algorithm:

1. Remove useless variables

2. Check if start variable S is useless

Infinite Language Question:

for context-free grammar G

find if $L(G)$ is infinite

Algorithm:

1. Remove useless variables

2. Remove unit and λ productions

3. Create dependency graph for variables

4. If there is a loop in the dependency graph then the language is infinite

Example:

$$S \to AB$$

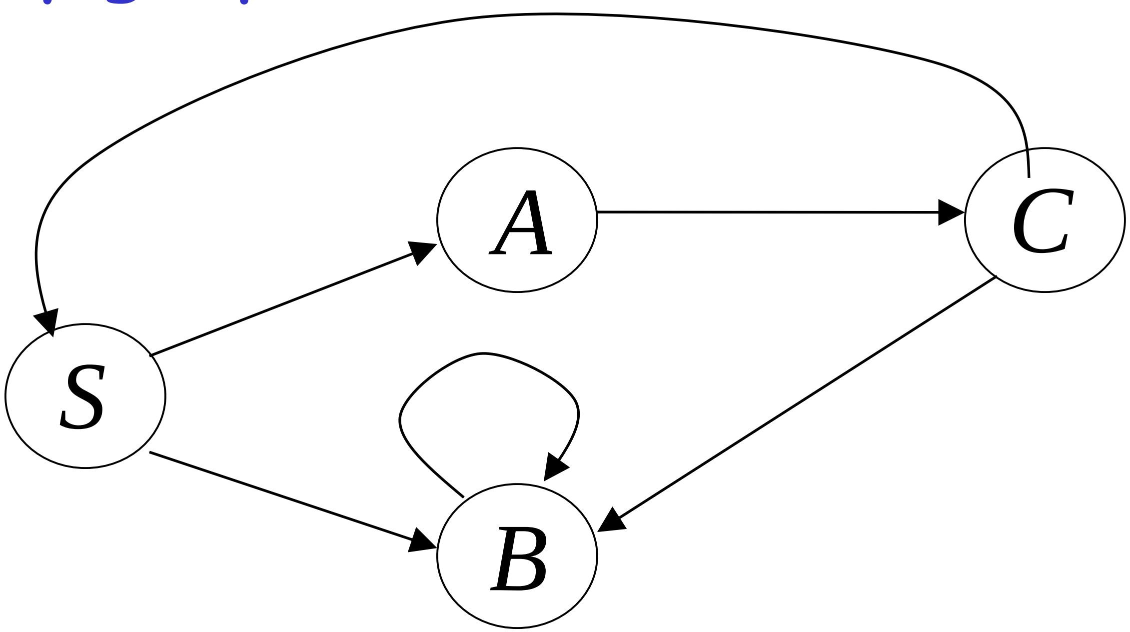
$$A \to aCb \mid a$$

$$B \to bB \mid bb$$

$$C \to cBS$$

Dependency graph

Infinite language



$$S \to AB$$

$$A \to aCb \mid a$$

$$B \to bB \mid bb$$

$$C \to cBS$$

$S \Rightarrow AB \Rightarrow aB \Rightarrow abB \Rightarrow ab^iB \Rightarrow ab^ibb$

$$S \to AB$$

$$A \to aCb \mid a$$

$$B \to bB \mid bb$$

$$C \to cBS$$

$$S \Rightarrow AB \Rightarrow aCbB \Rightarrow acBSbB \Rightarrow acbbSbbb$$

$$S \overset{*}{\Rightarrow} acbbSbbb \overset{*}{\Rightarrow} (acbb)^2 S (bbb)^2$$

$$\overset{*}{\Rightarrow} (acbb)^i S (bbb)^i$$

There is no algorithm to determine whether two context-free grammars generate the same language.

For the moment we do not have the technical machinery for defining the meaning of "there is no algorithm".