

Positive Properties of Context-Free languages

Union

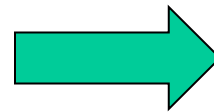
Context-free languages

are closed under:

Union

L_1 is context free

L_2 is context free



$L_1 \cup L_2$

is context-free

Example

Language

Grammar

$$L_1 = \{a^n b^n\}$$

$$S_1 \rightarrow aS_1b \mid \lambda$$

$$L_2 = \{ww^R\}$$

$$S_2 \rightarrow aS_2a \mid bS_2b \mid \lambda$$

Union

$$L = \{a^n b^n\} \cup \{ww^R\}$$

$$S \rightarrow S_1 \mid S_2$$

In general:

For context-free languages L_1, L_2
with context-free grammars G_1, G_2
and start variables S_1, S_2

The grammar of the **union** $L_1 \cup L_2$
has new start variable S
and additional production $S \rightarrow S_1 \mid S_2$

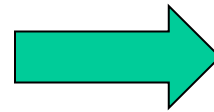
Concatenation

Context-free languages
are closed under:

Concatenation

L_1 is context free

L_2 is context free



L_1L_2

is context-free

Example

Language

Grammar

$$L_1 = \{a^n b^n\}$$

$$S_1 \rightarrow aS_1b \mid \lambda$$

$$L_2 = \{ww^R\}$$

$$S_2 \rightarrow aS_2a \mid bS_2b \mid \lambda$$

Concatenation

$$L = \{a^n b^n\} \{ww^R\}$$

$$S \rightarrow S_1 S_2$$

In general:

For context-free languages L_1, L_2
with context-free grammars G_1, G_2
and start variables S_1, S_2

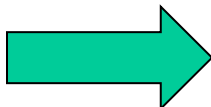
The grammar of the **concatenation** L_1L_2
has new start variable S
and additional production $S \rightarrow S_1S_2$

Star Operation

Context-free languages

are closed under:

Star-operation

L is context free  L^* is context-free

Example

Language

$$L = \{a^n b^n\}$$

Grammar

$$S \rightarrow aSb \mid \lambda$$

Star Operation

$$L = \{a^n b^n\}^*$$

$$S_1 \rightarrow SS_1 \mid \lambda$$

In general:

For context-free language L
with context-free grammar G
and start variable S

The grammar of the **star operation** L^*
has new start variable S_1
and additional production $S_1 \rightarrow SS_1 \mid \lambda$

Negative Properties of Context-Free Languages

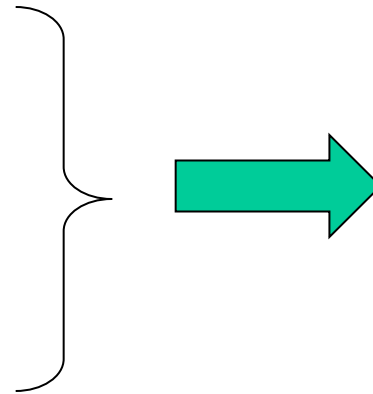
Intersection

Context-free languages
are not closed under:

intersection

L_1 is context free

L_2 is context free



$L_1 \cap L_2$

not necessarily
context-free

Example

$$L_1 = \{a^n b^n c^m\}$$

$$L_2 = \{a^n b^m c^m\}$$

Context-free:

$$S \rightarrow AC$$

$$A \rightarrow aAb \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

Context-free:

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \lambda$$

$$B \rightarrow bBc \mid \lambda$$

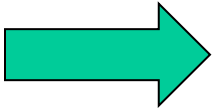
Intersection

$$L_1 \cap L_2 = \{a^n b^n c^n\} \quad \text{NOT context-free}$$

Complement

Context-free languages
are not closed under:

complement

L is context free  \bar{L} not necessarily
context-free

Example

$$L_1 = \{a^n b^n c^m\}$$

$$L_2 = \{a^n b^m c^m\}$$

Context-free:

$$S \rightarrow AC$$

$$A \rightarrow aAb \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

Context-free:

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \lambda$$

$$B \rightarrow bBc \mid \lambda$$

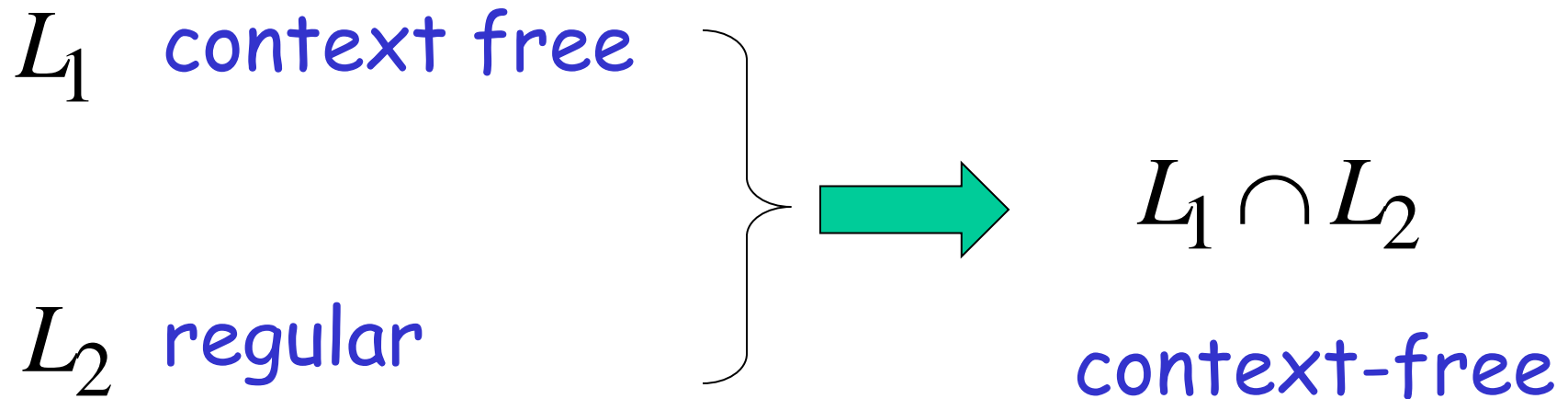
Complement

$$\overline{\overline{L_1} \cup \overline{L_2}} = L_1 \cap L_2 = \{a^n b^n c^n\}$$

NOT context-free

Intersection
of
Context-free languages
and
Regular Languages

The intersection of
a context-free language and
a regular language
is a context-free language



Machine M_1

NPDA for L_1
context-free

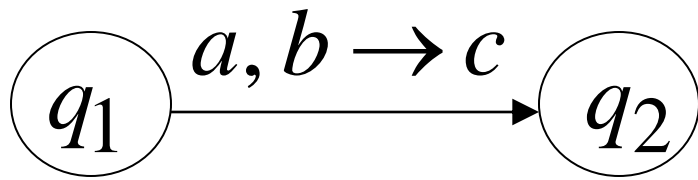
Machine M_2

DFA for L_2
regular

Construct a new NPDA machine M
that accepts $L_1 \cap L_2$

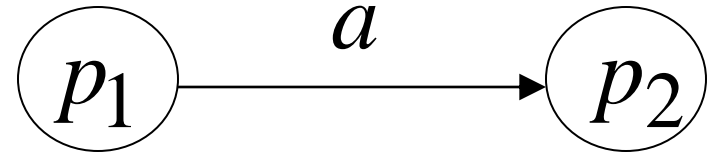
M simulates in parallel M_1 and M_2

NPDA M_1

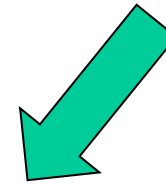
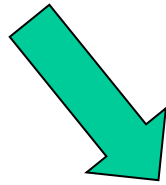


transition

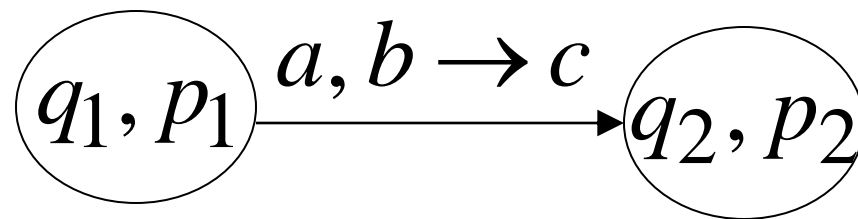
DFA M_2



transition

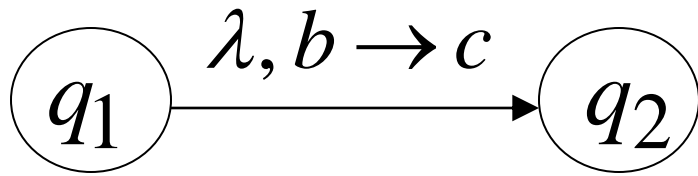


NPDA M



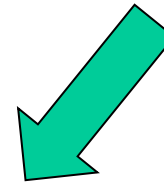
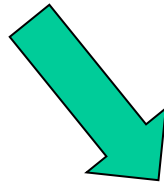
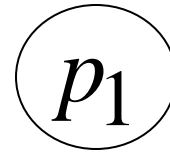
transition

NPDA M_1

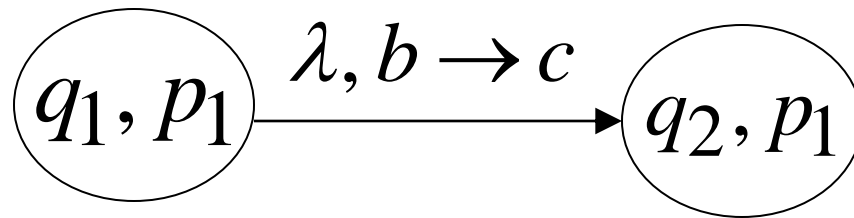


transition

DFA M_2

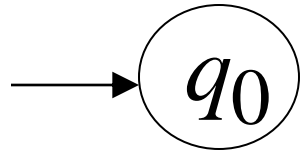


NPDA M



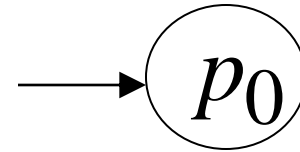
transition

NPDA M_1

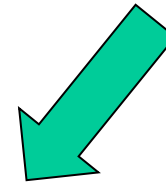


initial state

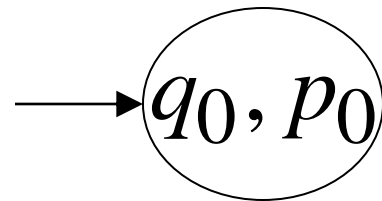
DFA M_2



initial state

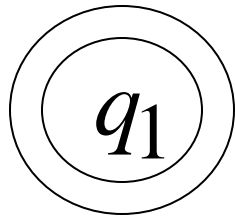


NPDA M



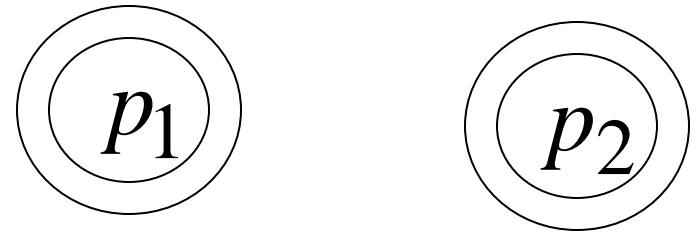
Initial state

NPDA M_1

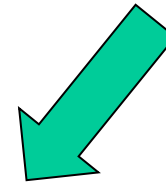


final state

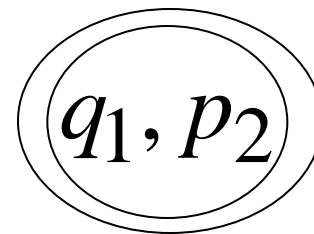
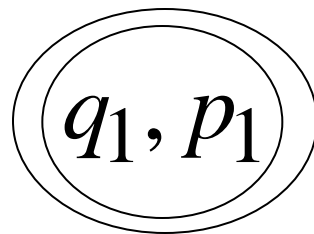
DFA M_2



final states



NPDA M



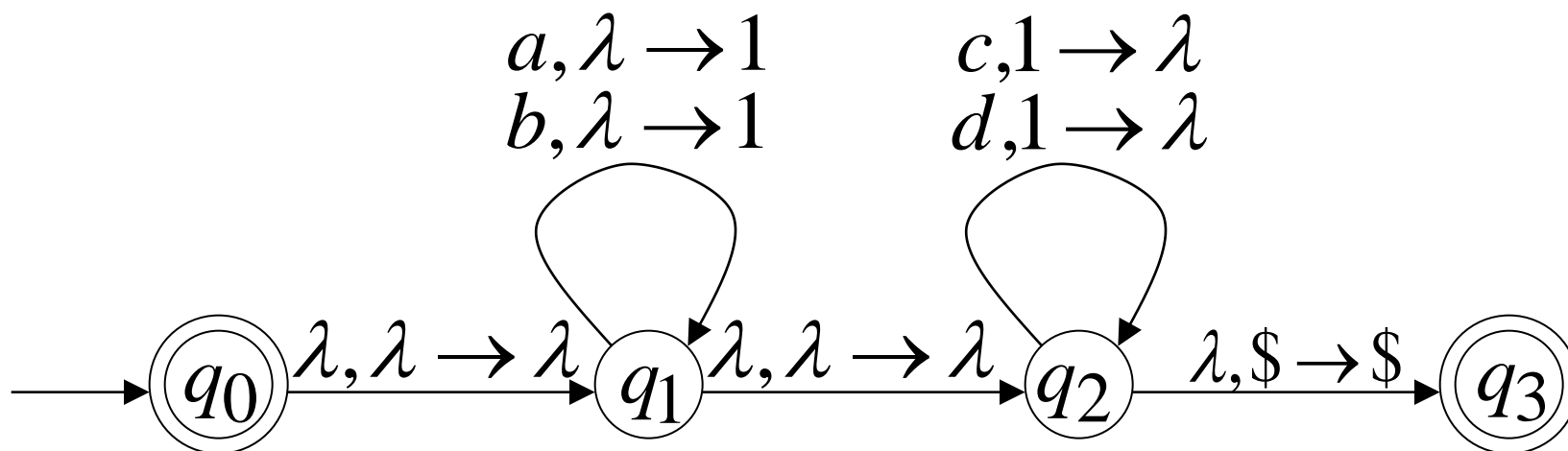
final states

Example:

context-free

$$L_1 = \{w_1w_2 : |w_1| = |w_2|, w_1 \in \{a,b\}^*, w_2 \in \{c,d\}^*\}$$

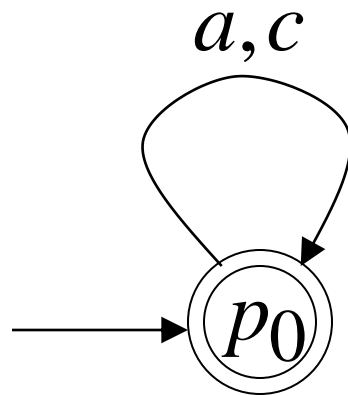
NPDA M_1



regular

$$L_2 = \{a, c\}^*$$

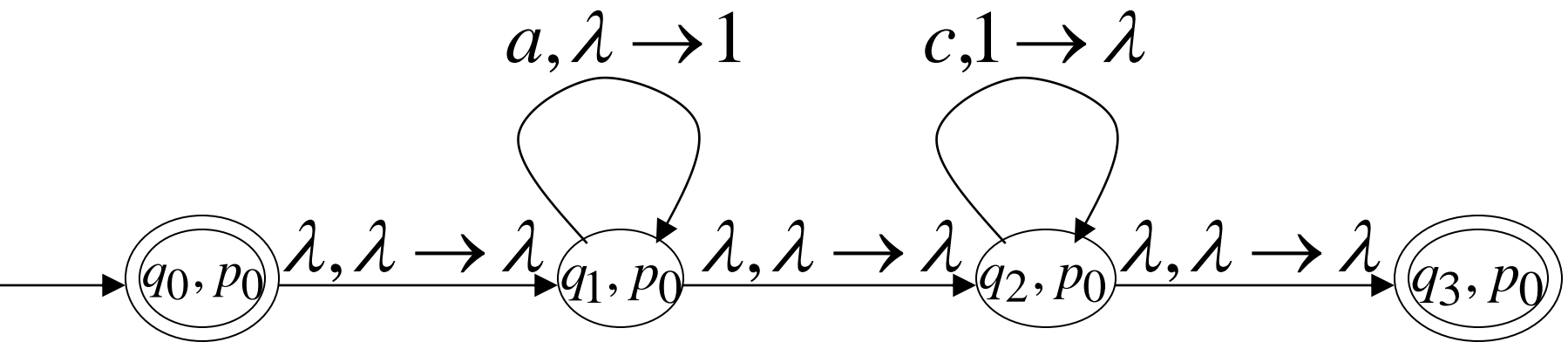
DFA M_2



context-free

Automaton for: $L_1 \cap L_2 = \{a^n c^n : n \geq 0\}$

NPDA M



In General:

M simulates in parallel M_1 and M_2

M accepts string w if and only if

M_1 accepts string w and

M_2 accepts string w

$$L(M) = L(M_1) \cap L(M_2)$$

Therefore:

M is NPDA



$L(M_1) \cap L(M_2)$ is context-free



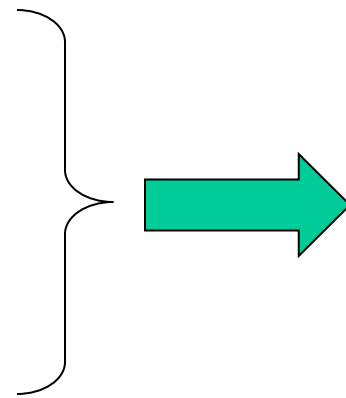
$L_1 \cap L_2$ is context-free

Applications of Regular Closure

The intersection of
a context-free language and
a regular language
is a context-free language

L_1 context free

L_2 regular



Regular Closure

$L_1 \cap L_2$

context-free

An Application of Regular Closure

Prove that: $L = \{a^n b^n : n \neq 100, n \geq 0\}$

is context-free

We know:

$\{a^n b^n : n \geq 0\}$ is context-free

We also know:

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



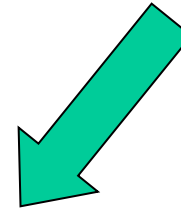
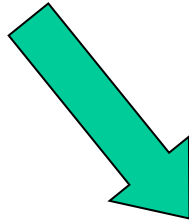
$\overline{L_1} = \{(a+b)^*\} - \{a^{100}b^{100}\}$ is regular

$$\{a^n b^n\}$$

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

context-free

regular



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



$$\{a^n b^n\} \cap \overline{L_1} = \{a^n b^n : n \neq 100, n \geq 0\} = L$$

is 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

CYK algorithm

- The Cocke-Younger-Kasami (CYK) algorithm determines whether a string can be generated by a given context-free grammar and, if so, how it can be generated. i.e. parsing the string.
- The standard version of CYK recognizes languages defined by context-free grammars written in Chomsky normal form (CNF).
- It is also possible to extend the CYK algorithm to handle some context-free grammars which are not written in CNF.

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: $S \rightarrow AB$

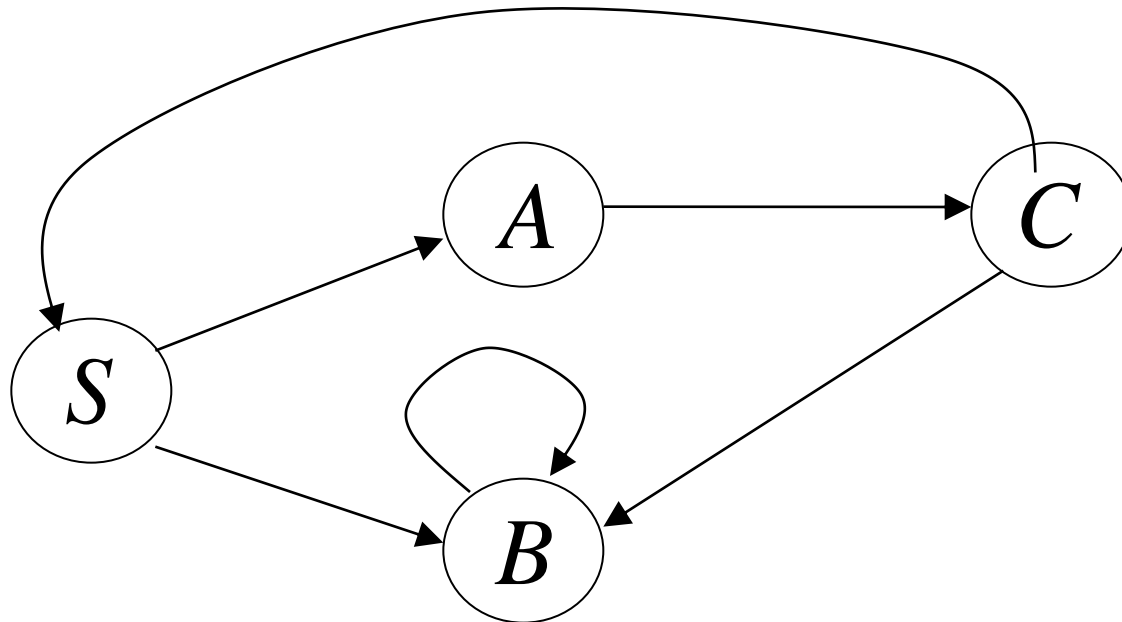
$A \rightarrow aCb \mid a$

$B \rightarrow bB \mid bb$

$C \rightarrow cBS$

Dependency graph

Infinite language



$$S \rightarrow AB$$

$$A \rightarrow aCb \mid a$$

$$B \rightarrow bB \mid bb$$

$$C \rightarrow cBS$$

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

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

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