CMSC 141 Automata and Language Theory

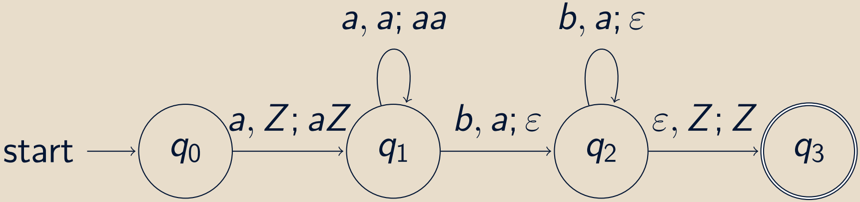
Context-Free Languages

Mark Froilan B. Tandoc

October 3, 2014

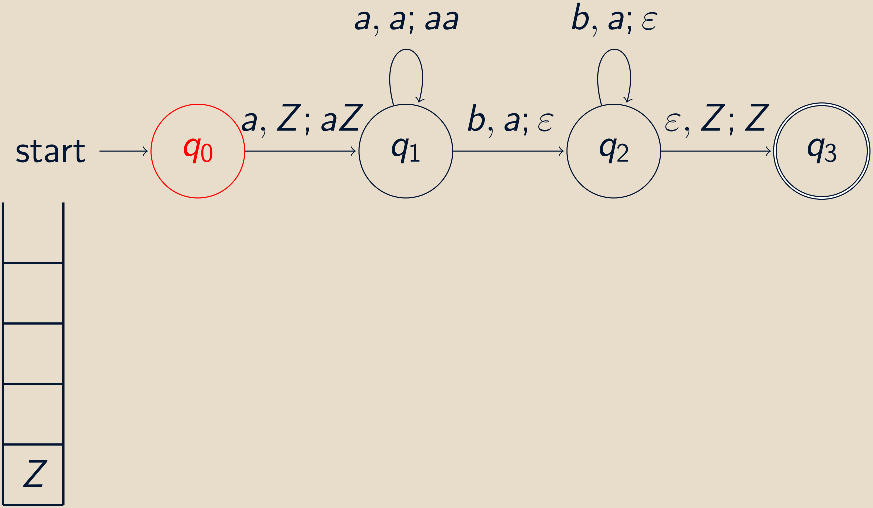
PDA FOR $\{a^n b^n : n > 0\}$

aabb



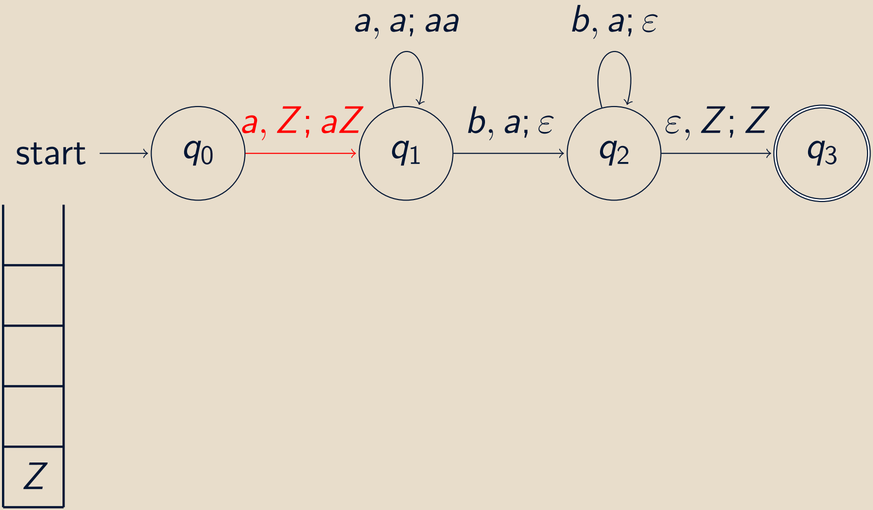
PDA for $\{a^n b^n : n > 0\}$

aabb



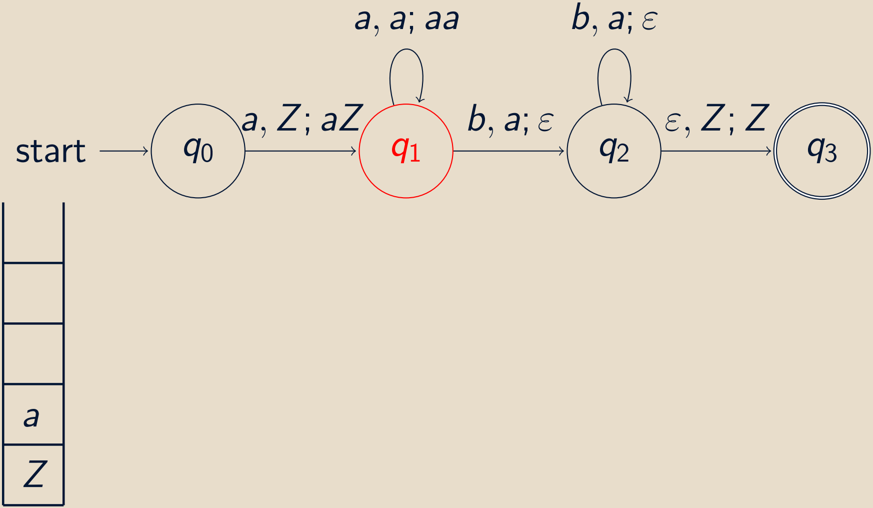
PDA for $\{a^n b^n : n > 0\}$

aabb



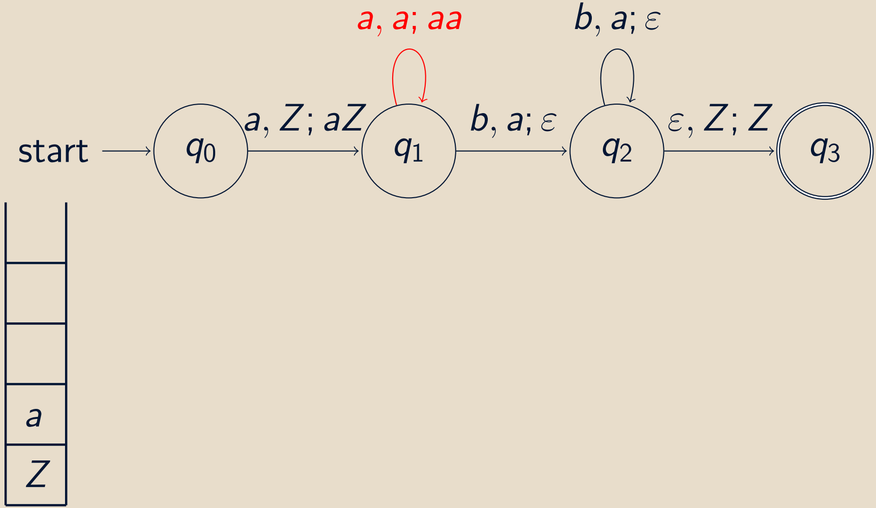
PDA for $\{a^n b^n : n > 0\}$

aabb



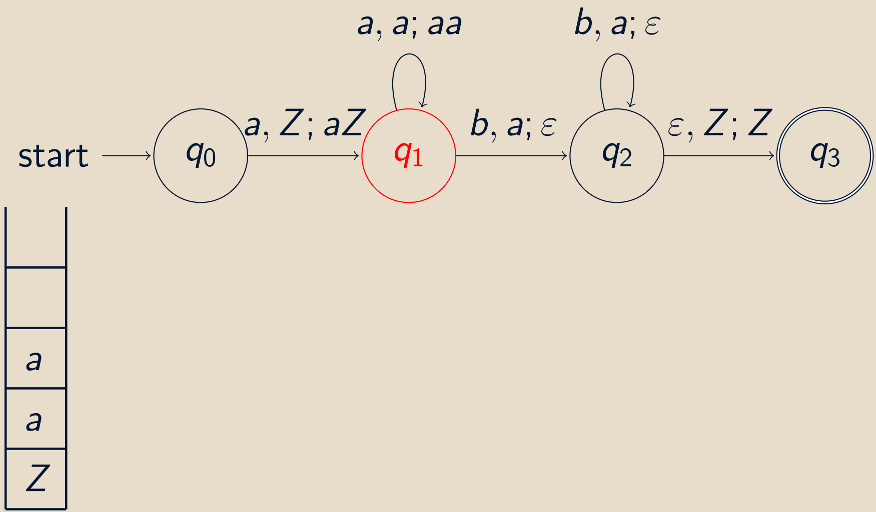
PDA for $\{a^n b^n : n > 0\}$

aabb



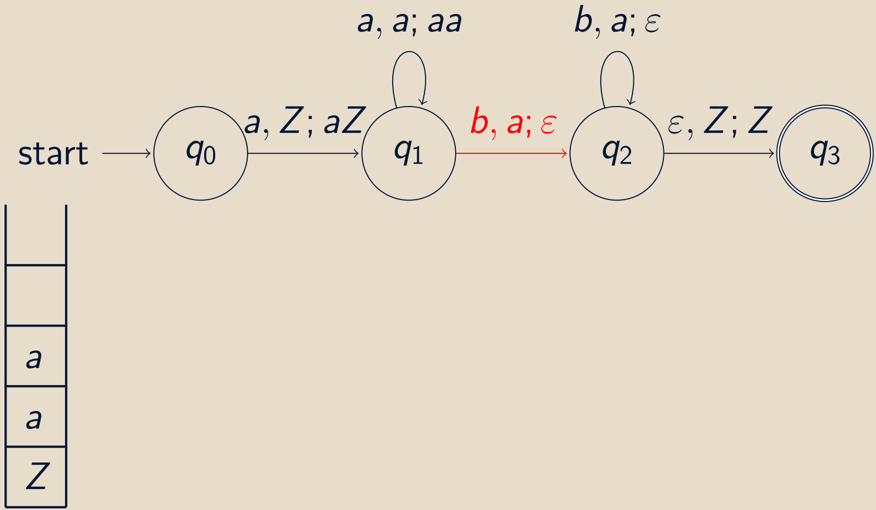
PDA FOR $\{a^n b^n : n > 0\}$

aabb



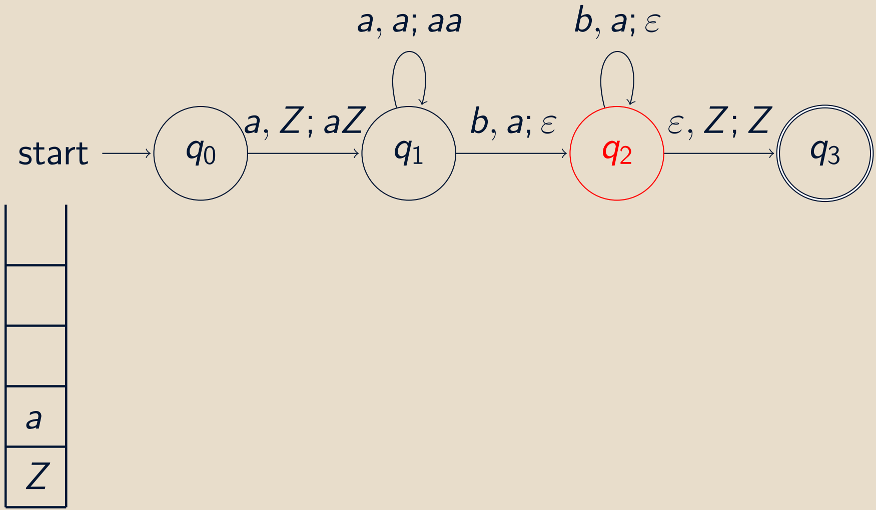
PDA for $\{a^n b^n : n > 0\}$

aabb



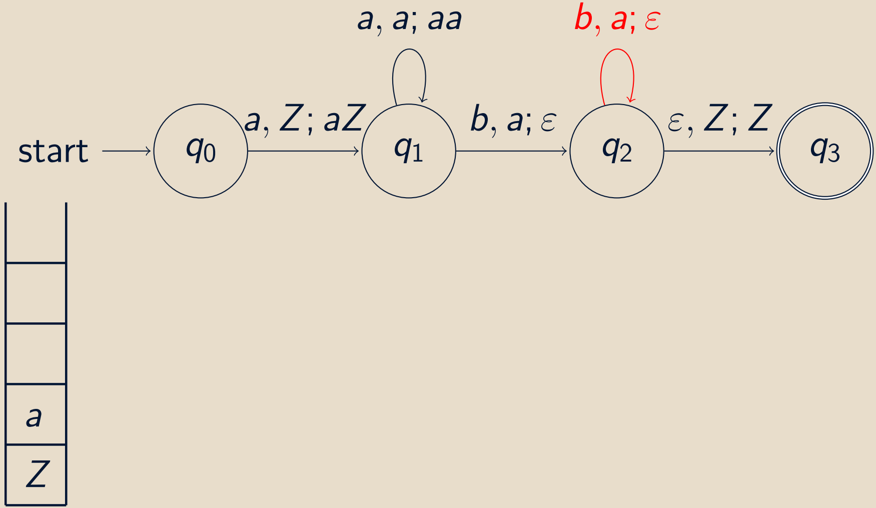
PDA for $\{a^n b^n : n > 0\}$

aabb



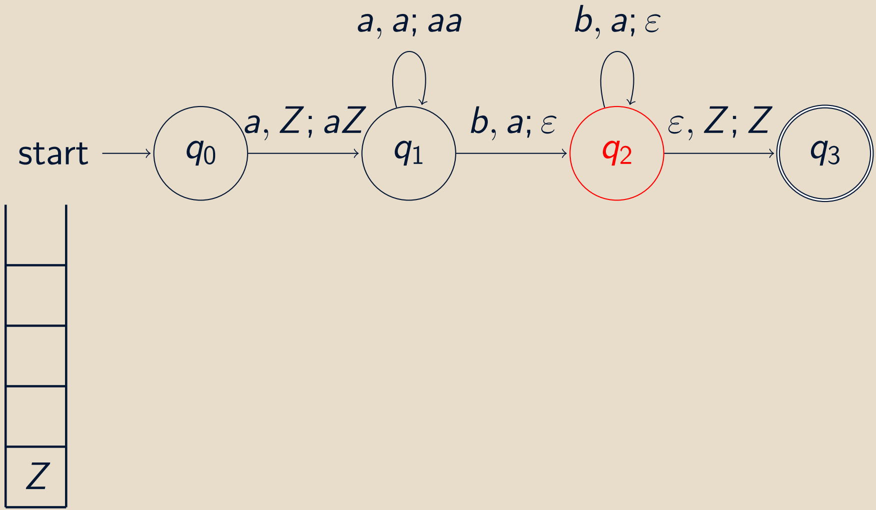
PDA for $\{a^n b^n : n > 0\}$

aabb



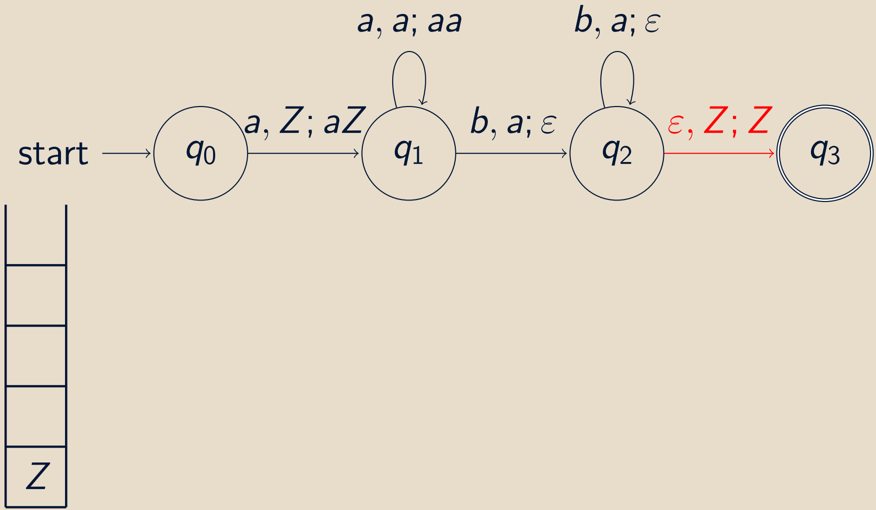
PDA for $\{a^n b^n : n > 0\}$

aabb



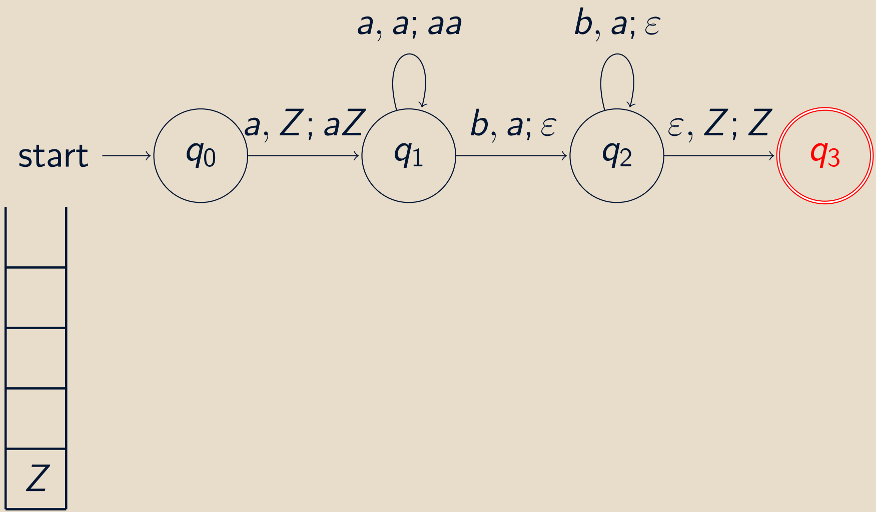
PDA for $\{a^n b^n : n > 0\}$

aabb



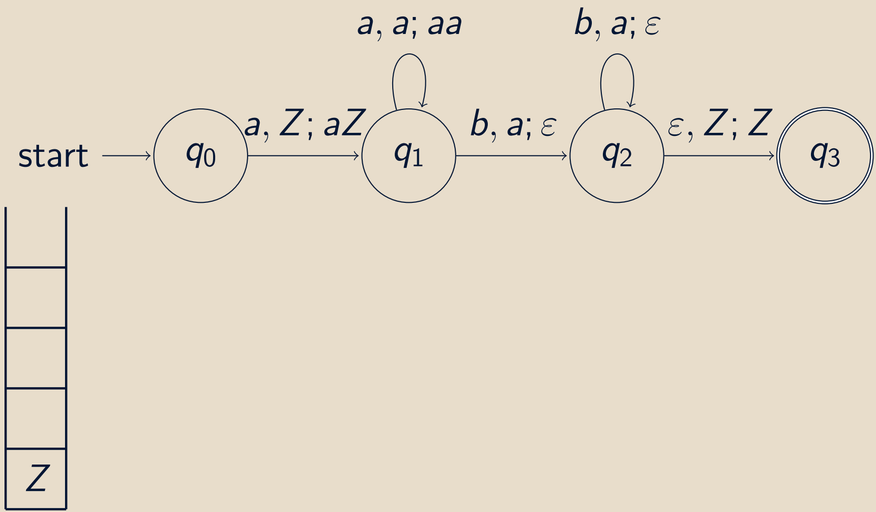
PDA for $\{a^n b^n : n > 0\}$

aabb



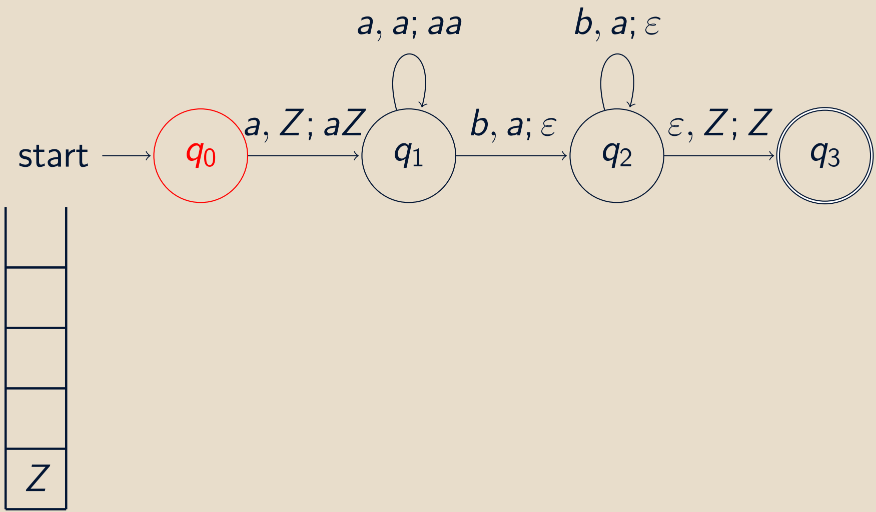
PDA for $\{a^n b^n : n > 0\}$

aab

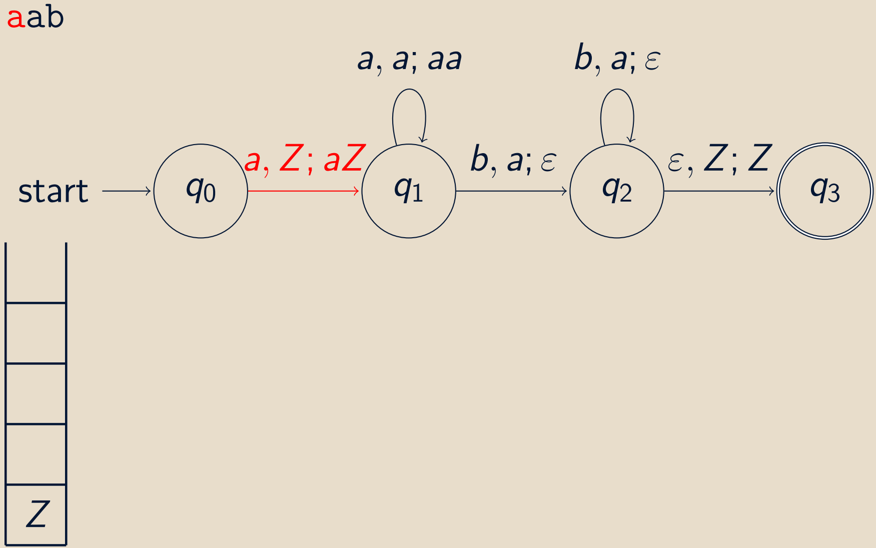


PDA for $\{a^n b^n : n > 0\}$

aab

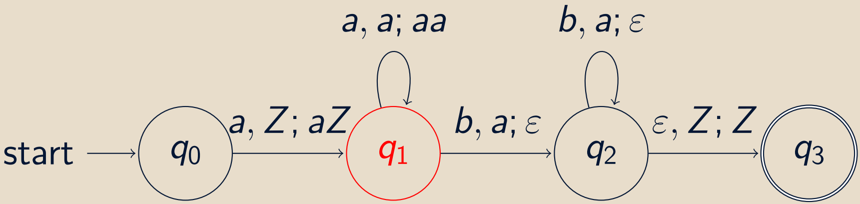


PDA for $\{a^n b^n : n > 0\}$



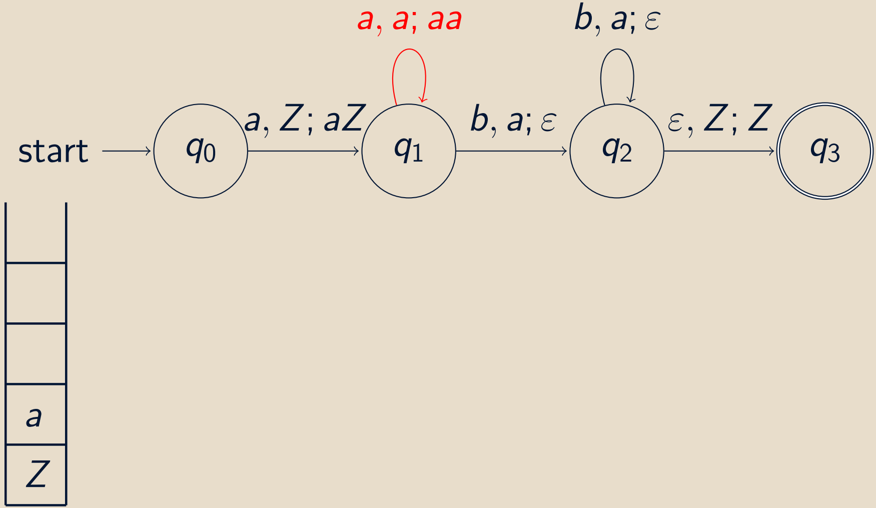
PDA FOR $\{a^n b^n : n > 0\}$

aab



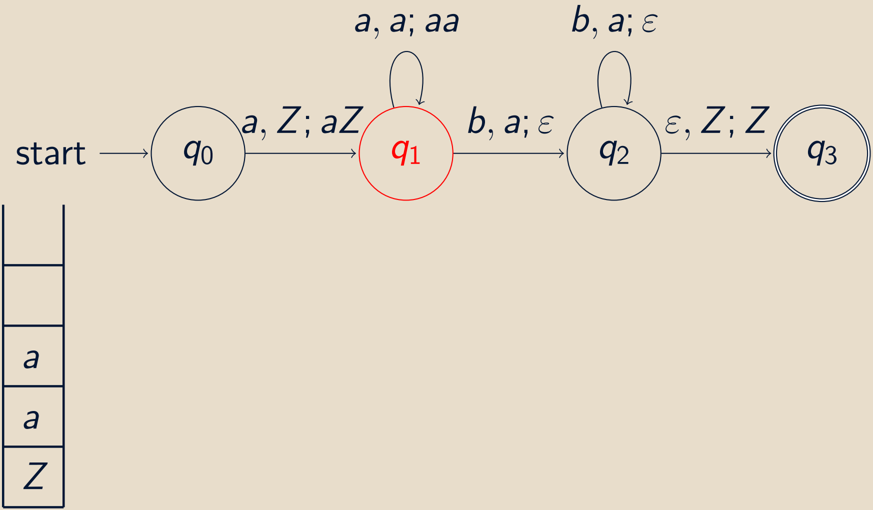
PDA for $\{a^n b^n : n > 0\}$

aab



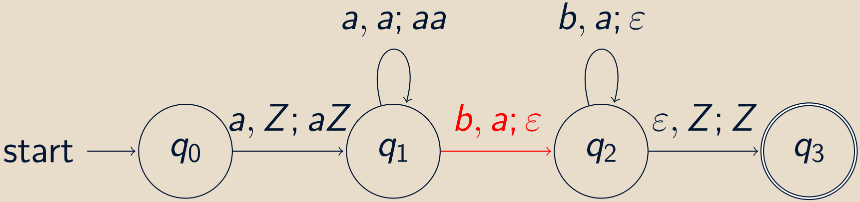
PDA for $\{a^n b^n : n > 0\}$

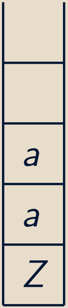
aab



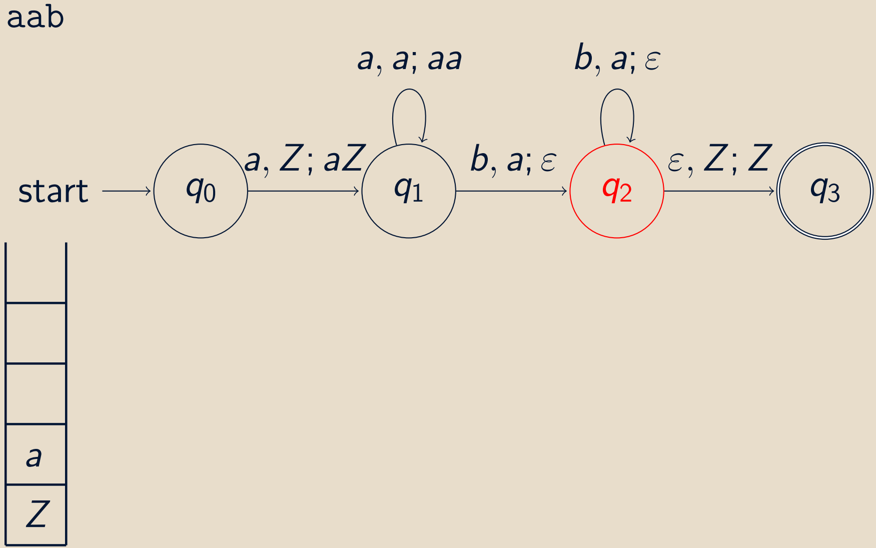
PDA for $\{a^n b^n : n > 0\}$

aab





PDA for $\{a^n b^n : n > 0\}$



Context-Free Grammars (CFG)

Context-Free Grammars (CFG)

- A grammar is a set of *string substitution rules* for producing a set of strings.

CONTEXT-FREE GRAMMARS (CFG)

- A grammar is a set of *string substitution rules* for producing a set of strings.
- Example: The context-free grammar that generates the language $\{a^n b^n : n > 0\} = \{ab, aabb, aaabbb, ...\}$ is shown below:

Context-Free Grammars (CFG)

- A grammar is a set of *string substitution rules* for producing a set of strings.
- Example: The context-free grammar that generates the language $\{a^n b^n : n > 0\} = \{ab, aabb, aaabbb, ...\}$ is shown below:

$$\begin{array}{lll} S & \rightarrow ab & \text{(base case)} \\ S & \rightarrow aSb & \text{(recursive rule)} \end{array}$$

Context-Free Grammars (CFG)

- A grammar is a set of *string substitution rules* for producing a set of strings.
- Example: The context-free grammar that generates the language $\{a^n b^n : n > 0\} = \{ab, aabb, aaabbb, ...\}$ is shown below:

$$\begin{aligned} S &\rightarrow ab && \text{(base case)} \\ S &\rightarrow aSb && \text{(recursive rule)} \end{aligned}$$

- The grammar can also be shorten by combining rules with the same left-hand side and using "|"
$S \rightarrow ab \mid aSb$

Derivation

Derivation

A string x can be derived from a grammar if x can be generated by successive applications of the production rules starting from the start symbol.

Example

Derivation

A string x can be derived from a grammar if x can be generated by successive applications of the production rules starting from the start symbol.

Example

Grammar:

$S \rightarrow ab$ (base case)

$S \rightarrow aSb$ (recursive rule)

Derivation

A string x can be derived from a grammar if x can be generated by successive applications of the production rules starting from the start symbol.

Example

Grammar:

$$\begin{array}{lll} S & \to ab & \text{(base case)} \\ S & \to aSb & \text{(recursive rule)} \end{array}$$

Derive: aaabbb

Derivation

A string x can be derived from a grammar if x can be generated by successive applications of the production rules starting from the start symbol.

Example

Grammar:

$S \rightarrow ab$ (base case)

$S \rightarrow aSb$ (recursive rule)

Derive: aaabbb

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaabbb$

Parse Trees (Derivation Tree)

Parse Trees (Derivation Tree)

A parse tree is a tree with the *start symbol as the root*, and the *target string forming the leaves* of the tree

Parse Trees (Derivation Tree)

A parse tree is a tree with the *start symbol as the root*, and the *target string forming the leaves* of the tree

Grammar: $S \rightarrow ab \mid aSb$

Parse Trees (Derivation Tree)

A parse tree is a tree with the *start symbol as the root*, and the *target string forming the leaves* of the tree

Grammar: $S \rightarrow ab \mid aSb$

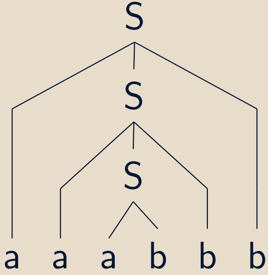
Derive: `aaabbb`

Parse Trees (Derivation Tree)

A parse tree is a tree with the *start symbol as the root*, and the *target string forming the leaves* of the tree

Grammar: $S \rightarrow ab \mid aSb$

Derive: `aaabbb`



Context-Free Languages

Context-Free Languages

- All strings that can be generated constitute the *language of the grammar*

Context-Free Languages

- All strings that can be generated constitute the *language of the grammar*
- We write $L(G)$ for the language of grammar G

Context-Free Languages

- All strings that can be generated constitute the *language of the grammar*
- We write $L(G)$ for the language of grammar G
- Any language that can be generated by some context-free grammar is called a *context-free language*

English Language Example

$$
\begin{aligned}
\text{SENTENCE} &\rightarrow \text{NOUN-PHRASE VERB-PHRASE} \\
\text{NOUN-PHRASE} &\rightarrow \text{CMPLX-NOUN} \\
&\rightarrow \text{CMPLX-NOUN PREP-PHRASE} \\
\text{VERB-PHRASE} &\rightarrow \text{CMPLX-VERB} \\
&\rightarrow \text{CMPLX-VERB PREP-PHRASE} \\
\text{PREP-PHRASE} &\rightarrow \text{PREP CMPLX-NOUN} \\
\text{CMPLX-NOUN} &\rightarrow \text{ARTICLE NOUN} \\
\text{CMPLX-VERB} &\rightarrow \text{VERB} \mid \text{VERB NOUN PHRASE} \\
\text{ARTICLE} &\rightarrow \text{a} \mid \text{the} \\
\text{NOUN} &\rightarrow \text{boy} \mid \text{girl} \mid \text{flower} \\
\text{VERB} &\rightarrow \text{touches} \mid \text{likes} \mid \text{sees} \\
\text{PREP} &\rightarrow \text{with}
\end{aligned}
$$

English Language Example

English Language Example

Sample strings we can derive from the grammar are:

- `a boy sees`
- `the boy sees a flower`
- `a girl with a flower likes the boy`

English Language Example

Sample strings we can derive from the grammar are:

- `a boy sees`
- `the boy sees a flower`
- `a girl with a flower likes the boy`

Try deriving them using the grammar

English Language Example

English Language Example

Derive: `a boy sees`

English Language Example

Derive: `a boy sees`

SENTENCE $\Rightarrow$ NOUN-PHRASE VERB-PHRASE

English Language Example

Derive: `a boy sees`

$$\begin{aligned}\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE}\\ &\Rightarrow \text{CMPLX-NOUN VERB-PHRASE}\end{aligned}$$

English Language Example

Derive: `a boy sees`

$$\begin{aligned} \text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE} \\ &\Rightarrow \text{CMPLX-NOUN VERB-PHRASE} \\ &\Rightarrow \text{ARTICLE NOUN VERB-PHRASE} \end{aligned}$$

English Language Example

Derive: `a boy sees`

$$\begin{aligned}\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE}\\ &\Rightarrow \text{CMPLX-NOUN VERB-PHRASE}\\ &\Rightarrow \text{ARTICLE NOUN VERB-PHRASE}\\ &\Rightarrow \text{a NOUN VERB-PHRASE}\end{aligned}$$

English Language Example

Derive: `a boy sees`

$$
\begin{aligned}
\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE} \\
&\Rightarrow \text{CMPLX-NOUN VERB-PHRASE} \\
&\Rightarrow \text{ARTICLE NOUN VERB-PHRASE} \\
&\Rightarrow \text{a NOUN VERB-PHRASE} \\
&\Rightarrow \text{a boy VERB-PHRASE}
\end{aligned}
$$

English Language Example

Derive: `a boy sees`

$$
\begin{aligned}
\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE} \\
&\Rightarrow \text{CMPLX-NOUN VERB-PHRASE} \\
&\Rightarrow \text{ARTICLE NOUN VERB-PHRASE} \\
&\Rightarrow \text{a NOUN VERB-PHRASE} \\
&\Rightarrow \text{a boy VERB-PHRASE} \\
&\Rightarrow \text{a boy CMPLX-VERB}
\end{aligned}
$$

English Language Example

Derive: `a boy sees`

$$
\begin{aligned}
\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE}\\
&\Rightarrow \text{CMPLX-NOUN VERB-PHRASE}\\
&\Rightarrow \text{ARTICLE NOUN VERB-PHRASE}\\
&\Rightarrow \text{a NOUN VERB-PHRASE}\\
&\Rightarrow \text{a boy VERB-PHRASE}\\
&\Rightarrow \text{a boy CMPLX-VERB}\\
&\Rightarrow \text{a boy VERB}
\end{aligned}
$$

English Language Example

Derive: `a boy sees`

$$
\begin{aligned}
\text{SENTENCE} &\Rightarrow \text{NOUN-PHRASE VERB-PHRASE} \\
&\Rightarrow \text{CMPLX-NOUN VERB-PHRASE} \\
&\Rightarrow \text{ARTICLE NOUN VERB-PHRASE} \\
&\Rightarrow \text{a NOUN VERB-PHRASE} \\
&\Rightarrow \text{a boy VERB-PHRASE} \\
&\Rightarrow \text{a boy CMPLX-VERB} \\
&\Rightarrow \text{a boy VERB} \\
&\Rightarrow \text{a boy sees}
\end{aligned}
$$

Formal Definition of CFG

Formal Definition of CFG

- A *context-free grammar* is a 4-tuple (V, Σ, R, S), where
  - V is a finite set of *variables* (or non-terminals)
  - Σ is a finite set of *terminals*
  - R is a finite set of *rules*
  - $S \in V$ is the start variable.

Formal Definition of CFG

- A *context-free grammar* is a 4-tuple (V, Σ, R, S), where
  - V is a finite set of *variables* (or non-terminals)
  - Σ is a finite set of *terminals*
  - R is a finite set of *rules*
  - $S \in V$ is the start variable.
- The rule for the *rules* is $V \to (V + T)^*$

Formal Definition of CFG

- A *context-free grammar* is a 4-tuple (V, Σ, R, S), where
  - V is a finite set of *variables* (or non-terminals)
  - Σ is a finite set of *terminals*
  - R is a finite set of *rules*
  - $S \in V$ is the start variable.
- The rule for the *rules* is $V \rightarrow (V + T)^*$
- Previous grammar is more formally defined as $G = (\{S\}, \{a, b\}), \{S \rightarrow ab, S \rightarrow aSb\}, S$

Chomsky Hierarchy

by Noam Chomsky

Chomsky Hierarchy

by Noam Chomsky
Containment hierarchy of classes of formal grammars

Chomsky Hierarchy

by Noam Chomsky

Containment hierarchy of classes of formal grammars

- Regular grammars (simplest, weakest)
  - $V \rightarrow T^*(V + \varepsilon)$

Chomsky Hierarchy

by Noam Chomsky

Containment hierarchy of classes of formal grammars

- Regular grammars (simplest, weakest)
    - $V \rightarrow T^*(V + \varepsilon)$
- Context-free grammars
    - $V \rightarrow (V + T)^*$

Chomsky Hierarchy

by Noam Chomsky

Containment hierarchy of classes of formal grammars

- Regular grammars (simplest, weakest)
  - $V \to T^*(V + \varepsilon)$
- Context-free grammars
  - $V \to (V + T)^*$
- Context-sensitive grammars

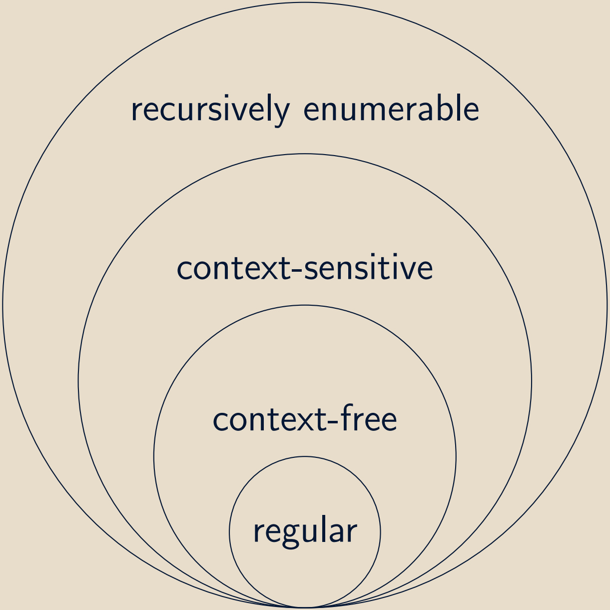
Chomsky Hierarchy

by Noam Chomsky
Containment hierarchy of classes of formal grammars

- Regular grammars
  (simplest, weakest)
  - $V \rightarrow T^*(V + \varepsilon)$
- Context-free grammars
  - $V \rightarrow (V + T)^*$
- Context-sensitive grammars
- Unrestricted grammars/Recursively enumerable grammars
  (most expressive)

Chomsky Hierarchy

Chomsky Hierarchy



References

- Previous slides on CMSC 141
- M. Sipser. Introduction to the Theory of Computation. Thomson, 2007.
- J.E. Hopcroft, R. Motwani and J.D. Ullman. Introduction to Automata Theory, Languages and Computation. 2nd ed, Addison-Wesley, 2001.
- E.A. Albacea. Automata, Formal Languages and Computations, UPLB Foundation, Inc. 2005
- JFLAP, www.jflap.org
- Various online LaTeX and Beamer tutorials