Language class	**Denotational tool**	**Computational model**
Regular languages	Regular expressions	DFAs and NFAs
Context-free languages	Context-free grammars	(N)PDAs

You have seen context free grammars before:

Mini-English grammar from DMFP:

$$Sentence \rightarrow NounPhrase\ Predicate\ PrepPhrase_{opt}$$

$$NounPhrase \rightarrow Article\ Adjective_{opt}\ Noun$$

$$Predicate \rightarrow Adverb_{opt}\ VerbPhrase$$

$$VerbPhrase \rightarrow \begin{cases} TransitiveVerb\ NounPhrase \\ IntransitiveVerb \\ LinkingVerb\ NounPhrase \end{cases}$$

$$PrepPhrase \rightarrow Preposition\ NounPhrase$$

Arithmetic expressions:

$$Expr \rightarrow Variable \mid Number \mid (Expr\ Op\ Expr)$$

$$Op \rightarrow + \mid - \mid * \mid \div$$

Appropriately nested strings of parentheses and brackets:

$$Expr \rightarrow \varepsilon \mid (Expr) \mid [Expr] \mid Expr\ Expr$$

A **context-free grammar** contains

- An alphabet Σ, the set of *terminal symbols*
- A set of non-terminal symbols
- Rules for expanding non-terminals
- A start symbol

(The book unites the terminal and non-terminal symbols into set V, which it calls the *alphabet*.)

All regular languages are context-free

- PDAs (§3.3) generalize NFAs
- Context-free languages are closed under union, concatenation, and Kleene star
- We can construct a CFG from a DFA

Not all context-free languages are regular

CFGs represent a strictly more powerful model than DFAs/NFAs.

Perspective: *We are taking DFAs, which have no memory, and equipping them with minimal memory*

Definition 3.3.1: A **pushdown automaton** is a sextuple $M = (K, \Sigma, \Gamma, \Delta, s, F)$, where

- K is a finite set of **states**
- Σ is an alphabet of **input symbols**
- Γ is a set of **stack symbols**
- $s \in K$ is the **initial state**
- $F \subseteq K$ is the set of **final states**
- Δ is the **transition relation**, a subset of

$$(K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma*) \times (K \times \Gamma*)$$

LP, pg 131

Ex 3.3.2.a. Construct a PDA to accept the language of strings with appropriately nested parenthesis and square brackets.

$K = \{s, q\}$, $\Gamma = \{(, [, b\}$ and $F = \{s\}$.

$$
\begin{array}{llllllll}
((& s, & (, & \varepsilon &), & (& q, & b(&)) \\
((& s, & [, & \varepsilon &), & (& q, & b[&)) \\
((& q, & (, & \varepsilon &), & (& q, & (&)) \\
((& q, & [, & \varepsilon &), & (& q, & [&)) \\
((& q, &], & b[&), & (& s, & \varepsilon &)) \\
((& q, &), & b(&), & (& s, & \varepsilon &)) \\
((& q, &], & [&), & (& q, & \varepsilon &)) \\
((& q, &), & (&), & (& q, & \varepsilon &))
\end{array}
$$

Ex 3.3.2.b Construct a PDA for the language of strings consisting in a certain number of occurrences of a followed by between as many and twice as many occurrences of b.

$$\{\mathtt{a}^m\mathtt{b}^n \mid m \leq n \leq 2m\}$$

$K = \{s, q, r, f\}$

$$\begin{array}{llllllll}
((& s, & a, & \varepsilon &), & (& q, & xaa &))\\
((& q, & a, & \varepsilon &), & (& q, & aa &))\\
((& q, & b, & aa &), & (& r, & \varepsilon &))\\
((& q, & b, & a &), & (& r, & \varepsilon &))\\
((& r, & b, & aa &), & (& r, & \varepsilon &))\\
((& r, & b, & a &), & (& r, & \varepsilon &))\\
((& r, & b, & xa &), & (& f, & \varepsilon &))\\
((& r, & b, & xaa &), & (& f, & \varepsilon &))
\end{array}$$

Main points of §3.(4 & 5)

Theorem 3.4.1: The class of languages accepted by *nondeterministic* pushdown automata equals the class of context-free languages.
(*Deterministic* pushdown automata are less powerful.)

Lemma 3.4.1: $CFG \subseteq PDA$. **Proof.** Construct a a PDA from a CFG.

Lemma 3.4.2: $PDA \subseteq CFG$. **Proof.** First simplify PDAs, then show the simplification doesn't change anything, then construct a CFG from a simplified PDA.

Some languages *aren't context-free*.

Theorem 3.5.1: CFGs are closed under union, concatenation, and Kleene star...

...but not under intersection or complementation.

LP, pg 136–139, 143

Lemma 3.5.1: The yield of any parse tree of G of height h has length at most $\phi(G)^h$.

Theorem 3.5.3: Let G be a context-free grammar. Then any string $w \in L(G)$ with length greater than $\phi(G)^{|V-\Sigma|}$ can be rewritten as $w = uvxyz$ in such a way that either v or y is nonempty and $uv^nxy^nz \in L(G)$ for every $n \geq 0$.

Define **fanout** of G, $\phi(G)$.

> **Proof outline.** *Imagine a parse tree. Each node has at most $\phi(G)$ children, so for height h, the length of the yielded string is at most $\phi(G)^h$.*
> *Context-free languages have a form of regularity:*
>
> $$u\ v^n\ x\ y^n\ z$$
>
> *So, contrapositively, if a language has no such regularity, it is not context free.*
> □