1. Let $P[i]$ be the least total penalty for traveling from the hotel (or home) at position i to the destination at position n. $P[0]$ is the entire problem. Then

$$P[i] = \begin{cases} 0 & \text{if } i = n \\ \min_{i<k\leq n}((200 - (a_k - a_i))^2 + P[k]) & \text{otherwise} \end{cases}$$

2. a.

$\forall\ L \in N$
$\quad \exists M \in DFA$
$\quad\quad |\ (\forall\ s \in L$
$\quad\quad\quad w \in L(M)$
$\quad\quad \text{and } (\forall w \in L(M)$
$\quad\quad\quad w \in L$

b.

$\forall\ L \in R$
$\quad \exists\ n \in \mathbb{W}$
$\quad\quad \forall\ w \in L,$
$\quad\quad\quad \text{if } |w| \geq n,$
$\quad\quad\quad\quad \exists\ x, y, z$
$\quad\quad\quad\quad\quad |\ w = xyz$
$\quad\quad\quad\quad\quad \text{and } \forall i \in \mathbb{W}$
$\quad\quad\quad\quad\quad\quad xy^iz \in L$

Date	Reading	Daily work problems
Fri, Nov 1	3.1 CFGs 3.2 Parse trees 3.3 PDAs	2.2.6. Make an NFA, convert to DFA
Mon, Nov 11	3.4 PDAs and CFGs 3.5 Languages not CF 4.1 Turing machines defined	3.3.2 Construct PDAs
Wed, Nov 13	4.2 Computing with TMs 4.3 Extensions to TMs 4.4 Random access TMs	4.1.1 Trace a TM computation
Fri, Nov 15	4.5 Nondeterministic TMs	4.5.1 Design an Nondet TM

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.

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

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

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

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.*
> $\square$

Turing machines

Criteria:

- They should be automata
- They should be as simple as possible to describe
- They should be as general as possible

The tape has a left end, but it extends indefinitely to the right.

LP pg 180

Formal definition:

A **Turing machine** is a quintuple $(K, \Sigma, \delta, s, H)$ where

- K is a finite set of states
- Σ is an alphabet, including $\sqcup$ (blank) and $\triangleright$ (left-end-of-tape), but not $\leftarrow$ or $\rightarrow$.
- $s \in K$ is the initial state
- $H \subseteq K$ is the set of halting states
- δ is the transition function from $(K - H) \times \Sigma$ to $K \times (\Sigma \cup \{\leftarrow, \rightarrow\})$
- For all $q \in K - H$, if $\delta(q, \triangleright) = (p, b)$, then $b = \rightarrow$
- For all $q \in K - H$ and $a \in \Sigma$, if $\delta(q, a) = (p, b)$, then $b \neq \triangleright$

Definition 4.1.2: Configuration:

$$K \times \triangleright\Sigma* \times (\Sigma*(\Sigma - \{\sqcup\}) \cup \{\varepsilon\})$$

Definition 4.1.3: $\vdash_M$ means *transition in one step* to a new state *and* either write, go left, or go right.

Definition 4.1.4:

- One configuration **yields** another: $C_0 \vdash_M^* C_2$
- A **computation** is a sequence of configurations
- A computation has **length** n or n **steps**, $C_0 \vdash_M^n C_n$.

Ex 4.1.1: $K = \{q_0, q_1, h\}, \Sigma = \{\mathtt{a}, \sqcup, \triangleright\}, s = q_0\}, H = \{h\}$

q	σ	$\delta(q,\sigma)$
q_0	$\mathtt{a}$	$(q_1, \sqcup)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$
q_1	$\mathtt{a}$	$(q_0, \mathtt{a})$
q_1	$\sqcup$	$(q_0, \rightarrow)$
q_1	$\triangleright$	$(q_1, \rightarrow)$

LP pg 182

Ex 4.1.1: $K = \{q_0, h\}, \Sigma = \{\mathtt{a}, \sqcup, \triangleright\}, s = q_0, H = \{h\}\}$

q	σ	$\delta(q, \sigma)$
q_0	$\mathtt{a}$	$(q_0, \leftarrow)$
q_0	$\sqcup$	$(h, \sqcup)$
q_0	$\triangleright$	$(q_0, \rightarrow)$

LP pg 183

$$
\begin{tikzcd}
> L_{\sqcup} \arrow[r] & R \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R^2_{\sqcup} a L^2_{\sqcup} a \arrow[l, bend right] \\
& R_{\sqcup} &
\end{tikzcd}
$$

LP pg 190. Figure 4-8, redrawn

$$\begin{tikzcd}
> L \arrow[r, "a \neq \sqcup"] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup} a L_{\sqcup} \arrow[l, bend right] \\
R_{\sqcup} &
\end{tikzcd}$$

LP pg 190. Figure 4-9, redrawn and corrected