Theory of Computation 6
Normalforms and Algorithms

Frank Stephan

Department of Computer Science

Department of Mathematics

National University of Singapore

fstephan@comp.nus.edu.sg

Repetition 1

Let $(\mathbf{Q_1}, \mathbf{\Sigma}, \delta_\mathbf{1}, \mathbf{s_1}, \mathbf{F_1})$ and $(\mathbf{Q_2}, \mathbf{\Sigma}, \delta_\mathbf{2}, \mathbf{s_2}, \mathbf{F_2})$ be dfas which recognise $\mathbf{L_1}$ and $\mathbf{L_2}$, respectively.

Consider $(\mathbf{Q_1} \times \mathbf{Q_2}, \mathbf{\Sigma}, \delta_\mathbf{1} \times \delta_\mathbf{2}, (\mathbf{s_1}, \mathbf{s_2}), \mathbf{F})$ with $(\delta_\mathbf{1} \times \delta_\mathbf{2})((\mathbf{q_1}, \mathbf{q_2}), \mathbf{a}) = (\delta_\mathbf{1}(\mathbf{q_1}, \mathbf{a}), \delta_\mathbf{2}(\mathbf{q_2}, \mathbf{a}))$. This automaton is called a product automaton and one can choose $\mathbf{F}$ such that it recognises the union or intersection or difference of the respective languages.

Union: $\mathbf{F} = \mathbf{F_1} \times \mathbf{Q_2} \cup \mathbf{Q_1} \times \mathbf{F_2}$;
Intersection: $\mathbf{F} = \mathbf{F_1} \times \mathbf{F_2} = \mathbf{F_1} \times \mathbf{Q_2} \cap \mathbf{Q_1} \times \mathbf{F_2}$;
Difference: $\mathbf{F} = \mathbf{F_1} \times (\mathbf{Q_2} - \mathbf{F_2})$;
Symmetric Difference: $\mathbf{F} = \mathbf{F_1} \times (\mathbf{Q_2} - \mathbf{F_2}) \cup (\mathbf{Q_1} - \mathbf{F_1}) \times \mathbf{F_2}$.

Repetition 2 and Gaps Filled

Regular languages are also closed under Kleene star, Kleene plus and concatenation: Use nfas for these and convert to dfas.

Context-free languages are closed under union, Kleene star, Kleene plus, concatenation and intersection with regular languages. They are in general not closed under intersection and complement.

Context-sensitive languages are closed under union, intersection, Kleene star, Kleene plus and concatenation. While these are easy to see, the following result is more difficult: They are also closed under complement (not part of this course).

Recursively enumerable languages are closed under union, intersection, Kleene star, Kleene plus and concatenation; they are not closed under complement.

Repetition 3: Palindromes

The members of the language $\{x \in \Sigma^* : x = x^{mi}\}$ are called palindromes. A palindrome is a word or phrase which looks the same from both directions.

An example is the German name “OTTO”; furthermore, when ignoring spaces and punctuation marks, a famous palindrome is the phrase “A man, a plan, a canal: Panama.” originating from the time when the canal in Panama was built.

The grammar with the rules $S \rightarrow aSa|aa|a|\varepsilon$ with a ranging over all members of Σ generates all palindromes; so for $\Sigma = \{0, 1, 2\}$ the rules of the grammar would be $S \rightarrow 0S0 \mid 1S1 \mid 2S2 \mid 00 \mid 11 \mid 22 \mid 0 \mid 1 \mid 2 \mid \varepsilon$.

The set of palindromes is not regular.

Repetition 4: Homomorphisms

Example
$\mathbf{Ascii(Year\ 2021) = 59656172203230323 1}$ represents each letter of "$\mathbf{Year\ 2021}$" by its two-digit hexadecimal ASCII representation.

Definition 5.28
A homomorphism is a mapping $\mathbf{h}$ with domain $\mathbf{\Sigma^*}$ for some alphabet $\mathbf{\Sigma}$ which preserves concatenation: $\mathbf{h(v \cdot w) = h(v) \cdot h(w)}$.

Proposition 5.29
The homomorphism is determined by the images of the single letters and $\mathbf{h(w) = h(a_1) \cdot h(a_2) \cdot \ldots \cdot h(a_n)}$ for a word $\mathbf{w = a_1 a_2 \ldots a_n}$; $\mathbf{h(\varepsilon) = \varepsilon}$.

Description 5.46
If $\mathbf{h}$ is a homomorphism then $\mathbf{h^{-1}(L) = \{w : h(w) \in L\}}$ is the inverse image of $\mathbf{L}$ under $\mathbf{h}$.

Repetition 5: Homomorphism-Laws

Theorem 5.32
The homomorphic images of regular and context-free languages are regular and context-free, respectively.

Theorem 5.38
Every language generated by some grammar is the homomorphic image of a context-sensitive language; in particular the class of context-sensitive languages is not closed under homomorphism.

Theorem 5.47
If $\mathbf{L}$ is on level $\mathbf{k}$ of the Chomsky hierarchy and $\mathbf{h}$ is a homomorphism then $\mathbf{h^{-1}(L)}$ is also on level $\mathbf{k}$.

Definition 6.1: Normal Forms

The normal forms are named after Noam Chomsky (born 7.12.1928) and Sheila Greibach (born 6.10.1939).

Assume that the language does not contain ε.

Chomsky Normal Form
All rules are of the form $\mathbf{A \to BC}$ or $\mathbf{A \to d}$ where $\mathbf{A, B, C}$ are non-terminals and $\mathbf{d}$ is a terminal.

Greibach Normal Form
All rules are of the form $\mathbf{A \to bw}$ where $\mathbf{b}$ is a terminal and $\mathbf{A}$ a non-terminal and $\mathbf{w}$ a (possibly empty) string of non-terminals.

If the language contains ε, one allows in both normal forms $\mathbf{S} \to \varepsilon$ for the start symbol $\mathbf{S}$ which then is not allowed to appear on the right side of a rule.

CNF Algorithm Steps 1 and 2

Given context-free grammar $(\mathbf{N_0}, \Sigma, \mathbf{P_0}, \mathbf{S})$.

1. Dealing with ε: Let $\mathbf{N_1} = \mathbf{N_0} \cup \{\mathbf{S'}\}$ and
$\mathbf{P_1} = \mathbf{P_0} \cup \{\mathbf{S'} \to \mathbf{S}\}$;
While there are rules $\mathbf{A} \to \mathbf{vBw}, \mathbf{B} \to \varepsilon$ in $\mathbf{P_1}$ with
$\mathbf{A} \to \mathbf{vw}$ not in $\mathbf{P_1}$ Do Begin $\mathbf{P_1} = \mathbf{P_1} \cup \{\mathbf{A} \to \mathbf{vw}\}$ End;
Remove all rules $\mathbf{A} \to \varepsilon$ with $\mathbf{A} \in \mathbf{N_0}$ from $\mathbf{P_1}$;
Continue with grammar $(\mathbf{N_1}, \Sigma, \mathbf{P_1}, \mathbf{S'})$.

2. Dealing with single terminal letters: Let $\mathbf{N_2} = \mathbf{N_1}$, $\mathbf{P_2} = \mathbf{P_1}$;
While there are $\mathbf{a} \in \Sigma$ and rule $\mathbf{A} \to \mathbf{vaw}$ in $\mathbf{P_2}$ with
$\mathbf{vw} \neq \varepsilon$ Do Begin Choose a new non-terminal $\mathbf{B} \notin \mathbf{N_2}$;
Replace in all rules in $\mathbf{P_2}$ all occurrences of $\mathbf{a}$ by $\mathbf{B}$;
Update $\mathbf{N_2} = \mathbf{N_2} \cup \{\mathbf{B}\}$ and $\mathbf{P_2} = \mathbf{P_2} \cup \{\mathbf{B} \to \mathbf{a}\}$ End;
Continue with grammar $(\mathbf{N_2}, \Sigma, \mathbf{P_2}, \mathbf{S'})$.

CNF Algorithm Steps 3 and 4

3. **Breaking long ride hand sides:** Let $\mathbf{N_3} = \mathbf{N_2}$ and $P_3 = \mathbf{P_2}$;
   While there is $\mathbf{A} \rightarrow \mathbf{Bw}$ in $\mathbf{P_3}$ with $|\mathbf{Bw}| \geq \mathbf{3}$
   Do Begin Choose a new non-terminal $\mathbf{C} \notin \mathbf{N_3}$ and let
   $\mathbf{N_3} = \mathbf{N_3} \cup \{\mathbf{C}\}$; Add the rules $\mathbf{A} \rightarrow \mathbf{BC}, \mathbf{C} \rightarrow \mathbf{w}$ into $\mathbf{P_3}$
   and remove the rule $\mathbf{A} \rightarrow \mathbf{Bw}$ from $\mathbf{P_3}$ End;
   Continue with grammar $(\mathbf{N_3}, \boldsymbol{\Sigma}, \mathbf{P_3}, \mathbf{S'})$.

4. **Removing rules $\mathrm{A} \rightarrow \mathrm{B}$:** Make a table of all $(\mathbf{A}, \mathbf{B}), (\mathbf{A}, \mathbf{c})$
   such that $\mathbf{A}, \mathbf{B} \in \mathbf{N_3}$, $\mathbf{c} \in \boldsymbol{\Sigma}$ and, in the grammar
   $(\mathbf{N_3}, \boldsymbol{\Sigma}, \mathbf{P_3}, \mathbf{S'})$, $\mathbf{A} \Rightarrow^* \mathbf{B}$ and $\mathbf{A} \Rightarrow^* \mathbf{c}$, respectively;
   Let $\mathbf{N_4} = \mathbf{N_3}$ and $\mathbf{P_4}$ contain the following rules:
   $\mathbf{S'} \rightarrow \varepsilon$ in the case that this rule is in $\mathbf{P_3}$;
   $\mathbf{A} \rightarrow \mathbf{a}$ in the case that $(\mathbf{A}, \mathbf{a})$ is in the table;
   $\mathbf{A} \rightarrow \mathbf{BC}$ in the case that there is $\mathbf{D} \rightarrow \mathbf{EF}$ in $\mathbf{P_3}$ with
   $(\mathbf{A}, \mathbf{D}), (\mathbf{E}, \mathbf{B}), (\mathbf{F}, \mathbf{C})$ in the table;
   The grammar $(\mathbf{N_4}, \boldsymbol{\Sigma}, \mathbf{P_4}, \mathbf{S'})$ is in Chomsky Normalform.

Example for Chomsky Normalform

The grammar $(\{\mathbf{S}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{S} \rightarrow \mathbf{0S0}|\mathbf{1S1}|\mathbf{00}|\mathbf{11}\}, \mathbf{S})$ generates all palindromes of even nonzero length.

Chomsky Normal Form
Non-terminals: $\{\mathbf{S}, \mathbf{T}, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$; terminals: $\{\mathbf{0}, \mathbf{1}\}$; start symbol: $\mathbf{S}$; rules: $\mathbf{S} \rightarrow \mathbf{TV}|\mathbf{UW}$, $\mathbf{T} \rightarrow \mathbf{VS}|\mathbf{0}$, $\mathbf{U} \rightarrow \mathbf{WS}|\mathbf{1}$, $\mathbf{V} \rightarrow \mathbf{0}$, $\mathbf{W} \rightarrow \mathbf{1}$.

The derivation $\mathbf{S} \Rightarrow \mathbf{0S0} \Rightarrow \mathbf{01S10} \Rightarrow \mathbf{010010}$ in the old grammar is equivalent to
$\mathbf{S} \Rightarrow \mathbf{TV} \Rightarrow \mathbf{VSV} \Rightarrow \mathbf{0SV} \Rightarrow \mathbf{0S0} \Rightarrow \mathbf{0UW0} \Rightarrow \mathbf{0WSW0} \Rightarrow \mathbf{0WS10} \Rightarrow \mathbf{01S10} \Rightarrow \mathbf{01TV10} \Rightarrow \mathbf{010V10} \Rightarrow \mathbf{010010}$
in the new grammar.

Exercises 6.4 – 6.6

Bring the following grammars into Chomsky Normal Form.

Exercise 6.4
$(\{\mathbf{S},\mathbf{T}\},\{\mathbf{0},\mathbf{1}\},\{\mathbf{S}\rightarrow\mathbf{TTTT},\mathbf{T}\rightarrow\mathbf{0T1}|\varepsilon\},\mathbf{S})$.

Exercise 6.5
$(\{\mathbf{S},\mathbf{T}\},\{\mathbf{0},\mathbf{1}\},\{\mathbf{S}\rightarrow\mathbf{ST}|\mathbf{T},\mathbf{T}\rightarrow\mathbf{0T1}|\mathbf{01}\},\mathbf{S})$.

Exercise 6.6
$(\{\mathbf{S}\},\{\mathbf{0},\mathbf{1}\},\{\mathbf{S}\rightarrow\mathbf{0SS11SS0},\mathbf{0110}\},\mathbf{S})$.

Always choose exercises of the right difficulty level: Students who are good in the field should take challenging exercises; students who have difficulties should take exercises they can master. So everyone should learn a bit from the exercises done by oneself; students should also try out exercises they do not present.

Removing Useless Non-Terminals

Let $(\mathbf{N_0}, \mathbf{\Sigma}, \mathbf{P_0}, \mathbf{S})$ be in Chomsky Normal Form.

1. **Removing non-terminating non-terminals:** Let $\mathbf{N_1} = \{\mathbf{A} \in \mathbf{N_0}$: there is a rule $\mathbf{A} \rightarrow \mathbf{a}$ or $\mathbf{A} \rightarrow \varepsilon$ in $\mathbf{P_0}\}$;
   While there is a rule $\mathbf{A} \rightarrow \mathbf{BC}$ in $\mathbf{P_0}$ with $\mathbf{A} \in \mathbf{N_0} - \mathbf{N_1}$ and $\mathbf{B}, \mathbf{C} \in \mathbf{N_1}$ Do Begin $\mathbf{N_1} = \mathbf{N_1} \cup \{\mathbf{A}\}$ End;
   Let $\mathbf{P_1} = \{\mathbf{A} \rightarrow \mathbf{w}$ from $\mathbf{P_0}$ with $\mathbf{A} \in \mathbf{N_1}$ and $\mathbf{w} \in \mathbf{N_1} \cdot \mathbf{N_1} \cup \mathbf{\Sigma} \cup \{\varepsilon\}\}$;
   If $\mathbf{S} \notin \mathbf{N_1}$ then terminate with empty grammar else continue with grammar $(\mathbf{N_1}, \mathbf{\Sigma}, \mathbf{P_1}, \mathbf{S})$.

2. **Selecting all reacheable non-terminals:** Let $\mathbf{N_2} = \{\mathbf{S}\}$;
   While there is a rule $\mathbf{A} \rightarrow \mathbf{BC}$ in $\mathbf{P_1}$ with $\mathbf{A} \in \mathbf{N_2}$ and $\{\mathbf{B}, \mathbf{C}\} \not\subseteq \mathbf{N_2}$ Do Begin $\mathbf{N_2} = \mathbf{N_2} \cup \{\mathbf{B}, \mathbf{C}\}$ End;
   Let $\mathbf{P_2} = \{\mathbf{A} \rightarrow \mathbf{w}$ from $\mathbf{P_1}$: $\mathbf{A} \in \mathbf{N_2}\}$;
   The grammar $(\mathbf{N_2}, \mathbf{\Sigma}, \mathbf{P_2}, \mathbf{S})$ does not contain any useless non-terminal.

Quiz 6.8

$\Sigma = \{0\}$; $N = \{Q, R, S, T, U, V, W, X, Y, Z\}$;
rules $S \to TU|UV$, $T \to UT|TV|TW$, $R \to VW|QQ|0$,
$Q \to 0$, $U \to VW|WX$, $V \to WX|XY|0$, $W \to XY|YZ|0$;
start symbol S.

Determine the set of reacheable and terminating non-terminals.

Exercises 6.9 and 6.10

Exercise 6.9: Consider the grammar

$$(\{S_0, S_1, \ldots, S_9\}, \{0\}, \{S_0 \to S_0S_0,\ S_1 \to S_2S_3,\ S_2 \to S_4S_6|0,\ S_3 \to S_6S_9,\ S_4 \to S_8S_2,\ S_5 \to S_0S_5,\ S_6 \to S_2S_8,\ S_7 \to S_4S_1|0,\ S_8 \to S_6S_4|0,\ S_9 \to S_8S_7\}, S_1).$$

Exercise 6.10: Consider the grammar

$$(\{S_0, S_1, \ldots, S_9\}, \{0\}, \{S_0 \to S_1S_1,\ S_1 \to S_2S_2,\ S_2 \to S_3S_3,\ S_3 \to S_0S_0|S_4S_4,\ S_4 \to S_5S_5,\ S_5 \to S_6S_6|S_3S_3,\ S_6 \to S_7S_7|0,\ S_7 \to S_8S_8|S_7S_7,\ S_8 \to S_7S_6|S_8S_6,\ S_9 \to S_7S_8|0\}, S_1).$$

For each of the two grammars, determine the set of reacheable and terminating non-terminals and explain the steps on the way to this set. What is the shortest word generated by the grammar?

Emptyness Check

The removing of useless non-terminals results in an empty grammar in the case that the grammar does not generate any word. Here an algorithm for a context-free grammar $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ which is not in any normalform.

Initialisation: Let $\mathbf{N}' = \emptyset$;

Loop: While there are $\mathbf{A} \in \mathbf{N} - \mathbf{N}'$ and a rule $\mathbf{A} \to \mathbf{w}$ with $\mathbf{w} \in (\mathbf{N}' \cup \mathbf{\Sigma})^*$
Do Begin $\mathbf{N}' = \mathbf{N}' \cup \{\mathbf{A}\}$ End;

Decision: If $\mathbf{S} \notin \mathbf{N}'$ then the language of the grammar is empty else the language of the grammar constains some word.

Finiteness Check

Let $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ be in Chomsky Normal Form. One computes $\mathbf{N}''(\mathbf{A})$ as the set of non-terminals occurring in a derivation of a terminal word from $\mathbf{A}$ after at least one step.

Initialisation 1: Let $\mathbf{N}' = \emptyset$;

Loop 1: While there are $\mathbf{A} \in \mathbf{N} - \mathbf{N}'$ and a rule $\mathbf{A} \to \mathbf{w}$ with
$\mathbf{w} \in (\mathbf{N}' \cup \mathbf{\Sigma})^*$
Do Begin $\mathbf{N}' = \mathbf{N}' \cup \{\mathbf{A}\}$ End;

Initialisation 2: For all $\mathbf{A} \in \mathbf{N}$, let $\mathbf{N}''(\mathbf{A}) = \emptyset$;

Loop 2: While there are $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbf{N}'$ and a rule $\mathbf{B} \to \mathbf{CD}$
with $\mathbf{B} \in \mathbf{N}''(\mathbf{A}) \cup \{\mathbf{A}\}$ and $\{\mathbf{C}, \mathbf{D}\} \not\subseteq \mathbf{N}''(\mathbf{A})$
Do Begin $\mathbf{N}''(\mathbf{A}) = \mathbf{N}''(\mathbf{A}) \cup \{\mathbf{C}, \mathbf{D}\}$ End;

Decision: If there is $\mathbf{A} \in \mathbf{N}''(\mathbf{S}) \cup \{\mathbf{S}\}$ with $\mathbf{A} \in \mathbf{N}''(\mathbf{A})$ then the language of the grammar is infinite else it is finite.

Exercise 6.13: Time Complexity

Polynomial time means that the algorithm uses time $\mathbf{p(n)}$ for inputs of length $\mathbf{n}$ to decide some property. If a grammar is in Chomsky Normal Form, $\mathbf{n}$ can just be the number of non-terminals.

The checks whether a grammar in CNF generates some word or generates infinitely many words are in polynomial time. How complex is it to not only check whether some word is generated but also to output one such witness? Possible answers are polynomial time, exponential time and double exponential time. Here the time to output a word of length $\mathbf{m}$ is $\mathbf{m}$ computation steps. Give reasons for the answer.

Exercises 6.14 – 6.17

For grammar $(\{S, T, U, V, W\}, \{0, 1, 2\}, P, S)$ with below rules determine how many words the grammar generates: (a) None, (b) One, (c) Two, (d) Three, (e) Finitely many and at least four, (f) Infinitely many?

Exercise 6.14: The rules are $S \rightarrow TT$, $T \rightarrow UU$, $U \rightarrow VW|WV$, $V \rightarrow 0$, $W \rightarrow 1$.

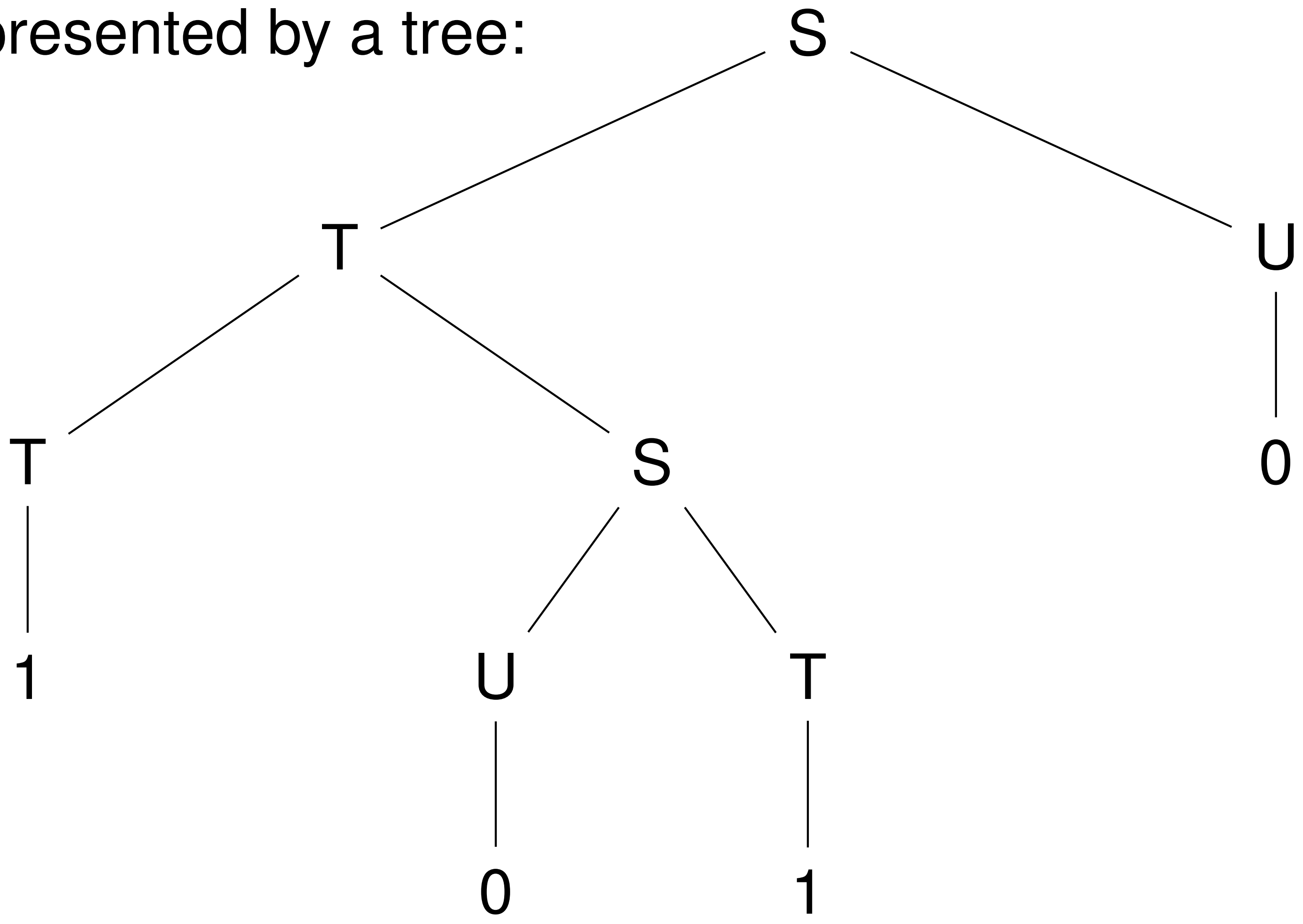
Exercise 6.15: The rules are $S \rightarrow ST$, $T \rightarrow TU$, $U \rightarrow UV$, $V \rightarrow VW$, $W \rightarrow 0$.

Exercise 6.16: The rules are $S \rightarrow UT|TU|2$, $T \rightarrow VV$, $U \rightarrow WW$, $V \rightarrow 0$, $W \rightarrow 1$.

Exercise 6.17: The rules are $S \rightarrow SS|TT|UU$, $T \rightarrow VV$, $U \rightarrow WW$, $V \rightarrow 0$, $W \rightarrow WW$.

Derivation Tree

For grammar $(\{\mathbf{S},\mathbf{T},\mathbf{U}\},\{\mathbf{0},\mathbf{1}\},\{\mathbf{S}\rightarrow\mathbf{SS}|\mathbf{TU}|\mathbf{UT},$ $\mathbf{U}\rightarrow\mathbf{0}|\mathbf{US}|\mathbf{SU},\ \mathbf{T}\rightarrow\mathbf{1}|\mathbf{TS}|\mathbf{ST}\},\mathbf{S})$, a derivation $\mathbf{S}\Rightarrow\mathbf{TU}\Rightarrow\mathbf{TSU}\Rightarrow\mathbf{TUTU}\Rightarrow\mathbf{1UTU}\Rightarrow\mathbf{10TU}\Rightarrow\mathbf{101U}\Rightarrow\mathbf{1010}$ can be represented by a tree:



Properties of Derivation Trees

Each node has a symbol in it.

If the symbol is a non-terminal then the successor nodes have the symbols of the rule applied at this symbol; if the symbol is a terminal then the node is a leaf.

The derivation does not depend on the order of rules on incomparable nodes are applied; however, the rule of each node has to be applied before it is applied to any successor.

The leaves below a node have the part of the word generated from the corresponding symbol.

Trees are more precise then derivations by lists of rules, as they explicitly say which non-terminal derives into which symbols and these are on the successor nodes.

Sometimes a derivation is ambiguous: For $\mathbf{TU} \Rightarrow \mathbf{TSU}$, the rule can be $\mathbf{T} \to \mathbf{TS}$ or $\mathbf{U} \to \mathbf{SU}$.

Number of Derivation Trees

As the rules to derive a word are not always unique but can be ambiguous, there can be different derivation trees for the same word. Indeed, one can prove that for some grammars it is impossible to have the derivation tree all the time unique. Consider the grammar $(\{S, T, U\}, \{0, 1\}, \{S \to SS|TU|UT,\ U \to 0|US|SU,$ $T \to 1|TS|ST\}, S)$, how many derivation trees has the word below?

Exercise 6.19: How many derivation trees are there for 011001?

Exercise 6.20: How many derivation trees are there for 000111?

Exercise 6.21

One can generalise the notion of derivation tree to all context-free grammars and might then have nodes with more than two successors. Consider the grammar

$$(\{S, T\}, \{0, 1, 2\}, \{S \to TT, T \to 0T1|2\}, S).$$

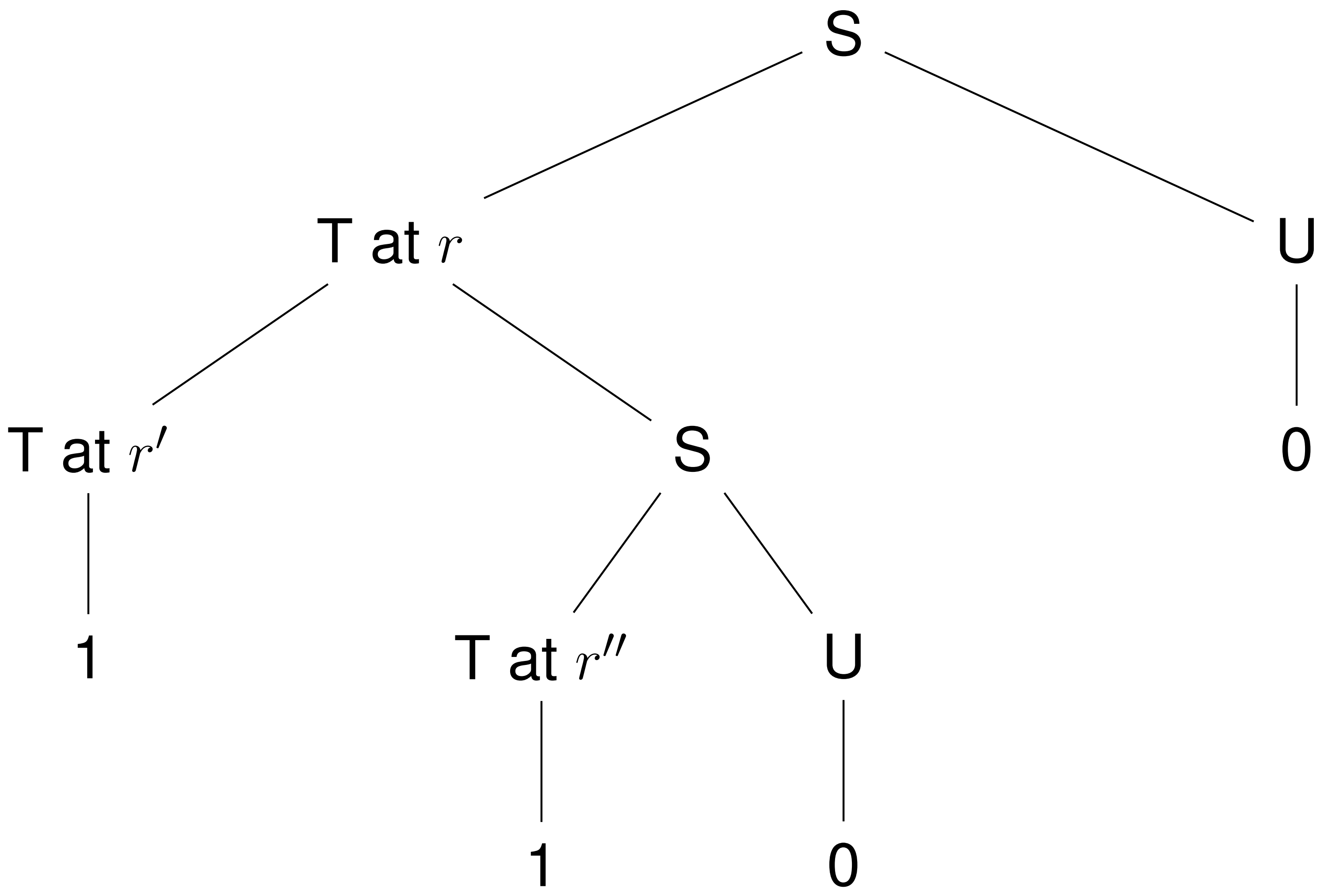
and draw the derivation tree for the word 00211021. Prove that all words in the language of this grammar have a unique derivation tree.

Pumping Lemma

Theorem 2.15 (b)
Let $\mathbf{L} \subseteq \mathbf{\Sigma}^*$ be an infinite context-free language generated by a grammar $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ in Chomsky Normal Form with $\mathbf{h}$ non-terminals. Then the constant $\mathbf{k} = \mathbf{2}^{\mathbf{h}+\mathbf{1}}$ satisfies that for every $\mathbf{u} \in \mathbf{L}$ of length at least $\mathbf{k}$ there is a representation $\mathbf{vwxyz} = \mathbf{u}$ such that $|\mathbf{wxy}| \leq \mathbf{k}$, $(\mathbf{w} \neq \varepsilon$ or $\mathbf{y} \neq \varepsilon)$ and $\mathbf{vw}^\ell\mathbf{xy}^\ell\mathbf{z} \in \mathbf{L}$ for all $\ell \in \mathbb{N}$.

Proof Idea
In the derivation tree of a word $\mathbf{u}$ longer than $\mathbf{2}^{\mathbf{h}+\mathbf{1}}$, find the lowest node having a non-terminal $\mathbf{A}$ which is also on a branch somewhere below. Below $\mathbf{A}$, no branch repeats a non-terminal and therefore each branch has at most the length $\mathbf{h}$. Thus there are at most $\mathbf{2}^{\mathbf{h}+\mathbf{1}}$ leaves below $\mathbf{A}$. Since $\mathbf{A}$ itself repeats, one has that $\mathbf{S} \Rightarrow^* \mathbf{vAz} \Rightarrow^* \mathbf{vwAyz}$ $\Rightarrow^* \mathbf{vwxyz}$ and $|\mathbf{wxy}| \leq \mathbf{2}^{\mathbf{h}+\mathbf{1}}$ and $\mathbf{A} \Rightarrow^* \mathbf{wAy}$ and $\mathbf{wy} \neq \varepsilon$.

Example of Derivation Tree



Lowest node $\mathbf{r}$ with non-terminal repeated below at $\mathbf{r'}$ or $\mathbf{r''}$.
Possible Pumpings: $\mathbf{1(10)^\ell 0}$, $\mathbf{1^\ell 10^\ell 0}$.

Ogden's Lemma

Marked Symbols

Mark the first four $\mathbf{1}$ in word $\mathbf{0000011111}$ as $\mathbf{0000011111}$. The word can be pumped such that at least one but at most four marked symbols are pumped or between the pumped parts: $\mathbf{000\,0^\ell\,01\,1^\ell\,111}$.

Theorem 6.22

Let $\mathbf{L} \subseteq \mathbf{\Sigma}^*$ be an infinite context-free language generated by a grammar $(\mathbf{N}, \mathbf{\Sigma}, \mathbf{P}, \mathbf{S})$ in Chomsky Normal Form with $\mathbf{h}$ non-terminals. Then the constant $\mathbf{k} = \mathbf{2^{h+1}}$ satisfies that for every $\mathbf{u} \in \mathbf{L}$ with at least $\mathbf{k}$ marked symbols, there is a representation $\mathbf{vwxyz} = \mathbf{u}$ such that $\mathbf{wxy}$ contains at most $\mathbf{k}$ marked symbols, $\mathbf{wy}$ contains at least $\mathbf{1}$ marked symbol and $\mathbf{vw^\ell xy^\ell z} \in \mathbf{L}$ for all $\ell \in \mathbb{N}$.

Example 6.23

The Language

Let $\mathbf{L}$ be the language of all words $\mathbf{w} \in \mathbf{1^+(0^+1^+)^+}$ with no two runs of zeroes of equal lengths. That is, words in $\mathbf{L}$ start and end with $\mathbf{1}$ and do not contain a subword $\mathbf{10^h1}$ twice for any $\mathbf{h} > \mathbf{0}$.

$\mathbf{L}$ contains words $\mathbf{1000101}$ and $\mathbf{1100110001}$ and $\mathbf{1111111}$.
$\mathbf{L}$ neither contains $\mathbf{100111001}$ nor $\mathbf{10100110100001}$.

Traditional Context-Free Pumping Lemma Satisfied

More precisely: Each word longer than $\mathbf{2}$ symbols in $\mathbf{L}$ can somewhere be pumped by a single one-symbol pump.
If $\mathbf{w} \in \mathbf{1^+0^+1^+}$ then pump some $\mathbf{0}$ in the middle.
If $\mathbf{w}$ contains a border of at least two $\mathbf{1}$ then pump a $\mathbf{1}$ in this border.
If $\mathbf{w}$ has a longest run of $\mathbf{0}$ separated by a single $\mathbf{1}$ from another run of $\mathbf{0}$ then pump this separating $\mathbf{1}$.

$\mathbf{L}$ does not satisfy Ogden's Lemma

If $\mathbf{k}$ is a constant supposed to work for Ogden's Pumping Lemma then consider $\mathbf{u = 1010^2 10^3 1 \ldots 10^{4k} 1}$ and mark the zeroes of the subword $\mathbf{10^k 1}$.
Let $\mathbf{vwxyz}$ split $\mathbf{u}$ into five parts.
If $\mathbf{w}$ or $\mathbf{y}$ in $\mathbf{\{0,1\}^+ - \{0\}^* - \{1\}^*}$ then $\mathbf{vw^4xy^4z}$ contains a subword $\mathbf{10^h 1}$ at least twice.
One of $\mathbf{w}$ and $\mathbf{y}$ must be in $\mathbf{0^+}$ and contain some of the marked zeroes.
In the word $\mathbf{vwwxyyz}$ pumping has replaced $\mathbf{10^k 1}$ by $\mathbf{10^{k+h} 1}$ for $\mathbf{h \in \{1, 2, \ldots, k\}}$ and thus the subword $\mathbf{10^{k+h} 1}$ in $\mathbf{u}$ must also be pumped and replaced by $\mathbf{10^{k+h+\ell} 1}$ for $\mathbf{\ell \in \{1, 2, \ldots, k+h\}}$. However, $\mathbf{10^{k+h+\ell} 1}$ is also a subword of $\mathbf{u}$ which occurs in $\mathbf{vwwxyyz}$ and thus $\mathbf{vwwxyyz \notin L}$ (as $\mathbf{h \leq k}$ and $\mathbf{\ell \leq 2k}$ and $\mathbf{h + k + \ell \leq 4k}$).

Square-Containing Words

Theorem 6.24 [Ehrenfeucht and Rozenberg 1983; Ross and Winklmann 1982]
The language $\mathbf{L}$ of square-containing words over $\{0, 1, 2\}$ is not context-free.

Proposition 6.25
The language $\mathbf{L}$ of square-containing words over $\{0, 1, 2\}$ satisfies Ogden's Lemma with constant $\mathbf{6}$.

If $\mathbf{u} = \mathbf{01201210201201210 2}$ then one can see that $\mathbf{012(01)^*2102012012102} \subseteq \mathbf{L}$. Details on next page.

Proof of Proposition

For given word $\mathbf{u} \in \mathbf{L}$ with at least six letters marked, consider those splittings $\mathbf{vwxyz} = \mathbf{u}$ where the following conditions are met:

- $\mathbf{v}$ ends with the same letter $\mathbf{a}$ with which $\mathbf{z}$ starts;
- $\mathbf{w}$ contains at least one marked letter;
- $\mathbf{x}$, $\mathbf{y}$ are ε.

Such splittings exist obviously when there is $\mathbf{a} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$ such that there are three marked letters $\mathbf{a}$ or there are two marked letters $\mathbf{a}$ with one other marked letter in between.

The remaining case is wlog of form $\mathbf{r_0 0 r_1 0 r_2 1 r_3 1 r_4 2 r_5 2 r_6}$. If $\mathbf{r_1}$ is in $\mathbf{0}^*$ then one let $\mathbf{w} = \mathbf{1}$ and $\mathbf{v}$ everything left of the first $\mathbf{1}$ and $\mathbf{z}$ everything right of it. If $\mathbf{r_1}$ contains a $\mathbf{b} \neq \mathbf{0}$ then one let $\mathbf{w}$ be all the part right of this $\mathbf{b}$ and left of the first marked $\mathbf{b}$ and $\mathbf{w}$ contains the marked letter $\mathbf{0}$.

Exercises 6.27 and 6.28

Exercise 6.27

Prove that the language

$\mathbf{L} = \{\mathbf{a}^{\mathbf{h}} \cdot \mathbf{w}$: $\mathbf{a} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}$, $\mathbf{w} \in \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^*$, $\mathbf{w}$ is square-free and $\mathbf{h} \in \mathbb{N}\}$

satisfies Theorem 2.15 (b) but does not satisfy Ogden's Pumping Lemma. The fact that there are infinitely many square-free words can be used without proof.

Exercise 6.28

Use the Block Pumping Lemma to prove the following variant of Ogden's Lemma for regular languages: If a language $\mathbf{L}$ satisfies the Block Pumping Lemma with constant $\mathbf{k} + \mathbf{1}$ then one can, for each word $\mathbf{u}$ having at least $\mathbf{k}$ marked symbols, find a splitting of the word into parts $\mathbf{x}, \mathbf{y}, \mathbf{z}$ such that $\mathbf{u} = \mathbf{xyz}$ and $\mathbf{xy}^*\mathbf{z} \subseteq \mathbf{L}$ and $\mathbf{y}$ contains at least $\mathbf{1}$ and at most $\mathbf{k}$ marked symbols.

Greibach Normal Form

Example 6.29

Consider $(\{\mathbf{S}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{S} \to \mathbf{0S0}|\mathbf{1S1}|\mathbf{00}|\mathbf{11}|\mathbf{0}|\mathbf{1}\}, \mathbf{S})$. For the Greibach Normal Form, one needs two additional non-terminals $\mathbf{T}, \mathbf{U}$ and updates the rules as follows:

$$\mathbf{S} \to \mathbf{0ST}|\mathbf{1SU}|\mathbf{0T}|\mathbf{1U}|\mathbf{0}|\mathbf{1},\ \mathbf{T} \to \mathbf{0},\ \mathbf{U} \to \mathbf{1}.$$

Consider $(\{\mathbf{S}\}, \{\mathbf{0}, \mathbf{1}\}, \{\mathbf{S} \to \mathbf{SS}|\mathbf{0S1}|\mathbf{1S0}|\mathbf{10}|\mathbf{01}\}, \mathbf{S})$. For the Greibach Normal Form, one needs two additional non-terminals $\mathbf{T}, \mathbf{U}$ and updates the rules as follows:

$$\mathbf{S} \to \mathbf{0SU}|\mathbf{0U}|\mathbf{1ST}|\mathbf{1T},\ \mathbf{T} \to \mathbf{0}|\mathbf{0S},\ \mathbf{U} \to \mathbf{1}|\mathbf{1S}.$$

Exercises 6.30 and 6.31

Construct Greibach Normal Form for the intersection of $\mathbf{0}^*\mathbf{1}^*\mathbf{0}^*\mathbf{1}^*$ with each of these two languages, respectively.

Languages and Derivatives

Exercise 6.32. Prove the following rules of the derivative with $\mathbf{a} \in \Sigma$ and $\mathbf{x} \in \Sigma^*$:

- $(\mathbf{L} \cup \mathbf{H})_\mathbf{x} = \mathbf{L}_\mathbf{x} \cup \mathbf{H}_\mathbf{x}$ and $(\mathbf{L} \cap \mathbf{H})_\mathbf{x} = \mathbf{L}_\mathbf{x} \cap \mathbf{H}_\mathbf{x}$;
- If $\varepsilon \in \mathbf{L}$ then $(\mathbf{L} \cdot \mathbf{H})_\mathbf{a} = \mathbf{L}_\mathbf{a} \cdot \mathbf{H} \cup \mathbf{H}_\mathbf{a}$ else $(\mathbf{L} \cdot \mathbf{H})_\mathbf{a} = \mathbf{L}_\mathbf{a} \cdot \mathbf{H}$;
- $(\mathbf{L}^*)_\mathbf{a} = \mathbf{L}_\mathbf{a} \cdot \mathbf{L}^*$.

The following theorem characterises the context-free languages by stating that all derivatives have to be formed from a finite set of languages using concatenation and union; furthermore, all the derivatives of these finitely many languages satisfy the same condition.

Theorem 6.33. A language $\mathbf{L}$ is context-free iff there is a finite list of languages $\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_\mathbf{n}$ with $\mathbf{L} = \mathbf{H}_1$ such that for every word $\mathbf{x}$ and every $\mathbf{H}_\mathbf{m}$, $(\mathbf{H}_\mathbf{m})_\mathbf{x}$ is a finite union of finite products of some $\mathbf{H}_\mathbf{k}$.

Exercise 6.34. Prove this theorem using Exercise 6.32 and Existence of Greibach Normal Form.