Formal Languages Example Sheet 1

OTOC: These exercises are bookwork. Do these only if you want to check your answers (Only TO Check).
OIFT: These exercises provide deeper insights or more practice (and are exam-level questions), but if you don't have time do not attempt them. (Only IF Time).
Optional: Attempt these only if you want more practice.
Advanced Optional: Attempt these only if you want a challenge or after the exams are over.

Warm up exercises

These warm up exercises are optional, though it would be good to quickly go through them.

Exercise 1 Remind yourself of the definitions for REs, DFAs and NFAs. Remind yourself why these are equivalent.

Exercise 2 Show that regular languages are closed under union, i.e. given languages L_1 and L_2 the language $L = L_1 \cup L_2$ is regular.

[Example Sheet 1 Formal Languages 1(a)]

Exercise 3 Show that regular languages are closed under concatenation, i.e. given languages L_1 and L_2 the language $L = L_1 L_2 = \{uv \mid u \in L_1, v \in L_2\}$ is regular.

[Example Sheet 1 Formal Languages 1(b)]

Exercise 4 Show that regular languages are closed under intersection, i.e. given languages L_1 and L_2 the language $L = L_1 \cap L_2$ is regular. *[**Hint:** Create a DFA that simulates the simultaneous execution of the two DFAs for the two languages.]*

[Example Sheet 1 Formal Languages 1(c)]

Exercise 5 Given a regular language L show that L^R, i.e. that language where all strings are reversed, is also regular. *[**Hint:** Consider a transformation on the DFA of the language.]*

Exercise 6 On **Lecture 1 slide 23**, it is mentioned that the recognition problem can be solved in $\mathcal{O}(n)$ time for regular languages, where n is the length of the input string. How can you do this?

Exercise 7 Are there any hidden assumptions in the $\mathcal{O}(n)$ recognition time for regular languages, that might be of practical importance?

Exercise 8 On **Lecture 1 slide 23**, it is mentioned that the recognition problem is undecidable for recursively enumerable languages. Why?

Exercise 9 State the pumping lemma for regular languages. Show that the following languages are not regular:

(a) $L = \{ab^n cd^n e \mid n \geq 1\}$,

(b) $L = \{a^n b^{n+1} \mid n \geq 1\}$,

(c) $L = \{ww \mid w \in \{a, b\}^*\}$.

[Example Sheet 1 Pumping Lemma 1]

Exercise 10 What is different in saying that: i) grammar G is X (e.g. for X being LL(1)) and ii) language L is X?

Exercise 11 Are the following statements true or false:

(a) Every subset of a regular language is regular.

(b) There exists a non-regular language with a subset of infinite size that is regular.

Phrase-structure grammars

Exercise 12 (**OTOC**) Define *phrase-structure grammars*.

Exercise 13 [Derivations](**OTOC**)

(a) Define a *derivation* in a phrase-structure grammar.

(b) Show the derivation for *abbccbba* in the CFG

$$S \to aSa \mid bSb \mid cSc \mid \epsilon$$

(c) Show two derivations for (())() and ()()() in the CFG

$$S \to SS \mid (S) \mid \epsilon$$

Exercise 14 [Algorithms for PSGs] What algorithmic problems are we interested in solving for PSGs?

Linear and Left/Right-linear languages

Exercise 15 In this exercise, you will prove that regular languages are equivalent to languages accepted by *left-linear* grammars, i.e. phrase structure grammars with rules of the form $A \to aB$ and $A \to a$ for $A, B \in \mathcal{N}$ and $a \in \Sigma$.

(a) By considering the DFA of a regular language, show how to generate a left-linear grammar that accepts the same strings. ***[Hint:** You need to define a non-terminal for each state of the DFA and carefully handle transitions to accepting states.]*

(b) Show how to generate a NFA for a given left-linear grammar.

(c) Argue that this shows that every regular language is also context free.

(d) How would you modify your proof for *right-linear* grammars. ***[Hint:** See warm up exercises.]*

Exercise 16 Consider the *linear grammar* consisting of rules of the form $A \to Ba$, $A \to aB$ and $A \to a$ for $A, B \in \mathcal{N}$ and $a \in \Sigma$. Is this grammar regular? Give a proof or find a counterexample.

Exercise 17 (**OIFT**) Show that adding rules of the form $A \to aBb$, for $A, B \in \mathcal{N}$ and $a, b \in \Sigma$ to linear grammars does not change the expressivity of the language.

Exercise 18(**OIFT**) Design a left-linear or right-linear language and show derivations and parse trees for:

(a) the language of unary multiples of k, $(0^k)^+$ (for $k = 3$)

(b) the language of binary strings with even number of 1s.

(c) the language of strings that end with 1011.

Exercise 19 (**OIFT**) Design a linear grammar for accepting binary palindromes of even length.

Exercise 20 [Open-ended] What are the advantages and disadvantages of using linear grammars to model natural languages? Which ones are solved by CFGs?

Context-free languages

Core exercises

Exercise 21 (**OTOC**) Define a *context-free grammar*.

Exercise 22 (**OTOC**) Show how to generate a CFG G that generates the same strings as that of a regular expression R.

Exercise 23 Define a CFG that generates:

(a) the strings consisting of balanced parentheses

(b) the valid regular expressions for the binary alphabet

(c) the arithmetic expressions with addition, multiplication and parentheses. Give a precedence to the operations and remove any ambiguity.

Exercise 24 [Closure under union] Given CFGs $G_i = (V_i, \Sigma, R_i, S_i)$, create a CFG G, so that $L(G) = L(G_1) \cup \ldots \cup L(G_n)$.

Exercise 25 [Closure under concatenation] Show that the concatenation of two context-free languages is also context-free.

Exercise 26 [Closure under reversal] (**OIFT**) Show that the reversal of the strings of a context-free language form a context-free language.

Exercise 27 [CNF]

(a) (**OTOC**) What does it mean for a language to be in *Chomsky Normal Form*?

(b) (**OIFT**) Show how to convert the following CFG to CNF:

$$S \to aAB$$
$$A \to aAa \mid bb$$
$$B \to a$$

(c) (**Optional**) How would you deal with unit productions, i.e. productions of the form $A \to B$?

Exercise 28 [Non-CFLs]

(a) State the pumping lemma for CFLs. How can it be used to prove that a language is not context-free?

(b) What other way can you use to prove that a language is not CFL?

(c) Show that the following languages are not CFLs:

i. $\{a^n b^n c^n \mid n \geq 1\}$,

ii. $\{a^n b^n c^m \mid n \leq m\}$,

iii. (**OIFT**) $\{a^n b^m c^n d^m \mid n \leq m\}$,
iv. (**OIFT**) $\{0^n 1^n 0^n \mid n \geq 0\}$,
v. (**OIFT**) $\{0^n \# 0^{2n} \# 0^{3n}\}$,
vi. (**OIFT**) The language of all palindromes over $\{0, 1\}$ with an equal number of 0s and 1s.

[Example Sheet 1 Pumping Lemma 2]

Exercise 29(**OIFT**)
(a) Show that the languages $L_1 = \{a^m b^n c^n \mid n, m \geq 0\}$ and $L_2 = \{a^n b^n c^m \mid n, m \geq 0\}$ are context-free.
(b) By considering their intersection, argue that context-free languages are not closed under intersection.
(c) Using De Morgan's rules, argue that context-free languages are not closed under complementation.

Pushdown-automata (PDA)

Exercise 30 (**OTOC**) Define a *push-down automaton.*

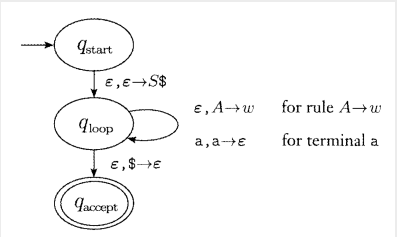
Exercise 31 Design automata for the following CFLs:
(a) $L = \{0^n 1^n \mid n \geq 0\}$,
(b) (**OIFT**) $L = \{a^i b^j c^k \mid i, j, k \geq 0, i = j \text{ or } i = k\}$,
(c) (**OIFT**) $L = \{ww^R \mid w \in \{0, 1\}^*\}$.
[Hint: *These examples are taken from pages 112 to 114 of "Introduction to Computation Theory" by Sipser (2nd Edition). You can check your solutions there.]*

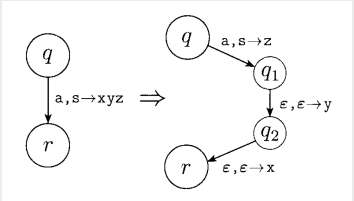
Exercise 32 Show that the intersection of a regular and a context-free language is also context-free. ***[Hint:*** *Try to construct a non-deterministic PDA].*

[Example Sheet 1 Formal Languages 2(a)]

Exercise 33 (**OIFT**) In this exercise, you will prove that any language accepted by a CFG can be accepted by some PDA. Consider the PDA defined to contain three main states q_{start}, q_{loop} and q_{accept}. Then add the following connections:

- From q_{start} to q_{loop} with $\epsilon, \epsilon \to S\$$
- From q_{loop} to q_{accept} with $\epsilon, \$ \to \epsilon$
- For each rule $A \to a_1, \ldots, a_k$, add a sequence of states $q_{loop} \to q_1 \to \ldots \to q_{k-1} \to q_{loop}$ with transitions $A, \epsilon \to a_k$, $\epsilon, \epsilon \to a_{k-1}$, ..., $\epsilon, \epsilon \to a_1$.
- For each rule $A \to a$ add from q_{loop} to q_{loop} with transition $a, a \to \epsilon$.





(a) Derive the equivalent PDA for the language

$$S \to aTb \mid b$$
$$T \to Ta \mid \epsilon$$

(b) Argue that this PDA accepts the same language as the one generated by the grammar.

Natural language

Exercise 34 Explain the meaning of the commonly used linguistic non-terminals (S, NP, PP, A, etc). ***[Note:** There is no standard for the names of these, so you might notice slight differences in notation (or even in structure).]*

Exercise 35 [English sentences] Define a CFG that generates the following linguistic concepts in a sensible manner and produce the derivation trees for the sentences:

(a) NP Preposition phrases
   i. *He saw the man with the telescope.*
   ii. *She saw the elephant with the telescope.*
   iii. *He saw the woman with the hat.*

(b) VP Preposition phrases
   i. *The Eiffel Tower is in Paris.*
   ii. *She drove down the street in the car.*

(c) Noun adjectives (A)
   i. *The new car outperforms the old car.*
   ii. *The fast car mechanic repaired the broken car.*

(d) Conjuction (CONJ)
   i. *The man and the woman walked.*
   ii. *The woman walked and sang.*
   iii. *The man shot the sheriff and the deputy.*
   iv. *The woman walked and sang with joy.*

(e) Subordinate clause (SBAR)
   i. *Wikipedia says that the company closed.*

(f) Indirect object
   i. *She bought her a present.*

Exercise 36 [Noun modifiers] (**OIFT**) A *noun modifier* is a noun that modifies or qualifies another noun. For example, "parking" in "parking slot" is a noun modifier of "slot".

(a) Construct meaningful examples with three and four consecutive nouns.

(b) Using rules of the form $N \to NN$, show how the different parse trees correspond to different interpretations.

(c) The number of parse trees for n noun modifiers is given by the $(n+1)$-th Catalan number. For $n = 4$ nouns, calculate the number of different parse trees and draw these parse trees.

(d) Find example noun phrases, for as many of the trees that you drew.

(e) What does this example demonstrate for the possible number of parses?

(f) (**Optional**) Catalan numbers count a number of items in important combinatorial classes. Go through one of the proofs in the wikipedia page (the sixth one is probably the simplest).

Exercise 37 [Centre embedding/Non-regularity] Describe the phenomenon of centre embedding and how it is used to prove that English is non-regular.

Exercise 38 Are LR(k), LL(k), SLR(k) or LALR(k) languages suitable for modelling natural languages?

Earley's parser

Exercise 39

(a) Describe the operation of Earley's parser.

(b) What speedups if any would you do for natural language grammars with large vocabularies?

(c) (**OIFT**) Earley's parser takes $O(N^3)$ time to parse a sentence with N tokens. Show that parsing a^N in the grammar $S \to SS \mid a$ takes $\Theta(N^3)$ time.

(d) The time complexity $O(N^3)$ is hiding some constants which might be of practical interest for a natural language grammar. What are these?

Exercise 40 (Practical) Write an implementation of the Earley parser that can use the toy grammar defined in **Lecture 3 slide 25** to parse the sentences below:

- *They can fish in rivers.*
- *They can fish in rivers in December.*

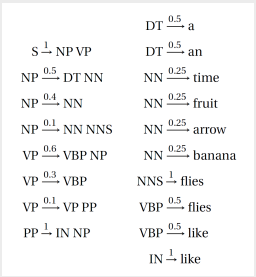
How many parses are there for each sentence?
As mentioned in the course's supervision sheet, do not overengineer this, i.e. you don't need to write code to print derivation trees and I don't need to see the code. Also, you may want to run `mains/earley_parser/earley_parser_main.cc` *which prints each step of the parser. You can modify the grammar to answer this question.*

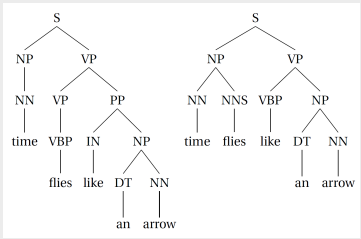
[Example Sheet 1 Top down parsing 1]

Exercise 41 (**OIFT**)

(a) Describe how you would modify Earley's parser to find the highest scoring derivation.

(b) Find the scoring of the following parse trees:





Example taken from here.

(c) Finding the highest scoring parse corresponds to which estimation method (cf Part IB Data Science)?

Past papers

(OIFT)

- **[2008P2Q8 (a),(b)]**

- **[2008P9Q9 (a),(b),(c)]**
- **[2006P11Q11 (a),(b),(c)]**
- **[2005P2Q9 (b)]**
- **[1995P5Q3]**
- **[1994P4Q3]**

(See also past papers from Compiler Construction)

Context-sensitive languages

Exercise 42 Consider the following context-sensitive grammar:

$$\begin{aligned} S &\to aSBC \\ S &\to aBC \\ CB &\to BC \\ aB &\to ab \\ bB &\to bb \\ bC &\to bc \\ cC &\to cc \end{aligned}$$

(a) Show the derivation for *aabbcc*.

(b) (**OIFT**) Argue informally that this grammar generates exactly the strings $a^n b^n c^n$ for $n > 0$.

Tree adjoining grammars

Exercise 43

(a) (**OTOC**) Define *tree adjoining grammars*.

(b) What does it mean for a word to be generated by a tree adjoining grammar.

(c) Why are they at least as powerful as CFGs?

(d) What is the linguistic motivation for tree adjoining grammars?

Exercise 44 For three sentences of your choice from Exercise 35, create a Tree Adjoining grammar and show the steps in the derivation.

Exercise 45(**Optional**) Consider the following sentences:

- Alice eats cakes.
- The caterpillar gives Alice cakes.
- The cat with a grin disappears.
- Alice paints white roses red.

Using the examples in the notes/slides to start you off, complete the following tasks:

(a) Define a context free grammar that could generate the sentences.

(b) Draw a dependency parse for the sentences.

(c) Define a tree adjoining grammar that could generate the sentences

[Example Sheet 1 Comparing grammar formalisms 1]

Exercise 46 (**Optional**) In this exercise, you will show that TAGs are more powerful than CFGs.

(a) Create a TAG that generates $L_1 = \{wc^n \mid w \in \{a,b\}^*, n = \#a(w) = \#b(w), \forall p \in \text{prefix}(w).\#a(p) \geq \#b(p)\}$.

(b) By considering the intersection with the regular language $a^*b^*c^*$, show that L_1 cannot be context-free.

Dependency grammars

Exercise 47 (**OTOC**) Define dependency grammars.

Exercise 48 (**OIFT**) A dependency tree is a (labelled) rooted tree over the words in a sentence.

(a) Design a linear-time algorithm to check if a graph is a rooted tree.

(b) What does it mean for a tree to be projective?

(c) Design an algorithm for checking if a tree is projective.

(d) Give an example of a sentence that has a non-projective dependency tree.

Exercise 49 [CFG equivalence]

(a) Given a projective dependency grammar, show how to construct a weakly equivalent CFG.

(b) Demonstrate your approach with a dependency grammar for the sentence "She gave her a present". How many valid parses does your grammar have?

Exercise 50 [Natural languages]

(a) Explain the meaning of commonly used dependency labels (root, nsubj, dobj, iobj, nmod). (See the univeral dependencies site and a database with many examples)

(b) For three sentences of your choice from Exercise 35, create a dependency grammar and show the dependency tree.

(c) Show the possible dependency trees for an ambiguous sentence.

Exercise 51 [Dependency grammar parsing]

(a) Show how CFG equivalence can be used to generate dependency trees.

(b) Argue that shift-reduce parsers produce non-projective dependency trees.

(c) How do data-driven dependency tree parsers work? Do they need a dependency grammar?

(d) What is beam search and what problem does it solve?

(e) (**Optional**) Read this blog post on Parsey McParseface.

Natural languages / Psycholinguistics

Exercise 52 [Ambiguity]

(a) Describe the different types of ambiguity in natural language and give examples for each.

(b) Describe the ambiguities in the following sentences:

i. *She fed her cat food.*

ii. *She saw the man with one eye.*

iii. *She saw the queen in the garden with the telescope.*

[Example Sheet 1 Natural Languages 1]

Exercise 53 [Processing difficulty] The following natural language sentences are difficult to process. Hypothesise what is causing the difficulty:

(a) *I told the girl the rabit knew the caterpillar would help her.*

(b) *The twins the rabbit the girl chased liked laughed.*

(c) *She shook the bottle containing the potion which had made her grow very tall up.*

Discuss how your hypotheses might be tested.

[Example Sheet 1 Natural Languages 2]

Exercise 54 [Complex sentences] (Open-ended) What makes a complex sentence? How does this relates to the information theoretic content of a sentence? Give examples.

Exercise 55 [Spoken vs written] How does spoken differ from written language? How do these differences affect NLP models targeted for speech/written language? Consider garden-path sentences, homophony, etc.

Exercise 56 What is Yngve's hypothesis? What evidence support it? Do you agree?

Exercise 57 [Chomsky's hierarchy]

(a) Create a Venn diagram for the languages: LR(1), LL(1), SLR(1), LALR(1), finite languages, regular languages, CFGs, TAGs, CSGs, REs.

(b) Where would you place natural languages? Why?

Additional optional exercises

The following exercsies are points to more advanced material, in case you want to go in more depth over the summer break.

Project 1 [Size of NFA] Read the 2.5-page paper "A lower bound technique for the size of nondeterministic finite automata" by Glaister and Shallit (1995).

(a) What does it mean to lower bound the size of an NFA?

(b) What is the technique that the authors propose?

(c) How does it apply to the languages i) $L_k = \{0^i 1^i 2^i \mid 0 \leq i < k\}$ and ii) $A_k = \{w \in \{0,1\}^k \mid w = w^R\}$?

Project 2 [CYK parsing algorithm]

(a) Read about the CYK parser for CFGs that are in CNF.

(b) How fast does it parse? How does it compare with Earley parser?

(c) Read this presentation for a modified version of CYK parser for TAGs.

Project 3 [Collins' parser]

(a) Read about Collins' parser from this presentation.

(b) Implement Collins' parser. You can have a look at the implementation in `languages/collins_parser.cc`.

(c) Read about Eisner's parser, which is an improvement over Collins' parser, from this .

(d) Implement Eisner's parser.