Exercise 1: Formal languages and grammars

You can earn up to 10 points on this exercise.
You may work as a group of up to 3 people, but please submit your own version.
You may use any programming language you wish, but any submission that we cannot run on our computers without installing things must be presented to the class.

Please email your solution to `claytong@coli.uni-saarland.de` *by* **November 4, 2015**.

In the following grammars, strings within double quotes ("...") denote terminal symbols.
All others are non-terminals.

Task 1

Consider the following grammar:

S	→	N	"fish"
N	→		"fish"

a. What kind(s) of formal language does the grammar admit? Explain your answer. (1 point)

b. Add one *sensible* rule to the grammar such that the grammar admits a non-finite, regular language. Explain the relevant properties of your rule. (1 point)

c. Describe the general form of the sentences belonging to the part (b) language. (1 point)

Task 2

Now consider the following grammar:

S	→	NP VP
NP	→	Det N
PP	→	P NP
VP	→	V NP
Det	→	"the"
N	→	"fish"
P	→	"with"
V	→	"moved"

a. What kind(s) of formal language does the grammar admit? Explain your answer. (1 point)

b. Add one *sensible* rule to the grammar such that the grammar admits a non-finite, deterministic context-free language. Explain the relevant properties of your rules. (1 point)

c. Describe the general form of the sentences belonging to the part (b) language. (1 point)

d. Add one *sensible* rule to the grammar such that the grammar admits a non-finite, *non*-deterministic context-free language. Explain the relevant properties of your rules. (1 point)

e. What does it mean that the grammar is in Chomsky Normal Form (CNF)? What kinds of languages can be described using grammars in CNF? Add a few (perhaps non-sensical) rules to the grammar that are context-free, but not in CNF. (1 point)

Task 3

a. What is the general form of context-sensitive grammar rules and why are they necessary to describe cross serial dependencies in Swiss German? (1 point)

b. Briefly explain the transformation system for affix hopping, as described in Noam Chomsky's *Syntactic Structures* (1957). What level of formal language is required to implement this system? (1 point)