Teoria da Computação

(Theoretical Computer Science)
Licenciatura em Engenharia Informática Exercises 2011-2012

version of December 12, 2011

1 Modeling with Sets and Logic

1. Consider the language of set theory enriched with the following ingredients

   - a set *BOOL* of all booleans {`TRUE`, `FALSE`};
   - a set *NAT* of all natural numbers $\{0, 1, 2, 3, \cdots, \}$;
   - the basic operations on natural numbers $(+, -, \times, \div, \%)$;
   - a set *CHAR* of all UNICODE characters;
   - a set *STRING* of all strings of characters ("hello", "tc", "jack", "mary", ...);

   Define the following sets

   (a) The set of all even natural numbers.

   (b) The set of all prime numbers.

   (c) The extension of the addition function on natural numbers.

   (d) The extension of the negation function *NOT* on booleans.

   (e) The extension of the conjuction function *AND* on booleans.

   (f) The extension of the disjunction *OR* and implication *IMPLIES* function on booleans, based on *AND* and *NOT*.

   (g) A contact card contains a name, a set of phone numbers, and a set of emails. Define the set of all contact cards.

(h) A function *FIND* that given a set of contact cards and a name, returns the corresponding phone numbers and emails, or `FALSE`, if the given name is not in any contact card.

To what set does *FIND* belong?

(i) A function that given two contact books gives the contact book obtained by merging the two sets of contacts.

(j) The set of all pairs of strings.

(k) The set M of all functions from the set of natural numbers to the set of all sets of natural numbers.

(l) The set M defined in the previous item contains many many elements. Define the element g of M that maps each natural n to the set of all naturals less than n. Check that g as you have defined indeed belongs to M.

2. Consider the following candidate definition of an "ordered pair"

$$(x, y) \triangleq \{x, \{y\}\}$$

Why does this not really work?

Hint: find sets a, a', b, b' where $a \neq a'$ and $b \neq b'$ such that $(a, b) = (a', b')$ according to the candidate definition.

3. Model the following data with sets

(a) the set of dates since 1 Jan 1904 until the end of the XXth century.

(b) the set of integers supported by the Sun Java Virtual Machine.

(c) the set of floating point numbers representable in IEEE 754-2008 half precision format.

4. Model the following system with a structure.

A lamp with two states `ON` and `OFF`.

(a) Model the set of states of a lamp with a set $SLAMP$.

(b) Define a function in $SLAMP \rightarrow SLAMP$ that models the "turn on" operation.

(c) Define a function in $SLAMP \rightarrow SLAMP$ that models the "turn off" operation.

(d) Define a function in $SLAMP \rightarrow BOOL$ that returns the current state of the lamp.

5. Model the following system with a structure.

   A counter keeps the count of cars inside a tunnel by keeping track if cars entering the tunnel and cars exiting the tunnel.

   (a) Model the set of states of a counter with a set $SCOUNTER$.
   (b) Define a function in $SCOUNTER \to SCOUNTER$ that models the "car enter" operation.
   (c) Define a partial function in $SCOUNTER \to SCOUNTER$ that models the "car exit" operation.
   (d) Define a function in $SCOUNTER \to NAT$ that yields the number of cars currently inside the tunnel.

6. Model the following data with sets

   (a) The set of all states of a bank account, including the owner, the account number, and the balance.
   (b) The set of all transactions of bank accounts, each transaction includes the account number, the type of operation (deposit or withdrawal, and the exchanged amount.
   (c) Define a function $JOIN$ that given a set of bank accounts, and two account numbers, yields a set of bank accounts identical to the given one, except that the two given accounts are merged in a new account, under the number of (and owner of) smallest account number.
   (d) To what set belongs the function $JOIN$?
   (e) The set of all states of a bank account with history. A bank account with history is like a bank account, but also keeps a set of performed transactions.
   (f) A bank account is correct if its set of transactions only refers to the given account. Define a function that given a set of bank accounts with history, returns its subset of correct bank accounts.

7. Model the following system with a structure.

   A stock list keeps a list of products, each one with its identifier (name), price, and stock available.

   (a) Model the set of states a stock list with a set $SLIST$.

Figure 1: Arpanet (the embrionic Internet, 1969).

(b) Model (with a function or relation) the operation that adds a product to a stock list.

(c) Model (with a function or relation) the operation on stock lists that removes a product.

(d) Model (with a function or relation) the operation on stock lists that given a product returns its price.

(e) Model (with a function or relation) the operation on stock lists that given a product returns its price and stock.

(f) Model (with a function or relation) the operation on stock lists that gives the set of products out of stock.

(g) Model (with a function or relation) the operation on stock lists that increases the stock of product by N items.

(h) Model (with a function or relation) the operation on stock lists that decreases the stock of product by N items.

8. Model the following data with sets

   (a) A graph is a structure that comprises a set of nodes and a set of arcs, which are arrows connecting nodes. A simple example of a graph is a computer network, where the nodes are the computers and the arcs represent the connections between computers. Define a set modeling the graph in Figure 1.

   (b) Define a function *GRAPHS* that given a set of nodes (assume that nodes are simply represented by strings), yields the set of all graphs over that set of nodes.

   (c) To what set belongs the function *GRAPHS*?

   (d) Define a function that given a graph, returns the graph obtained by reversing the direction of all arcs.

   (e) Consider a graph G, modeled as you have done above, and a node N of G. Define a sequence of sets $S_0, S_1, S_2, ...$ such that S_0 contains only N, and for all $k \geq 0$, S_{k+1} contains the set of notes reachable by transversing a single arc from a node in S_k.

9. Model the following system with a structure.

A weather station keeps measuring the temperature using a sensor, and reports on the values detected. Each measurement consists of the time of the sample, and the temperature.

(a) Model the set of states a weather station with a set $SWEATHER$.

(b) Model (with a function or relation) the operation that performs a sample and registers the temperature value and measurement time.

(c) Model (with a function or relation) that clears all performed measurements.

(d) Model (with a function or relation) an operation that gives the number of recorded measurements.

(e) Model (with a function or relation) an operation that gives the maximum and minimal temperature measured.

(f) Model (with a function or relation) an operation that gives the maximum and minimal temperature measured in a given time interval within the recording period.

10. Model the following system with a structure.

    Consider a system that keeps track of the access permissions regarding a set of printers within a network. Consider that for each user there is a set printers that s/he is allowed to use.

    (a) Model the set of states of the access system with a set $APRINTER$.

    (b) Model (with a function or relation) the operation that gives a user access to a printer.

    (c) Model (with a function or relation) the operation that blocks access of an user to a printer.

    (d) Model (with a function or relation) the operation that checks if a user has access to a specific printer.

    (e) Model (with a function or relation) an operation that returns the users that can access more printers.

11. Model the following system with a structure.

    A dictionary mapping portuguese words into (finite) sets of portuguese sentences. For example, a dictionary entry contains the following information,

"aluno"
====================
"aquele que recebe doutrem educação e instrução"; "discipulo"; "aprendiz"; "educando"

(a) Model the set of states of the dictionary with a set *SDICT*. Before defining this, look to the list of operations the dictionary is expected to support.
(b) Model (with a function or relation) the operation that adds an entry to the dictionary. If an entry for the given word already exists, the new synonyms should be added to the existing entry.
(c) Model (with a function or relation) the operation that checks if a given word is defined in the dictionary.
(d) Model (with a function or relation) the operation that adds replaces the definitions associated to an existing entry for a given word with a different set of sentences.
(e) Model (with a function or relation) the operation that removes an entry from the dictionary.
(f) Model (with a function or relation) the operation that lookups in the dictionary the meaning of a word.

=======

12. Model the following system with a structure.

    A stack of text notes. Each note may be represented by a string. As usual, only the top of the stack is accessible to add or remove elements.

    (a) Model the set of states of a stack of notes with a set *SSTACK*.
    (b) Model (with a function or relation) the operation that pushes a note in the stack.
    (c) Model (with a function or relation) the operation that pops and returns the top node in the stack.
    (d) Model (with a function or relation) an operation that observes if the stack is empty or not.

13. Model the following system with a structure.

    Consider a system that keeps track of playlists. A playlist includes a name and a set of songs. Each song is defined by a title, an artist or band name, a duration, and a rating (1 to 5).

(a) Model the set of states of the system with a set *SPLAYLIST*.

(b) Model (with a function or relation) the operation that adds a song to a playlist.

(c) Model (with a function or relation) the operation that removes a song from a playlist.

(d) Model (with a function or relation) the operation that removes a song from all playlists.

(e) Model (with a function or relation) the operation that determines the name of the favorite song (or songs) of all playlists.

(f) Model (with a function or relation) the operation that given a song name returns the playlist names that include the song.

14. Model the following system with a structure.

    An array of names.

    A name array consists of a collection names, each name being associated to an index.

    Indexes start at 0, are incremented by 1 for each name, and are consecutive (0, 1, 2, . . .).

    (a) Model the set of states of the name array with a set *SARRAY*.

    (b) Model (with a function or relation) the operation `add` that adds a name to the array (at the last free index).

    (c) Model (with a function or relation) the operation `find` that returns the first index of a given name. If no such name exists, the operation returns `false` .

    (d) Model (with a function or relation) the operation `remove` that removes the first occurrence of a name from an array (and decrements all indexes greater than the removed name's index). If no such name exists in the array, it is left unchanged.

15. Model the following system with a structure.

    Consider a simplified version of *Google*$^+$ Circles where a user organizes friends using circles.

    A circle is simply a collection of users identified by a name.

    A user can be added to more than one circle.

    Assume that each user is identified by an email.

(a) Model SCircle, the set of all states of the system.

(b) Define (with a function or relation) the operation on SCircle that adds a user to a set of circles

(c) Define (with a function or relation) the operation on SCircles that retrieves all people in a circle

(d) Define (with a function or relation) the operation on SCircle that blocks a user, i.e. deletes a user from all circles.

(e) Define (with a function or relation) the operation on SCircle that retrieves all the circles associated with a given user.

(f) Define (with a function or relation) the operation on SCircle that given a user, retrieves its larger circle

(g) Define (with a function or relation) the operation on SCircle that given two circles names, retrieves the users that belong to the intersection.

(h) An extended circle of an user are its circle's circles, i.e. people who are at one degree of distance.

Define (with a function or relation) the operation on SCircle that given a user retrieves its extended circle.

(i) Imagine a super extended circle that is defined by all people who are connected at any arbitrary degree of distance.

Define (with a function or relation) the operation on SCircle that given a user retrieves its super extended circle.

16. Provide an inductive definition for the following set

$$\mathit{PowersOfTwo} \triangleq \{n \in \mathit{NAT} \mid n = 2^p \wedge (p >= 0)\}$$

17. Provide an inductive definition for the function

$$\mathit{exp2} \in \mathit{NAT} \to \mathit{NAT}$$

such that

$$\mathit{exp2}(n) = 2^n$$

Give a bottom up justification that $4 \mapsto 16 \in \mathit{exp}2$ (or equivalently, that $\mathit{exp}2(4) = 16$).

18. Provide an inductive definition for the following sets

$$\mathit{EvenSequences} \triangleq \{s \in \mathit{SEQ} \mid \mathit{len}(s)\%2 = 0\}$$

$$OddSequences \triangleq \{s \in SEQ \mid len(s)\%2 = 1\}$$

N.B. Recall the (indutive) definition of the set *SEQ* of all sequences of natural numbers, and the function *len* that gives the length of a sequence (both are defined in the lecture notes).

Write down a top-down justification that $(1, 2, 3) \in OddSequences$.

Write down a bottom-up justification that $(0, 0) \in EvenSequences$.

19. Provide an inductive definition for the set *SortedSequences* of the increasingly sorted sequences of natural numbers. For example,

$$(2, 5, 6, 6, 7) \in SortedSequences$$

$$(1, 2, 1, 3, 4) \notin SortedSequences$$

20. Provide an inductive definition for the function

$$addLast \in SEQ \times NAT \to SEQ$$

such that $addLast(s_1, n) = s_2$ if and only if s_2 is obtained from s_1 by adding n as last element. For example,

$$addLast(((1, 3, 5), 9)) = (1, 3, 5, 9)$$

21. Provide an inductive definition for the relation

$$Reverse \subset SEQ \times SEQ$$

such that $(s_1, s_2) \in Reverse$ if and only if s_2 is the reverse sequence of s_1. For example,

$$((1, 3, 5), (5, 3, 1)) \in Reverse$$

$$((1, 3, 5), (2, 3)) \notin Reverse$$

22. Model the following system with a structure.

Consider a (very) simplified version of Facebook. The system maintains a set of users, and the information about who is friend of each user.

   (a) Model SFace, the set of all states of the system.
   (b) Define (with a function or relation) the operation on SFace that adds a user to the system.

(c) Define (with a function or relation) the operation on SFace that sets a user as friend of other user.

(d) Define (with a function or relation) the operation on SFace that says if a user if a friend of other user.

(e) Define (with a function or relation) the operation on SFace that returns the set of friends of some user.

(f) Define (with a function or relation) the operation on SFace that says if a user is connected to other user by a sequence of friends.

(g) Define (with a function or relation) the operation on SFace that returns the network of a user (the set of all users that are connected by a chain of friendship to the given user).

23. Justify whether the following sets are finite, countable, or uncountable.

(a) The set of all functions from $SLAMP$ to $SLAMP$.

(b) The set of all functions $SLAMP$ to NAT.

(c) The set of all real numbers which are roots of some polynomial with natural coeficients.

(d) The set of all real numbers which are roots of some polynomial with rational coeficients.

(e) The set of all functions $NAT \to NAT$.

(f) The set of all graphs with less that 128 nodes.

(g) The set of states of any stock list $S \in SLIST$.

(h) $SLIST$, the set of all stock lists.

(i) $SCOUNT$, the set of all counters.

(j) The set of all correctly written Java programs that fit in a 2GB flash pen drive.

(k) The set of all real numbers.

2 Computational Machines and Specifications

1. Specify a Deterministic Finite Automaton (DFA) over the alphabet

$$\Sigma = \{\mathtt{X}\}$$

that checks if a sequence of `X`'s is of even length, in other words, that only accepts the words over Σ of even length.

2. Specify a DFA over the alphabet

$$\Sigma = \{\mathtt{X}\}$$

that checks if a sequence of `X`'s is of odd length.

3. Consider the alphabet

$$DIGITS \triangleq \{\mathtt{0}, \mathtt{1}, \mathtt{2}, \mathtt{3}, \mathtt{4}, \mathtt{5}, \mathtt{6}, \mathtt{7}, \mathtt{8}, \mathtt{9}\}$$

Specify a DFA over the alphabet $DIGITS$ that checks if a word over $DIGITS$ is a possible pin for a mobile phone.

4. Consider the alphabet

$$DNA \triangleq \{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}\}$$

Specify a DFA over the alphabet DNA that only accepts the words over DNA that contain at least one occurrence of `ACT` as a substring.

5. Specify a DFA over the alphabet $DIGITS$ that checks if a word over $DIGITS$ is the representation of an even natural number.

6. Specify a DFA over the alphabet

$$\Sigma = \{ \begin{array}{l} \mathtt{insertcard}, \\ \mathtt{pin}, \\ \mathtt{checkbalance}, \\ \mathtt{widthdraw}, \\ \mathtt{moreops?}, \\ \mathtt{yes}, \\ \mathtt{no}, \\ \mathtt{retrievecard} \quad \} \end{array}$$

that checks if a word (trace) over Σ describes a valid interaction between a user and an ATM.

7. Consider the alphabet

$$AB \triangleq \{\texttt{a}, \texttt{b}\}$$

Specify a DFA over the alphabet AB that only accepts the words over AB that contain an odd number of `b`'s.

8. Specify a DFA over the alphabet $\Sigma \triangleq \mathit{DIGITS} \cup \{-, +\}$ that checks if a word over Σ is the representation of an integer literal in the Java language (such as $+\texttt{120}$, or $-\texttt{99}$).

9. Consider the alphabet

$$\mathit{LETTERS} \triangleq \{\texttt{a}, \texttt{b}, \ldots \texttt{z}\}$$

Specify a DFA over the alphabet $\Sigma \triangleq \mathit{DIGITS} \cup \mathit{LETTERS} \cup \{., :, /\}$ that checks if a word over Σ can represent the URL of a web site.

10. Consider the alphabet

$$AB \triangleq \{\texttt{a}, \texttt{b}\}$$

Specify a DFA over the alphabet AB that only accepts the words over AB that contain an odd number of `b`'s and an even number of `a`'s.

11. Consider the alphabet

$$\mathit{DNA} \triangleq \{\texttt{A}, \texttt{T}, \texttt{C}, \texttt{G}\}$$

Specify a DFA over the alphabet DNA that only accepts the words over DNA that contain at least one occurrence of `ATACA` as a substring.

12. Specify a DFA over the alphabet

$$\Sigma = \{ \begin{array}{l} \texttt{coffee}, \\ \texttt{tea}, \\ \texttt{cappuccino}, \\ \texttt{10cents}, \\ \texttt{20cents}, \\ \texttt{cancel} \end{array} \}$$

that checks if a word (trace) over Σ describes a valid interaction between a user and a vending machine. Assume that the machine requires exact change and that coffee, tea, and cappuccino cost, respectively, 20 cents, 30 cents, and 40 cents.

13. Specify a DFA over the alphabet

$$\begin{aligned}\Sigma \triangleq\ & \mathit{DIGITS} \cup \mathit{LETTERS} \\ & \cup \{call, contact, first\_name, last\_name, end\}\end{aligned}$$

that checks if a word (trace) over Σ describes a valid interaction between a user and a cell phone: after inserting a phone number, the user can either call it or save it. If the user chooses to save the number, he/she must specify a first name and/or a surname (at least one of the two names must be specified), and then can either call the number or end it in order to return to the main menu. The following sequences are valid interactions:

$123\, contact\, first\_name\, Joana\, end$,

$123\, contact\, first\_name\, Joana\, last\_name\, Pereira\, call$,

$123\, call$.

14. Consider the alphabet $\mathit{MOVES} = \{\texttt{up}, \texttt{down}, \texttt{left}, \texttt{right}\}$.

Check if the following words belong to the language denoted by the regular expression $(\texttt{up} + \texttt{left})^*(\texttt{up} + \texttt{down} + \texttt{right})^+$ over MOVES.

- ()
- `up right`
- `right up`
- `down up right`
- `up up down left`
- `up up left down`

15. Consider the alphabet $\Sigma = \{\texttt{talk}, \texttt{bob}, \texttt{alice}, \texttt{over}\}$ and the following regular expression over Σ:

$$(\texttt{talk bob over} + \texttt{talk alice over})^*$$

Define a DFA that accepts the language denoted by this regular expression. Note: you are not supposed to use any formal method to define the DFA, we would like you to think about what is the language denoted by the expression, and construct a solution you find correct. Hint: define first a DFA that recognizes the language denoted by

$$\texttt{talk bob over} + \texttt{talk alice over}$$

16. Consider the following regular expressions over *MOVES*

    - $(\texttt{up} + \texttt{down})^*$
    - $(\texttt{up}^* \; \texttt{down}^*)^*$

    Do you think they represent the same language? Discuss and justify your opinion.

17. Consider the alphabet $\Sigma = \{a, b, c\}$.

    Check if the following words belong to the language denoted by the regular expression $(a + b)^*(ab + bc)^*$ over Σ.

    - $()$
    - $ababbc$
    - $bcbca$
    - $bcbcab$
    - $bbbaaabc$
    - $bbbaaaba$
    - $bbbabcba$

18. Specify a regular expression over the alphabet

    $$\Sigma = \{\texttt{X}\}$$

    that denotes the set of words over Σ of even length.

19. Recall the alphabet

    $$\Sigma = \{ \begin{array}{l} \texttt{insertcard}, \\ \texttt{pin}, \\ \texttt{checkbalance}, \\ \texttt{widthdraw}, \\ \texttt{moreops?}, \\ \texttt{yes}, \\ \texttt{no}, \\ \texttt{retrievecard} \quad \} \end{array}$$

    Using a regular expression over Σ, specify the language of valid interaction traces between a user and an ATM.

20. Consider the alphabet

$$DIGITS \triangleq \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}, \texttt{8}, \texttt{9}\}$$

Define a regular expression over the alphabet $DIGITS$ that denotes the language of all 4-digin pins.

21. Consider the alphabet

$$DNA \triangleq \{\texttt{A}, \texttt{T}, \texttt{C}, \texttt{G}\}$$

Specify a regular expression over the alphabet DNA that defines the language over DNA of all words that contain at least one occurrence of `ACACG` as a substring.

22. Specify a regular expression over the alphabet $DIGITS \cup \{\texttt{E}, +, -, .\}$ that denotes all numbers in IEEE754 notation.

23. Translate the regular expression $(XX)^*$ over the alphabet $\Sigma = \{X\}$ into a NFA. Then, translate the NFA into a DFA.

24. Translate the next regular expression over the alphabet $MOVES$ into a NFA. Then, translate the NFA into a DFA.

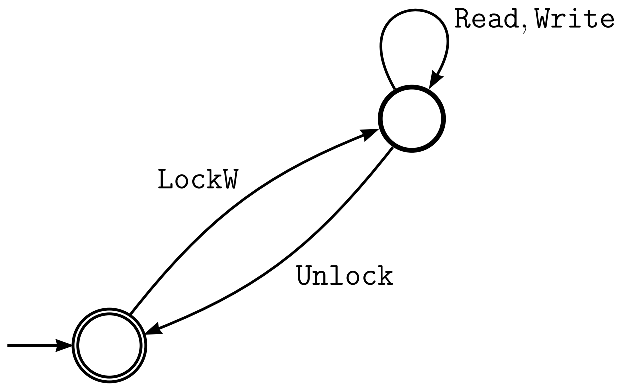
$$(open\ read^* + open\ write^*)close$$

25. Translate the regular expression over the alphabet $\Sigma = \{0, 1\}$ into a NFA. Then, translate the NFA into a DFA.

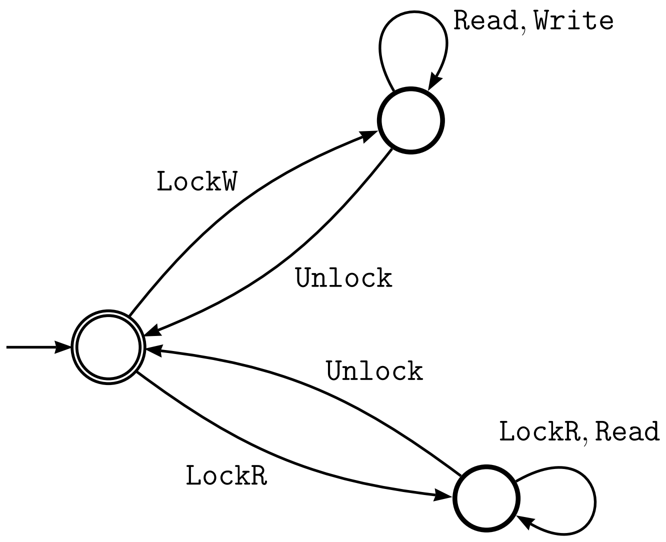
$$((0 + 1)^*111) + (101(1 + 0)^*)$$

26. Translate the regular expression, over the alphabet DNA, obtained in exercise 21 into a NFA. Then translate the NFA into a DFA.

27. Translate the following DFA into a regular expression:



28. Translate the following DFA into a regular expression:



29. Translate the DFA obtained in exercise 10 into a regular expression.

30. Define a context free grammar (CFG) $\mathcal{G} = \langle V, \Sigma, S, R\rangle$ that accepts the same language as the DFA of exercise 28.

31. Explain how given any DFA A, you may define a CFG that recognizes exactly the same language (inspire yourself in the approach you followed in the previous exercise). Conclude that

    - Any regular language can be specified with some CFG, and so CFGs are at least so powerful as DFAs.
    - Any regular language is also a context-free language.

32. Consider the following grammar $HTML = \langle V, \Sigma, C, R\rangle$ where

$$\begin{array}{rcl} V & = & \{C, S, R, T\} \\ \Sigma & = & \{ <, h, \backslash, >, c, x, \} \end{array}$$

and R contains the following rules:

$$\begin{array}{rcl} C & \rightarrow & TR \\ S & \rightarrow & < h > C < \backslash h > \\ R & \rightarrow & \epsilon \\ R & \rightarrow & TR \\ T & \rightarrow & S \\ T & \rightarrow & c \\ T & \rightarrow & x \end{array}$$

For each one of the following words, check, by exhibiting derivations, which ones belong to the language of *HTML*. For the words that you think do not belong, give your best explanation about that.

- $cxcxcx$
- $< h > xcx < h >$
- $< h > c < \backslash h >$
- $< hh >$
- $< h > ccxx < h > xx < \backslash h >< \backslash h >$

33. Consider the following CFG $\mathcal{G} = \langle V, \Sigma, P, R\rangle$ where

$$\begin{array}{rcl} V & = & \{P, S, E\} \\ \Sigma & = & \{\texttt{id}, \texttt{num}, \{, ;, \}, \texttt{if}, \texttt{then}, \texttt{else}, :=, \texttt{while}, (,), +, =\} \end{array}$$

and R contains the following rules:

$$\begin{array}{rcl} P & \rightarrow & S \\ S & \rightarrow & \texttt{id} := E \\ S & \rightarrow & \texttt{if}\ (E)\ S \\ S & \rightarrow & \texttt{if}\ (E)\ S\ \texttt{else}\ S \\ S & \rightarrow & \texttt{while}\ (E)\ S \\ S & \rightarrow & \{S\} \\ S & \rightarrow & S; S \\ E & \rightarrow & E + E \\ E & \rightarrow & E = E \\ E & \rightarrow & (E) \\ E & \rightarrow & \texttt{id} \\ E & \rightarrow & \texttt{num} \end{array}$$

(a) Construct a derivation of the word

$$\{\texttt{id} := \texttt{num} + \texttt{id}; \texttt{if}(\texttt{id} + \texttt{id})\{\texttt{id} := \texttt{num}\}\}$$

in the given grammar. Indicate the rule used at each step.

34. Define a CFG than accepts the language over alphabet $\Sigma = \{\texttt{push}, \texttt{pop}\}$ that consists of all words that contains the same number of `push` and `pop` symbols.

35. Consider the alphabet $\Sigma = \{\texttt{begin}, \texttt{end}\}$. Based on this alphabet, we may define a context free language by the grammar $G = \langle V, \Sigma, S, R\rangle$ where

$$V = \{S, T\}$$

and R contains the following rules:

$$\begin{array}{lcl} S & \rightarrow & TS \\ S & \rightarrow & \epsilon \\ T & \rightarrow & \texttt{begin}\ S\ \texttt{end} \end{array}$$

(a) Construct a derivation of the word

`begin end begin begin end end`

in the given grammar. Indicate the rule used at each step.

(b) In this part you are required to produce a deterministic LL(1) parser for the above grammar.

The main job is to define the transition function

$$\delta : V \times (\Sigma \cup \{\$\}) \rightarrow (R \cup \{\texttt{SYNTAXERR}\})$$

- Compute the set $First(w)$ for each rule $A \rightarrow w$ in the grammar.
- Compute the set $Follow(H)$ for each nonterminal $H \in V$.
- Construct the transition table for δ. Explain why the grammar is $LL(1)$.
- Show step by step how the deterministic stack-based parser based on the transition table recognizes the word given above.

36. Consider the alphabet

$$\Sigma = \{\texttt{the}, \texttt{dog}, \texttt{fox}, \texttt{eats}, \texttt{barks}, \texttt{banana}, \texttt{that}\}$$

Based on this alphabet, we define a context free language by the grammar $G = \langle V, \Sigma, S, R\rangle$ where

$$V = \{S, NP, VP, RC, IV, TV, DET, N\}$$

and R contains the following rules:

$$\begin{array}{lcl} S & \rightarrow & NP\ VP \\ NP & \rightarrow & DET\ N\ RC \\ VP & \rightarrow & IV \\ VP & \rightarrow & TV\ NP \\ RC & \rightarrow & \epsilon \\ RC & \rightarrow & \texttt{that}\ VP \\ DET & \rightarrow & \texttt{the} \\ IV & \rightarrow & \texttt{barks} \\ TV & \rightarrow & \texttt{eats} \\ N & \rightarrow & \texttt{dog} \\ N & \rightarrow & \texttt{fox} \\ N & \rightarrow & \texttt{banana} \end{array}$$

(a) Construct a derivation of the words of over Σ

`the banana barks`

in the given grammar. Indicate the rule used at each step.

(b) Construct a derivation of the words of over Σ

`the dog that barks eats the banana`

in the given grammar. Indicate the rule used at each step.

(c) In this part you are required to produce a deterministic LL(1) parser for the above grammar.

The main job is to define the transition function

$$\delta : V \times (\Sigma \cup \{\$\}) \to (R \cup \{\texttt{SYNTAXERR}\})$$

- Compute the set $First(w)$ for each rule $A \to w$ in the grammar.
- Compute the set $Follow(H)$ for each nonterminal $H \in V$.
- Construct the transition table for δ. Explain why the grammar is $LL(1)$.
- Show step by step how the deterministic stack-based parser based on the transition table recognizes the second sentence given above in (b)

37. Consider the alphabet

$$\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{h}, \texttt{f}, \texttt{g}\}$$

Based on this alphabet, we define a context free language by the grammar $G = \langle V, \Sigma, S, R\rangle$ where

$$V = \{S, B, C, D, E, F\}$$

and R contains the following rules:

$$\begin{array}{lcl}
S & \to & \texttt{a}\, B\, D\, \texttt{h} \\
B & \to & \texttt{c}\, C \\
C & \to & \texttt{b}\, C \\
C & \to & \epsilon \\
D & \to & E\, F \\
E & \to & \texttt{g} \\
E & \to & \epsilon \\
F & \to & \texttt{f} \\
F & \to & \epsilon
\end{array}$$

(a) Construct a derivation of the words of over Σ

`a c b f h`

in the given grammar. Indicate the rule used at each step.

(b) In this part you are required to produce a deterministic LL(1) parser for the above grammar.

The main job is to define the transition function

$$\delta : V \times (\Sigma \cup \{\$\}) \to (R \cup \{\texttt{SYNTAXERR}\})$$

- Compute the set $First(w)$ for each rule $A \to w$ in the grammar.
- Compute the set $Follow(H)$ for each nonterminal $H \in V$.
- Construct the transition table for δ. Explain why the grammar is $LL(1)$.
- Show step by step how the deterministic stack-based parser based on the transition table recognizes the second sentence given above in (a).

38. Consider the alphabet

$$\Sigma = \{\texttt{num}, \texttt{id}, *, +, (,)\}$$

Based on this alphabet, we define a context free language by the grammar $G = \langle V, \Sigma, E, R\rangle$ where

$$V = \{E, E', T, T', F\}$$

and R contains the following rules:

$$\begin{array}{lcl} E & \to & T\ E' \\ E' & \to & +\ T\ E' \\ E' & \to & \epsilon \\ T & \to & F\ T' \\ T' & \to & *\ F\ T' \\ T' & \to & \epsilon \\ F & \to & (E) \\ F & \to & \texttt{id} \\ F & \to & \texttt{num} \end{array}$$

(a) Construct a derivation of the words of over Σ

`(num * id ) + num`

in the given grammar. Indicate the rule used at each step.

(b) In this part you are required to produce a deterministic LL(1) parser for the above grammar.

The main job is to define the transition function

$$\delta : V \times (\Sigma \cup \{\$\}) \to (R \cup \{\texttt{SYNTAXERR}\})$$

- Compute the set $First(w)$ for each rule $A \to w$ in the grammar.
- Compute the set $Follow(H)$ for each nonterminal $H \in V$.
- Construct the transition table for δ. Explain why the grammar is $LL(1)$.
- Show step by step how the deterministic stack-based parser based on the transition table recognizes the second sentence given above in (a).

39. Consider the alphabet

$$\Sigma = \{\texttt{f}, \texttt{const}, (,), \texttt{comma}, \texttt{digit}\}$$

where `digit` may be realized by any of the characters `0`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `const` may be realized by the character "`a`", and `comma` may be realized by the character ",".

Based on this alphabet, we define a context free language by the grammar $G = \langle V, \Sigma, E, R \rangle$ where

$$V = \{I, S, ARGS, ARGR, DR\}$$

and R contains the following rules:

$$\begin{array}{lcl}
I & \to & S \\
S & \to & \texttt{f}\ DR\ (\ ARGS\ ) \\
S & \to & \texttt{const} \\
ARGS & \to & S\ ARGR \\
ARGS & \to & \epsilon \\
ARGR & \to & \texttt{comma}\ S\ ARGR \\
ARGR & \to & \epsilon \\
DR & \to & \texttt{digit}\ DR \\
DR & \to & \epsilon
\end{array}$$

Examples of accepted words:

```
a
f()
f123(a)
f(a,f(a))
f12(a)
f33(f43(),f8282(a,f(a)))
```

(a) You are required to implement (in Java) a deterministic LL(1) parser for the above grammar. The main job is to define the transition function

$$\delta : V \times (\Sigma \cup \{\$\}) \to (R \cup \{\texttt{SYNTAXERR}\})$$

- Compute the set $First(w)$ for each rule $A \to w$ in the grammar.
- Compute the set $Follow(H)$ for each nonterminal $H \in V$.
- Construct the transition table for δ. Explain why the grammar is $LL(1)$.
- Implement the parser in Java, representing the symbols as a proper Java class, and the LL(1) deterministic automata using a stack along the lines defined in the course. Your program should read an input string from the stdin, in a terminated line, and then write either `SyntaxError` or `Accepted` in a single line as answer. The code may contain more than one file, but the main class (containing the `main` function) should be called `Main`.

IMPORTANT NOTICE: The full Java code for the last item of this exercise is to be found in the CLIP system. The TP3 will follow exactly the same structure of this exercise, so that you may find there some inspiration by carefully studying the code.

40. Specify a Stack Based Turing Machine (SBTM) that in the end of its execution contains the value $(\texttt{first}, \texttt{second})$ in the memory cell M_1.

41. Specify a SBTM that in the end of its execution contains the value $(((\texttt{5},\texttt{7}),\texttt{13}),(\texttt{21},\texttt{30}))$ in the memory cell M_1

42. Specify a SBTM that, given two lists, one in the memory cell M_1 and another in the memory cell M_2, computes the concatenation of the two lists and puts it in the memory cell M_3.

43. Consider a list L stored in the memory cell M_1 and an element X stored in the memory cell M_2. Specify a SBTM that in the end of its execution contains the value `true` in the memory cell M_3 if X belongs to L, and `false` otherwise.

44. Consider a list `Dictionary` that contains pairs of (`word`, `definition`) stored in the memory cell M_1, and a specific `word` stored in the memory cell M_2. Specify a SBTM that in the end of its execution contains in the memory cell M_3 the value `definition` corresponding to the `word` stored in M_2, or `null` if the `word` is not in the `Dictionary`.