Department of Information and Computing Sciences
Utrecht University

INFOB3TC – Exam 1

Johan Jeuring

Friday, 12 December 2014, 08:30–10:30

Preliminaries

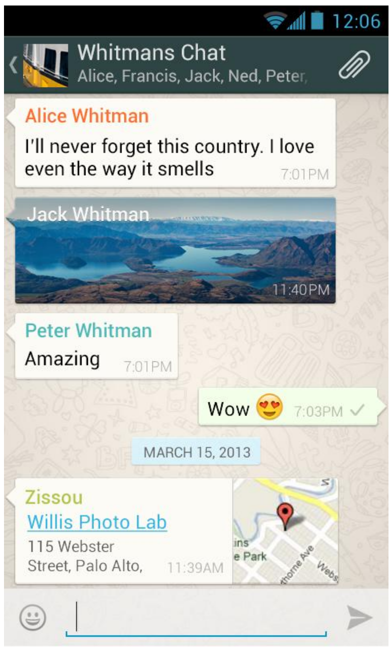
- The exam consists of 10 pages (including this page). Please verify that you got all the pages.
- Fill out the answers **on the exam itself**.
- Write your **name** and **student number** here:

- The maximum score is stated at the top of each question. The total amount of points you can get is 90.
- Try to give simple and concise answers. Write readable text. Do not use pencils or pens with red ink. You may use Dutch or English.
- When writing grammar and language constructs, you may use any set, sequence, or language operations covered in the lecture notes.
- When writing Haskell code, you may use Prelude functions and functions from the following modules: *Data.Char*, *Data.List*, *Data.Maybe*, and *Control.Monad*. Also, you may use all the parser combinators from the `uu-tc` package. If you are in doubt whether a certain function is allowed, please ask.

Good luck!

Questions

A group chat in Whatsapp looks as follows:


12:06

Whitmans Chat

Alice, Francis, Jack, Ned, Peter,

Alice Whitman

I'll never forget this country. I love even the way it smells 7:01PM

Jack Whitman

11:40PM

Peter Whitman

Amazing 7:01PM

Wow 😍 7:03PM ✓

MARCH 15, 2013

Zissou

Willis Photo Lab

115 Webster Street, Palo Alto, 11:39AM

These group chats are stored on Whatsapp servers, and sent to the computers (phones, tablets) of the members of the groupchat. The internal representation of the above group chat (or at least a part of it) might look as follows:

```
GROUPCHAT
  NAME    Whitmans Chat
  MEMBERS Alice, France, Jack, Ned, Peter, Zissou
  MESSAGES
    MESSAGE
      NAME Alice
      TIME 7:01 PM, March 14, 2013
      CONTENT I'll never forget this country. I love even the way it smells END
   MESSAGE
      NAME Jack
      TIME 11:40 PM, March 14, 2013
      CONTENT mountains.jpg END
   MESSAGE
      NAME Peter
      TIME 7:01 PM, March 14, 2013
      CONTENT Amazing END
   MESSAGE
      NAME Ned
```

```
         TIME 7:03 PM, March 14, 2013
         CONTENT Wow U+1F60D END
      MESSAGE
         NAME Zissou
         TIME 11:39 AM, March 15, 2013
         CONTENT http:\\www.willis.com\ END
```

In this exercise we look at the language of group chats.

A group chat consists of the keyword `GROUPCHAT` followed by:

- the keyword `NAME` followed by a name,
- the keyword `MEMBERS` followed by a non-empty list of members, separated by comma's,
- the keyword `MESSAGES` followed by a list of messages, where a message consists of the keyword `MESSAGE`, followed by the keyword `NAME`, a name, the keyword `TIME`, a time, the keyword `CONTENT`, some content, and the keyword `END`.

1 (12 points)**.** Give a concrete syntax (a context-free grammar) of this language for group chats. You may use nonterminal *Identifier* to recognise a single name, and *String* to recognise the content of a message (a string not containing "END"). •

The abstract syntax of the language for groups chats is given by the following (data)types:

```
type GroupChat = (Name, Members, Messages)
type Name      = [String]
type Members   = [Name]
type Messages  = [Message]
type Message   = (Name, Time, Content)
type Time      = (Hours, Minutes, APM, Date)
type Hours     = Int
type Minutes   = Int
data APM       = AM | PM deriving Show
type Date      = (Day, Month, Year)
type Day       = Int
type Month     = Int
type Year      = Int
type Content   = String
```

2 (12 points)**.** Define a parser *pGroupChat* :: *Parser Char GroupChat* that parses sentences from the language of groupchats. •

3 (12 points)**.** Whatsapp only offers chats and groupchats. I can imagine it would be useful to have a hierarchy of chats. For example, Utrecht University might start a chat, with seven subchats for the seven faculties. So people can chat at the university level, or at their own faculty level. Each faculty chat consists of faculty wide chats, but also of chats at the various departments of the faculty, and so on. In this exercise I encode a slightly simplified version of this situation. A *GroupChatTree* is either a single *GroupChat*, or it collects a number of *GroupChatTree*'s under a particular name.

```
data GroupChatTree = Fork Name [GroupChatTree]
                   | Single GroupChat
```

Define the algebra type, and the *fold* for the datatype *GroupChatTree*. You may assume that the type *GroupChat* is a constant type such as *Int* and *String*, that is, you don't have to define a *fold* for *GroupChat*. •

4 (9 points). Define a function *nrOfMessagesGCT* :: *GroupChatTree* → *Int* that returns the number of messages present in a *GroupChatTree*. Define function *nrOfMessagesGCT* as a *fold* on the datatype *GroupChatTree*.

•

5 (9 points). Define a function *messagesMember* :: *GroupChatTree* → *Member* → *Messages* that returns the messages of a particular member in a *GroupChatTree*. Define function *messagesMember* as a *fold* on the datatype *GroupChatTree*. •

6 (9 points).

(a) Give an example of a grammar that can be left-factorized

(b) Left-factorize this grammar

(c) Explain why left-factorization may be a useful grammar transformation

•

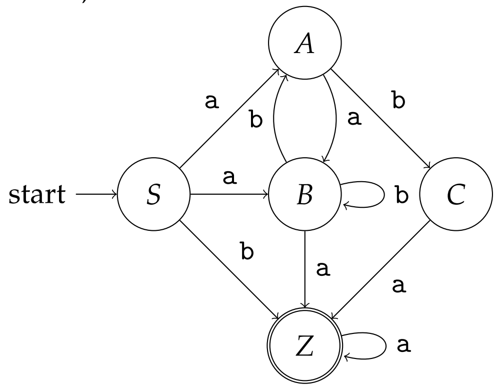
7 (9 points).

(a) Give an example of a grammar that is left-recursive

(b) Remove this left-recursion

(c) Explain why removing left-recursion may be a useful grammar transformation

Consider the following NFA (Nondeterministic Finite-state Automaton), with start state *S*, and final state *Z*.



8 (6 points)**.** Construct a regular grammar with the same language.

9 (6 points). Construct a DFA (Deterministic Finite-state Automaton) with the same language (you may draw a DFA). •

10 (6 points). Suppose we have two context-free grammars $G_1 = (T_1, N_1, R_1, S_1)$ and $G_2 = (T_2, N_2, R_2, S_2)$, where the intersection of N_1 and N_2 is empty. Define $G = (T_1 \cup T_2, N_1 \cup N_2 \cup \{S\}, R_1 \cup R_2 \cup \{S \rightarrow S_1\ S_2\}, S)$, where S is the new startsymbol.

(a) What is the language of G?

(b) This construction does not work for regular grammars. Why not?

(c) Describe the construction of a grammar with the same language as G, which is regular if both G_1 and G_2 are regular.

●