Programming Language Technology

Exam, 08 January 2018 at 8.30{12.30 in J

Course codes: Chalmers DAT150/151, GU DIT231. As re-exam, also TIN321 and DIT229/230.
Exam supervision: Andreas Abel (+46 31 772 1731), visits at 9:30 and 11:30.

Grading scale : Max = 60p, VG = 5 = 48p, 4 = 36p, G = 3 = 24p.
Allowed aid : an English dictionary.
Exam review : Tuesday 16 January 2017 at 10.30-12 in room EDIT 6128.

Please answer the questions in English. Questions requiring answers in code can be answered in any of: C, C++, Haskell, Java, or precise pseudocode.

For any of the six questions, an answer of roughly one page should be enough.

uestion 1 (Grammars): Write a BNF grammar that covers the following kinds of constructs in Java/C++:

- statements:
  - blocks: lists of statements (possibly empty) in curly brackets
  - variable initialization statement: a type followed by an identifier, the equals sign, an initializing expression, and a semicolon, e.g.
  - return statements: an expression between keyword and a semicolon, e.g.
- types:
- expressions:
  - integer literals
  - subtraction -
  - multiplication *
  - pre-increment of variables
  - function calls with zero or more arguments
  - parenthesized expressions .
- Both arithmetic operations are left associative. Multiplication binds stronger than subtraction.

An example statement is:

```
{ int x=f(); return x*-2-3-4*y;
  int y=5; return y;}
```

You can use the standard BNFC categories and . *Do not* use list categories or / rules. *Do not* use either. (10p)

Question 4 [*8 points total*]

On pages 5 7 of this document you nd an OWL ontology in turtle format.
All parts of this question refer to that ontology.

(a) [*4 pts*] Draw an Entity-Relationship diagram (E-R diagram) that shows the classes, relationships and attributes that are de ned in the ontology.

(b) [*1 pts*] Give OWL statements that declare inverses of object properties takesPartIn and isStageOf.

(c) [*3 pts*] Suppose that a database based on the ontology le is implemented in Neo4j. Give a Cypher query that nds the the name of the rider who wins a stage that nishes in Paris.

Question 5 [*4 points total*]

A pizza company uses AI to check the quality of its products:

> DOM Pizza Checker uses advanced machine learning, arti cial intelligence and sensor technology to identify pizza type, even topping distribution and correct toppings.

(Source: https://dompizzachecker.dominos.com.au/)

Discuss whether deploying the DOM Pizza Checker raises potential ethical issues.

Question 6 [*15 points total*]

Assume you need to analyze 27 TB of historical log- le information from a large telecommunication equipment manufacturer. The log les are text les; each row of comma separated values has the same structure,

Error class, Error code, Log Message.

Here error class indicates one of four subsystems memory, CPU, network, power encoded as strings MEMERR, CPUERR, NETERR, PWRERR, respectively in which the error occurred, error type is one of about 26000 numerical 5-digit error codes (in the range 10000,...,37000), and log message is a natural language text giving further details.

The data is made available on a cluster consisting of twenty servers with 64 cores each and local disks, where the 27TB of data is stored in a distributed manner in the Hadoop le system (HDFS).

Solve the problem of reporting the 50 most-frequent error codes in each error class and, additionally, the 10 most-frequent error codes irrespective of error class. Your approach should be ef cient. That is use parallel computations as much as possible, and avoid unnecessary communication.

Sketch your solution either graphically, with bullet points, or commented pseudo-code (or Python code) and explain how the communication between parallel tasks is handled. You do not have to explain File I/O. Do explain the data ow and how the execution progresses to the nal result. Be speci c, so that someone with familiarity with Python, MapReduce and Spark would have enough details to implement your solution. In particular, specify the keys you propose to use in MapReduce or Spark *explicitly*.

(a) [*6 pts*] Propose a solution using the MapReduce-framework. Please give your best-case estimate (that is, ignoring pathological inputs such as 27TB of identical lines in the log) on how many cores/servers each speci c step of your MapReduce solution can be executed in parallel.

(b) [*5 pts*] Propose a solution using Spark. Again, give your best-case estimate on how many cores/servers each speci c step of your solution can be executed in parallel.

(c) [*4 pts*] Discuss the differences and argue whether the Spark model of parallel programming has an advantage over MapReduce for this problem. Please be speci c.

Question 7 [*2 points total*]

As a junior data scientist recently hired to Reynholm Industries you are tasked with overhauling and redesigning the IT infrastructure and accelerating existing computational workloads. A lot of the existing analytics code is utilizing one core. Outline your decision process, including key variables, when deciding whether a speci c program can and should be parallelized.

Question 3 (Parsing): Consider the following BNF-Grammar (written in bnfc). The starting non-terminal is S.

```
S1.   S ::= S ";" C ;
S2.   S ::= C ;

C1.   C ::= C "," A ;
C2.   C ::= A ;

AA.   A ::= "alice" ;
AB.   A ::= "bob" ;
AC.   A ::= "chris" ;
```

Step by step, trace the LR-parsing of the expression

```
alice ; bob , chris
```

showing how the stack and the input evolves and which actions are performed. (8p)

SOLUTION: The actions are S (shift), R (reduce with rule(s)), and Accept. Stack and input are separated by a dot.

```
a = alice
b = bob
c = chris

              . a ; b , c    -- S          AA        C2        S2
a             .   ; b , c    -- RRR:  a ----> A ----> C ----> S
S             .   ; b , c    -- SS         AB        C2
S ; b         .       , c    -- RR:    b ----> A ----> C
S ; C         .       , c    -- SS         AC
S ; C , c     .              -- R:     c ----> A
S ; C , A     .              -- R with C1
S ; C         .              -- R with S1
S             .              -- Accept
```