Examination: Modelling of Complex Systems

Monday 14/01/2019

This exam consists of two parts:

- **CLOSED BOOK**: a written theoretical part (8pts).
- **OPEN BOOK**: a written exercise part (12pts).

The procedure is as follows:

- **For every question, use a new page. Put your name and student number on each page.**
- Solve the theoretical part. When finished, give your solution to the assistant.
- You will then receive a print-out of your project.
- Then you can take your course materials, and start with the exercise part. Allowed materials are:
  - Copies of slides
  - Course notes
  - Nothing more.

The duration of the exam is at most 4 hours.

Studenten die de nederlandstalige master volgen mogen antwoorden in het nederlands..

Success !

Prof. Marc Denecker

1 Theory part (8pt)

1. (4pts) State and prove the inexpressivity theorem and prove the theorem that "A can reach B through graph G" is not expressible in FO. Give one logic in which it can be expressed and show how?

   Or, for a bonus point (5pts), state and prove the inexpressivity theorem for FO and prove that the proposition "The graph G is cyclic" is not expressible in FO. In what logic can it be expressed and how?

2. Small questions (4pts)

   (a) In the course, it is stated that the CTL* state formula $\mathrm{E}[\mathrm{G}\,\mathrm{F}\,p]$ cannot be expressed in CTL nor in LTL. What does it express? Prove that it cannot be expressed in LTL.

   (b) What are the components of $Th(DB)$, the theory of a database structure DB.

   (c) Consider the following FO(.) definition of unary predicate A in terms of parameter $G/2$, a graph predicate.

   $$\left\{\ \forall x(A(x) \leftarrow \forall y(G(x,y) \Rightarrow A(y)))\ \right\}$$

   What is the set A defined here? Draw a small structure in which A is not the complete domain.

   (d) Explain what is the two watched literal technique in SAT-solving.

2 Exercise part (7+5pt)

Some students have exemption of exercise question 2 or 3 or both. They only make the part of the exam they are not exempted of.

2.1 Modeling DMN in FO(.) (5pts)

Decision Modeling Notation (DMN) is a well-known tabular knowledge representation format used in business applications. The DMN tables 1 and 2 represent knowledge for a ship to enter or refuel at an unnamed Dutch harbour.

- The first row in each table specifies the names of the attributes and the possible values.
- The first column in each table is just a numbering of rows.
- The table "Ship Clearance" specifies whether a ship may enter the harbour for loading or unloading as expressed by the attribute "Enter". It is the "output" attribute of this table. As for the other attributes, "Cer. Exp." stands for Certificate Expiration Date; the attribute "Cargo" is a measure of the residual load of the boat; "Length", "Draft" (in Dutch: diepgang), "Capacity" have the obvious meaning.

Ship Clearance						
U	Cer. Exp. (date)	Length (m)	Draft (m)	Capacity (TEU)	Cargo (mg/cm^2)	Enter
	≥ 0	≥ 0	≥ 0	≥ 0	≥ 0	y, n
1	$\leq$ `today`	–	–	–	–	n
2	$>$ `today`	< 260	< 10	< 1000	–	y
3	$>$ `today`	< 260	< 10	≥ 1000	–	n
4	$>$ `today`	< 260	$[10, 12]$	< 4000	≤ 0.75	y
5	$>$ `today`	< 260	$[10, 12]$	< 4000	> 0.75	n
6	$>$ `today`	$[260, 320)$	$(10, 13]$	< 6000	≤ 0.5	y
7	$>$ `today`	$[260, 320)$	$(10, 13]$	< 6000	> 0.5	n
8	$>$ `today`	$[320, 400)$	≥ 13	> 4000	≤ 0.25	y
9	$>$ `today`	$[320, 400)$	≥ 13	> 4000	> 0.25	n

Table 1: DMN representation of the *ship clearance* decision of Figure 1b

Refuel Area Determination				
U	Enter	Length (m)	Cargo (mg/cm^2)	Refuel Area
	y,n	≥ 0	≥ 0	none, indoor, outdoor
1	n	–	–	none
2	y	≤ 350	–	indoor
3	y	> 350	≤ 0.3	indoor
4	y	> 350	> 0.3	outdoor

Table 2: DMN representation of the *refuel area determination* decision of Figure 1b

- E.g., the second row specifies that a ship with valid certificate and length less than $260m$, draft less than $10m$ and capacity less than 100 *TEU* may enter.
- The table "Refuel Area Determination" specifies whether and where a ship may refuel: not, or in the indoor fuel station or in the outdoor fuel station. Note that "Enter" defined in the first table, is an input attribute of this one.

The questions are to express (part of) these tables in FO(.) and use the theories to solve questions. A model should represent a correct decision regarding entry and fuel station of the ship. Solve the following questions:

1. Introduce your vocabulary: you may use types such as rational numbers and dates; for the attribute *Enter* use a constant symbol; for the attribute *RefuelArea*, use a unary predicate; for the other attributes, you may choose. Indicate for each table whether it is a definition or a set of constraints. If it is a definition, express it as a definition. For the first table express line 2 and 8. For the second table, express lines 2 and 3.
2. Now assume that line 3 of the second table is extended with a second possible value 'outdoor', so that in this case, it has value 'indoor' or 'outdoor'. Both choices are acceptable. What changes? Adapt your representation.
3. Assume that for a given ship we know its expiration date and cargo, and furthermore, only that it belongs to the Panamax class of ships for which length, draft and capacity is known to range over given intervals $[l_1, l_2]$, $[d_1, d_2]$ and $[c_1, c_2]$. E.g., $l_1 = 190; l_2 = 250$.

How to represent this information and use your theory and these data to decide whether and where to enter and refuel? Describe input and output of inference tasks in detail.

2.2 Linear Time Calculus: Movable Load (4pts)

In this exercise, you will implement an extension to the first part of the original Rush Hour project and answer some questions about this extension.

You are asked to modify the project according to specification. Those that wish can build from the model solution, at a penalty of loosing 1/3 of the points on that part of the question.

The extension proposed here, is to associate a transferable load with one of the cars. Each instance of the Rush Hour puzzle has one initial car carrying a load in the starting position. This load has a goal car, ie. a car the load should end up on in order to solve the puzzle, in addition to any winning requirements mentioned in the original assignment. A new *transfer* action is introduced, that allows transferring the load from the car carrying this load to a car horizontally or vertically adjacent to the first car.

1. Model the extension in the LTC formalism.

   In order to model this extension, you use the following additional vocabulary (a line starting in an asterisk is a line that was present in the original assignment but has been altered):

```
LTCvocabulary  PackageVoc {
    Init_HasPackage:Car
    HasPackage(Time):Car
    PackageGoal:Car
    CanTransferPackage(Time,Car,Car)

    *type Action constructed from
         { M(Car, Dir, Distance),T(Car,Car) }
    *ActionAt(Time,Action)
}
```

   with the following intended interpretation:

   Init_HasPackage is the car that has the load at time Start.

   HasPackage(t) is the car that has the load at time t.

   PackageGoal is the car that the load should be delivered to in the end. The game cannot end unless this car is carrying the load.

   CanTransferPackage(t,c,c) holds iff. the first car can transfer its load to the second car. This predicate only becomes true if the first car has the package at time t and if both cars are horizontally or vertically adjacent in some way (ie. they touch keep in mind that this can also occur if one car is positioned vertically and the other one horizontally!).

(adapted) Action is the renamed version of the type Move from the original assignment. Represents either a move `M(Car,Dir,Distance)` as before, or a load transfer `T(Car,Car)` from the first car to the second car.

(adapted) Actionat(t,a) is the renamed version of `MoveAt(t,m)` from the original assignment. Holds iff. a is the action performed at time t (if there is any action).

You are also allowed to define any auxiliary predicates and functions (but **not** types) you might require to complete this assignment. Do not forget to change **any** pre-existing aspects of your theory that require changing due to this extension and discuss everything you change.

2. Inferences: Concisely formulate an answer to the following questions:

   (a) Consider the statement 'There is SOME (not predetermined) car that can never obtain the load (within given finite time)'. How would you verify if this statement holds for a given partial structure? Use an FO($\cdot$)-sentence for part of your solution.

   (b) Say we have a partial structure specifying the starting position of a Rush Hour game, where only the position of a single car is not yet known. We want to determine all possible starting positions for this car so that the resulting game is winnable. Explain how to do this using optimal propagation inference.

2.3 Temporal logic and Refinement (3pts)

1. Express the following sentences (not necessarily valid) both in LTL and CTL (if it is possible). For a car c, use the predicates `driveLeft`(c), `driveRight`(c), `driveUp`(c) and `driveDown`(c) to mean that in the current state we will apply the events with the same name. Use the predicate `drive`(c) to denote the disjunction of the four drive predicates. Use the functions `carPos`(c) and `goalPos`(c) to denote the position of the car c, respectively, the goal position of the car c. If you experience ambiguity, explain what you understood by the sentence.

   (a) If a car c drives infinite times to the left, then it drives infinite times to the right.

   (b) Once a car reaches its goal, it doesn't drive anymore.

   What does the following CTL formula mean in natural language for a car c:

$$\mathtt{EG}(\mathtt{A}(\mathtt{driveRight}(c)\ \mathtt{U}\ \mathtt{driveLeft}(c)))$$

2. Refinement: how does the following relate to the original assignment: which of the refinement properties hold?

- Cars with a goal now have an intermediate goal. The game can only be won, if each car with a goal has been on its intermediate goal and is currently on its goal.