Models of Software Systems
Qualifying Examination

Jan 9, 2015

1	/ 40
2	/ 60
Total	/ 100

1. **(40 points) Logic and Proof**

Translate the following argument into Predicate Logic formulas and prove its validity using Natural Deduction. Use only the inference rules in page 4.

(a) Every decision has both positive aspects and negative aspects.
(b) A rational person weighs both positive aspects and negative aspects of each decision.
(c) Some person does not weigh positive aspects and negative aspects of any decision.

(d) Therefore, some person is not rational.

Hint) Use the following set of domains:

D: set of decisions
A: set of aspects
H: set of persons

Use the following predicates:

P(x,y): y is the set of positive aspects of x
N(x,y): y is the set of negative aspects of x
R(x): x is rational
W(x,y,z): x weighs y and z to compare

For example, a translation of (b) might be

$\forall x{:}H;\ d{:}D;\ y,z : \mathbb{P}A \bullet R(x) \wedge P(d,y) \wedge N(d,z) \rightarrow W(x,y,z)$

∧-intro
$$\frac{A, B}{A \wedge B}$$

∧-elim
$$\frac{A \wedge B}{A}$$

∧-elim
$$\frac{A \wedge B}{B}$$

∨-intro
$$\frac{A}{A \vee B}$$

∨-intro
$$\frac{B}{A \vee B}$$

∨-elim
$$\frac{\Sigma, A \vdash C \qquad \Sigma, B \vdash C \qquad A \vee B}{\Sigma \vdash C}$$

→ -intro
$$\frac{\Sigma, A \vdash B}{\Sigma \vdash A \rightarrow B}$$

→ -elim
$$\frac{A, A \rightarrow B}{B}$$

¬-intro
$$\frac{\Sigma, A \vdash B, \qquad \Sigma, A \vdash \neg B}{\Sigma \vdash \neg A}$$

¬-elim
$$\frac{\neg\neg A}{A}$$

∀-intro
$$\frac{P(x)}{\forall x\, P(x)}$$

∀-elim
$$\frac{\forall x\, P(x)}{P(t)}$$

∃-intro
$$\frac{P(t)}{\exists x\, P(x)}$$

∃-elim
$$\frac{\Sigma, P(x) \vdash C}{\Sigma, \quad \exists x\, P(x) \vdash C}$$

Σ is a list of formulas
t is a term
x is not free in C

3. (60 points) State Machines and Temporal Logic

Consider a simple state machine model of an elevator system with three floors. Assume that the elevator has two state variables: *floor*, which represents which floor the elevator is on; and *requested*, which represents which floors have been requested for stops by users. When the elevator attempts to service a floor request, it visits all (and only those) intermediate floors that have outstanding requests.

The values for *floor* are one of the following:

- *On*1: The elevator is on floor 1;
- *On*2*Up*: The elevator is on floor 2 and will give priority to requests to go to the third floor before requests to go to the first floor;
- *On*2*Down*: The elevator is on floor 2 and give priority to requests to go to the first floor before requests to go to the third floor; and
- *On*3: The elevator is on the third floor.

The values for *requested* are *sets* of zero or more of the values f_1, f_2, and f_3 (representing requests for floor one, two and three, respectively). Values are added to the set when a user presses one of the elevator buttons (either inside the elevator, or on a floor), and are removed when an elevator visits the floor requested.

(1) This model abstracts away many aspects associated with a real elevator. List three aspects of a real elevator that are missing from this model.

(2) For each of the following, (i) express the claim in linear temporal logic, and (ii) say whether it is true of this model. (You may use floor values as predicates. For example, *On*1 can be used as a predicate to indicate that the elevator is on floor 1. Similarly, you may use f_i as a predicate to indicate that floor *i* has been requested.)

(a) The elevator visits floor 1 only finitely many times.

(b) In all traces if f_2 is pressed, eventually f_2 will be visited.

(d) The elevator will never go directly from the *On*1 state to the *On*2*Down* state.

(e) The elevator will remain on floor 1 until it moves from floor 1 to floor 2.