Practice On Formulating Conditions

For the project seminar:
"Online Monitoring of Complex Conditions for Event-based Distributed Architectures"

Lucas Sakizloglou

May 21, 2019

Let L be an imaginary temporal logic that supports the following operators:

$$\wedge, \vee, \neg, \exists, \forall, \Box, \rightarrow, U_I, \Diamond_I, S_I, \blacklozenge_I$$

where I is an interval over integers. L is defined over abstract event types of the form:

$$Ent(a_1, a_2, ..., a_n)$$

where *Ent* is an entity/action related to the system at hand and $a_1, a_2, ..., a_n$ are attributes of the entity/action in question. The attribute data type can either be an integer or a string. L also supports basic arithmetic operators, i.e. $=, <, \leq, >, \geq$ for comparison among attributes.

Use L to understand and formulate properties in the following examples.

Example 1 (Business Report Approval)**.** The following example deals with the approval requirements in a business that generates business reports.

1.1 Whenever a report is published, it must be published by an accountant and the report must be approved by her manager within at most 10 timeunits prior to publication.

Let's derive the event types that express the basic actions/entities referred to by the condition:

$$acc(acc_{id})$$
$$mgr(mgr_{id}, acc_{id})$$
$$publish(acc_{id}, r_{id})$$
$$approve(mgr_{id}, r_{id})$$

All the data types of the events in this case are of type *int*. Based on these event types, the property formulation is:

$$1.1 : \Box \forall acc_{id}.\, \forall r_{id}.\, publish(acc_{id}, r_{id}) \rightarrow$$
$$acc(acc_{id}) \wedge \blacklozenge_{[0,10]} \exists mgr_{id}.\, mgr(mgr_{id}, acc_{id}) \wedge approve(mgr_{id}, r_{id})$$

which in natural language reads: Globally (now and for ever) for all accountants and for all reports such that[1] event *published* is true, implies that once in the previous 10 timeunits (starting at this moment) there exists a manager such that the *mgr* event is true for the accountant in question (this means the accountant is managed by this manager) and the manager has approved the report in question.

Example 2 (Operating System Task Handling)**.** The following example deals with an Operating System environment:

2.1 Whenever a handler has been spawned, it means a task has been created in the last 5 timeunits.

2.2 The handler has to be done in the next 10 timeunits.

2.3 Until the handler is done, there should be no termination signal.

Event types:

$$task(t_{id})$$
$$handlerSpawned(h_{id}, t_{id})$$
$$terminate(h_{id})$$
$$handlerDone(h_{id})$$

All the data types of the events in this case are of type *int*. Property formulation:

$$2.1 : \Box \forall h_{id}.\, handlerSpawned(h_{id}, t_{id}) \rightarrow \blacklozenge_{[0,5]}\, \exists t_{id}.\, task(t_{id})$$
$$2.2 : \Box \forall h_{id}.\, handlerSpawned(h_{id}) \rightarrow \Diamond_{[0,10]}\, handlerDone(h_{id})$$

Try to formulate the property 2.3 .

Example 3 (Bank Transactions)**.** The next example is about compliance policy for a banking system that processes customer transactions.

3.1 Executed transactions t of any customer c must be reported as suspicious within at most 5 timeunits if the transferred money a exceeds a given threshold th.[2]

3.2 Transactions that exceed the threshold must be authorized at least 2 timeunits and at most 20 timeunits by some employee e before they are executed.

Try to derive the event types and properties of this example. Reminder: event types can have more than two attributes.

[1] The dot (".") should be read as "such that". Here the two dots are expressed as one for readability.

[2] Assume *th* is a constant number that can be compared to with basic arithmetic operators.