Formal specification development and verification of hybrid systems based on CafeOBJ

Daigo Yamagishi (210097)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 13, 2004

Keywords: CafeOBJ, hybrid systems, HOTS, activity, modeling.

1 Introduction

I have proposed new model which can formally model hybrid systems and formally verify that ones have desired properties. I have ascertained that the model is suitable for this purpose from two experiences based on case studies.

A hybrid system is a system which contains both of digital (or discrete activities) and analogue (or continuous ones) components. Most of such systems are safety critical, but analysing behavior of them with ad hoc ways are difficult. Therefore, we need some formal model and formal analyses based on it.

CafeOBJ is one of formal specification languages. CafeOBJ is able to describe the abstract machines over algebra. Besides, the model of abstract machines which is able to describe used on CafeOBJ is called OTS (Observational Transition System).

In OTS, the abstract machines are expressed by the state space and the group of three set, a set of observable values, the set of initial state and a set of conditional transition rule. States are acquired by observing out of the assumed state space Υ. Observable values are acquired by the observations which functions are returning discrete data-type. Initial state is

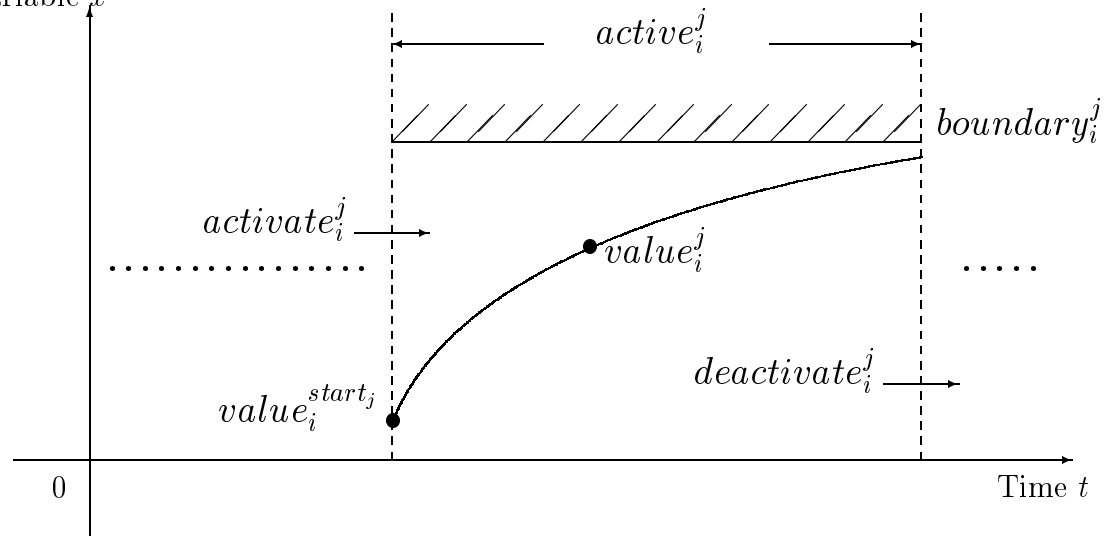

Figure 1: Diagram of an activity

subset of Υ. The values of current states are changed by the conditional transition rules which functions are returning Υ. In addition, extended OTS which describes real-time systems is called TOTS (Timed OTS). New additions from OTS are real variables and time advancing transition rules *tick*. Safety properties with OTS and HOTS are verified by induction.

Although real-time systems which is including TOTS are classified of one of hybrid systems models, it doesn't describe a real variables evolution except for time. In TOTS, it which has only one real observable variable *now* is evolved by *tick*. Therefore I extended OTS which expresses the real variables evolutions called HOTS (Hybrid OTS). Furthermore, for two case studies, I have specified based on HOTS, described using CafeOBJ and verified of safety properties with help of CafeOBJ so that HOTS use like OTS.

2 Concept of the activities

HOTS has two set to the aspects and the activities. One aspect has one or multiple activities. More detail, for aspects, on the following case, it is called aspect, that is, "if we pick up the arbitrary real valuables into an aspect, the variables has existence and uniqueness" Moreover, a set of the activities which controls the values into an aspect includes an aspect. The total number of the aspects and the activities are defined by α and n_i ($i = \{1, ..., \alpha\}$).

In the activities, there are four observations and two conditional transi-

tion rules in an activity. The observations are the $value_i^{start_j}$, the $value_i^j$, the $boundary_i^j$ and the $active_i^j$. The conditional transition rules are the $activate_i^j$ and the $deactivate_i^j$. ($j = \{1, ..., n_i\}$) Fig. 1 is the diagram of an activity. Explanation of the observations is defined as the following.

- The $value_i^{start_j}$ means the initial values in the activities. In an activity, a real variable starts from this number.
- The $value_i^j$ means the current values in the activities. In an activity, a real variable evolves in keeping with a differential equation.
- The $boundary_i^j$ means the invariant values in the activities. In an activity, a real variable doesn't exceed this number.
- The $active_i^j$ means the flags in the activities. In an activity, if a flag is true, then a real variable evolves in an activity. if a flag is false, then a real variable doesn't evolve in an activity.

In addition to the above, HOTS includes a special observation. It is called *now*. The *now* is the master clock in the systems. Explanation of conditional transition rules are defined as the following.

- The $activate_i^j$ means the discrete transitions which allows the activities to evolve the real variables. In j's of i, the conditions of the $activate_i^j$ needs to at least satisfy that all of the $active_i^j$ with the i is false. After the $activate_i^j$, if the condition are satisfied, the real variables are set to the value of the i^{start_j} and the flags of the $active_i^j$ are set to true and the $boundary_i^j$ are set to the arbitrary values.
- The $deactivate_i^j$ means the discrete transitions which stops to evolve the real variables. In j's of i, the conditions of the $activate_i^j$ needs to at least satisfy that the j's $active_i^j$ is false. After the $deactivate_i^j$, if the conditions are satisfied, the flags of the $active_i^j$ are set to false and the $boundary_i^j$ are set to a infinity or a negative infinity.

Each of the conditional transition rules, if the conditions aren't satisfied, the states of the systems keeps up former states. Moreover, HOTS includes a special conditional transition rule. It is called $tick_r$. The r are elements of non-negative real numbers. The $value_i^j$ and the *now* are evolved by the $tick_r$. Conditions of the $tick_r$ are defined as the following,

- The $value_i^j$ after the $tick_r$ does't exceed the $boundary_i^j$.

And then the $value_i^j$ and the *now* are defined as the following,

- The *now* evolves a r.
- In $value_i^j$, if the flags in j's of i are true, The $value_i^j$ are evolved by the differential equation.

3 Conclusion and future works

Case studies pick up the thermostat and crossing-gate examples. Thermostat has one aspect. In contrast, crossing-gate has two aspects. In particular, crossing-gate has two aspects which refer each other . Both examples have satisfyingly modeled by HOTS and described by CafeOBJ. In verification of the safety property, these have needed to find the lemma to prove the time-advancing rule *tick*. In particular, the one of lemma in crossing-gate which needed the difficulty predict. Specifically, lemma was found by the trace of compositional differential equation about aspects.

HOTS is useful model with the examples. Thus, now, HOTS verifies following class, that is,

- liner order differential equation.

Incidentally, in thermostat example, HOTS definition corresponded with hybrid automata definition. Consequently, I guess to describe hybrid automata using CafeOBJ.

Present future works are two. First, to suggest the more general HOTS model, more explicitly,

- calculation of n's order differential results constant.

Second, to propose the decision procedure from continuous to discrete, furthermore, it reflects to the CafeOBJ.