

STU: Stateflow to Uppaal Appendix

Yixiao Yang¹, Yu Jiang^{1,2}, Ming Gu¹, Jianguang Sun¹

Department of Computer Science, University of Illinois at Urbana-Champaign, USA²

School of Software, Tsinghua University, TNLIST, KLISS, Beijing, China¹

Email: {yangyixiaofirst@163.com}

Abstract—This file is used to show the name convention of the Stateflow and Uppaal. The name of a state or variable of Stateflow is changed in Uppaal. We show the change rule in this paper in detail.

Keywords—Stateflow, Uppaal, Conversion, Verification.

I. STATEFLOW UPPAAL DICTIONARY – OVERVIEW

In this appendix, we show the correspondence between elements of Stateflow and Uppaal. In order to verify properties of Stateflow, we need to know what the elements would be after transformed from Stateflow to Uppaal. Variables, names of states and the data which represents total run time of the model are the most cared elements. The maps of these elements from Stateflow to Uppaal are described as follows.

A. Stateflow Uppaal Dictionary – State Map

In Stateflow, any child states of one state must have distinct names. Except for this constraint, any two states could have the same name. Stateflow can distinguish these states with the same name due to their different hierarchies. Uppaal has no hierarchy. Thus two states in Stateflow with the same name of different hierarchies will puzzle Uppaal. In order to distinctively identify each state by its name, we give each state a unique 'full name'. The formal method to construct the 'full name' of a state is described in two steps. First, put all the ancestral states of this state and this state in order. Second, connect the names of these ordered states with the letter '_'.

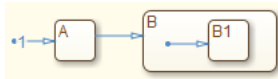


Fig. 1. State Map Example

Here is an example. As is shown in Figure 1.

State A is contained in the 'toppest' state Chart. So A's ancestor is Chart. A's full name is 'Chart_A'.

Similar to state A, state B's full name is 'Chart_B'.

State B1 has two ancestors which are Chart and B. So B1's full name is 'Chart_B_B1'.

B. Stateflow Uppaal Dictionary – Variable Map

Variables in Stateflow are defined in a state. We name the state in which the variable is defined 'the parent state of the variable'. A scope is assigned to every variable in Stateflow, which is just the region in which the variable is valid. In detail, the scope of a variable in Stateflow is the variable's

parent state itself and its descendants. In Stateflow, there may be two variables with same name defined in different scopes. But in Uppaal, a variable does not possess scopes. In order to distinguish variables with same name in different scopes, we need to add a prefix to the name of the variable when translating it to Uppaal. The prefix is just the full name of the variable's parent state. The 'full name' of a state has already been introduced in the previous sub-section 'State Map'. To Make it look better, we add an underline between the prefix and the raw name of variable. Here is an example.

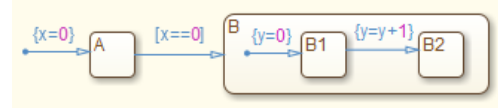


Fig. 2. Variable Map Example

Variable x is defined in the 'toppest' state which is also called 'Chart'. So the name of x in Uppaal is 'Chart_x'.

Variable y is defined in state B. The full name of state B is 'Chart_B'. So the name of y in Uppaal is 'Chart_B_y'.

C. Stateflow Uppaal Dictionary – Time Map

Here the word 'Time' refers to the implicit Stateflow variable which represents the current run time of the model. This variable is shown in the status bar of Stateflow UI.



Fig. 3. Time Representation in Stateflow

The variable 'T=8.000' in the Figure shown above is just what we are talking about. In many cases, we may want to check whether some properties are held at some specific time. In these time critical situations, this variable is especially useful.

We map this variable to an integer variable in Uppaal named as 'mTotalTime'. This variable 'mTotalTime' is defined in the Daemon Automaton with the fixed name 'Process_all_controller'.

The automaton 'Process_all_controller' which is also called 'Daemon Automaton' is a fixed automaton which is same in any transformations of any Stateflow models. This automaton is designed to simulate the running environment of Stateflow.

When we want to refer to this variable, for example, we want to judge whether this variable equals to 2, we just write the following : 'Process_all_controller.mTotalTime == 2'.