A computational model for biomedical system.

Sungeetha D†, Vasumathi K Narayanan

1Sathyabama University Chennai, India
2Department of Information Technology, St. Joseph’s College of Engineering, Chennai, India

Abstract

In this paper, we discuss about the specification of timed state transition systems that communicate by binary and rendezvous interactions. The observed approach unfolds the state machines according to synchronization and timing requirements. The unfolded state transition diagrams has been used to approximate a global time and to verify the safety, liveness and fairness properties of the system. A Master, Worker and Assembler of laser speckle image processing system in biomedical engineering are used as running example to illustrate the approach and properties.

Keywords: Timed CFSMs, Unfolded CFSMs, State explosion, Interleaving semantics, Partial-ordered semantics.

Accepted on January 16, 2017

Introduction

With the increasing use of computers in medical imaging, there is a pressing need for designing more reliable systems. As a result, developing formal methods for the design and analysis of concurrent systems has become an active area in computer science and biomedical engineering research. Traditional testing method such as simulation is inadequate for developing bug-free complex and concurrent systems. One approach to ensure correctness is to employ automatic verification method. A verification formalism consists of [1]:

• A formal semantics to assign mathematical meaning to system components and its correctness criteria;
• A language for describing the constructs for combining the components;
• A specification language for expressing the correctness requirements;
• A verification algorithm to ensure that correctness criteria are fulfilled in every possible execution of the system.

Real-time system performances are used in medical image processing system. Failures in such systems can be very expensive and even life-threatening and consequently there is a great demand for formal methods that should be applied to real-time systems. Different formalisms have been proposed to reason reactive systems. These include process algebra, temporal logics, Petri Nets, automata theoretic techniques and partial-order models [1]. In the traditional approach for verification of concurrent programs, the correctness of the program is expressed by a formula in first-order temporal logic. The verification reduces to proving a theorem in a deductive system. Model-checking provides a different approach to check the properties of finite state systems [1-8].

The model-checking approach used for program verification is probably the most exciting advance in the theory of program correctness in recent years. The main difficulty in using model-checking approach is the state-space explosion problem. The size of the global state-transition graph grows exponentially with the number of components in the system. There are many ways to avoid this problem [1-8]. In this work, the partial-ordered unfoldings of CFSMs (Communicating finite state machine) are constructed to avoid the state-explosion in constructing a single total-ordered global-state transition graph (also known as product machine). The above method is achieved by simulating each local CFSMs in the presence of non-local CFSMs and performing the synchronizations. Section 2 defines the preliminaries of the CFSMs and their unfolding using a real time laser speckle image processing in biomedical image processing system.

Timed CFSMs and their Unfoldings

Here it is assumed that specification of n CFSMs are non-terminating [2]. The CFSMs interact by synchronous message-passing via a binary rendezvous based on the seminal work of CSPs (Communicating Sequential Processes) [9,10]. The simultaneous n-ary (multiparty) interaction is split into a sequence of two party rendezvous interactions using a queue.

In a set of n communicating and non-terminating FSMs, each CFSM is defined as a 7-tuple:

Definition 1. A CFSM $F_i = (s_{0fi}, S_{fi}, A_{fi}, C_{fi}, R_{dfi}, R_{async_{fi}}, R_{async_{0fi}})$ where,
$S_0$ is the finite set of states of CFMS $F_i$, $s_{0i} \in S_0$ being the initial state.

$A_0$ is the finite set of asynchronous and synchronous actions of $F_i$.

$C_0$ is the set of clock variables of $F_i$

$R_{0i}$ is a transition relation such that:

In i-transition $(s_{fi}, a_{fi}, g_{fi}, u_{fi}, s'_{fi}) \in R_{0i}$, $s_{fi}$ is called the input state and $s'_{fi}$ the output state, $a_{fi}$ is the action label, $g_{fi}$ is the guard which is the enabling condition of the transition based on clock variables $C_{fi}$, (default being True), $u_{fi}$ is the update statement again based on clock variables (default being Null statement) executed upon the transition. If the enabling condition $g_{fi}$ is True, the transition is considered instantaneous and if $g_{fi}$ is of the form $(x_i \geq t_i)$, $x_i \in C_{fi}$, the transition takes at least $t_i$ time units to enter the next state $s'_{fi}$ from $s_{fi}$. In the case of a synchronous transition, the send action is denoted as $A_0^i$ and the corresponding receive action as $A_0^i$? in the partner CFMS $F_{j}$, $i \neq j$.

• is a binary relation which relates the output states of synchronous transitions.

• relates the set of pairs of initial states: All the initial states are assumed to be in pairwise synchronous with each other to begin with.

• Figure 1 shows an example of worker, master and assembler CFMS [11]. When there is more than one worker in a simultaneous approach, the master serves one worker at a time, making the other worker wait in a queue.

The unfolding of CFMSs

Definition 2: The unfolded CFMS $M_i = (s_{0i}, A_i, C_i, R_{0i}, R_{synci}, R_{sync0i})$, whose 7-tuple entities are generated as instances of corresponding entities of CFMS $F_i$, $i \in \{1..n\}$, unfolded in time.

Figure 2 shows the unfolded CFMSs of the worker $t_i$, master and the assembler. The master is considered as the anchoring agent as it synchronizes both the worker agent and assembler agent/process. $T_{ti}$, $T_{mi}$ and $T_{ai}$ are the variables monitoring the local times of the worker, master and assembler agents, respectively. The local time of each agent at every state-entry, is initialized to zero. Thus $T_{ti}=0$, $T_{mi}=0$ and $T_{ai}=0$, respectively at states $Sw_{0i}$, $Sm_{0i}$ and $Sa_{0i}$. At every local output state entry, for instantaneous transition, the time remains the same as that of the input state entry. For non-instantaneous transition the local time is updated corresponding to the duration of the transition as dictated by its guard $g_{fi}(C_{fi})$. For instance, if the synchronous transition pair appri! and appri? respectively from states $Sw_{1i}$ and $Sm_{0i}$ are instantaneous then $T_{ti}=0$ and $T_{mi}=0$ at the respective output states $St_{1i}$ and $Sm_{1i}$. At every rendezvous/synchronization point, the state entry times of both synchronous output states are normalized to maximum of the corresponding local times. After the transition pair lower! and lower? from states $Sm_{1i}$ and $Sa_{0i}$ respectively, the local clock constraint of guard $z<1$ dictates $T_{mi}=1$ at $Sm_{2i}$. Even though there is no such constraint for the assembler agent’s lower? action, the local time $T_a$ at $Sa_{1i}$ is normalized to the maximum one, viz., $T_{mi}=1$, same as master. Thus the local time is propagated at synchronization point and becomes logically a semi-global time. Hence the anchoring agent master, propagates the worker’s local time to the assembler agent and vice versa. Rest of the local time updates are shown in Figure 2.

Well-founded, partially-ordered causality generation

When the CFMS graphs are unwind in their mutual global environments into trees by simulating each of the former in their respective non-local environments, it tends to generate a global causality which is a partial-order. The global, temporal causality order is composed using the binary relations $R_{synci}$ and $R_{0i}$, $i \in \{1..n\}$ as follows [2]:

$$\leq ::= (R_{0i} U R_{synci})$$

The binary successor relation $R_{0i}$ is $R_{0i}$ with input and output states related. The binary relation represents the partially ordered, and well-founded causality relation among the states of CFMSs in the unfolding based on their points of entry in time. The $R_{synci}$ relations capture the equality in time of the synchronous output states.

The unfolded timing diagram is shown in Figure 2. Each local state of the unfolding CFMS stores the local time, which is updated according to local clock constraint after asynchronous local transitions. After every synchronous transition, the local time is updated to the maximum of the two synchronizing FSMs. Thus, an approximate global time is synthesized by propagating the maximum advanced local times of the communicating FSMs.

Figure 1. Graphical representation of the 3 CFMSs of worker, master and assembler system.

Initially the worker-i is in state $Sw_{i0}$ and the master is in state $Sm_{0i}$ and the assembler in $Sa_{0i}$. The approach of the first worker is captured by the pair of transitions approach!(i) by the sending worker and approach? by the receiving master from the corresponding local states $Sw_{i0}$ and $Sm_{0i}$ respectively. If there is more than one worker approaching simultaneously, one of them is selected from the head of the list/queue $L$ and the others are made to wait in the queue. $T_{ti}$, the local time of worker $ti$ is set to 0, $T_{mi}=0$, and similarly the local time of master is set to 0, $T_{ai}=0$ after the instantaneous transition pair approach!/approach? reaching states $Sw_{1i}$ and $Sm_{1i}$. Next, the master sends lower! signal to the assembler which receives it by its corresponding lower? transition within a maximum of 1 minute. Therefore at the end of lower!/lower? pair of synchronous transitions, the local
times of master and assembler are, \( T_m = 1 \) and \( T_a = 1 \) at states \( S_m_2 \) and \( S_a_3 \) respectively. Next, the worker \( t_i \) makes a local asynchronous transition in \( x > 2 \) minutes. If \( T_{ti} = 3 \), at state \( S_w_{i2} \) following the local clock constraint of \( x > 2 \). Then, worker \( t_i \) makes another local transition out which is instantaneous, reaching \( S_w_{i3} \). The local time of worker \( t_i \) remains \( T_{ti} = 3 \) in state \( S_w_{i3} \) and so on. The remaining transitions and local times are shown in Figure 2. Because of the fact that the master is synchronizing with assembler and the worker \( t_i \) independently and at each synchronization point the local times are updated to the maximum, deducing the logical global time of the observer, observing all the three parties/processes [12-14].

Figure 2. The three unfolded CFSMs of Worker, Master and Assembler system.

The elimination of state-explosion due to non-deterministic interleaving in unfolded CFSMs

Consider a conventional product machine composed of two component CFSMs. Assume a state vector \((s_1, s_2)\) of the product machine with \(s_1\) representing the state of CFSM1 and \(s_2\) representing that of CFSM2. Consider two local/ asynchronous state transitions \((s_1, a_1, s'_1)\) and \((s_2, a_2, s'_2)\) of CFSM1 and CFSM2, respectively. In the traditional construction of the product machine, these two transitions are modeled by choosing either one of the two transitions to happen first followed by the other in sequence. This would lead to the following product machine state transitions as shown in Figure 3. Consisting of \(2^2 = 4\) transitions representing the 2 independent local/asynchronous transitions of CFSM1 and CFSM2. To extend this representation of non-deterministic interleaving to \(n\) different local/asynchronous transitions of \(n\) respective CFSMs, CFSM1, CFSM2 … CFSMn, it would need \(2^n\) transitions to represent \(n\) independent asynchronous transitions of the \(n\) CFSMs in the case of product machine. On the other hand, in the case of the unfolded CFSMs, the \(n\) local/ asynchronous transitions of the \(n\) CFSMs would be modeled by exactly \(n\) transitions. This is the main advantage of the partially-ordered CFSM unfoldings where the exponential explosion of states due to non-deterministic interleaving is avoided due to maintains of the locality of each CFSM.

Conclusion

A specification of timed CFSMs with a worker, assembler, master of laser speckle image processing system in biomedical engineering, is illustrated to specify and model-check the safety, liveness and fairness properties of a timed communicating agent. With the help of temporal logic verification model checking is to be done in author’s future work. The state-explosion problem of model-checking is eliminated by avoiding the product machine construction of interleaving semantics, using the unfolded CFSMs of partial-order semantics.

References


*Correspondence to
Sungeetha D
Sathyabama University Chennai
India