

# Stuttering Refinement on Partial Systems

Shiva Nejati     Arie Gurfinkel

Department of Computer Science, University of Toronto,  
Toronto, ON M5S 3G4, Canada.

Email: {shiva, arie}@cs.toronto.edu

We discuss the problem of stating and refining partial specifications of software systems at different levels of abstraction. In general, refinement is the process of deriving an implementation from a specification while verifying the correctness of the transformation.

In this approach, we define a refinement relation to relate a specification, given as a partial state transition system with some variables and transitions marked as incomplete, and an implementation, which takes the form of a complete (or fully-defined) state transition system. This refinement relation may also be used as a basis for stepwise-refinement, where at each level we replace a partial system with a more complete one.

Partial specifications are desirable because they do not force us to commit to all decisions made about the behavior of the system in initial specifications. Here, partiality of the specification is obtained by a knowledge ordering relation describing the *degree of certainty* for the values of variables and transition labels. The degree of certainty allows us to express what parts of the specification are complete, and what parts are still unknown.

We also need to check the correctness of the implementation against the initial specification. Hence, the refinement relation is required to preserve properties of the partial specification in the implementation. Therefore, a logical characterization of the proposed refinement relation is studied.

We use finite-state transition systems with 3-valued transitions and propositions for our modeling formalism. Some examples of such systems are:  $\lambda$ Kripke structures of [4], partial Kripke structures of [2, 3], Kripke Modal Transition Systems of [6], and Modal Transition Systems of [7]. In this approach,  $M$ , i.e. the third truth value of the 3-valued logic, is interpreted as *unknown* and represents uncertainty. The truth and knowledge orderings for the 3-valued logic are shown in Figure 1(a).

Some examples of our modeling formalism and the refinement relations that we want to characterize are illustrated in Figure 1. Figure 1(b) represents a partial system whose status is controlled by a button. In this partial system, it is admissible (but not guaranteed) that the system can

change its status from  $\neg active$  to  $active$  no matter whether  $Btn_1$  is  $T$  or  $F$ . A possible refinement of this system (Figure 1(c)) starts with  $Btn_1 = F$  and guarantees that the status can change from  $\neg active$  to  $active$  when the button is pressed.

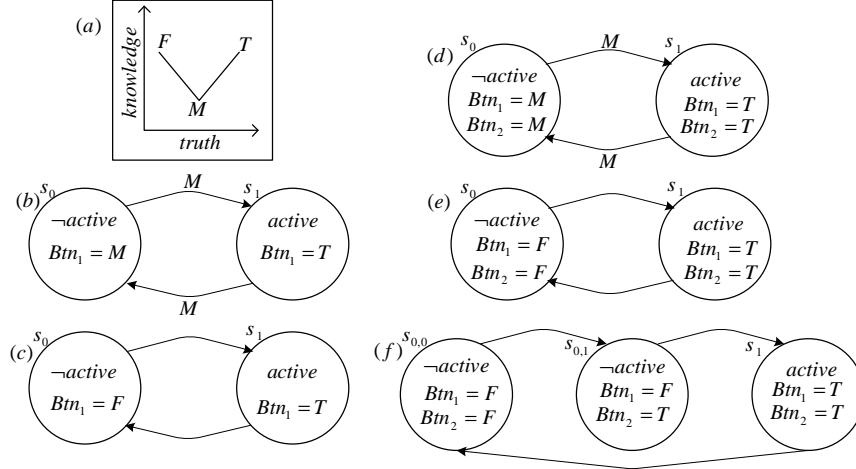
The refinement relation between models (in Figures 1(b) and 1(c)) can be expressed by extending the notion of bisimulation relation [8] from the context of complete (2-valued) models to the context of partial (3-valued) models. The completeness preorder proposed in [2, 3] and the refinement preorders proposed in [6, 7] are examples of such refinement relations.

Note that at each refinement step we compare systems at different levels of abstraction, where a single state of the partial system may be related to several states of a more complete system. For this reason, the refinement relations that we consider do not correspond to bisimulation.

For example, Figure 1(d) illustrates a partial system which is similar to the system in Figure 1(b), but this time two buttons that are wired sequentially control the status of the system, i.e. the system is  $active$  when both buttons are pressed. The model in Figure 1(e) is a possible refinement of the partial system in Figure 1(d) that further requires both buttons to be pressed simultaneously. Alternatively, consider the system in Figure 1(f) in which buttons can be pressed one at a time. Clearly it is also a possible refinement of the system in Figure 1(d); however, in this refinement, the system enters the  $active$  state after two transitions.

The refinement relation between models in Figure 1(d) and 1(f) has to be insensitive to stuttering (finite repetition of states), because state  $s_0$  in Figure 1(d) is related by two states, i.e.  $s_{0,0}$  and  $s_{0,1}$  in Figure 1(f) in such a way that each variable in  $s_{0,0}$  and  $s_{0,1}$  has a degree of certainty higher than that in state  $s_0$ .

Therefore, in order to capture different relationships between models at different levels of abstraction, we need to define a refinement preorder relation that takes finite stuttering into account. In [1, 5], an equivalence relation with respect to finite stuttering is defined. It is shown that states related by the stuttering equivalence relation satisfy the same



**Figure 1. Different refinement relations: (a) 3-valued logic; (b) partial system with one button; (c) refinement of system (b); (d) partial system with two buttons; (e) refinement of system (d); (f) stuttering refinement of system (d);**

$\text{CTL}_{-X}$  properties, i.e. CTL properties without the next-time operators.

We define a stuttering refinement preorder, denoted  $\preceq_{stut}$ , which is sensitive to divergence (infinite internal computation) over 3-valued  $\lambda$ Kripke models. At each refinement step, we only allow structures whose states are never deadlocked, i.e. each state always has at least one non-false outgoing transition. We establish the following theorem which shows that stuttering refinement preorder is logically characterized by  $\text{CTL}_{-X}$  properties.

**Theorem 1**  $s \preceq_{stut} t$  if and only if  $\forall \varphi \in \text{CTL}_{-X}$

$$s \models \varphi \Rightarrow t \models \varphi \text{ and } s \models \neg \varphi \Rightarrow t \models \neg \varphi$$

Intuitively the above theorem means that every  $\text{CTL}_{-X}$  property that evaluates to true or false in a partial state  $s$  has the same truth value when evaluated in a more complete state  $t$ , but properties that evaluate to  $M$  in state  $s$  have an arbitrary truth value in state  $t$ . More precisely, a state  $s$  is less refined than a state  $t$  with respect to stuttering refinement if and only if the value of every  $\text{CTL}_{-X}$  formula  $\varphi$  in state  $s$  has a lower degree of certainty than that in state  $t$ .

In conclusion, we developed the notion of stuttering refinement and showed a logical characterization for it. This notion can be used either for determining the relationship between partial models at different levels of abstraction or for deriving sound abstractions from low-level programs to verify arbitrary formulas [2, 3, 6]. We plan to continue our work in automating the proposed refinement relation and extending the idea to other multivalued logics with uncertain values. We also plan to study how fairness constraints affect refinement relations over partial models.

**Acknowledgements:** We thank Marsha Chechik for her extensive assistance in formalizing our ideas and improving their presentation for this paper.

## References

- [1] M. C. Browne, E. M. Clarke, and O. Grumberg. “Characterizing Finite Kripke Structures in Propositional Temporal Logic”. *Theoretical Computer Science*, 59(1-2):115–131, 1988.
- [2] G. Bruns and P. Godefroid. “Model Checking Partial State Spaces with 3-Valued Temporal Logics”. In *Proceedings of CAV’99*, LNCS 1633, pages 274–287. Springer, 1999.
- [3] G. Bruns and P. Godefroid. “Generalized Model Checking: Reasoning about Partial State Spaces”. In *Proceedings of CONCUR’00*, LNCS 1877, pages 168–182. Springer, 2000.
- [4] M. Chechik, B. Devereux, S. Easterbrook, and A. Gurfinkel. “Multi-Valued Symbolic Model-Checking”. *ACM Transactions on Software Engineering and Methodology*, 2003. (Accepted for publication.)
- [5] R. de Nicola and F. Vaandrager. “Three Logics for Branching Bisimulation”. *Journal of the ACM (JACM)*, 42(2):458–487, 1995.
- [6] M. Huth, R. Jagadeesan, and D. A. Schmidt. “Modal Transition Systems: A Foundation for Three-Valued Program Analysis”. In *Proceedings of ESOP’01*, LNCS 2028, pages 155–169. Springer, 2001.
- [7] K. Larsen and B. Thomsen. “A Modal Process Logic”. In *Proceedings of LICS’88*, pages 203–210. IEEE Computer Society Press, 1988.
- [8] R. Milner. “A Calculus of Communicating Systems”. LNCS 92. Springer-Verlag, 1980.