Hybrid Modal Model Checking

Alessandro Mosca, Daniele Codecasa, Ignazio Gentile, Luca Manzoni

Department of Computer Science, Systems and Communication (DISCo)
University of Milano-Bicocca, Viale Sarca 336, 20126 Milano, Italy
alessandro.mosca@disco.unimib.it

The term Hybrid Logic covers a number of logics obtained by adding further expressive power to ordinary modal logic. The most basic hybrid logic is obtained by adding so-called nominals which are propositional symbols of a new sort, each being true at exactly one possible world.
(Torben Braner, *Hybrid Logic*, The Stanford Encyclopedia of Philosophy (2008 Edition), Edward N. Zalta (ed.))

The current technological trend depicts a scenario in which space, and more generally the environment in which the computation takes place, represents a key aspect that must be considered in order to improve systems context awareness, even if the kind of information processed is not only of spatial nature.

Past works by one of the authors of this paper have shown that Hybrid Logics are a powerful and rich formalism to model Qualitative Spatial Reasoning for context aware systems. Classes of commonsense spatial models can be defined as classes of relational structures according to the formal properties of the involved spatial relations. The relational structures can therefore provide the semantics for spatial hybrid languages, whose formulas represent contextual information (see on this, S. Bandini, A. Mosca, M. Palmonari, *Commonsense Spatial Reasoning for Information Correlation in Pervasive Computing*, Applied Artificial Intelligence, Taylor&Francis, Vol. 21/3-4, April 2007).

The present research project deals with the design and the implementation of Model Checker for Hybrid Modal Logics. With this aim in mind, special attention will be given to the definition and the management of the models and of the hybrid formulas.

The functionality of the model checker is based on back-end modules that receive as input a file containing a hybrid modal formula and a file with the specification of a labeled graph structure, representing a model. Models have been implemented as XML files, by means of the definition of suitable DTD schema. We implement two different DTD, the first one is compatible with the Dragone's models (see on this, the work *Model checking for hybrid logics (with an application to semistructured data)* by M. Franceschet and M. de Rijke, published in the Journal of Applied Logic, 4:3, 2006), while the second extends the first one by allowing the possibility to introduce n-ary relations among worlds. The figure below shows an example of the XML-based specification of a model we proposed, together with its diagrammatic representation.

Formulas have been specified in a 'lisp-like' format, the features of which are

```
<hl-kripke-struct name="M">
     <world label="w1"/>
     <world label="w2"/>
     <world label="w3"/>
     <modality label="pi1" property="reflexivity">
          <acc-pair to-world-label="w2" from-world-label="w1"/>
          <acc-pair to-world-label="w3" from-world-label="w1"/>
     </modality>
     <modality label="pi2" property="simmetry">
          <acc-pair to-world-label="w3" from-world-label="w2"/>
     </modality>
     <modality label="pi3" property="transitivity reflexivity">
          <acc-pair to-world-label="w1" from-world-label="w2"/>
          <acc-pair to-world-label="w2" from-world-label="w3"/>
     </modality>
     <prop-sym label="p" truth-assignments="w1 w2"/>
     <prop-sym label="q" truth-assignments="w3"/>
     <prop-sym label="r" truth-assignments="w1 w3"/>
     <nominal label="i" truth-assignment="w1"/>
     <nominal label="j" truth-assignment="w2"/>
     <nominal label="k" truth-assignment="w2"/>
     <relation label="r">
          <relation-element vector="w1 w2 w3"/>
     </relation>
</hl-kripke-struct>
```

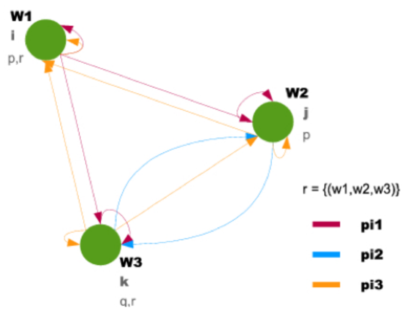

Figure 1: A model for a hybrid logic is a graph structure, made of a set of nodes (or 'possible worlds'), and a set of arcs (or 'accessibility relations'). To each node of the structure is associated a label, and the set of propositions true at that world. An arc is identified by a label, we called 'modality', that represent the modal operator to which it is associated, by an attribute formally characterizing the type of the arc (e.g. simmetric, reflexive, antisymmetric), and by the set of couple of worlds it link together.

constrained by a clear defined BNF grammar. According to the possibility to introduce n-ary relations in the model, the specification of the formulas support the exploitation of n-ary modal operators (of course, formulas may contain also 'binders', 'existential' and 'universal' quantification).

The implementation of the back-end allows to answer different reasoning tasks, with respect to those inputs, i.e. validity in a model, satisfiability in a model, and unsatisfiability. The model checker therefore support the checking of the satisfiability of a formula with respect to some specified world of the model, as well as with respect to all the set of worlds. Finally, in the case of a unsatisfiable formula, the model checker is able to provide the list of the worlds of the models that falsify the formula.

The latest version of the model checker we implemented provides a parallel implementation of the model checking tasks (as a 'multi-threaded execution'), thus optimizing the use of the resources and increasing the efficiency of the model checker. Further works will consider the introduction of specific optimization techniques to treat with long formulas and formulas with several nested operators.