Local Model Checking Games
(Extended Abstract)

Colin Stirling*
Department of Computer Science
University of Edinburgh
Edinburgh EH9 3JZ, UK
email: cps@uk.ac.ed.dcs

1 Introduction

Model checking is a very successful technique for verifying temporal properties of finite state concurrent systems. It is standard to view this method as essentially algorithmic, and consequently a very fruitful relationship between temporal logics and automata has been developed. In the case of branching time logics the connection has not been quite so tight as tree automata are not naturally the correct semantics of programs. Hence the introduction of amorphous automata with varying branching degrees, and the use of alternating automata.

Local model checking was proposed as a proof system approach to verification which also applies to infinite-state concurrent systems. In part this was because it predominantly uses the process algebra model of concurrency (such as CCS) where a concurrent system is presented as an expression of the calculus. The question of verification is then whether a particular expression has a temporal property (rather than all the states of a transition system which have a property). In local model checking the proof system is developed in a goal directed fashion, that is a top down approach. When a property holds, there is a proof tree which witnesses this truth. It also allows there to be proofs for infinite-state systems. Moreover local model checking permits compositional reasoning, when a proof tree may be guided by the algebraic structure of the system as well as the logical structure of the formula expressing the property.

In this talk we show that in the finite-state case these two approaches, model checking as essentially algorithmic and model checking as a proof system, can be combined using games as an underlying conceptual framework that can enjoy the best of both worlds. The automata theoretic approach is captured via the resulting game graph (which is an alternating automaton), and on the other hand a witness for non-emptiness of a game graph is just a proof tree. Alternatively a game graph can be translated into a formula of boolean fixed point logic, which has provided a variant framework for model checking. Game playing also provides a very perspicuous basis for understanding branching time temporal logics and model checking of finite and infinite-state systems.

An important question is whether finite-state model checking of modal mu-calculus properties can be done in polynomial time. Emerson showed that this

* Research in part supported by ESPRIT BRA project Concur2.
problem belongs to NP ∩ co-NP. Games provide a very direct proof of this result. It appears that finer structure needs to be exposed to improve upon this. We believe that new insights may come from the relationship between these games and other graph games. For example model checking games can be reduced to simple stochastic games, an observation due to Mark Jerrum, whose decision procedure also belongs to NP ∩ co-NP.

2 Modal Mu-Calculus

Modal mu-calculus, modal logic with extremal fixed points, was introduced by Kozen [13]. Formulas of the logic given in positive form are defined by

\[ \Phi ::= Z \mid \Phi_1 \land \Phi_2 \mid \Phi_1 \lor \Phi_2 \mid [K]\Phi \mid \langle K \rangle \Phi \mid \nu Z.\Phi \mid \mu Z.\Phi \]

where \( Z \) ranges over a family of propositional variables, and \( K \) over subsets of an action set \( \mathcal{A} \). The binder \( \nu Z \) is the greatest whereas \( \mu Z \) is the least fixed point operator.

Modal mu-calculus with action labels drawn from \( \mathcal{A} \) is interpreted on labelled transition systems \((\mathcal{P}, \{ \xrightarrow{a} : a \in \mathcal{A} \})\) where \( \mathcal{P} \) is a countable but non-empty set of processes (or states), and each \( \xrightarrow{a} \) is a binary transition relation on \( \mathcal{P} \). Labelled transition systems are popular structures for modelling concurrent systems especially process calculi such as CCS [17]; \( \mathcal{P} \) is then a transition closed set of process expressions and \( E \xrightarrow{a} F \) means that \( E \) may evolve to \( F \) by performing the action \( a \).

Assume a fixed transition system \((\mathcal{P}, \{ \xrightarrow{a} : a \in \mathcal{A} \})\), and let \( \nu \) be a valuation which assigns to each variable \( Z \) a subset \( \nu(Z) \) of processes in \( \mathcal{P} \). Let \( \nu[\mathcal{E}/Z] \) be the valuation \( \nu' \) which agrees with \( \nu \) everywhere except possibly \( Z \) when \( \nu'(Z) = \mathcal{E} \). The subset of processes in \( \mathcal{P} \) satisfying an arbitrary formula \( \Psi \) under the valuation \( \nu \) is inductively defined as the set \( \| \Psi \|_\mathcal{P} \) where for ease of notation we drop the superscript \( \mathcal{P} \) which is assumed fixed throughout:

\[
\begin{align*}
\|Z\|_\nu &= \nu(Z) \\
\|\Phi \land \Psi\|_\nu &= \|\Phi\|_\nu \cap \|\Psi\|_\nu \\
\|\Phi \lor \Psi\|_\nu &= \|\Phi\|_\nu \cup \|\Psi\|_\nu \\
\|\langle K \rangle \Phi\|_\nu &= \{E \in \mathcal{P} : \text{if } a \in K \text{ and } E \xrightarrow{a} F \text{ then } F \in \|\Phi\|_\nu\} \\
\|\{K\}\Phi\|_\nu &= \{E \in \mathcal{P} : E \xrightarrow{a} F \text{ for some } a \in K \text{ and } F \in \|\Phi\|_\nu\} \\
\|\nu Z.\Phi\|_\nu &= \bigcup\{\mathcal{E} \subseteq \mathcal{P} : \mathcal{E} \subseteq \|\Phi\|_{\nu[\mathcal{E}/Z]}\} \\
\|\mu Z.\Phi\|_\nu &= \bigcap\{\mathcal{E} \subseteq \mathcal{P} : \|\Phi\|_{\nu[\mathcal{E}/Z]} \subseteq \mathcal{E}\}
\end{align*}
\]

Any formula \( \Phi \) determines the monotonic function \( \lambda \mathcal{E} \subseteq \mathcal{P} \). \( \|\Phi\|_{\nu[\mathcal{E}/Z]} \) with respect to the variable \( Z \), the valuation \( \nu \), and the set \( \mathcal{P} \). Hence the meaning of the greatest fixed point is the union of all postfixed points, and it is the intersection of all prefixed points in the case of the least fixed point. One consequence is that the meaning of \( \nu Z.\Phi \) is the same as its unfolding \( \Phi(\sigma Z.\Phi/Z) \) (where

\footnote{It is very convenient to allow sets of labels to appear in modalities instead of the usual single labels.}

\footnote{If \( E \in \mathcal{P} \) and \( E \xrightarrow{a} F \) then \( F \in \mathcal{P} \).}