Verification of Open Systems Based on Explicit State Space Representation [1]

Irem Aktug Dilian Gurov

Inst. Microelectronics and Information Technology
Royal Institute of Technology
Stockholm, Sweden
E-mail: {irem,dilian}@ imit.kth.se

Abstract

When verifying behavioural properties of a system, one has sometimes to deal with components for which there is no implementation available yet, but only a specification of its behaviour. Such a *component specification* denotes a potentially infinite set of possible implementations of the component, which induces a potentially infinite set of possible system behaviours. A behavioural property of such an *open system* is then a property shared by *all* possible system behaviours. To support verification of open system properties based on state space exploration, these behaviours have to be suitably represented in a finite structure.

In this paper we propose a representation of the state space of open systems for which the behaviour of the yet unavailable components is specified in the modal μ-calculus, and a proof system for verifying open system properties written in the same logic. Following earlier work on modular verification for various fragments of the logic, we base the state space representation on the notion of modal transition systems, suitably extended to deal with disjunctive assumptions and with least fixed point assumptions. The latter are captured through a set of prohibited infinite runs. The representation, called *extended modal transition system*, has been tailored to allow (a) convenient construction from an open system description, and (b) verification of open system properties. For the latter task, we adapt an existing proof system for infinite state systems due to Stirling and Bradfield. Soundness and (relative) completeness of our proof system are established through a translation between the two systems. The problem of constructing a finite state space representation from a given open system description will be addressed elsewhere.

Key words: modal transition systems, verification, proof system

[1] Work partially supported by the SEFROS project of the Swedish Research Council.

This is a preliminary version. The final version will be published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

1 Introduction

When verifying the behaviour of a system based on state space exploration, it is usually assumed that the system's implementation is available at verification time, and thus an accurate representation of the system's behavior can be extracted. Such a *closed world* assumption, however, cannot always be made, since sometimes the implementation of some of the components is not available at verification time. This is the case, for example, when certain components join the system after the latter has been put in operation, as for example when applications are loaded on a smart card after the card has been issued (see e.g. [SGH04]). Such systems are usually referred to as *open*. Still, even if the implementation of certain components is not yet available, an open system can be verified on the basis of the components' specifications, i.e. the properties which they are assumed to satisfy. Furthermore, proving system properties relative to given assumptions on some of its components is a standard element of compositional reasoning. Such reasoning can for example be employed for verifying systems with dynamically changing configuration due to dynamic process spawning [DFG98]. Finally, replacing a component implementation with a specification can be used as a means of abstraction in order to control state space explosion.

State space exploration based verification of open systems requires the state space of the open system to be represented in a suitable fashion. A component specification denotes a set of possible behaviours of the component. Hence, the state space of an open system corresponds to a (potentially infinite) set of possible behaviours, as induced by the component specifications. A behavioural property of an open system is then a property which is satisfied by all such behaviours. These behaviours have to be conveniently represented in a structure, which is at the same time suitable for verification. We propose *extended modal transition systems* (EMTS) for these purposes, when component specifications are written in the modal μ-calculus. These are based on the notion of modal transition systems, introduced by Larsen [Lar89,BL90] as a graphical specification language. A similar structure, called a Kripke modal transition system, was first introduced by Huth [HJS01], and later refined by Shoham and Grumberg [SG04] as a means for representing state space abstractions in an abstraction refinement framework for CTL. As in the latter approach, EMTSs have sets of states (instead of single states) as targets to transitions – an extension which is needed for dealing with disjunctive assumptions. In addition, we add well-foundedness constraints to the structure to handle least fixed point assumptions. These serve a purpose dual to the one of fairness constraints (see e.g. [CGL94]).

The notion of EMTS is planned to be used in a larger context to be developed for the verification of open systems using state space exploration. The process begins with specifying the system as a process algebra term with assumptions on variables representing the underspecified components. This model is to be converted to an EMTS in such a way that, ideally, all possible behaviours of the model are captured by the EMTS. Finally, to check whether the system has a certain property, the property is checked for this EMTS.

Related Work. Besides the beforementioned work on modal transition systems and its variants, our work is closely related to the approach of using maximal model constructions for modular verification, as pioneered by Grumberg and Long [GL94] in the context of ACTL, extended to ACTL* by Kupferman and Vardi [KV00], and applied by Sprenger et al to the fragment of the modal μ-calculus without least fixed points and diamond modalities [SGH04]. These approaches provide an algorithmic solution to the problem of verifying open systems for the particular logics, all of which are proper fragments of the modal μ-calculus, when the same logic is used for specifying the properties of both the yet unavailable components and the open system. For the modal μ-calculus, however, no such algorithmic solution is possible. In [DG00], a proof system is presented to verify properties of open CCS terms parameterized on variables with modal mu-calculus assumptions. The proof is guided by the formula to be verified and the state space is implicit in the proof tree. In our approach, we separate the tasks of constructing a finite representation of the state space of an open system (to be addressed elsewhere) from the task of verifying properties of the representation (addressed here) allowing different fragments of the modal μ-calculus to be explored for these two tasks. Having this seperation has additional benefits like state space visualisation and possibility of proof reuse.

Organization. In this paper, we first give a motivating example for EMTSs, followed by the definition of this notion. We also define a simulation relation in Section 2 which specifies the set of states of a given labelled transition system (LTS) denoted by an EMTS state. Section 3 gives background information about modal μ-calculus along with a proof system due to Bradfield and Stirling [BS92,Sti01] for checking properties of labelled transition systems in this logic. In Section 4 we introduce our proof system for verifying modal μ-calculus properties of EMTS states, which is an adaptation of Bradfield and Stirling's. Soundness and relative completeness of our proof system is shown by providing a translation of proofs in our proof system to proofs in the other system. Finally, Section 6 concludes the paper with an outline of future work.

2 Extended Modal Transition Systems

Motivation. Consider a concurrent server that spawns off *Handle* processes for handling requests to produce a result, modelled by the action $\overline{out}$. Although *Handle* processes can receive further input, modelled by the action in, they do not process these. We would like to prove that such a system eventually stabilizes – i.e., always comes to a state where only input actions are enabled – provided that the server does not receive any new requests. Eventual stabilization can be formalized in the modal μ-calculus as $\nu X.\mu Y.\,[in]\,X \wedge [\overline{out}]\,Y$.

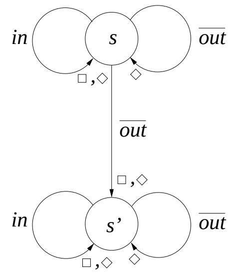

Fig. 1. EMTS representation of an open system.

This system can be specified as the open system, $x \mid \mathit{Handle}$, which consists of component *Handle* and any eventually stabilizing component x. In order to verify the above open system, we propose an abstract representation of its behaviour as the EMTS shown in Figure 1. The EMTS consists of two "abstract" states, each state denoting the set of "concrete" states (processes) which it *simulates*, and two transition relations called *may* and *must*, depicted by $\longrightarrow^{\Diamond}$ and $\longrightarrow^{\Box}$ arrows, respectively. The must transitions are a subset of the may transitions, defining an interval of possible behaviours. Thus, state s' in the example denotes all processes which can engage in arbitrary interleavings of in and $\overline{\mathit{out}}$ actions, but so that in has to be enabled throughout while $\overline{\mathit{out}}$ has not. In addition, the EMTS specifies explicitly a set W of infinite runs which are deemed impossible. In this example, W consists of the infinite runs stabilizing on $\overline{\mathit{out}}$ actions. A proof of eventual stabilization of the system using this representation can be found in Section 4.

EMTS. We introduce the central notion of the present paper, aimed at representing the state space of an open system for the purposes of verifying temporal properties. The notion of EMTS is based on Modal Transition Systems (MTS) of Larsen [Lar89], and borrows ideas from some generalizations [GL94,SG04]. The formal definition of EMTS has been chosen to support in a natural way the construction of an EMTS from an open system specification: it includes may and must transitions to deal with modal assumptions, sets of states (instead of single states) as targets to transitions to deal with disjunctive assumptions, and well-foundedness constraints to deal with least fixed point assumptions.

Definition 2.1 (EMTS) *An* extended modal transition system *is a structure*

$$\mathcal{E} = (S_{\mathcal{E}}, A, \longrightarrow^{\Diamond}_{\mathcal{E}}, \longrightarrow^{\Box}_{\mathcal{E}}, W)$$

where (i) $S_{\mathcal{E}}$ is a set of abstract states, (ii) A is a set of actions, (iii) $\longrightarrow^{\Diamond}_{\mathcal{E}}$, $\longrightarrow^{\Box}_{\mathcal{E}}$ $\subseteq$ $S_{\mathcal{E}} \times A \times 2^{S_{\mathcal{E}}}$ are may *and* must *transition relations, and (iv) W is a set of infinite $\mathcal{E}$-runs.*

A *run*, or may–run, of $\mathcal{E}$ is a possibly infinite sequence of transitions $\rho_{\mathcal{E}} = s_0 \xrightarrow{a_0}_{\mathcal{E}} s_1 \xrightarrow{a_1}_{\mathcal{E}} s_2 \xrightarrow{a_2}_{\mathcal{E}} \ldots$ where for every $i \geq 0$, $s_i \xrightarrow{a_i}{}^{\Diamond}_{\mathcal{E}} S_e$ for some S_e such that $s_{i+1} \in S_e$. Must–runs are defined similarly. We distinguish between two kinds of *a-derivatives*: $\partial^{\Diamond}_a(s) \triangleq \{S \mid s \xrightarrow{a}{}^{\Diamond}_{\mathcal{E}} S\}$, and $\partial^{\Box}_a(s) \triangleq \{S \mid s \xrightarrow{a}{}^{\Box}_{\mathcal{E}} S\}$.

For purposes of generality, the definition of W is extentional; there are various possibilities for giving W a finite characterization.

We define a simulation relation between the states of an EMTS, a form of fair simulation (cf. e.g. [GL94,BG02]).

Definition 2.2 (Simulation) *$R \subseteq S_{\mathcal{E}} \times S_{\mathcal{E}}$ is a* simulation relation *between the states of an $\mathcal{E}$ if:*

(i) *whenever sRt and $a \in A$:*
  (a) *if $s_1 \xrightarrow{a}{}^{\Diamond}_{\mathcal{E}} S_1$, then there is a S_2 such that $s_2 \xrightarrow{a}{}^{\Diamond}_{\mathcal{E}} S_2$ and for each $s'_1 \in S_1$, there exists a $s'_2 \in S_2$ such that $s'_1 R s'_2$;*
  (b) *if $s_2 \xrightarrow{a}{}^{\Box}_{\mathcal{E}} S_2$, then there is a S_1 such that $s_1 \xrightarrow{a}{}^{\Box}_{\mathcal{E}} S_1$ and for each $s'_1 \in S_1$, there exists a $s'_2 \in S_2$ such that $s'_1 R s'_2$.*

(ii) *If $\rho_t = t \xrightarrow{a_1}_{\mathcal{E}} t_1 \xrightarrow{a_2}_{\mathcal{E}} t_2 \xrightarrow{a_3}_{\mathcal{E}} \ldots$ is in W, then every infinite run $\rho_s = s \xrightarrow{a_1}_{\mathcal{E}} s_1 \xrightarrow{a_2}_{\mathcal{E}} s_2 \xrightarrow{a_3}_{\mathcal{E}} \ldots$ such that $s_i R t_i$ for all $i \geq 1$ is also in W.*

The choice for (ii) is made to guarantee soundness of the proof system presented in Section 4. We say that abstract state t *simulates* abstract state s, denoted $s \preceq t$, if there is a simulation relation R such that sRt. Simulation can be generalized to two different EMTSs $\mathcal{E}_1$ and $\mathcal{E}_2$ in the natural way.

The standard notion of *labelled transition system* (LTS) can be viewed as a special kind of EMTS, where: $\longrightarrow^{\Box}_{\mathcal{E}} = \longrightarrow^{\Diamond}_{\mathcal{E}}$, the target sets of the transition relation are singleton sets of states, and set W is empty. We give the meaning of an abstract state relative to a given LTS, as the set of concrete LTS states simulated by the abstract state.

Definition 2.3 (Denotation) *The* denotation *of an abstract state $s_{\mathcal{E}}$ w.r.t. a given LTS $\mathcal{T}$ is the set $[\![s_{\mathcal{E}}]\!]_{\mathcal{T}} \triangleq \{t \in S_{\mathcal{T}} \mid t \preceq s_{\mathcal{E}}\}$.*

The notion is lifted to sets of abstract states $S' \subseteq S_{\mathcal{E}}$ in the natural way: $[\![S']\!]_{\mathcal{T}} \triangleq \bigcup\{[\![s]\!]_{\mathcal{T}} \mid s \in S'\}$. In the rest of the paper, we shall assume that EMTSs obey the following *consistency* restrictions: $\longrightarrow^{\Box}_{\mathcal{E}} \subseteq \longrightarrow^{\Diamond}_{\mathcal{E}}$, $s \xrightarrow{a}{}^{\Box}_{\mathcal{E}} S$ implies S is non-empty, and W does not contain runs corresponding to infinite must–runs of the EMTS.

3 Modal μ-Calculus

In this section we give a summary of the modal μ-calculus [Koz83], a process logic which we consider for specifying assumptions about the behavior of components, as well as for specifying the system properties to be verified. We present a known proof system, which we refer to for showing soundness and completeness of the proof system which we propose in the next section.

Definition 3.1 (μ-calculus syntax) *A formula Φ of the modal μ-calculus is generated by the following grammar, where a ranges over a set of actions,* *and Z over a set of propositional variables.*

$$\Phi ::= \mathrm{tt} \mid \mathrm{ff} \mid Z \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid [a]\,\Phi \mid \langle a \rangle\,\Phi \mid \nu Z.\Phi \mid \mu Z.\Phi$$

A formula Φ is a *normal* formula if $\sigma_1 Z_1$ and $\sigma_2 Z_2$ are two different occurrences of binders in Φ then $Z_1 \neq Z_2$ and no occurrence of a free variable Z is also used in a binder σZ in Φ. Let Φ be a normal formula and $\sigma_1 X.\Psi_1$ and $\sigma_2 Z.\Psi_2$ be subformulas of Φ, then X *subsumes* Z if $\sigma_2 Z.\Psi_2$ is a subformula of $\sigma_1 X.\Psi_1$.

The semantics of the μ-calculus is standard (see e.g. [Sti01]), defining a satisfaction relation $\models^{\mathcal{T}}_{\mathrm{V}}$ relative to a LTS $\mathcal{T} = (S_{\mathcal{T}}, A, \longrightarrow_{\mathcal{T}})$ and a valuation V mapping propositional variables Z to a sets of states $\mathrm{V}(Z) \subseteq S_{\mathcal{T}}$. Satisfaction is then lifted to sets of states in the natural way.

The proof system $\Sigma_{\mathcal{T}}$ given below has been proposed by Bradfield and Stirling [BS92,Sti01] for showing that a set of states in a given LTS $\mathcal{T}$ (or, as in the original text, a set of closed CCS process terms) satisfies a property given as a μ-calculus formula. A proof tree is constructed starting from the goal using the rules in a goal-directed fashion, checking at each step if a terminal node was reached. A proof tree is a proof if all the terminal nodes are successful. In the rules below, σ ranges over μ and ν. We give a single-action version of Bradfield and Stirling's K notation. The original set of rules has been augmented with an admissable Cut rule for later purposes. The rules are presented in a goal-directed fashion.

The proof system is sound and complete (cf. [BS92,Sti01]). In the next section, we present a similar proof system for states of extended modal transition systems. We rely on the properties of the above proof system to show that our system enjoys these properties as well.

$$\wedge\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \wedge \Psi}{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \quad \mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi} \qquad \vee\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \vee \Psi}{\mathrm{S}_1 \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \quad \mathrm{S}_2 \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi}\ \mathrm{S} = \mathrm{S}_1 \cup \mathrm{S}_2$$

$$\sigma Z\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \sigma Z.\Phi}{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} Z} \qquad \Box_a\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} [a]\,\Phi}{\partial_a(\mathrm{S}) \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}$$

$$Z\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} Z}{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ Z \text{ identifies } \sigma Z.\Phi \qquad \Diamond_a\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \langle a \rangle\,\Phi}{f_a(\mathrm{S}) \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ f_a : \mathrm{s} \mapsto \mathrm{s}' \in \partial_a(\mathrm{s})$$

$$\mathrm{Thin}\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}{\mathrm{R} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \mathrm{S} \subset \mathrm{R} \qquad \mathrm{Cut}\ \frac{\mathrm{S} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}{\mathrm{S}_1 \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \quad \mathrm{S}_2 \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \mathrm{S} = \mathrm{S}_1 \cup \mathrm{S}_2$$

A successful tableau (or proof) is a finite proof tree having successful terminals as leaves. A node n in a proof tree labelled by a sequent $S \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi$ is denoted $n : S \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi$. If $n : S \vdash^{\mathcal{T}}_{\mathbf{V}} Z$ is a node where Z identifies a fixed point formula $\sigma Z.\Phi$, and there is a ancestor node $n' : S' \vdash^{\mathcal{T}}_{\mathbf{V}} Z$ above n with at least one application of a rule other than Thin and Cut in between, $S' \supseteq S$ and for any other fixed point variable Y on this path, Z subsumes Y, then node n is called a σ-terminal. So no further rules are applied to it[2]. The most recent node making n a σ-terminal is called n's companion. The conditions for a leaf sequent $R \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi$ to be a successful (resp. unsuccessful) terminal are as follows

Successful Terminals

(i) $\Psi = \mathrm{tt}$, or else $\Psi = Z$, Z is free in the initial formula, and $R \subseteq \mathbf{V}(Z)$

(ii) $R = \emptyset$

(iii) $\Psi = Z$ where Z identifies a fixed point formula $\sigma Z.\Phi$, and the leaf sequent is a σ-terminal with companion node $n : \ S \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi$, then
  (a) If $\sigma = \nu$, then the terminal is successful.
  (b) If $\sigma = \mu$, then the terminal is successful if there is no infinite chain of composable trails $T_0 \circ T_1 \circ T_2 \ldots$ of n (see Definition 3.3 below).

Unsuccessful Terminals

(i) $\Psi = \mathrm{ff}$, or $\Psi = Z$ and Z is free in the initial formula with R $\not\subseteq \mathbf{V}(Z)$

(ii) $\Psi = \langle a \rangle\, \Phi$ and for some $r \in R$, $\partial_a(\{r\}) = \emptyset$

(iii) $\Psi = Z$ where Z identifies a minimal fixed point formula $\mu Z.\Phi$, there is an ancestor node $S \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi$ with at least one application of a rule other than Thin and Cut in between, $S \subset R$ and for any other fixed point variable Y on this path, Z subsumes Y.

We now explain the notion of trail used in the above definition.

Definition 3.2 (Dependent) *If node $n' : \ S' \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi'$ is an immediate successor of node $n : \ S \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$, then state $s' \in S'$ at n' is a* dependant *of state $s \in S$ at n if:*

- *$s = s'$ and the rule applied to n is $\wedge$, $\vee$, σZ, Z, or Thin, or*
- *$s \xrightarrow{a}_{\mathcal{T}} s'$ and the rule is $\Box_a$, or*
- *$s' = f_a(s)$ and the rule is $\Diamond_a$ applied with choice function f_a.*

Definition 3.3 ($\mathcal{T}$-Trail) *Assume that node n_k:$S_k \vdash^{\mathcal{T}}_{\mathbf{V}} Z$ is a μ-terminal and node n_0:$S_0 \vdash^{\mathcal{T}}_{\mathbf{V}} Z$ is its companion. A trail T of the companion node n_0 is a sequence of state–node pairs $(s_0, n_0), \ldots, (s_k, n_k)$ from state $s_0 \in S_0$ at n_0 to $s_k \in S_k$ at n_k, such that for all $0 \leq i < k$, one of the following holds:*

(i) *$s_{i+1} \in S_{i+1}$ at n_{i+1} is a dependent of $s_i \in S_i$ at n_i, or*

(ii) *n_i is the immediate predecessor of a σ-terminal node $n' \neq n_k$ whose companion is n_j for some $j : 0 \leq j \leq i$, and $n_{i+1} = n_j$ and $s_{i+1} \in S_{i+1}$ at n' is a dependant of $s_i \in S_i$ at n_i.*

[2] In order to show completeness for our proof system, we relax this condition without harming the soundness and completeness results.

Two trails T_1 and T_2 of the same companion node are *composable*, if the last pair of T_1 and the first pair of T_2 mention the same state; in this case their composition is denoted by $T_1 \circ T_2$.

Trails mention state–node pairs. Later in this paper, we will work with actual sequences of transitions (runs). We therefore define a mapping α to extract the corresponding run from a given trail.

Definition 3.4 (α: Trail to Run Conversion) *Let $T = (s_0, n_0), \ldots, (s_k, n_k)$ be a $\mathcal{T}$-trail of proof tree Σ. The corresponding run $\alpha(T)$ is inductively defined as follows:*

$$\alpha(s, n) \triangleq \varepsilon$$

$$\alpha((s_1, n_1) \cdot (s_2, n_2) \cdot T) \triangleq \begin{cases} (s_1 \xrightarrow{a}_{\mathcal{T}} s_2) \cdot \alpha((s_2, n_2) \cdot T) & \Box_a \textit{ or } \Diamond_a\textit{-rule is applied to } n_1 \\ \alpha((s_2, n_2) \cdot T) & \textit{otherwise.} \end{cases}$$

4 A Proof System for EMTSs

Satisfaction of a temporal property by an EMTS state cannot be defined directly. The reason for this is the presence of the set W in EMTSs, which makes it impossible to define the denotation of a fixed point formula compositionally (that is, inductively on the structure of the formula). We therefore give an indirect definition of satisfaction by means of the denotation $[\![s]\!]_{\mathcal{T}}$ of a state s.

Definition 4.1 *Let $\mathcal{E}$ be an EMTS, and let $s \in S_{\mathcal{E}}$. We define satisfaction $s \models^{\mathcal{E}}_{\mathcal{V}} \Phi$ to hold if and only if $[\![s]\!]_{\mathcal{T}} \models^{\mathcal{T}}_{\mathrm{V}} \Phi$ holds for any LTS $\mathcal{T}$ and valuation $\mathcal{V}$ mapping propositional variables Z to sets of states $\mathcal{V}(Z) \subseteq S_{\mathcal{E}}$, where valuation V is defined via $\mathcal{V}$ by $\mathrm{V}(Z) \triangleq \bigcup\{[\![s]\!]_{\mathcal{T}} \mid s \in \mathcal{V}(Z)\}$.*

We present a proof system $\Sigma_{\mathcal{E}}$ for verifying whether a state s of an EMTS $\mathcal{E}$ satisfies a modal μ-calculus formula Φ. Since the transitions of the EMTS result in sets of states, the latter are immediately split into separate goals.

$$\frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \wedge \Psi}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \quad s \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi} \qquad \frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \vee \Psi}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi} \qquad \frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \vee \Psi}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi}$$

$$\frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \sigma Z.\Phi}{s \vdash^{\mathcal{E}}_{\mathcal{V}} Z} \qquad \frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} [a]\,\Phi}{s_1 \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \ \ldots \ s_n \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi} \{s_1, \ldots, s_n\} = \cup\, \partial^{\Diamond}_a(s)$$

$$\frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} Z}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi} Z \text{ identifies } \sigma Z.\Phi \qquad \frac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \langle a \rangle\,\Phi}{s_1 \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \ \ldots \ s_n \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi} \{s_1, \ldots, s_n\} \in \partial^{\Box}_a(s)$$

The conditions for a leaf node of a proof tree in $\Sigma_{\mathcal{E}}$ to be a successful terminal are similar to the ones specified for $\Sigma_{\mathcal{T}}$ (see Section 3), and are even somewhat simpler since single states are considered instead of sets of states.

Considering node $r \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi$:

Successful Terminals

(i) $\Psi = \mathrm{tt}$, or else $\Psi = Z$, Z is free in the initial formula, and $r \in \mathcal{V}(Z)$

(ii) $\Psi = [a]\,\Phi$ and $\cup\partial^{\Diamond}_a(r) = \emptyset$

(iii) $\Psi = Z$ where Z identifies a fixed point formula $\sigma Z.\Phi$, and the sequent is a σ-terminal with companion node $n :\ r \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi$, then

(a) If $\sigma = \nu$, then the terminal is successful.

(b) If $\sigma = \mu$, then the terminal is successful if every infinite run w_{n_0} of the EMTS that corresponds to an infinite sequence of $\mathcal{E}$-trails of the companion node n_0 is in W. That is, if $w_{n_0} = \alpha(E_1)\circ\alpha(E_2)\circ\alpha(E_3)\ldots$ is such that the first state of trail E_i is r for all $i \geq 1$, then $w_{n_0} \in W$ (the notion of $\mathcal{E}$-trail is explained below).

Unsuccessful Terminals

(i) $\Psi = \mathrm{ff}$, or else $\Psi = Z$, Z is free in the initial formula, and $r \notin \mathcal{V}(Z)$

(ii) $\Psi = \langle a\rangle\,\Phi$ and $\cup\partial^{\Box}_a(r) = \emptyset$

(iii) $\Psi = Z$ where Z identifies the minimal fixed point formula $\mu Z.\Phi$, and the sequent is a σ-terminal with companion node $n :\ r \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi$, then the terminal is unsuccessful if some infinite run w_{n_0} of the EMTS that corresponds to an infinite sequence of $\mathcal{E}$-trails of the companion node n_0 is not in W.

The notion of $\mathcal{E}$-trails of an EMTS is defined analogously to $\mathcal{T}$-trails of an LTS. Since the nodes of the proof trees mention single states, the notion of dependant becomes superfluous. What is more, $\mathcal{E}$-trails of a node always begin and end with the same state, and are thus always composable. The mapping α from Definition 3.4 which converts an $\mathcal{T}$-trail to an $\mathcal{T}$-run can be applied exactly the same way to $\mathcal{E}$-trails and proofs trees in $\Sigma_{\mathcal{E}}$.

Consider the example from the Introduction, where the set W consists of all infinite runs which have the suffix $s \xrightarrow{\overline{out}} s \xrightarrow{\overline{out}} \ldots$ or $s' \xrightarrow{\overline{out}} s' \xrightarrow{\overline{out}} \ldots$. Using the above proof system, eventual stabilization of all processes denoted by the abstract state s is established with the successful tableau below.

Node n_6 is discharged with companion node n_1, and similarly n_{17} is discharged with n_{12} without appealing to W since X identifies a greatest fixed point formula. To discharge node n_8 with n_3, however, we need to make sure that all infinite runs of $\mathcal{E}$ corresponding to infinite sequences of $\mathcal{E}$-trails of n_3 are in W. Node n_3 has only one $\mathcal{E}$-trail, giving rise to a single infinite run $s \xrightarrow{\overline{out}} s \xrightarrow{\overline{out}} \ldots$. Since this run is in W, the terminal is successful. Node n_{19} with companion n_{14}, and node n_{21} with companion n_9 are discharged similarly.

$$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \nu X.\mu Y.\,[in]\,X \wedge [\overline{out}]Y}{\dfrac{n_1 : s \vdash^{\mathcal{E}}_{\mathcal{V}} X}{\dfrac{n_2 : s \vdash^{\mathcal{E}}_{\mathcal{V}} \mu Y.\,[in]\,X \wedge [\overline{out}]Y}{\dfrac{n_3 : s \vdash^{\mathcal{E}}_{\mathcal{V}} Y}{\dfrac{n_4 : s \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X \wedge [\overline{out}]Y}{\Box\,\dfrac{n_5 : s \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X}{n_6 : s \vdash^{\mathcal{E}}_{\mathcal{V}} X} \quad \dfrac{n_7 : s \vdash^{\mathcal{E}}_{\mathcal{V}} [\overline{out}]Y}{\Box\; n_8 : s \vdash^{\mathcal{E}}_{\mathcal{V}} Y \quad \dfrac{n_9 : s' \vdash^{\mathcal{E}}_{\mathcal{V}} Y}{\dfrac{n_{10} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X \wedge [\overline{out}]Y}{\Box\,\dfrac{n_{11} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X}{\dfrac{n_{12} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} X}{\dfrac{n_{13} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} \mu Y.\,[in]\,X \wedge [\overline{out}]Y}{\dfrac{n_{14} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} Y}{\dfrac{n_{15} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X \wedge [\overline{out}]Y}{\Box\,\dfrac{n_{16} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [in]\,X}{n_{17} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} X} \quad \Box\,\dfrac{n_{18} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [\overline{out}]Y}{n_{19} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} Y}}}}}} \quad \Box\,\dfrac{n_{20} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} [\overline{out}]Y}{n_{21} : s' \vdash^{\mathcal{E}}_{\mathcal{V}} Y}}}}}}}}}$$

5 Soundness and Completeness

To prove soundness of the proof system $\Sigma_{\mathcal{E}}$ presented above, we show that every successful tableau for sequent $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ in $\Sigma_{\mathcal{E}}$ can be transformed into a successful tableau for sequent $[\![s]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$ in $\Sigma_{\mathcal{T}}$ for any given $\mathcal{T}$. Soundness of $\Sigma_{\mathcal{E}}$ follows then from the soundness of $\Sigma_{\mathcal{T}}$ by Definition 4.1. Full proofs of results in this section has been left out due to lack of space and can be found in [AG05].

We achieve this transformation by defining a translation from rule instances in $\Sigma_{\mathcal{E}}$ to proof trees in $\Sigma_{\mathcal{T}}$ and we show that its extension to proof trees in $\Sigma_{\mathcal{E}}$ translates successful tableaux in $\Sigma_{\mathcal{E}}$ to successful tableaux in $\Sigma_{\mathcal{T}}$.

Definition 5.1 (Translation) *Translation $\pi_{\mathcal{T}}$, mapping rule instances in $\Sigma_{\mathcal{E}}$ to proof trees in $\Sigma_{\mathcal{T}}$ for a given $\mathcal{T}$, is defined through Table 1. In the definition, f_a is a choice function.*

5.1 Soundness

Lemma 5.2 (Correctness) *For each rule instance in $\Sigma_{\mathcal{E}}$, translation $\pi_{\mathcal{T}}$ assigns a correct proof tree in $\Sigma_{\mathcal{T}}$, so that each premise (resp. conclusion), $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$, of the rule is matched by a leaf (resp. root), $[\![s]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$, and all unmatched leaves of the constructed proof tree are successful terminals.*

Translation $\pi_{\mathcal{T}}$ extends to proof trees in $\Sigma_{\mathcal{E}}$ in the obvious way as defined by the matching sequents.

Corollary 5.3 *For proof tree $A_{\mathcal{E}}$ with root sequent $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ in $\Sigma_{\mathcal{E}}$, $A_{\mathcal{T}} =$*

$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_1 \wedge \Phi_2}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_1 \quad s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_2}$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_1 \wedge \Phi_2}{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_1 \quad \llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_2} \wedge$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_1 \vee \Phi_2}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_1}$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_1 \vee \Phi_2}{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_1} \vee$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_1 \vee \Phi_2}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_2}$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_1 \vee \Phi_2}{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_2} \vee$
$s \vdash^{\mathcal{E}}_{\mathcal{V}} [a]\,\Phi$ and $\cup\, \partial^{\Diamond}_a(s) = \emptyset$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} [a]\,\Phi}{\emptyset \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \Box_a$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} [a]\,\Phi}{s_1 \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \quad \ldots \quad s_n \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi}\ \{s_1, \ldots, s_n\} = \cup\, \partial^{\Diamond}_a(s)$	$\dfrac{\dfrac{\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} [a]\,\Phi}{\partial_a(\llbracket s \rrbracket_{\mathcal{T}}) \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \Box_a}{\llbracket \cup\, \partial^{\Diamond}_a(s) \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \text{Thin*}}{\dfrac{\llbracket s_1 \rrbracket_{\mathcal{T}} \cup \ldots \cup \llbracket s_{n-1} \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}{\dfrac{\vdots}{\llbracket s_1 \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \quad \llbracket s_2 \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\,\text{Cut}}\ \text{Cut} \qquad \llbracket s_n \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \text{Cut}$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \langle a \rangle\,\Phi}{s_1 \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi \quad \ldots \quad s_n \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi}\ \{s_1, \ldots, s_n\} \in \partial^{\Box}_a(s)$	$\dfrac{\dfrac{\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \langle a \rangle\,\Phi}{f_a(\llbracket s \rrbracket_{\mathcal{T}}) \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \Diamond_a}{\llbracket \{s_1, \ldots, s_n\} \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \text{Thin*}}{\dfrac{\llbracket s_1 \rrbracket_{\mathcal{T}} \cup \ldots \cup \llbracket s_{n-1} \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}{\dfrac{\vdots}{\llbracket s_1 \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi \quad \llbracket s_2 \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\,\text{Cut}}\ \text{Cut} \qquad \llbracket s_n \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi}\ \text{Cut}$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} Z}{s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi} Z$ identifies $\sigma Z.\Phi$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} Z}{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi} \sigma Z$
$\dfrac{s \vdash^{\mathcal{E}}_{\mathcal{V}} \sigma Z.\Phi}{s \vdash^{\mathcal{E}}_{\mathcal{V}} Z}$	$\dfrac{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \sigma Z.\Phi}{\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} Z} Z$

Table 1
Translation $\pi_{\mathcal{T}}$

$\pi_{\mathcal{T}}(A_{\mathcal{E}})$ is a proof tree with root sequent $\llbracket s \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$ in $\Sigma_{\mathcal{T}}$, such that each leaf $s_i \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi_i$ of $A_{\mathcal{E}}$ is matched by a leaf $\llbracket s_i \rrbracket_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi_i$ in $A_{\mathcal{T}}$, and all unmatched leaves in $A_{\mathcal{T}}$ are successful terminals.

Definition 5.4 (β:Trail Translation) *Given a proof tree $A_{\mathcal{E}}$ with nodes labeled $m_0 \ldots m_l$, the function β converts a trail, $T = (t_0, n_0), \ldots, (t_k, n_k)$, of the corresponding LTS tree $\pi_{\mathcal{T}}(A_{\mathcal{E}})$ to the $\mathcal{E}$-trail, $E = (s_0, n_0), \ldots, (s_{k'}, n_{k'})$ by replacing each node with the matching one and deleting the remaining ones:*

$$\beta(\varepsilon) \triangleq \varepsilon$$

$$\beta((t_1, n_1) \cdot T) \triangleq \begin{cases} ((s_{1'}, n_{1'}) \cdot \beta(T)) & \text{if } n_{1'} \text{ matches } n_1 \text{ and } s_{1'} \text{ is in } n_{1'} \\ \beta(T) & \text{if no node matches } n_1 \text{ in } A_{\mathcal{E}} \end{cases}$$

Notice that whenever a pair (t_i, n_i) is replaced by $(s_{i'}, n_{i'})$, $t_i \in [\![s_{i'}]\!]_{\mathcal{T}}$, since node n_i contains the sequent $[\![s_{i'}]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi'$.

Lemma 5.5 *$s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ implies $[\![s]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$ for any LTS $\mathcal{T}$. That is, if there is a successful tableaux for $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ in $\Sigma_{\mathcal{E}}$, then for any $\mathcal{T}$ there is a successful tableaux for $[\![s]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$ in $\Sigma_{\mathcal{T}}$.*

Proof. Assume $A_{\mathcal{E}}$ is a successful tableau for sequent $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$, and assume $\mathcal{T}$ is an LTS. To establish the result it suffices, due to Corollary 5.3, to show that each leaf of tableaux $A_{\mathcal{T}} = \pi_{\mathcal{T}}(A_{\mathcal{E}})$ matching a (successful) terminal of $A_{\mathcal{E}}$ is a successful terminal. Consider the successful terminal $m : r \vdash^{\mathcal{E}}_{\mathcal{V}} \Psi$ in $A_{\mathcal{E}}$ and the matching node of $A_{\mathcal{T}}$, $n : [\![r]\!]_{\mathcal{T}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Psi$. The proof that the latter is also a successful terminal is trivial when $\Psi = \mathrm{tt}$, when $\Psi = Z$ and Z is free in the initial formula, when Z identifies the formula $\nu Z.\Psi'$, or when $\Psi = [a]\,\Psi'$ and $\cup\partial^{\Diamond}_{a}(r) = \emptyset$.

The only interesting case occurs when $\Psi = Z$ and Z identifies $\mu Z.\Psi'$. In such a case, the condition of subset inclusion is trivially satisfied, so it remains to show that no infinite sequences of composable trails of the companion node n' exist in $A_{\mathcal{T}}$, i.e. there is no $\kappa_{n'} = T_1 \circ T_2 \circ \ldots$, such that for all $i \geq 1$, T_i begins with a pair (t_i, n'), where $t_i \in [\![r]\!]_{\mathcal{T}}$.

Assume such an infinite sequence, $\kappa_{n'} = T_1 \circ T_2 \ldots$, exists. Since the trails are composable, we also know that for all $i \geq 1$, T_i ends with the pair $(t_{i'}, n)$ for some $t_{i'} \in [\![r]\!]_{\mathcal{T}}$. Then the corresponding sequence of the EMTS for terminal node m and companion node m', $w_{m'} = \beta(T_1) \circ \beta(T_2) \ldots$ is an infinite sequence of $\mathcal{E}$-trails, and for all $i \geq 1$, $\beta(T_i)$ starts and ends with pairs (r,m') and (r, m) respectively. Because $m : r \vdash^{\mathcal{E}}_{\mathcal{V}} Z$ is a successful terminal, the corresponding infinite run $\alpha(w_{m'})$ is guaranteed to be in W. Then, by the definitions of simulation and denotation, the run $\alpha(\kappa_{n'})$ of $\mathcal{T}$ cannot be infinite, and therefore neither can $\kappa_{n'}$ be infinite. We thus reached a contradiction. □

Theorem 5.6 (Soundness) *$s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ implies $s \models^{\mathcal{E}}_{\mathcal{V}} \Phi$.*

Proof. Follows directly from Lemma 5.5, the soundness of Bradfield and Stirling's proof system (cf. [BS92,Sti01]), and Definition 4.1. □

5.2 Completeness

We base our completeness argument on the existence of a *universal* LTS $\mathcal{U}$, having the property that every LTS $\mathcal{T}$ is isomorphic to a transition–closed sub–structure of $\mathcal{U}$. Because of the shape of our disjunction proof rule, completeness can only be shown for formulas Φ, all subformulas of which are *prime* (cf. [BL90]). A formula Ψ is prime if whenever it logically implies a disjunction $\Psi_1 \vee \Psi_2$ then it also implies one of the disjuncts.

By the completeness proof of Bradfield and Stirling's proof system ([BS92], [Sti01]), whenever $[\![s]\!]_{\mathcal{U}} \models^{\mathcal{U}}_{\mathbf{V}} \Phi$ we know that there is a space of canonical proofs of $[\![s]\!]_{\mathcal{U}} \vdash^{\mathcal{U}}_{\mathbf{V}} \Phi$, which differ in the choice functions that are made use of in the $\Diamond$-rule applications. For the case when $\mathcal{E}$ is finite–state and all subformulas of Φ are prime, we show how to construct a proof $A_{\mathcal{U}}$ of $[\![s]\!]_{\mathcal{U}} \vdash^{\mathcal{U}}_{\mathbf{V}} \Phi$ that captures several of these canonical proofs using the branching provided by Cut-rule. This proof is constructed mutually with another proof $A^*_{\mathcal{U}}$ for the same goal that is built using "macro"s from $\pi_{\mathcal{U}}$ instead of single rules. Both proofs are built in the weaker version of Bradfield and Stirling's proof system, where repeat nodes are not necessarily terminals. In the final step of the Completeness proof, $A^*_{\mathcal{U}}$ is translated backward into a proof of $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$ using the reverse function $\pi^{-1}_{\mathcal{U}}$.

The constructions of $A_{\mathcal{U}}$ and $A^*_{\mathcal{U}}$ guide each other. At each step, the rule application in $A_{\mathcal{U}}$ determines the corresponding subtree (or "macro") application taken from the range of translation $\pi_{\mathcal{U}}$ in $A^*_{\mathcal{U}}$ except when the rule is Thin. In turn, $A^*_{\mathcal{U}}$ determines the number of Cut-rules that are to be applied after each $\Box$ and $\Diamond$ rule in $A_{\mathcal{U}}$. As a result, each rule application in $A_{\mathcal{U}}$ is matched by the same rule application in $A^*_{\mathcal{U}}$ except for Thin. In order to describe this, we define in the construction process a γ function which matches nodes of $A_{\mathcal{U}}$ and $A^*_{\mathcal{U}}$. If the rule/macro to be used is $\Diamond$, $A^*_{\mathcal{U}}$ additionally constraints the choice function used in $A_{\mathcal{U}}$ so that the extension of the former to the set mentioned in the matching node of $A_{\mathcal{U}}$ gives the latter. Finally, when a repeat node of $A_{\mathcal{U}}$ is to be taken as a terminal is also determined by whether the matching node is a terminal in $A^*_{\mathcal{U}}$.

$A_{\mathcal{U}}$ is similar to the canonical proofs from the completeness proof of $\Sigma_{\mathcal{T}}$ [BS92,Sti01] in the judicious application of the Thin rule, and in that the validity of the sequent is preserved at each application. Besides being applied to the goal as the first rule, Thin is applied in the rest of the tree only if the goal is of the form $S \vdash^{\mathcal{U}}_{\mathbf{V}} \sigma Z.\Psi$. The application of Thin should reduce the goal $S \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$ to the subgoal $||\Phi||^{\mathcal{T}}_{\mathbf{V}} \vdash^{\mathcal{T}}_{\mathbf{V}} \Phi$, where $||\Phi||^{\mathcal{T}}_{\mathbf{V}}$ is defined as the set of all states in $S_{\mathcal{T}}$ that satisfy Φ under valuation $\mathbf{V}$. $A_{\mathcal{U}}$ may include applications of a special version of the Cut-rule, while no Cut-rule applications occur in canonical proofs. Cut may be applied after $\Box$ and $\Diamond$ rule in $A_{\mathcal{U}}$, in such a way that the set mentioned in each of the subgoals produced will be identical to the set mentioned in the original node. (Note that this application merely duplicates the current subgoal hence giving the possibility to combine several proof trees for the same subgoal in a single proof tree.)

Theorem 5.7 (Completeness) *Let $\mathcal{E}$ be a finite–state EMTS, $s \in S_{\mathcal{E}}$, and let Φ have prime subformulas only. Then $s \vDash^{\mathcal{E}}_{\mathcal{V}} \Phi$ implies $s \vdash^{\mathcal{E}}_{\mathcal{V}} \Phi$.*

Proof. See [AG05]. □

6 Conclusion

We propose extended modal transition systems as a means for representing the set of possible behaviours of an open system in the presence of not yet available components for which we are given behavioural specifications (assumptions) written in the modal μ-calculus. The extentions to the standard notion of modal transition systems are needed to deal with disjunctive assumptions and with liveness assumptions. The latter are captured through a set of well-founded (that is, prohibited infinite) runs, a notion dual to the familiar notion of fairness constraints. We present a proof system for local model checking of EMTS properties written in the modal μ-calculus. We define a translation of proofs in our proof system to proofs in a known sound and complete proof system for infinite state systems due to Bradfield and Stirling, and use this translation to show these properties for our proof system.

This paper presents work in progress towards state space representation and verification of open systems. Current work focuses on developing a formalism to model open systems, and on the problem of constructing a finite EMTS from such a specification, exploring both interactive techniques when dealing with the full logic and algorithmic techniques for certain fragments. The notion of an EMTS will be modified to include a suitable finite representation for the set of well-founded runs W, which in turn will make it possible to give an algorithm for verifying that an EMTS state satisfies a temporal property. The theory will be extended to deal with assumptions on data variables, in addition to assumptions on process variables. Finally, tool development will be followed by practical applications and verification case studies.

Acknowledgement. The authors are indebted to Christoph Sprenger for his valuable comments on an earlier draft.

References

[AG05] I. Aktug and D. Gurov. Verification of open systems based on explicit state space representation. Technical report, ICT, KTH, `http://www.nada.kth.se/~irem/sefros/techrep05.ps`, August 2005.

[BG02] Doron Bustan and Orna Grumberg. Applicability of fair simulation. In *TACAS '02: Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 401–414. Springer-Verlag, 2002.

[BL90] G. Boudol and K. G. Larsen. Graphical versus logical specifications. In André Arnold, editor, *CAAP '90, 15th Colloquium on Trees in Algebra and Programming, Copenhagen, Denmark, May 15-18, 1990, Proceedings*, volume 431 of *Lecture Notes in Computer Science*, pages 57–71. Springer, 1990.

[BS92] J.C. Bradfield and C.P. Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96:157,174, 1992.

[CGL94] E.M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. *ACM-TOPLAS*, 16(5):1512–1542, 1994.

[DFG98] M. Dam, L.-Å. Fredlund, and D. Gurov. Toward parametric verification of open distributed systems. In H. Langmaack, A. Pnueli, and W.-P.

de Roever, editors, *Compositionality: the Significant Difference*, volume 1536 of *Lecture Notes in Computer Science*, pages 150–185. Springer, 1998.

[DG00] M. Dam and D. Gurov. Compositional verification of CCS processes. In *Proc. PSI'99*, volume 1755 of *Lecture Notes in Computer Science*, pages 247–256, 2000.

[GL94] O. Grumberg and D. Long. Model checking and modular verification. *ACM Transactions on Programming Languages and Systems*, 16(3):843–871, 1994.

[HJS01] M. Huth, R. Jagadeesan, and D. A. Schmidt. Modal transition systems: A foundation for three-valued program analysis. In *ESOP '01: Proceedings of the 10th European Symposium on Programming Languages and Systems*, volume 2028, pages 155–169, London, UK, 2001. Springer-Verlag.

[Koz83] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[KV00] Orna Kupferman and Moshe Vardi. An automata-theoretic approach to modular model checking. *ACM Transactions on Programming Languages and Systems*, 22(1):87–128, 2000.

[Lar89] K. G. Larsen. Modal specifications. *Automatic Verification Methods for Finite State Systems*, pages 232–246, 1989.

[SG04] S. Shoham and O. Grumberg. Monotonic abstraction-refinement for CTL. In *TACAS'04: Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2988 of *Lecture Notes in Computer Science*, pages 546–560. Springer-Verlag, April 2004.

[SGH04] C. Sprenger, D. Gurov, and M. Huisman. Compositional verification for secure loading of smart card applets. In C. Heitmeyer and J.-P. Talpin, editors, *Proc. MEMOCODE'04*, pages 211–222. IEEE, 2004.

[Sti01] C. Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer-Verlag, 2001.