Expressiveness and Completeness in Abstraction

Maciej Gazda and Tim A.C. Willemse
Department of Computer Science, Eindhoven University of Technology (TU/e),
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

We study two notions of expressiveness, which have appeared in abstraction theory for model checking, and find them incomparable in general. In particular, we show that according to the most widely used notion, the class of Kripke Modal Transition Systems is strictly less expressive than the class of Generalised Kripke Modal Transition Systems (a generalised variant of Kripke Modal Transition Systems equipped with hypertransitions). Furthermore, we investigate the ability of an abstraction framework to prove a formula with a finite abstract model, a property known as *completeness*. We address the issue of completeness from a general perspective: the way it depends on certain abstraction parameters, as well as its relationship with expressiveness.

1 Introduction

Model checking [3] is one of the key technologies for formal software verification. Given a model of a program or a process and a specification of the required behaviour in the form of a logical formula, a model checker can automatically verify whether the model satisfies the specification. A model checker typically explores the entire state space of a program. Such a state space is enormous in most practical applications.

Abstract interpretation [5, 2] is among the most important techniques designed to handle the state space explosion problem, making many instances of the model checking problem tractable. It works by approximating the artefacts of the original model, the so-called *concrete model*, by simpler *abstract objects*. A model transformed in this way has a smaller abstract state space. The loss of detail in this model may allow a model checker to successfully verify the property, but it can also give rise to an inconclusive answer. The cause of the inconclusive answer may be resolved by successive refinements of the abstraction [4], leading to finer-grained abstract models.

Assuming that, as most works on abstraction do, concrete models are modelled by *Kripke Structures*, we investigate two key properties of abstraction formalisms for Kripke Structures. First, we study the *expressiveness* of the formalism. This gives us the information about the classes of concrete structures that can be described by abstract models. Second, we study the *completeness* of the formalism. In abstraction, completeness is the degree to which properties of a concrete model can be proved using a *finite* abstraction.

A systematic survey of the literature reveals that there is an abundance of different abstraction formalisms for Kripke Structures. Kripke Structures equipped with the usual simulation relation themselves form one of the first studied abstraction formalisms, but their power is rather limited. Below, we list the most important families of abstraction formalisms:

1. *Modal Transition Systems* (MTSs) [15] with *may* and *must* transitions and a built-in consistency requirement, and related formalisms, see e.g. [11], such as *Kripke Modal Transition Systems* (KMTSs) [13].
2. *Mixed Transition Systems* (MixTSs) [6], a modelling formalism similar to KMTSs, but with the added capability of expressing inconsistent specifications.

B. Luttik and M.A. Reniers (Eds.): Combined Workshop on Expressiveness in
Concurrency and Structural Operational Semantics (EXPRESS/SOS 2012)
EPTCS 89, 2012, pp. 49–64, doi:10.4204/EPTCS.89.5

3. *Generalised Kripke Modal Transition Systems* (GTSs) [17] with *must hypertransitions*; similar structures were already introduced by Larsen and Xinxin in [16] under the name of Disjunctive Transition Systems [16], although there, these structures served a different purpose.

4. Tree Automata (TA) [8], the most expressive and complete of all of the listed formalisms.

In this paper, we mostly restrict ourselves to GTSs and KMTSs. This is because in practice, GTSs and KMTSs are the key abstraction formalisms used. Tree Automata, while being the "most complete" among the abstraction formalisms, are mainly of theoretical importance due to the complexity of computing an abstraction using this formalism.

Expressiveness of abstraction formalisms has been studied before. In fact, Wei *et al.* [19] proved that the formalisms from the KMTS family with may and must transitions have the same expressiveness as GTSs with must hypertransitions; they claim to:

> "...complete the picture by showing the expressive equivalence *between* these families."

At first glance, this seems rather odd: the GTS abstraction formalism is more liberal than any member of the KMTS family of abstraction formalisms. While the arguments in [19] are sound, a closer inspection of their results reveals that their notion of expressiveness can be regarded as non-standard. It is therefore not immediately clear whether their results are comparable to the expressiveness results reported by, e.g. Godefroid and Jagadeesan in [11].

We show that the notion of expressiveness *does* make a difference: using the notion used by e.g. Godefroid and Jagadeesan we conclude that the GTS abstraction formalism is *strictly more expressive* than members of the KMTS family of abstraction formalisms. The expressiveness results claimed by Wei *et al* are therefore likely to become a source of confusion. We henceforth refer to the notion of expressiveness used by Wei *et al* as *contextual expressiveness*.

The aforementioned paper [19] suggests the expressiveness and completeness of an abstraction formalism are closely related, given the following quote:

> "The work of Godefroid and Jagadeesan, and Gurfinkel and Chechik showed that the models in the KMTS family have the same expressive power and are equally precise for SIS. Dams and Namjoshi showed that the three families considered in this paper are subsumed by tree automata. We completed the picture by proving that the three families are equivalent as well. Specifically, we showed that KMTSs, MixTSs and GKMTSs are relatively complete (in the sense of [Dams and Namjoshi]) with one another."

Since Dams and Namjoshi's paper [7] to which they refer only studies completeness, we can only conclude that Wei *et al* consider completeness and expressiveness as equivalent notions. Since they are defined in different ways, it is therefore interesting to know the exact relationship between the two, if one exists.

We have been able to establish only a weak link between expressiveness and completeness, namely, only when assuming a *thorough semantics* for logical formulae, more expressive abstraction formalisms are more complete. Since in [19], thorough semantics is not used, our findings are not in support of the claims in the above quote.

Finally, we investigate the notion of completeness itself in more detail. For instance, it is known that GTSs are complete for the fragment of least-fixpoint free μ-calculus formulae [1]. However, it is not known whether those are the only formulae for which GTSs are complete. Our investigations reveal that the answer to this open question is ambiguous and depends on the setting that is used.

Related work. Expressiveness of modelling formalisms for abstraction has been first studied by Godefroid and Jagadeesan in [11]. There, it has been proved that Partial KSs (Kripke structures with possibly unknown state labels), MTSs (without state labels) and KMTSs are equally expressive. The proof consists of defining 3 translations that preserve a variant of mixed simulation called a 'completeness preorder'. In [12], Gurfinkel and Chechik have shown that Partial classical Kripke Structures (with each atomic proposition either always "true" or "false", or always "maybe") are expressively equivalent to the above formalisms. Subsequently, Wei, Gurfinkel and Chechik [19] have studied expressiveness in the context of a fixed abstraction; we come back to this notion in more detail in Section 3.

Expressiveness of various modelling formalisms in the context of refinement has been studied in [9]. In contrast to our work, the authors consider only deterministic structures as proper concretisations of a model.

Dams and Namjoshi were the first to explicitly address the question whether there are abstraction frameworks that are complete for the entire μ-calculus. They answer this question in [7] by introducing abstraction based on *focused transition systems*. In their follow-up work [8], they show that these focused transition systems are in fact variants of μ-automata, enabling a very brief and elegant argument for completeness of their framework.

The GTS/DMTS framework has received a considerable interest from the abstraction community. Shoham and Grumberg studied the precision of the framework in [18]; Fecher and Shoham, in [10] used the framework for a more algorithmic approach to abstraction, by performing abstraction in a lazy fashion using a variation on parity games.

Outline. In Section 2, we introduce the abstraction formalisms and the basic mathematical machinery needed to understand the remainder of the paper. We investigate the expressiveness of the GTS and KMTS abstraction formalisms in Section 3. We then proceed to study completeness in Section 4; there, we provide a formal framework that allows to compare completeness of different formalisms, we study the relation between completeness and expressiveness and we more accurately characterise the set of formulae for which GTSs are complete. We wrap up with concluding remarks and issues for future work in Section 5.

2 Preliminaries

The first basic ingredient of an abstraction theory is the class of concrete structures $\mathscr{C}$, representing the objects (programs, program models) that we wish to analyse. Throughout this paper, we restrict ourselves to the setting where $\mathscr{C}$ consists of all Kripke Structures (KSs), or possibly some subclasses of KSs (e.g. the set of finitely branching Kripke Structures).

Let AP denote an arbitrary, fixed set of *atomic propositions*, used to specify properties of states; propositions and their negations constitute the set of *literals* $\mathsf{Lit} = \mathsf{AP} \cup \{\neg p \mid p \in \mathsf{AP}\}$. Below, we recall the definition of Kripke Structures.

Definition 1 A Kripke Structure is a tuple $\langle S, S^0, R, L\rangle$ where:

- S is a set of states,
- $S^0 \subseteq S$ is a set of initial states
- $R \subseteq S \times S$ is the transition relation;
- $L{:}S \to 2^{\mathsf{Lit}}$ is a labelling function such that $L(s)$ contains exactly one of p and $\neg p$ for all $p \in \mathsf{AP}$.

The class of all Kripke Structures is denoted **KS**.

The concrete structures are described using abstract models; as a convention, we use symbols $\mathscr{M}_{(i)}$ to denote classes of abstract models and $M_{(i)}$ for instances of models. We assume that models consist of states with a distinguished set of initial states, and additional structural components such as transitions and labels. Formally, models are tuples of the form $M = (S, S^0, \Sigma)$, where Σ represents the aforementioned structural artefacts. For a given model M, $\mathbf{states}(M)$ is the set of states underlying M. The notation $\langle M, s\rangle$ will be used to represent the state s of a model M. We also make the general assumption that abstract models considered in this paper are finite.

Properties of concrete and abstract models are expressed using a certain logic L. In case of concrete structures, we will use μ-calculus ($\mathscr{L}_\mu$) with its standard semantics. The same logic is used for our abstract models; however, for such models, semantics of $\mathscr{L}_\mu$, given by the definition of the satisfaction relation by $\models^\alpha$, may vary (e.g. inductive or thorough semantics). Below, we first present the syntax of $\mathscr{L}_\mu$ and its semantics for Kripke Structures.

Definition 2 A μ-calculus formula (in positive form) is a formula generated by the following grammar:

$$\varphi, \psi ::= \top \mid \bot \mid l \mid X \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Box\varphi \mid \Diamond\varphi \mid \nu X.\ \varphi \mid \mu X.\ \varphi$$

where $l \in \mathsf{Lit}$ and $X \in \mathscr{V}$ for a set of propositional variables $\mathscr{V}$. The μ and ν symbols denote the least and greatest fixpoint respectively. The semantics of a formula φ is an inductively defined function $[\![\_]\!]$, in the context of a Kripke Structure $M = \langle S, S_0, R, L\rangle$ and an environment $\eta{:}\mathscr{V} \to 2^S$ assigning sets of states to the propositional variables:

$$\begin{array}{lcllcl}
[\![\top]\!]^\eta & = & S & [\![\bot]\!]^\eta & = & \emptyset \\
[\![l]\!]^\eta & = & \{s \in S \mid l \in L(s)\} & [\![X]\!]^\eta & = & \eta(X) \\
[\![\varphi \wedge \psi]\!]^\eta & = & [\![\varphi]\!]^\eta \cap [\![\psi]\!]^\eta & [\![\varphi \vee \psi]\!]^\eta & = & [\![\varphi]\!]^\eta \cup [\![\psi]\!]^\eta \\
[\![\Diamond\varphi]\!]^\eta & = & \tilde{\Diamond}[\![\varphi]\!]^\eta & [\![\Box\varphi]\!]^\eta & = & \tilde{\Box}[\![\varphi]\!]^\eta \\
[\![\mu X.\varphi]\!]^\eta & = & \mu U.[\![\varphi]\!]^{\eta[X:=U]} & [\![\nu X.\varphi]\!]^\eta & = & \nu U.[\![\varphi]\!]^{\eta[X:=U]}
\end{array}$$

Here, we used the following two abbreviations:

$$\tilde{\Diamond}U = \{s \in S \mid \exists q \in S.\ s\,R\,q \wedge q \in U\} \qquad \text{and} \qquad \tilde{\Box}U = \{s \in S \mid \forall q \in S.\ s\,R\,q \Rightarrow q \in U\}$$

In case the formula φ is *closed* (i.e., does not contain free propositional variables: variables that are not bound by a surrounding fixpoint binding the variable), its semantics is independent of the environment η, and we drop η from the semantic brackets. We say that a closed formula φ is *true* in a state $s \in S$, denoted $\langle M, s\rangle \models \varphi$ if $s \in [\![\varphi]\!]$; φ is *false* in s, denoted $\langle M, s\rangle \not\models \varphi$ if $s \notin [\![\varphi]\!]$.

The μ-calculus is a highly expressive logic, subsuming temporal logics such as LTL, CTL and CTL*. It is capable of expressing safety properties, liveness properties and (complex) fairness properties. Typically, least fixpoint subformulae express *eventualities*, whereas greatest fixpoint subformulae express *invariance*. For instance, the formula $\nu X.(\Box X \wedge l)$ expresses that l holds invariantly on all computation paths, whereas $\mu X.((\Box X \wedge \Diamond\top) \vee l)$ expresses that property l holds eventually on all computation paths. By mixing least and greatest fixpoints one can construct (computationally and intuitively) more complex formulae such as $\nu X.\mu Y.((\Diamond X \wedge l) \vee \Diamond Y)$, which expresses that there is a computation path on which l holds infinitely often.

An essential component of an abstraction formalism is the notion of *description*, *approximation* or *refinement*, which provides a link between concrete and abstract *structures*. Typically, there is a "meta-relation" between classes of structures (here denoted with $\preceq$). Such a meta-relation is typically a relation on states or languages; for instance, *mixed simulation* or *language containment*.

As already mentioned, the final ingredient of an abstraction formalism is the semantics of the logic on abstract models. The abstract $\mathscr{L}_\mu$ semantics should include the definition of both *satisfaction* (denoted with $\models^\alpha$) and *refutation* $\not\models^\alpha$ predicates. Unlike in Kripke Structures, because of the loss of precision it might be the case that neither $M \models^\alpha \varphi$ nor $M \not\models^\alpha \varphi$ is true for an abstract model M. However, we assume that the models are *consistent*, i.e. for no model M both $M \models^\alpha \varphi$ and $M \not\models^\alpha \varphi$ are true. This can typically be ensured by imposing a certain syntactic restriction; concretely, in our definitions of GTSs and KMTSs, we require that all must transitions are matched by the corresponding may transitions.

Given an $\mathscr{L}_\mu$ semantics for abstract models, the refinement relation has to meet a soundness property (also known as a weak preservation property). That is, we require that whenever $\langle K, s_K\rangle \rho \langle M, s_M\rangle$ holds for some specific states and relation ρ of type $\preceq$, then $\langle M, s_M\rangle \models^\alpha \varphi$ implies $\langle K, s_K\rangle \models \varphi$ for every $\varphi \in \mathscr{L}_\mu$.

Before we give the formal definitions of the GTS and KMTS abstraction frameworks, we formally define the notion of an *abstraction formalism*.

Definition 3 We define an *abstraction formalism* $\mathscr{F}$ as a quadruple $\langle \mathscr{C}, \mathscr{M}, \preceq, \models^\alpha\rangle$, where $\mathscr{C}$ is a class of concrete structures (i.e. a subclass of **KS**); $\mathscr{M}$ is a class of models, $\preceq$ is a class of (structural) refinement relations between $\mathscr{C}$ and $\mathscr{M}$, and $\models^\alpha$ specifies the semantics of the logic (μ-calculus) on abstract models.

Below, we define the class of Generalised Kripke Modal Transition Systems (GTSs) [17]. The class of Kripke Modal Transition Systems (KMTSs) can be viewed as a specialisation of GTSs; in turn, Kripke Structures can be considered as a specialisation of KMTSs.

Definition 4 A *Generalised Kripke Modal Transition System* (GTS) is a tuple $M = \langle S, S^0, R^+, R^-, L\rangle$ where:

- S is a set of states,
- $S^0 \subseteq S$ is a set of initial states,
- $R^- \subseteq S \times S$ is the *may* transition relation,
- $R^+ \subseteq S \times 2^S$ is the *must* transition relation; we require that $s\, R^+\, A$ implies $s\, R^-\, t$ for all $t \in A$,
- $L{:}S \to 2^{\mathsf{Lit}}$ is a labelling function; we require that $L(s)$ contains at most one of p and $\neg p$ for all $s \in S, p \in \mathsf{AP}$.

The system M is called a *Kripke Modal Transition System* (KMTS) if for all $s \in S, A \subseteq S$ for which $s\, R^+\, A$ we have $|A| = 1$. The class of all GTSs will be denoted with **GTS** and the class of all KMTSs is denoted by **KMTS**.

Note than an KMTS M can be identified with a Kripke Structure, if for all $s, s' \in S$ we have $s\, R^+\, \{s'\}$ iff $s\, R^-\, s'$ and $L(s)$ contains precisely one of p and $\neg p$ for all $p \in \mathsf{AP}$.

The intuition behind a must hypertransition sR^+A is that it is guaranteed that there is a transition from s to some state in A, but the exact state in A to which this transition leads is not determined upfront, offering some extra flexibility. In contrast, the extra condition that is imposed on KMTSs ensures that the destination of a must hypertransition *is* determined.

Next, we formalise the notion of approximation, or refinement, between concrete and abstract structures. The *de-facto* approximation relation for GTSs is *mixed simulation*; this approximation relation

also appears under the names "completeness preorder" [11, 8] and "refinement preorder" [19]. Given that GTSs generalise Kripke Structures, we define the notion of mixed simulation between abstract structures only.

Definition 5 Let $M_1 = \langle S_1, S_1^0, R_1^+, R_1^-, L_1 \rangle$ and $M_2 = \langle S_2, S_2^0, R_2^+, R_2^-, L_2 \rangle$ be two GTSs. A relation $H \subseteq S_1 \times S_2$ is a mixed simulation if $s_1\ H\ s_2$ implies

- $L_2(s_2) \subseteq L_1(s_1)$,
- if $s_1\ R_1^-\ s_1'$, then there exists $s_2' \in S_2$ such that $s_2\ R_2^-\ s_2'$ and $s_1'\ H\ s_2'$,
- if $s_2\ R_2^+\ A_2$, then there exists $A_1 \subseteq S_1$ such that $s_1\ R_1^+\ A_1$, and for every $s_1' \in A_1$ there is some $s_2' \in A_2$ such that $s_1'\ H\ s_2'$.

We write $\langle M_1, s_1 \rangle \leq_{mix} \langle M_2, s_2 \rangle$ if $s_1\ H\ s_2$ for some mixed simulation H. Mixed simulation between models M_1 and M_2, denoted $M_1 \leq_{mix} M_2$, holds iff for all initial states $s_1^0 \in S_1^0$ there is a corresponding initial state $s_2^0 \in S_2^0$ such that $\langle M_1, s_1^0 \rangle \leq_{mix} \langle M_2, s_2^0 \rangle$.

We proceed to define the *standard inductive semantics* (SIS) of a μ-calculus formula in the setting of GTSs. This can be done in more than one way; our definition is taken from Shoham and Grumberg [18].

Definition 6 The standard inductive semantics of a formula φ is inductively defined by two functions $[\![\_]\!]_{\mathtt{tt}}$ and $[\![\_]\!]_{\mathtt{ff}}$, in the context of a GTS $M = \langle S, R^+, R^-, L \rangle$ and an environment $\eta : \mathcal{V} \to 2^S$ assigning sets of states to propositional variables:

$$
\begin{array}{lcl}
[\![\top]\!]^\eta_{\mathtt{tt}} & = & S \\
[\![\bot]\!]^\eta_{\mathtt{tt}} & = & \emptyset \\
[\![l]\!]^\eta_{\mathtt{tt}} & = & \{s \in S \mid l \in L(s)\} \\
[\![X]\!]^\eta_{\mathtt{tt}} & = & \eta(X) \\
[\![\varphi \wedge \psi]\!]^\eta_{\mathtt{tt}} & = & [\![\varphi]\!]^\eta_{\mathtt{tt}} \cap [\![\psi]\!]^\eta_{\mathtt{tt}} \\
[\![\varphi \vee \psi]\!]^\eta_{\mathtt{tt}} & = & [\![\varphi]\!]^\eta_{\mathtt{tt}} \cup [\![\psi]\!]^\eta_{\mathtt{tt}} \\
[\![\Diamond\varphi]\!]^\eta_{\mathtt{tt}} & = & \tilde{\Diamond}[\![\varphi]\!]^\eta_{\mathtt{tt}} \\
[\![\Box\varphi]\!]^\eta_{\mathtt{tt}} & = & \tilde{\Box}[\![\varphi]\!]^\eta_{\mathtt{tt}} \\
[\![\mu X.\varphi]\!]^\eta_{\mathtt{tt}} & = & \mu U.[\![\varphi]\!]^{\eta[X:=U]}_{\mathtt{tt}} \\
[\![\nu X.\varphi]\!]^\eta_{\mathtt{tt}} & = & \nu U.[\![\varphi]\!]^{\eta[X:=U]}_{\mathtt{tt}}
\end{array}
\qquad
\begin{array}{lcl}
[\![\top]\!]^\eta_{\mathtt{ff}} & = & \emptyset \\
[\![\bot]\!]^\eta_{\mathtt{ff}} & = & S \\
[\![l]\!]^\eta_{\mathtt{ff}} & = & \{s \in S \mid \neg l \in L(s)\} \\
[\![X]\!]^\eta_{\mathtt{ff}} & = & \eta(X) \\
[\![\varphi \wedge \psi]\!]^\eta_{\mathtt{ff}} & = & [\![\varphi]\!]^\eta_{\mathtt{ff}} \cup [\![\psi]\!]^\eta_{\mathtt{ff}} \\
[\![\varphi \vee \psi]\!]^\eta_{\mathtt{ff}} & = & [\![\varphi]\!]^\eta_{\mathtt{ff}} \cap [\![\psi]\!]^\eta_{\mathtt{ff}} \\
[\![\Diamond\varphi]\!]^\eta_{\mathtt{ff}} & = & \tilde{\Box}[\![\varphi]\!]^\eta_{\mathtt{ff}} \\
[\![\Box\varphi]\!]^\eta_{\mathtt{ff}} & = & \tilde{\Diamond}[\![\varphi]\!]^\eta_{\mathtt{ff}} \\
[\![\mu X.\varphi]\!]^\eta_{\mathtt{ff}} & = & \nu U.[\![\varphi]\!]^{\eta[X:=U]}_{\mathtt{ff}} \\
[\![\nu X.\varphi]\!]^\eta_{\mathtt{ff}} & = & \mu U.[\![\varphi]\!]^{\eta[X:=U]}_{\mathtt{ff}}
\end{array}
$$

Here, we used the following two abbreviations:

$$\tilde{\Diamond}U = \{s \in S \mid \exists A \subseteq S.\ s\ R^+\ A \wedge A \subseteq U\} \qquad \text{and} \qquad \tilde{\Box}U = \{s \in S \mid \forall t \in S.\ s\ R^-\ t \Rightarrow t \in U\}$$

In case the formula φ is *closed*, its semantics is independent of the environment η, and we drop η from the semantic brackets. A closed formula φ is *true* in a state $s \in S$, denoted $\langle M, s \rangle \models^{SIS} \varphi$ (or simply $\langle M, s \rangle \models \varphi$ if the setting is clear from the context) if $s \in [\![\varphi]\!]_{\mathtt{tt}}$; φ is *false* in s, denoted $\langle M, s \rangle \not\models^{SIS} \varphi$ ($\langle M, s \rangle \not\models \varphi$) if $s \in [\![\varphi]\!]_{\mathtt{ff}}$ and it is *unknown* in s otherwise.

Note that if M is a Kripke structure, it always holds that either $\langle M, s \rangle \models \varphi$ or $\langle M, s \rangle \not\models \varphi$, and the satisfaction relation coincides with the usual semantics for the μ-calculus.

By abuse of notation, we make **GTS** also stand for the abstraction formalism $\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle$ Likewise, we make **KMTS** stand for the abstraction formalism $\langle \mathbf{KS}, \mathbf{KMTS}, \leq_{mix}, \models^{SIS} \rangle$.

In several works on abstraction [19, 18], the authors restrict themselves to a particular abstraction setting of the form $\langle C, \rho, S \rangle$, in which we are given two sets of concrete and abstract objects and a

fixed abstraction relation; properties of interest (precision, see [18], consistency but also expressiveness, see [19]) are then analysed relatively to such a fixed abstraction relation. The abstraction setting we define below formalises the approach of Wei *et al.* [19]; it differs from [18] in that C is assumed to be a set of concrete states without a structure, whereas in [18] it is assumed to be a Kripke Structure.

Definition 7 An *abstraction setting* is a pair $\langle \mathscr{F}, \langle C, \rho, S \rangle \rangle$, such that $\mathscr{F}$ is an abstraction formalism, C and S are sets, and $\rho \subseteq C \times S$ is an approximation relation.

3 On the Expressiveness of GTSs and KMTSs

In Section 3.1 we first define the notion of expressiveness, based on the definition of semantics of an abstract model that is used by e.g. Godefroid and Jagadeesan in [11]. Then, in Section 3.2, we formalise the notion of expressiveness used by Wei *et al* in [19].

3.1 Expressiveness

The expressiveness of an abstraction framework characterises classes of concretisations (refinements, implementations), that one is able to capture using abstract models from the framework. The definition is clear and intuitive; however, it still depends on the choice of what constitutes the "proper" concrete semantics of an abstract model. In most cases [11, 12, 14], the latter is defined simply as the class of all those Kripke structures that refine a given model. Formally, the definition is as follows:

Definition 8 Given an abstraction formalism $\mathscr{F} = \langle \mathscr{C}, \mathscr{M}, \preceq, \models^{\alpha} \rangle$ the class of *concretisations* of a state $\langle M, s \rangle$, where $M \in \mathscr{M}$ is defined as:

$$\mathbb{C}_{\mathscr{F}}[\langle M, s \rangle] \triangleq \{ \langle K, s_K \rangle \in \mathscr{C} \mid \langle K, s_K \rangle \preceq \langle M, s \rangle \}$$

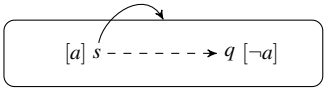
Having fixed the semantics of an abstract model, we may now define the corresponding notion of expressiveness.

Definition 9 Let $\mathscr{F}_1 = \langle \mathscr{C}, \mathscr{M}_1, \preceq_1, \models^{\alpha}_1 \rangle$ and $\mathscr{F}_2 = \langle \mathscr{C}, \mathscr{M}_2, \preceq_2, \models^{\alpha}_2 \rangle$ be two abstraction formalisms. We say that $\mathscr{F}_2$ is *more expressive than* $\mathscr{F}_1$, denoted $\mathscr{F}_1 \sqsubseteq_{ex} \mathscr{F}_2$, if for all $\langle M_1, s_1 \rangle \in \mathscr{M}_1$ there exists $\langle M_2, s_2 \rangle \in \mathscr{M}_2$ such that $\mathbb{C}_{\mathscr{F}_1}[\langle M_1, s_1 \rangle] = \mathbb{C}_{\mathscr{F}_2}[\langle M_2, s_2 \rangle]$. We say that $\mathscr{F}_2$ is *strictly more expressive than* $\mathscr{F}_1$, denoted $\mathscr{F}_1 \sqsubset_{ex} \mathscr{F}_2$, if $\mathscr{F}_1 \sqsubseteq_{ex} \mathscr{F}_2$ and there is some $\langle M_2, s_2 \rangle \in \mathscr{M}_2$ such that $\mathbb{C}_{\mathscr{F}_2}[\langle M_2, s_2 \rangle] \neq \mathbb{C}_{\mathscr{F}_1}[\langle M_1, s_1 \rangle]$ for every $\langle M_1, s_1 \rangle \in \mathscr{M}_1$.

Theorem 1 GTSs are strictly more expressive than KMTSs, i.e. **KMTS** $\sqsubset_{ex}$ **GTS**.

Proof: GTSs subsume KMTSs syntactically, and both abstraction formalisms use the same approximation relation. Therefore they are at least as expressive as KMTSs, i.e. **KMTS** $\sqsubseteq_{ex}$ **GTS**. We will prove that they are strictly more expressive than KMTSs by showing that a certain GTS cannot be matched by a semantically equivalent KMTS.

Consider the GTS G depicted below, where $L(s) = \{a\}$ and $L(q) = \{\neg a\}$; the labelling is denoted with square brackets, the must hypertransitions are depicted by solid transitions pointing to a rectangle containing one or more states, whereas dashed arrows are the may transitions.



Suppose, towards a contradiction, that for a certain $M \in \mathbf{KMTS}$, and a certain state s_M of M, we have $\mathbb{C}_{\mathbf{KMTS}}[\langle M, s_M\rangle] = \mathbb{C}_{\mathbf{GTS}}[\langle G, s\rangle]$. First, observe that for every state q_M, reachable from $\langle M, s_M\rangle$, one of the following holds:

1. $a \in L(q_M)$ and there is a must transition starting at q_M
2. $\neg a \in L(q_M)$ and there is no transition starting at q_M

To prove the above, one can show that these two properties, which hold on all (two) reachable states of G, are preserved "backwards" by mixed simulation, so they hold in all states of all refinements of $\langle G, s\rangle$. On the other hand, if one of them did not hold in some state of M, then M would have a refinement outside $\mathbb{C}_{\mathbf{GTS}}[\langle G, s\rangle]$.

The next observation is that both states p_1 and p_2 depicted below are valid concretisations of $\langle G, s\rangle$:

Let us now focus on M. Since the deadlocked process is not a concretisation of $\langle G, s\rangle$, there has to be a must transition starting in $\langle M, s_M\rangle$ (case 1 above), say, $\langle M, s_M\rangle R^+ \langle M, s'_M\rangle$. Consider two cases:

1. If there is a must transition starting at s'_M, then p_2 is not a concretisation of $\langle M, s_M\rangle$, a contradiction.
2. If $\neg a \in L(q_M)$, then p_1 is not a concretisation of $\langle M, s_M\rangle$, a contradiction.

Both cases lead to the desired contradiction. □

Note that in the above proof we did not make any assumption on the finiteness of M. That means that the GTS G cannot be semantically matched even by an infinite-state KMTS, i.e. our expressiveness result is rather robust.

3.2 Contextual Expressiveness

We already mentioned that the definition of expressiveness crucially depends on the choice of concretisations of an abstract model. Apart from taking the entire class, we may consider some restricted subclasses, for instance only deterministic implementations [9].

From the model checking perspective, the most important alternative definition was given in [19]. There, properties of an abstraction framework are always considered in the context of a specific abstraction, namely a relation $\rho \subseteq C \times S$ for some particular sets C and S of concrete and abstract objects, see the definition of the *abstraction setting*.

Given a model whose states consist of elements of S, we only consider concretisations from the given abstraction setting (based on ρ). The definition below is based on [19].

Definition 10 Given a fixed abstraction setting $\langle \mathscr{F}, \langle C, \rho, S\rangle\rangle$, we define the set of concretisations of a state, $\langle M, s\rangle$, where $M \in \mathscr{M}$, and $\mathbf{states}(M) \subseteq S$, *in the context of* ρ, as:

$$\mathbb{C}_{\mathscr{F}}[\langle M, s\rangle, \rho] \triangleq \{\langle K, s_K\rangle \in C \mid \mathbf{states}(K) \subseteq C,\ \langle K, s_K\rangle \rho \langle M, s\rangle \text{ and } \rho \text{ is a refinement relation in the sense of } \preceq \text{ between } K \text{ and } M\}$$

With the semantics of abstract models restricted to the specific instance of abstraction, we automatically obtain another notion of expressiveness. The one given in [19] is somewhat ambiguous:

> "Two partial modelling formalisms are *expressively equivalent* if and only if for every transition system M from one formalism, there exists a transition system M' from the other, such that M and M' are semantically equivalent."

The semantic equivalence to which the quote refers means having equal sets of concretisations in the context of a specific abstraction (in our terms, the equality of the $\mathbb{C}_{\mathscr{F}}[\langle M,s\rangle,\rho]$ sets).

For M, we can assume a specific setting $\langle\mathscr{F},\langle C,\rho,S\rangle\rangle$. However, it is not immediately obvious whether there are restrictions on the refinement relation ρ' with which M' is supposed to match M: can it be arbitrary, or should it be in some way related to ρ? We believe the best way is to allow for an arbitrary abstraction setting $\langle C,\rho',S'\rangle$ for M', with one practical assumption that whenever $S'\cap S\neq\emptyset$, then ρ' conservatively extends ρ on $S\cap S'$, i.e. $\{s_C \mid \exists s\in S'\cap S.\, s_C\ \rho'\ s\} = \{s_C \mid \exists s\in S'\cap S.\, s_C\ \rho\ s\}$.

The following definition of expressiveness, which we dub *contextual expressiveness*, so as to avoid confusion with the notion of expressiveness we considered in the previous section, is based on [19]; we clarify their notion by making the dependence on the abstraction context explicit.

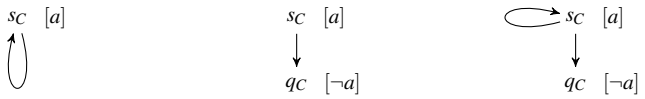
Definition 11 Let $\mathscr{F}_1=\langle\mathscr{C},\mathscr{M}_1,\preceq_1,\models_1^\alpha\rangle$ and $\mathscr{F}_2=\langle\mathscr{C},\mathscr{M}_2,\preceq_2,\models_2^\alpha\rangle$ be two abstraction formalisms. Then $\mathscr{F}_2$ is *contextually more expressive than* $\mathscr{F}_1$, denoted $\mathscr{F}_1\sqsubseteq_{cex}\mathscr{F}_2$ if for all abstraction settings $\langle\mathscr{F}_1,\langle C,\rho,S\rangle\rangle$ and all model-state pairs $\langle M_1,s_1\rangle\in\mathscr{M}_1$ with $\mathbf{states}(M_1)\subseteq S$, there is an abstraction setting $\langle\mathscr{F}_2,\langle C,\rho',S'\rangle\rangle$ $\langle M_2,s_2\rangle\in\mathscr{M}_2$, with $\mathbf{states}(M_2)\subseteq S'$ such that $\mathbb{C}_\rho[\langle M_1,s_1\rangle,\mathscr{F}_1]=\mathbb{C}_{\rho'}[\langle M_2,s_2\rangle,\mathscr{F}_2]$.

Theorem 2 **GTS** and **KMTS** are contextually equally expressive i.e. **GTS** $\sqsubseteq_{cex}$ **KMTS** and **KMTS** $\sqsubseteq_{cex}$ **GTS**, see [19].

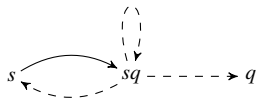
We believe it is at this point instructive to explain why the GTS we used to prove strictness of expressiveness in the previous section does not work in the setting of contextual expressiveness. For this, we consider the transformation GTOK defined in [19], which, given a GTS in a specific abstraction setting $\langle C,\rho,S\rangle$, produces a contextually equally expressive KMTS.

We first show that the application of GTOK does not produce an expressively equivalent (in our sense) KMTS. For this, we need to define an abstraction setting. Let us consider a very simple one, with concrete and abstract sets consisting of two elements, namely $C=\{s_C,q_C\}$ and $S=\{s,q\}$, and the description relation defined as $\rho=\{(s_C,s),(q_C,q)\}$.

Now consider the GTS G from the proof of Thm. 1. The set of concretisations of the state $\langle G,s\rangle$ in the context of ρ, $\mathbb{C}_\rho[\langle G,s\rangle,\mathbf{GTS}]$, consists of the following three models:



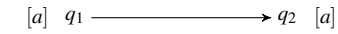
Next, we consider the transformation GTOK, and apply it to the GTS G from the proof of Thm.1. The starting point of the transformation is the KMTS consisting of the same states as the original GTS, with every original standard transition (with one state as a target). Then for every true hypertransition $p\,R^+\,A$, GTOK introduces a new state that mimics the target of the hypertransition p_A such that $\gamma(p_A)=\bigcup_{p'\in A}\gamma(p')$. Finally, the proper may transitions are added to the newly introduced states. This results in the following KMTS for our GTS G:



It is not too hard to check that $\mathbb{C}_\rho[\langle G,s\rangle,\mathbf{GTS}]=\mathbb{C}_{\rho'}[\langle\text{GTOK}(G),s\rangle,\mathbf{KMTS}]$, where ρ' is the abstraction relation resulting from the transformation, i.e. $\rho'=\{(s_C,s),(q_C,q),(s_C,sq),(q_C,sq)\}$. This means that G does not show that **GTS** is contextually strictly more expressive than **KMTS** (of course, this could

never be the case as the transformation GTOK is used to prove that an *arbitrary* GTS can be converted to a contextually equivalent KMTS).

On the other hand, the KMTS above is not expressively equivalent to G, because for instance the following Kripke structure is a concretisation of $\text{GTOK}(G)$:



From the fact that **GTS** and **KMTS** are contextually equally expressive but **GTS** are strictly more expressive than **KMTS**, it follows that contextual expressiveness does not imply expressiveness in general. In fact, if we consider arbitrary abstraction frameworks, these two notions are incomparable. The reason is that a certain framework may only allow maximal refinement relations (e.g. only maximal mixed simulations) between concrete and abstract models, which will make it less contextually expressive than if we allow all possible relations.

Theorem 3 Expressiveness and contextual expressiveness do not imply one another.

Proof: For an example showing that contextual expressiveness does not imply expressiveness we can use the example of GTSs and KMTSs.

To prove that the implication does not hold in the other direction as well, we can use frameworks based on the same class of models, but different refinement relations. Suppose that $\mathscr{F}_1$ is a standard KMTS framework with mixed simulation as a meta-refinement relation, and $\mathscr{F}_2$ differs from $\mathscr{F}_1$ in that it allows only those mixed simulations that are maximal, for a given concrete and abstract model. Clearly, $\mathscr{F}_1$ and $\mathscr{F}_2$ are expressively equivalent. On the other hand, they are not equivalent when it comes to contextual expressiveness.

In order to see this, consider a very simple KMTS M^{all} depicted below, and its concretisation K. Note that both states of M^{all} abstract the entire class of Kripke structures.

Let us now define an abstraction setting, $\langle \mathscr{F}_1, \langle \{q_1,q_2\}, \rho, \{s_1^{all}, s_2^{all}\}\rangle\rangle$, where $\rho(q_1) = s_1^{all}$ and $\rho(q_2) = s_2^{all}$ (on the left picture above). It is a valid abstraction setting; indeed, ρ is a mixed simulation, so it is a proper refinement in the framework $\mathscr{F}_1$. However, ρ is not a refinement in $\mathscr{F}_2$, because is is contained the relation $\{p_1,p_2\} \times \{s_1,s_2\}$ (depicted above on the right), which is the only maximal mixed simulation between K and M^{all}. Since this maximal refinement is the only one that we are allowed to consider as a candidate for ρ' to match ρ in concretisations of s_1^{all} and s_2^{all}, we observe that it is impossible to find a ρ' with the property that $\mathbb{C}_\rho[\langle M^{all}, s_1^{all}\rangle, \mathscr{F}_1] = \mathbb{C}_{\rho'}[\langle M^{all}, s_1^{all}\rangle, \mathscr{F}_2]$. Hence $\mathscr{F}_1$ and $\mathscr{F}_2$ are not contextually expressively equivalent. □

4 Completeness

Arguably the most important property of any abstraction formalism is its degree of completeness. We will first provide a formal measure of completeness; for a given formalism, it is defined as the set of formulae that can be proved with a finite abstract model.

Definition 12 Given an abstraction formalism $\mathscr{F} = \langle \mathscr{C}, \mathscr{M}, \preceq, \models^{\alpha} \rangle$. The *completeness set* of $\mathscr{F}$, denoted $\mathbf{compl}(\mathscr{F})$ is defined as follows:

$$\mathbf{compl}(\mathscr{F}) \triangleq \{\varphi \in \mathscr{L}_{\mu} \mid \varphi \text{ is satisfiable and } \forall \langle K, s_K \rangle \in \mathscr{C}. \langle K, s_K \rangle \models \varphi \Rightarrow \exists \langle M, s_M \rangle \in \mathscr{M}. \langle K, s_K \rangle \preceq \langle M, s_M \rangle \wedge \langle M, s_M \rangle \models^{\alpha} \varphi\}$$

Furthermore, we define the *relative completeness preorder* between abstraction formalisms $\mathscr{F}_1, \mathscr{F}_2$ as:

$$\mathscr{F}_1 \sqsubseteq_{cmp} \mathscr{F}_2 \overset{def}{\Leftrightarrow} \mathbf{compl}(\mathscr{F}_1) \subseteq \mathbf{compl}(\mathscr{F}_2)$$

Note that we have assumed that all abstract models in $\mathscr{M}$ are finite.

4.1 Expressiveness and completeness

We start by showing some simple facts concerning the relationship between expressiveness and completeness. One can observe an implication in one direction, if we assume the setting with thorough semantics, i.e. when the satisfaction of a formula by an abstract model depends on whether it is satisfied by all its concretisations. (note that this semantics differs from the standard inductive semantics we introduced in Section 2).

Definition 13 Given an abstraction formalism $\langle \mathscr{C}, \mathscr{M}, \preceq, \models^{t} \rangle$, we say that $\models^{t}$ is a *thorough semantics* if it is defined as (or satisfies):

$$\langle M, s \rangle \models^{t} \varphi \Leftrightarrow \forall \langle K, s' \rangle \in \mathbb{C}_{\mathscr{F}}[\langle M, s \rangle]. \langle K, s' \rangle \models \varphi$$

Proposition 1 Assuming thorough semantics, expressiveness containment implies relative completeness containment. More precisely, if $\mathscr{F}_1 = \langle \mathscr{C}, \mathscr{M}_1, \preceq_1, \models^{t}_1 \rangle$ and $\mathscr{F}_2 = \langle \mathscr{C}, \mathscr{M}_2, \preceq_2, \models^{t}_2 \rangle$, where $\models^{t}_1$ and $\models^{t}_2$ are thorough semantics, then:

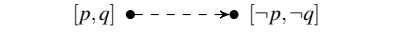
$$\mathscr{F}_1 \sqsubseteq_{ex} \mathscr{F}_2 \Rightarrow \mathscr{F}_1 \sqsubseteq_{cmp} \mathscr{F}_2$$

Proof: Take any $\varphi \in \mathbf{compl}(\mathscr{F}_1)$, and an arbitrary $\langle K, s_K \rangle \in \mathbf{KS}$ such that $\langle K, s_K \rangle \models \varphi$. From the fact that $\varphi \in \mathbf{compl}(\mathscr{F}_1)$ it follows that there exists some $\langle M_1, s_1 \rangle \in \mathscr{M}_1$ such that $\langle K, s_K \rangle \in \mathbb{C}_{\mathscr{F}_1}[\langle M_1, s_1 \rangle]$ and $\langle M_1, s_1 \rangle \models^{t}_1 \varphi$. Because of thorough semantics, we then have $\forall \langle K', s' \rangle \in \mathbb{C}_{\mathscr{F}_1}[\langle M_1, s_1 \rangle]. \langle K', s' \rangle \models \varphi$.

Since $\mathscr{F}_1 \sqsubseteq_{ex} \mathscr{F}_2$, there is a model $\langle M_2, s_2 \rangle \in \mathscr{M}_2$ such that $\mathbb{C}_{\mathscr{F}_1}[\langle M_1, s_1 \rangle] = \mathbb{C}_{\mathscr{F}_2}[\langle M_2, s_2 \rangle]$. Because of this equality $\langle M_1, s_1 \rangle$ and $\langle M_2, s_2 \rangle$ satisfy the same formulae under thorough semantics, and hence $\langle M_2, s_2 \rangle \models^{t}_2 \varphi$. Since K was chosen arbitrarily, we obtain $\varphi \in \mathbf{compl}(\mathscr{F}_2)$. □

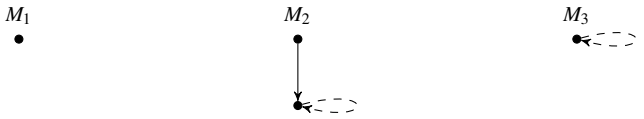
The necessity of a thorough semantics of both formalisms can be understood from the following counterexample that shows that, in general, equal expressiveness does not imply equal completeness. Take as $\mathscr{F}_1$ the standard class of **KMTS** with inductive semantics and as $\mathscr{F}_2$ also KMTSs, but this time with a thorough semantics. Since inductive semantics does not preserve thorough semantics, these two formalisms have different relative completeness.

Consider the following example, taken from [11]; the formula $\Box p \wedge \neg \Box q$ is *false* on all refinements of the KMTS depicted below, but it is unknown on the KMTS itself.



Theorem 4 In general, more completeness does not imply more expressiveness, even in formalisms with thorough semantics.

Proof: Consider two formalisms, consisting of simple KMTSs depicted below (we assume a setting with mixed simulation, and inductive or thorough semantics).



Let $\mathscr{M}_1 = \{M_1, M_2\}$ and $\mathscr{M}_2 = \{M_1, M_2, M_3\}$. We have: $\mathbf{compl}(\mathscr{F}_1) = \mathbf{compl}(\mathscr{F}_2) = \{\top, \Diamond\top, \Box\bot\}$ (up to semantic equivalence). However, $\mathscr{F}_1$ is strictly less expressive, because it cannot express the class of all Kripke Structures with a single model (as is the case with $\mathscr{F}_2$ and M_3). □

4.2 Characterisation of completeness sets for GTSs

The GTS framework is known to be complete for the set of least fixpoint free formulae, see [1]. However, it is not known whether there are other subsets of $\mathscr{L}_\mu$ for which GTSs are complete, too. In this section, we attempt to provide a more accurate characterisation of the completeness set of the GTS framework.

From hereon, we only consider GTSs or their subclasses with standard inductive semantics of $\mathscr{L}_\mu$. By $\mathscr{L}_\mu^\nu$ we will denote the fragment of μ-calculus in which only the greatest fixpoint is allowed.

Let $\equiv$ denote the semantic equivalence of formulae, i.e. $\varphi \equiv \psi$ if for every GTS M, $M \models^{SIS} \varphi \Leftrightarrow M \models^{SIS} \psi$. For any sublanguage $\mathscr{L}' \subseteq \mathscr{L}_\mu$, we define the completion of $\mathscr{L}'$ with respect to semantic equivalence as $[\mathscr{L}']_\equiv \triangleq \{\varphi \in \mathscr{L}_\mu \mid \exists \varphi' \in \mathscr{L}'. \varphi \equiv \varphi'\}$.

Before we prove our main result, we need to discuss certain technical issues. Inspired by the standard "naive" decision procedure for μ-calculus, that allows one to compute the semantics of a formula using approximants, we observe that, whenever a finite GTS satisfies a formula φ, it is witnessed by a least-fixpoint free *syntactic* approximant. These syntactic approximants can be obtained by successive unfoldings of μ-variables of φ.

By $\mathscr{V}(\varphi)$ we will denote the set of all variables occurring in φ, and $\mathbf{Sub}(\varphi)$ will denote all subformulae of φ. We will call a formula φ *well-formed*, if for every $X \in \mathscr{V}(\varphi)$, there is at most one subformula $\sigma X.\psi_X \in \mathbf{Sub}(\varphi)$. For a well-formed formula φ, the unfolding of a bound variable $X \in \mathscr{V}(\varphi)$, denoted by $\mathbf{unfold}_\varphi(X)$, is the formula ψ_X such that $\sigma X.\psi_X \in \mathbf{Sub}(\varphi)$. The set of bound μ-variables (resp. ν-variables) of a well-formed formula φ is denoted with $\mathscr{V}_\mu(\varphi)$ (resp. $\mathscr{V}_\nu(\varphi)$).

We first provide an auxiliary notion, namely the approximant of a formula with respect to *one* fixed μ-variable.

Definition 14 Fix a closed formula $\psi \in \mathscr{L}_\mu$. Let $\varphi \in \mathbf{Sub}(\psi)$ be a subformula of ψ and let $k \in \mathbb{N}$ be a positive natural number. Let γ be a word over $\mathbb{N}$; we will use it as a subscript in recursion variables to ensure well-formedness by introducing unique names. We will use the convention that the variables X, Y range over the variables of the original formula ψ; in the definition below, they are formally identified with $X_\varepsilon, Y_\varepsilon$, where ε denotes the empty word. A *simple k-th approximant* of φ with respect to $X \in \mathscr{V}_\mu$,

denoted $\mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi)$, is defined as follows:

$$
\begin{array}{lcll}
\mathbf{approx\text{-}var}^{X,0}_{\psi}(X_\gamma) & \triangleq & \bot & \\
\mathbf{approx\text{-}var}^{X,0}_{\psi}(\mu X_\gamma.\varphi) & \triangleq & \bot & \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi) & \triangleq & \varphi & \varphi \in \{\top,\bot\} \cup \mathsf{Lit} \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi_1 \oplus \varphi_2) & \triangleq & \mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi_1) \oplus \mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi_2) & \oplus \in \{\wedge,\vee\} \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(\star\varphi) & \triangleq & \star\mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi) & \star \in \{\Diamond,\Box\} \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(\sigma Y_\gamma.\varphi) & \triangleq & \sigma Y_{\gamma k}.\mathbf{approx\text{-}var}^{X,k}_{\psi}(\varphi) & Y \neq X, \sigma \in \{\mu,\nu\} \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(X_\gamma) & \triangleq & \mathbf{approx\text{-}var}^{X,k-1}_{\psi}(\mathbf{unfold}_\psi(X)) & \\
\mathbf{approx\text{-}var}^{X,k}_{\psi}(Y_\gamma) & \triangleq & Y_{\gamma k} & Y \neq X
\end{array}
$$

Note that in the above definition, the original formula ψ is kept as a parameter of the **approx-var** operator, so that we are able to retrieve the unfolding of the recursion variable X in ψ.

Proposition 2 For any well-formed formula $\psi \in \mathscr{L}_\mu$, $\mu X.\varphi_X \in \mathbf{Sub}(\psi)$ and any $k \in \mathbb{N}$, the μ-calculus formula $\mathbf{approx\text{-}var}^{X,k}_{\psi}(\mu X.\varphi_X)$ is well-formed.

Proof: We use induction on k.

- Base case: trivial.
- Induction. Observe that $\mathbf{approx\text{-}var}^{X,k+1}_{\psi}(\mu X.\varphi_X)$ preserves the structure of φ_X apart from the occurrences of X, which are unfolded with the "lower" approximants. Since $\mu X.\varphi_X$ is well-formed, the only place where duplicate definition of recursion variables could occur are the unfoldings. However, the indexing scheme guarantees that recursion variables defined in the lower approximants have different names. Together with the inductive assumption, this yields well-formedness. □

We are now in a position to define the general syntactic approximants of a μ-calculus formula, which are constructed by unfolding every μ variable a finite number of times (specified for each variable by a function α), so that the resulting formula is least-fixpoint free.

Definition 15 For a given formula $\varphi \in \mathscr{L}_\mu$ and $\alpha : \mathscr{V}_\mu(\varphi) \longrightarrow \mathbb{N}$ we define an approximant of φ, denoted with $\mathbf{approx}(\varphi,\alpha)$ as:

$$
\mathbf{approx}(\varphi,\alpha) \triangleq \begin{cases} \mathbf{approx}(\mathbf{approx\text{-}var}^{X,\alpha(X)}_{\varphi}(\varphi),\alpha) & \text{if } X_\gamma \in \mathscr{V}_\mu(\varphi) \text{ for some } \gamma \in \mathbb{N}^* \text{ and} \\ & \neg\exists Y_{\gamma'} \in \mathscr{V}_\mu(\varphi).\, \mu X_\gamma.\varphi' \in \mathbf{Sub}(\mathbf{unfold}_\varphi(Y_{\gamma'})) \\ \varphi & \text{if } \mathscr{V}_\mu(\varphi) = \emptyset \end{cases}
$$

Intuitively, as long as there are μ-variables in the formula, we take one (say X) such that for some $\gamma \in \mathbb{N}^*$, X_γ has the outermost occurrence and apply the **approx-var** operator with X as a parameter, by which we obtain a formula that does not contain any variable of the form $X_{\gamma'}$ for any $\gamma' \in \mathbb{N}^*$.

Proposition 3 For any well-formed formula $\psi \in \mathscr{L}_\mu$ and any $\alpha : \mathscr{V}_\mu(\varphi) \longrightarrow \mathbb{N}$, $\mathbf{approx}(\psi,\alpha)$ is a well-formed and least fixpoint-free μ-calculus formula.

Proposition 4 For any GTS M, $M \models \varphi \Leftrightarrow M \models \mathbf{approx}(\varphi,\alpha)$, where $\alpha(X) = |M|$ for every $X \in \mathscr{V}_\mu(\varphi)$.

From the existence of least-fixpoint free approximants we can deduce the following fact: for every formula containing a "true" least fixpoint property $\varphi \in \mathscr{L}_\mu \setminus [\mathscr{L}^{\mathsf{V}}_\mu]_\equiv$, and for an arbitrarily large number $n \in \mathbb{N}$, we can always find a concrete structure, which needs a GTS of size at least n to prove φ using a GTS abstraction. We will use this fact to prove that, assuming that our concrete structures are the most general class **KS**, the completeness set of GTS is exactly the set of least-fixpoint free formulae. First, we introduce some auxiliary notation, needed to facilitate proving this result.

Definition 16 Let $\mathscr{F} = \langle \mathscr{C}, \mathscr{M}, \preceq, \models^{\alpha} \rangle$ be an abstraction formalism, and $\varphi \in \mathscr{L}_{\mu}$ a formula. For an arbitrary Kripke Structure $K \in \mathbf{KS}$ we define

$$\begin{array}{lcl} \mathbf{minmodel}_{\mathscr{F}}(\varphi, K) & \triangleq & \begin{cases} inf\{|M| \mid M \in \mathscr{M} \wedge K \preceq M \models^{\alpha} \varphi\} & \text{if } \varphi \in \mathbf{compl}(\mathscr{F}) \\ \infty & \text{otherwise} \end{cases} \\ \mathbf{maxminmodel}_{\mathscr{F}}(\varphi) & \triangleq & sup\{\mathbf{minmodel}_{\mathscr{F}}(\varphi, K) \mid K \in \mathbf{KS} \wedge K \models \varphi\} \end{array}$$

Lemma 1 For any $\varphi \in \mathscr{L}_{\mu} \setminus [\mathscr{L}_{\mu}^{\nu}]_{\equiv}$, we have $\mathbf{maxminmodel}_{\mathbf{GTS}}(\varphi) = \infty$.

Proof: Suppose, towards a contradiction, that for some $\varphi \in \mathscr{L}_{\mu} \setminus [\mathscr{L}_{\mu}^{\nu}]_{\equiv}$, $\mathbf{maxminmodel}_{\mathbf{GTS}}(\varphi) = n$ for some $n \in \mathbb{N}$. Then for every Kripke Structure K satisfying φ there is a GTS M with a size at most n such that $K \leq_{mix} M \models^{SIS} \varphi$. From Prop. 4 we know that $M \models^{SIS} \varphi$ implies that $M \models \mathbf{approx}(\varphi, \alpha_{|M|})$ (where $\alpha_{|M|}(X) = |M|$ for all $X \in \mathscr{V}_{\mu}(\varphi)$), and hence $K \models \mathbf{approx}(\varphi, \alpha_{|M|})$. This is because of the fact that mixed simulation preserves properties of abstract models to concrete, i.e. the soundness. From the above observations we obtain that for every $K \in \mathbf{KS}$, we have $K \models \varphi \Leftrightarrow K \models \mathbf{approx}(\varphi, \alpha_{|M|})$, so $\varphi \equiv \mathbf{approx}(\varphi, \alpha_{|M|})$. But $\mathbf{approx}(\varphi, \alpha_{|M|}) \in \mathscr{L}_{\mu}^{\nu}$, hence $\varphi \in [\mathscr{L}_{\mu}^{\nu}]_{\equiv}$, a contradiction. □

We are now ready to give the exact characterisation of the completeness set of GTS, in case concrete structures are the most general class **KS**.

Theorem 5 For the general class **KS**, where an arbitrary, possibly infinite number of initial states is allowed, the class of formulae for which GTSs are complete, is, up to semantic equivalence, exactly $\mathscr{L}_{\mu}^{\nu}$, i.e. $\mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle) = [\mathscr{L}_{\mu}^{\nu}]_{\equiv}$.

Proof: The fact that $\mathscr{L}_{\mu}^{\nu} \subseteq \mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$ follows from the known result that GTSs are complete for $\mathscr{L}_{\mu}^{\nu}$ [1]. To prove the inclusion in the other direction, we proceed by contradiction. Assume that $\varphi \in (\mathscr{L}_{\mu} \setminus [\mathscr{L}_{\mu}^{\nu}]_{\equiv}) \cap \mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$.

Since φ is satisfiable, there is a KS K_1 such that $K_1 \models \varphi$, and from $\varphi \in \mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$ there exists a model $M_1 \in \mathbf{GTS}$ of size n_1 such that $K_1 \leq_{mix} M_1 \models^{SIS} \varphi$. Assume that M_1 is the smallest GTS with this property. Because $\varphi \in (\mathscr{L}_{\mu} \setminus [\mathscr{L}_{\mu}^{\nu}]_{\equiv})$, we know from Lem. 1 that $\mathbf{maxminmodel}_{\mathbf{GTS}}(\varphi) = \infty$, so there is some KS K_2, for which the corresponding smallest GTS M_2 proving φ has size $n_2 > n_1$. We can continue this construction ad infinitum, obtaining a sequence (K_i,M_i,n_i), such that $K_i \models \varphi$, $K_i \leq_{mix} M_i \models^{SIS} \varphi$, M_i is the smallest GTS proving φ on K_i and $n_i = |M_i|$.

Let us now define $K = \bigcup_{i=1}^{\infty} K_i$ with $S^0 = \bigcup_{i=1}^{\infty} S_i^0$. Since for all $K_i \models \varphi$, we have for all $i \in \mathbb{N}$, $s \in S_i^0$ $s \models \varphi$, therefore $K \models \varphi$. Since GTSs are complete for φ, there is some $M \in \mathbf{GTS}$ such that $K \leq_{mix} M \models^{SIS} \varphi$. But this means that for all $i \in \mathbb{N}$ $\mathbf{minmodel}_{\mathbf{GTS}}(\varphi, K_i) \leq |M|$, a contradiction. □

Note that in the proof of the above theorem, it is essential that one is allowed to have an infinite set of initial states. If we restrict to Kripke Structures with only finitely many initial states, then we obtain a richer completeness set. We can consider two subcases, depending on whether the concrete structures are finitely branching or not. While we believe that it is difficult to provide an exact characterisation of completeness sets in these cases, we can at least prove that the completeness hierarchy is strict.

By $\mathbf{KS}_{fi}$ we denote the subclass of **KS** with finitely many initial states; by $\mathbf{KS}_{fb}$ we denote the subclass of $\mathbf{KS}_{fi}$, consisting of finitely branching structures.

Theorem 6 We have both $\mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle) \subset \mathbf{compl}(\langle \mathbf{KS}_{fi}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$, and $\mathbf{compl}(\langle \mathbf{KS}_{fi}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle) \subset \mathbf{compl}(\langle \mathbf{KS}_{fb}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$.

Proof: Obviously, smaller classes of concrete structures give rise to larger completeness sets in general. That the inclusions are strict, can be proved with the following counterexamples.

- $\mathbf{compl}(\langle \mathbf{KS}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle) \subset \mathbf{compl}(\langle \mathbf{KS}_{fi}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$:
  By Thm. 5, it suffices to show that there is at least one "true" least fixpoint formula that belongs to $\mathbf{compl}(\langle \mathbf{KS}_{fi}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$. Consider the simple reachability formula $\varphi = \mu X.P \vee \Diamond X$ and suppose that $K \models^{SIS} \varphi$. Then a state labelled with P is reachable from every initial state with a finite number of steps. For any state s of K, let $\text{STEPS}(s)$ denote the minimal path length from s to a P-labelled state, and let n be the largest value of $\text{STEPS}(s^0)$, among all initial states s^0. We can group the states together according to the value of $\text{STEPS}(s)$, and collapse all states with $\text{STEPS}(s) > n$ into one abstract state. We complete the construction of the KMTS by adding must transitions corresponding to decrementing $\text{STEPS}(s)$, and may transitions whenever necessary. It is not too hard to see that such an KMTS mixed simulates K and allows to prove φ.
- $\mathbf{compl}(\langle \mathbf{KS}_{fi}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle) \subset \mathbf{compl}(\langle \mathbf{KS}_{fb}, \mathbf{GTS}, \leq_{mix}, \models^{SIS} \rangle)$:
  Consider a formula expressing that all computations terminate: $\varphi = \mu X.\Box X$. Firstly, observe that there exists an infinitely branching model with one initial state, on which this property holds but cannot be proved with a finite GTS. In its initial state there is a choice between executing n transitions for all $n \in \mathbb{N}$. Every execution path is bound to terminate, but since GTSs are unable to compress the counting of steps, no GTS can finitely approximate this Kripke Structure in a way that would allow to prove φ.

□

5 Conclusions and future work

Abstraction is often a key instrument for turning intractable model checking problems into tractable problems. In the theoretical studies on abstraction frameworks, the expressivity and completeness of the frameworks are the main indicators of the power of the frameworks.

A major problem is that the notion of expressiveness of an abstraction framework is not defined unambiguously in the literature. Wei *et al* [19] established that using a certain notion of expressiveness, the GTS abstraction formalism is equally expressive as the KMTS abstraction formalism. In this paper, we showed that using another common notion of expressiveness, occurring in the literature on abstraction, the GTS abstraction formalisms is actually strictly more expressive than the KMTS abstraction formalism.

The same paper occasionally uses the notions of completeness and expressivity interchangeably. Since both notions are defined differently, we set out to investigate their relations. We proved that only under specific conditions, there is a relation between completeness and expressivity.

Finally, we studied the problem of completeness in more detail. We give tighter characterisations of the completeness of the GTS framework. Among others, we showed that GTSs are complete for exactly the least fixpoint-free fragment of the μ-calculus under the condition that concrete models are Kripke Structures with potentially an infinite number of initial states. We showed that these characterisations change when imposing different requirements on the concrete models.

Several lines of future research are still open. For one, it would be interesting to provide conditions under which expressiveness and completeness are in some way related. A second avenue of research would be investigating exact characterisations of completeness for subclasses of Kripke Structures as concrete models. We expect that this is a very difficult, yet challenging problem to solve.

Acknowledgements The authors would like to thank the anonymous reviewers for their detailed feedback which have helped to improve the paper. This research has been partially funded by the Netherlands

Organisation for Scientific Research (NWO) under grant number 600.065.120 (the *Verification of Complex Hierarchical Systems* project).

References

[1] L. de Alfaro, P. Godefroid & R. Jagadeesan (2004): *Three-Valued Abstractions of Games: Uncertainty, but with Precision*. In: *LICS*, pp. 170–179, doi:10.1109/LICS.2004.1319611.

[2] E. M. Clarke, O. Grumberg & D. E. Long (1992): *Model Checking and Abstraction*. In: *POPL*, pp. 342–354, doi:10.1145/143165.143235.

[3] E. M. Clarke, Jr., O. Grumberg & D. A. Peled (1999): *Model checking*. MIT Press, Cambridge, MA, USA.

[4] E.M. Clarke, O. Grumberg, S. Jha, Y. Lu & H. Veith (2003): *Counterexample-guided abstraction refinement for symbolic model checking*. *J. ACM* 50(5), pp. 752–794, doi:10.1145/876638.876643.

[5] P. Cousot & R. Cousot (1977): *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints*. In: *POPL*, pp. 238–252, doi:10.1145/512950.512973.

[6] D. Dams, R. Gerth & O. Grumberg (1997): *Abstract interpretation of reactive systems*. *ACM Trans. Program. Lang. Syst.* 19(2), pp. 253–291, doi:10.1145/244795.244800.

[7] D. Dams & K. S. Namjoshi (2004): *The Existence of Finite Abstractions for Branching Time Model Checking*. In: *LICS*, pp. 335–344, doi:10.1109/LICS.2004.1319628.

[8] D. Dams & K.S. Namjoshi (2005): *Automata as Abstractions*. In: *VMCAI*, *LNCS* 3385, pp. 216–232, doi:10.1007/978-3-540-30579-8_15.

[9] H. Fecher, D. de Frutos-Escrig, G. Luttgen & H. Schmidt (2009): *On the Expressiveness of Refinement Settings*. In: *FSEN*, *LNCS* 5961, pp. 276–291, doi:10.1007/978-3-642-11623-0_16.

[10] H. Fecher & S. Shoham (2011): *Local abstraction-refinement for the μ-calculus*. *STTT* 13(4), pp. 289–306, doi:10.1007/s10009-011-0202-1.

[11] P. Godefroid & R. Jagadeesan (2003): *On the Expressiveness of 3-Valued Models*. In: *VMCAI*, pp. 206–222, doi:10.1007/3-540-36384-X_18.

[12] A. Gurfinkel & M. Chechik (2005): *How Thorough Is Thorough Enough?* In Dominique Borrione & Wolfgang J. Paul, editors: *CHARME*, *Lecture Notes in Computer Science* 3725, Springer, pp. 65–80, doi:10.1007/11560548_8.

[13] M. Huth, R. Jagadeesan & D.A. Schmidt (2001): *Modal Transition Systems: A Foundation for Three-Valued Program Analysis*. In: *ESOP*, *LNCS* 2028, pp. 155–169, doi:10.1007/3-540-45309-1_11.

[14] K. G. Larsen, U. Nyman & Andrzej Wasowski (2007): *On Modal Refinement and Consistency*. In Luís Caires & Vasco Thudichum Vasconcelos, editors: *CONCUR*, *Lecture Notes in Computer Science* 4703, Springer, pp. 105–119, doi:10.1007/978-3-540-74407-8_8.

[15] K. G. Larsen & B. Thomsen (1988): *A Modal Process Logic*. In: *LICS*, IEEE Computer Society, pp. 203–210, doi:10.1109/LICS.1988.5119.

[16] K. G. Larsen & L. Xinxin (1990): *Equation Solving Using Modal Transition Systems*. In: *LICS*, pp. 108–117, doi:10.1109/LICS.1990.113738.

[17] S. Shoham & O. Grumberg (2004): *Monotonic Abstraction-Refinement for CTL*. In: *TACAS*, *LNCS* 2988, pp. 546–560, doi:10.1007/978-3-540-24730-2_40.

[18] S. Shoham & O. Grumberg (2008): *3-Valued abstraction: More precision at less cost*. *Inf. Comput.* 206(11), pp. 1313–1333, doi:10.1016/j.ic.2008.07.004.

[19] O. Wei, A. Gurfinkel & M. Chechik (2011): *On the consistency, expressiveness, and precision of partial modeling formalisms*. *Inf. Comput.* 209(1), pp. 20–47, doi:10.1016/j.ic.2010.08.001.