Theoretical Computer Science 642 (2016) 24–53

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Nondeterministic Modal Interfaces ☆,☆☆

Ferenc Bujtor [a], Sascha Fendrich [b], Gerald Lüttgen [b], Walter Vogler [a]

[a] Institut für Informatik, University of Augsburg, Germany
[b] Software Technologies Research Group, University of Bamberg, Germany

ARTICLE INFO

Article history:
Received 9 February 2015
Received in revised form 15 April 2016
Accepted 10 June 2016
Available online 16 June 2016
Communicated by R. van Glabbeek

Keywords:
Interface theories
Modal Interface Automata
Component based design
Modal Transition Systems
Disjunctive must-transitions

ABSTRACT

Interface theories are employed in the component-based design of concurrent systems. They often emerge as combinations of Interface Automata (IA) and Modal Transition Systems (MTS), e.g., Nyman et al.'s IOMTS, Bauer et al.'s MIO, Raclet et al.'s MI or our MIA. In this paper, we generalise MI to *nondeterministic* interfaces, for which we properly resolve the longstanding conflict between unspecified inputs being allowed in IA but forbidden in MTS. With this solution we achieve, in contrast to related work, an *associative* parallel composition, a *compositional* preorder, a conjunction on interfaces with *dissimilar alphabets* supporting perspective-based specifications, and a quotienting operator for decomposing *nondeterministic* specifications in a single theory. In addition, we define a hiding and a restriction operator, complement conjunction with a disjunction operator and illustrate our interface theory by means of a simple example.

1. Introduction

Interface theories support the component-based design of concurrent systems and offer a semantic framework for, e.g., software contracts [2] and web services [3]. Several such theories are based on de Alfaro and Henzinger's *Interface Automata* (IA) [4], whose distinguishing feature is a parallel composition on labelled transition systems with inputs and outputs, where receiving an unexpected input is regarded as an error, i.e., a communication mismatch. In so-called *pessimistic* interface theories [5], a parallel composition of components is not defined, if such a mismatch occurs. In *optimistic* theories [6–10], such as the ones we consider here, a communication mismatch is acceptable as long as the system environment prevents that it can be reached; technically, all those states of the parallel composition are pruned from which entering an error state cannot be prevented by any so-called *helpful* environment.

Various researchers have combined IA with Larsen's *Modal Transition Systems* (MTS) [11], featuring may- and must-transitions to express allowed and required behaviour, resp. In a refinement of an interface, all required behaviour must be preserved and no disallowed behaviour may be added. Whereas in IA outputs are optional, they may now be enforced in theories combining IA and MTS, such as Nyman et al.'s IOMTS [8], Bauer et al.'s MIO [5], Raclet et al.'s *Modal Interfaces* (MI) [10] and our *Modal Interface Automata* (MIA) [9,12]. In this article we extend MI to nondeterministic systems, yielding the most general approach to date and permitting new applications, since nondeterminism arises, e.g., from races in

☆ An extended abstract of this paper appeared in [1].

☆☆ Research support was provided by the DFG (German Research Foundation) under grants LU 1748/3-1 and VO 615/12-1.

E-mail addresses: ferenc.bujtor@informatik.uni-augsburg.de (F. Bujtor), sascha.fendrich@swt-bamberg.de (S. Fendrich), gerald.luettgen@swt-bamberg.de (G. Lüttgen), walter.vogler@informatik.uni-augsburg.de (W. Vogler).

http://dx.doi.org/10.1016/j.tcs.2016.06.011
0304-3975/

networks. We build upon our prior work in [12], from which we adopt disjunctive must-transitions, which are needed for operationally defining conjunction on interfaces. Conjunction is a key operator in interface theories, supporting perspective-based specification and corresponding to the greatest lower bound wrt. refinement. We also consider the dual disjunction operator.

Combining IA and MTS is, however, problematic due to a conflict between unspecified inputs being forbidden in MTS but allowed in IA with arbitrary behaviour afterwards. In IOMTS [8], the MTS-view was adopted and, as a consequence, compositionality of refinement wrt. the parallel operator was lost. In [12] we followed the IA-view but found that reconciling the two views is essential for a more flexible conjunction. Flexibility is needed regarding the alphabets of the conjuncts that are to be composed; intuitively, each conjunct models a different perspective (i.e., a single system requirement) that only refers to the actions relevant to that perspective.

Here, we propose a middle way to reconcile the IA- and MTS-views by adding the option to treat an input i in a state p according to the IA-approach: If i should be allowed with subsequent arbitrary behaviour, we add an i-may-transition from p to a special *universal* state e that can be refined in any way. We need this option, in particular, when defining parallel composition. In contrast, if there is no i-transition originating in p, then i is forbidden in p according to the MTS-view. The idea behind e is similar to the one presented for MI in [10], where an ordinary state that has a may-loop for each action is added to a parallel composition. This way, however, associativity of parallel composition is lost. We avoid this problem since e is treated specially in our notion of refinement, which has far reaching consequences for many of the proofs; see Sec. 3.2 for a more detailed discussion of e. Now, with the universal state e and unlike the approach in [9,12], our interface theory, which we continue to call MIA, allows for a proper distinction between may- and must-transitions for inputs. This enables us to define the desired, more flexible conjunction using a simple alphabet extension mechanism in the sense of [10].

Our proposed reconciliation results in an interface theory that generalises the fully deterministic MI, where also internal actions are forbidden, to *nondeterministic* interfaces. Unlike IA and our previous work [9,12], we also do away with determinism on input-transitions. As in MI, our MIA theory is equipped with a multicast parallel composition, where one output can synchronise with several inputs. This is accompanied by hiding and restriction operators for scoping actions [13,14]. Parallel composition and hiding together (cf. [15]) are more expressive than the binary parallel composition of IA used in [5,8,9,12]. We also develop a quotienting operator $/\!\!/$ as a kind of inverse of parallel composition $\|$. For a specification P and a given component D, quotienting constructs the most general component Q such that $Q \| D$ refines P. Quotienting is a practical operator: it can be used for decomposing concurrent specifications stepwise, specifying contracts [16] and reusing components. In contrast to [10], our quotienting permits *nondeterministic* specifications and complements $\|$ rather than a simpler parallel product without pruning.

In summary, our new interface theory MIA generalises and improves upon existing theories combining IA and MTS: parallel composition is commutative and associative (cf. Sec. 3), quotienting also works for nondeterministic specifications (cf. Sec. 4), conjunction properly reflects perspective-based specification (cf. Secs. 5 and 6), and refinement (cf. Sec. 2) is compositional and permits alphabet extension (cf. Sec. 6). We illustrate the utility of MIA by means of a simple example (cf. Sec. 7).

2. Modal Interface Automata: the setting

In this section we define *Modal Interface Automata* (MIA) and the supported operations. Essentially, MIAs are state machines with disjoint input and output alphabets, as in Interface Automata (IA) [4], and two transition relations, *may* and *must*, as in Modal Transition Systems (MTS) [11]. May-transitions describe permitted behaviour, while must-transitions describe required behaviour. Unlike previous versions of MIA [9,12] and also unlike other similar theories, we introduce a special *universal state* e capturing arbitrary behaviour.

Definition 1 *(Modal Interface Automata).* A *Modal Interface Automaton* (MIA) is a tuple $(P, I, O, \longrightarrow, \dashrightarrow, p_0, e)$, where

- P is the set of states including the *initial state* p_0 and the *universal state* e,
- I and O are disjoint sets, the alphabets of *input* and *output actions*, not containing the special internal action τ, and $A =_{\text{df}} I \cup O$ is called the *alphabet*,
- $\longrightarrow \subseteq P \times (A \cup \{\tau\}) \times (\mathcal{P}(P) \setminus \emptyset)$ is the *disjunctive must-transition* relation, with $\mathcal{P}(P)$ being the powerset of P,
- $\dashrightarrow \subseteq P \times (A \cup \{\tau\}) \times P$ is the *may-transition* relation.

We require the following conditions:

1. For all $\alpha \in A \cup \{\tau\}$, $p \xrightarrow{\alpha} P'$ implies $\forall p' \in P'.\ p \overset{\alpha}{\dashrightarrow} p'$ *(syntactic consistency)*,
2. e appears in transitions only as the target state of input may-transitions *(sink condition)*.

A MIA P is called *universal* if $P = (\{e\}, I, O, \emptyset, \emptyset, e, e)$ for alphabets I, O.