Behavioral Specification Theories: an Algebraic Taxonomy

Uli Fahrenberg[1] and Axel Legay[2]

[1] École polytechnique, Palaiseau, France
[2] Université Catholique de Louvain, Belgium

Abstract. We develop a taxonomy of different behavioral specification theories and expose their algebraic properties. We start by clarifying what precisely constitutes a behavioral specification theory and then introduce logical and structural operations and develop the resulting algebraic properties. In order to motivate our developments, we give plenty of examples of behavioral specification theories with different operations.

1 Introduction

Behavioral specification theories are specification formalisms for formal models which are enriched with logical and structural operations. This allows for incremental and compositional design and verification and has shown itself to be a viable way to avoid the habitual state-space explosion problems associated with the verification of complex models.

Behavioral specification theories have seen significant attention in recent years [1, 6, 8, 12–14, 17–19, 40, 41, 44, 46]. Generally speaking, they have the property that the specification formalism is an extension of the modeling formalism, so that specifications have an operational interpretation and models are verified by comparing their operational behavior against the specification's behavior.

Popular examples of behavioral specification theories are modal transition systems [6, 18, 40], disjunctive modal transition systems [12–14, 17, 30–32, 44], and acceptance automata [19, 46]. Also relations to contracts and interfaces have been exposed [8, 47], as have extensions for real-time, probabilistic, and quantitative specifications and for models with data [9–11, 15, 20, 22, 23, 27–29].

Except for the work by Vogler *et al.* in [17, 18] and our own [31], behavioral specification theories have been developed only to characterize bisimilarity (or variants like timed or probabilistic bisimilarity). While bisimilarity is an important equivalence relation on models, there are many others which also are of interest. Examples include nested and k-nested simulation [2,34], ready or $\frac{2}{3}$-simulation [43], trace equivalence [36], impossible futures [51], or the failure semantics of [16–18, 45, 50] and others. We have addressed some of these equivalences in [31].

In this survey we take a step back and develop a *systemization* or *taxonomy* of different behavioral specification theories and expose their *algebraic* properties. As an example, the most basic ingredient of a behavioral specification theory is a preorder of *refinement* on specifications, turning the set of specifications into

a partial order up to $\equiv$, the equivalence generated by refinement. Now if the refinement preorder admits least upper bounds, then this binary operation is usually called *conjunction*, and the set of specifications becomes a meet-semilattice up to $\equiv$. Conjunction is a useful ingredient of any specification theory, but some also admit *disjunctions*, thus turning them into *distributive lattices* up to $\equiv$.

We believe that a systemization as we set out for here is useful to clarify which properties one needs or expects of behavioral specification theories, and that it may help in developing new behavioral specification theories, both for equivalence relations different from bisimilarity and for more intricate models such as real-time, probabilistic, or hybrid systems.

To develop our systemization, we first have to clarify what precisely *is* a behavioral specification theory. Here we follow the seminal work of Pnueli [45], Hennessy and Milner [35], and Larsen [41] and argue that a behavioral specification theory is built on an *adequate* and *expressive* specification formalism equipped with a mapping from models to their *characteristic formulae*, which provides the extension of the modeling formalism by the specification formalism. This is the theme of Sections 2 and 3.

Section 4 then introduces behavioral specification theories, and Section 5 makes precise what it means to have logical operations on specifications. Section 6 is concerned with *structural* operations on specifications: composition and quotient. When present in a specification theory, these can be used for compositional design and verification. Algebraically, a specification theory which has all the logical and structural operations forms a *residuated lattice* up to $\equiv$, a well-understood algebraic structure [37] which also appears in linear logic [33] and other areas.

All throughout Sections 2 to 6, we give plenty of examples, taken from our own work in [14, 27, 31], of specification theories which have the required operations. In the final Section 7 we survey a few other behavioral specification theories, for real-time and probabilistic models, in order to expose their particular algebraic properties. We make no claim to completeness of this survey; indeed there are many other examples which we do not treat here. The paper finishes with a scheme which sums up the relevant algebraic structures and an overview of the properties of the different behavioral specification theories encountered.

2 Models and Specifications

Let Spec be a set of *specifications*, Mod a set of *models*, $\models \subseteq \mathsf{Mod} \times \mathsf{Spec}$ a relation between specifications and models, and $\sim \subseteq \mathsf{Mod} \times \mathsf{Mod}$ an equivalence relation on Mod. The intuition is that Spec is to provide specifications for the models in Mod through the relation $\models$, but *up to* $\sim$, so that two models which are equivalent cannot be distinguished by their specifications. We will make this precise below.

We will generally use $\mathcal{S}$ for specifications and $\mathcal{M}$ for models. For $\mathcal{S} \in \mathsf{Spec}$, let $[\![\mathcal{S}]\!] = \{\mathcal{M} \in \mathsf{Mod} \mid \mathcal{M} \models \mathcal{S}\}$ denote its *set of implementations*. For $\mathcal{M} \in \mathsf{Mod}$,

let $\mathsf{Th}(\mathcal{M}) = \{\mathcal{S} \in \mathsf{Spec} \mid \mathcal{M} \models \mathcal{S}\}$ denote its *set of theories*. We record the following trivial fact:

Lemma 1. *For any $\mathcal{S} \in \mathsf{Spec}$ and $\mathcal{M} \in \mathsf{Mod}$, the following are equivalent:*

(1) $\mathcal{M} \models \mathcal{S}$;
(2) $\mathcal{M} \in [\![\mathcal{S}]\!]$;
(3) $\mathcal{S} \in \mathsf{Th}(\mathcal{M})$.

Example 2. A common type of models is given by *labeled transition systems* (LTS). These are structures $\mathcal{M} = (S, s^0, T)$ consisting of a finite set of states S, an initial state $s^0 \in S$, and transitions $T \subseteq S \times \Sigma \times S$ labeled with symbols from a fixed finite set Σ.

LTS are often considered modulo *bisimilarity*: A *bisimulation* between two LTS $\mathcal{M}_1 = (S_1, s_1^0, T_1)$ and $\mathcal{M}_2 = (S_2, s_2^0, T_2)$ is a relation $R \subseteq S_1 \times S_2$ such that $(s_1^0, s_2^0) \in R$ and for any $(s_1, s_2) \in R$,

(1) for all $(s_1, a, s_1') \in T_1$ there exists $(s_2, a, s_2') \in T_2$ such that $(s_1', s_2') \in R$;
(2) for all $(s_2, a, s_2') \in T_2$ there exists $(s_1, a, s_1') \in T_1$ such that $(s_1', s_2') \in R$;

and then $\mathcal{M}_1$ and $\mathcal{M}_2$ are said to be *bisimilar* if there exists a bisimulation between them.

A common specification formalism for LTS is *Hennessy-Milner logic* [35]. It consists of formulae generated by the abstract syntax

$$\mathsf{HML} \ni \phi, \psi ::= \mathbf{tt} \mid \mathbf{ff} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle a \rangle \phi \mid [a]\phi \quad (a \in \Sigma)\,,$$

with semantics defined by $[\![\mathbf{tt}]\!] = \mathsf{LTS}$, $[\![\mathbf{ff}]\!] = \emptyset$, $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$, $[\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!]$, and

$$[\![\langle a \rangle \phi]\!] = \{(S, s^0, T) \in \mathsf{LTS} \mid \exists (s^0, a, s) \in T : (S, s, T) \in [\![\phi]\!]\}\,;$$
$$[\![[a]\phi]\!] = \{(S, s^0, T) \in \mathsf{LTS} \mid \forall (s^0, a, s) \in T : (S, s, T) \in [\![\phi]\!]\}\,.$$

The Hennessy-Milner theorem [35] then states that HML specifies LTS up to bisimilarity, that is, $\mathcal{M}_1 \sim \mathcal{M}_2$ precisely when $\mathsf{Th}(\mathcal{M}_1) = \mathsf{Th}(\mathcal{M}_2)$. □

Definition 3 ([35]). $(\mathsf{Spec}, \models)$ *is* adequate *for* $(\mathsf{Mod}, \sim)$ *if it holds for any* $\mathcal{M}_1, \mathcal{M}_2 \in \mathsf{Mod}$ *that* $\mathcal{M}_1 \sim \mathcal{M}_2$ *iff* $\mathsf{Th}(\mathcal{M}_1) = \mathsf{Th}(\mathcal{M}_2)$.

3 Characteristic Formulae

Let $\mathcal{M} \in \mathsf{Mod}$. A specification $\mathcal{S} \in \mathsf{Spec}$ is a *characteristic formula for* $\mathcal{M}$ [45] if it holds for any $\mathcal{M}' \in \mathsf{Mod}$ that $\mathcal{M}' \models \mathcal{S}$ iff $\mathsf{Th}(\mathcal{M}') = \mathsf{Th}(\mathcal{M})$.

Lemma 4. *If $\mathcal{S}_1, \mathcal{S}_2 \in \mathsf{Spec}$ are characteristic formulae for $\mathcal{M} \in \mathsf{Mod}$, then* $[\![\mathcal{S}_1]\!] = [\![\mathcal{S}_2]\!]$.

Proof. For any $\mathcal{M}' \in \mathsf{Mod}$, $\mathcal{M}' \in [\![\mathcal{S}_1]\!]$ iff $\mathsf{Th}(\mathcal{M}') = \mathsf{Th}(\mathcal{M})$, iff $\mathcal{M}' \in [\![\mathcal{S}_2]\!]$. □

Definition 5 ([45]). $(\mathsf{Spec}, \models)$ *is* expressive *for* $(\mathsf{Mod}, \sim)$ *if every* $\mathcal{M} \in \mathsf{Mod}$ *admits a characteristic formula.*

Example 6. It is known [3] that HML is *not* expressive for LTS with bisimilarity. Indeed, the simple transition system $(\{s^0\}, s^0, (s^0, a, s^0)\}$ consisting only of a loop at the initial state does not admit a characteristic formula in HML.

The standard remedy [42] for this expressivity failure is to add recursion and maximal fixed points to the logic. For a finite set X of variables, let $\mathsf{HML}(X)$ be the set of formulae generated as follows:

$$\mathsf{HML}(X) \ni \phi, \psi ::= \mathbf{tt} \mid \mathbf{ff} \mid \phi \wedge \psi \mid \phi \vee \psi \mid \langle a \rangle \phi \mid [a]\phi \mid x \quad (a \in \Sigma, x \in X)$$

That is, $\mathsf{HML}(X)$ formulae are HML formulae which additionally may contain variables from X.

A *recursive Hennessy-Milner formula* [14, 42] is a tuple $\mathcal{H} = (X, X^0, \Delta)$ consisting of finite sets $X \supseteq X^0$ of *variables* and *initial* variables, respectively, and a *declaration* $\Delta : X \to \mathsf{HML}(X)$. The set of such formulae is denoted $\mathsf{HML}^{\mathsf{R}}$. The semantics of a formula $\mathcal{H} \in \mathsf{HML}^{\mathsf{R}}$ is a set $[\![\mathcal{H}]\!] \in \mathsf{LTS}$ which is defined as a maximal fixed point, see [3, 42]; we do not go into these details here because we will give another, equivalent, semantics below.

The *characteristic formula* [42] of $(S, s^0, T) \in \mathsf{LTS}$ is now the $\mathsf{HML}^{\mathsf{R}}$ formula $(S, \{s^0\}, \Delta)$ given by

$$\Delta(s) = \bigwedge_{(s,a,t) \in T} \langle a \rangle t \wedge \bigwedge_{a \in \Sigma} [a] \Big(\bigvee_{(s,a,t) \in T)} t \Big) .$$

Note how $\Delta(s)$ precisely specifies all labels which must be available from s (the first part of the conjunction) and, for each label, which properties must be satisfied after its occurrence (the second part of the conjunction). □

4 Specification Theories

Definition 7 ([31]). *A* behavioral specification theory *for* $(\mathsf{Mod}, \sim)$ *consists of a set* Spec *of specifications, a relation* $\models \subseteq \mathsf{Mod} \times \mathsf{Spec}$*, a mapping* $\chi : \mathsf{Mod} \to \mathsf{Spec}$*, and a preorder* $\leq$ *on* Spec*, called* refinement*, subject to the following conditions:*

(1) $(\mathsf{Spec}, \models)$ *is adequate for* $(\mathsf{Mod}, \sim)$*;*
(2) for every $\mathcal{M} \in \mathsf{Mod}$*,* $\chi(\mathcal{M})$ *is a characteristic formula for* $\mathcal{M}$*;*
(3) for all $\mathcal{M} \in \mathsf{Mod}$ *and all* $\mathcal{S} \in \mathsf{Spec}$*,* $\mathcal{M} \models \mathcal{S}$ *iff* $\chi(\mathcal{M}) \leq \mathcal{S}$*.*

We will generally omit "behavioral" from now and only speak about *specification theories.*

The equivalence relation $\equiv$ on Spec defined as $\leq \cap \geq$ is called *modal equivalence*. Some comments on the different ingredients above are in order.

1. By (2), $(\mathsf{Spec}, \models)$ is also *expressive* for $(\mathsf{Mod}, \sim)$.
2. χ is a *section* of $\models$: for all $\mathcal{M} \in \mathsf{Mod}$, $\mathcal{M} \models \chi(\mathcal{M})$.
3. (3) can be seen as *defining* $\models$, so we may omit $\models$ from the signature of specification theories.
4. For any $\mathcal{M} \in \mathsf{Mod}$, $\mathsf{Th}(\mathcal{M}) = \{\mathcal{S} \in \mathsf{Spec} \mid \chi(\mathcal{M}) \leq \mathcal{S}\} = \chi(\mathcal{M})\uparrow$ is the *upward closure* of $\chi(\mathcal{M})$ with respect to $\leq$.

Lemma 8 ([31]). *Let* $(\mathsf{Spec}, \chi, \leq)$ *be a specification theory for* $(\mathsf{Mod}, \sim)$.

(1) For all $\mathcal{S}_1, \mathcal{S}_2 \in \mathsf{Spec}$, $\mathcal{S}_1 \leq \mathcal{S}_2$ *implies* $[\![\mathcal{S}_1]\!] \subseteq [\![\mathcal{S}_2]\!]$.
(2) For all $\mathcal{M}_1, \mathcal{M}_2 \in \mathsf{Mod}$, $\mathcal{M}_1 \sim \mathcal{M}_2$ *iff* $\chi(\mathcal{M}_1) \leq \chi(\mathcal{M}_2)$.

Proof. For the first claim, $\mathcal{M} \in [\![\mathcal{S}_1]\!]$ implies $\chi(\mathcal{M}) \leq \mathcal{S}_1 \leq \mathcal{S}_2$, hence $\mathcal{M} \in [\![\mathcal{S}_2]\!]$.

For the second claim, we have $\mathcal{M}_1 \sim \mathcal{M}_2$ iff $\mathcal{M}_1 \models \chi(\mathcal{M}_2)$ (as $\chi(\mathcal{M}_2)$ is characteristic for $\mathcal{M}_2$), iff $\chi(\mathcal{M}_1) \leq \chi(\mathcal{M}_2)$ by (3). □

Example 9. [14] introduces a *normal form* for $\mathsf{HML}^{\mathsf{R}}$ formulae, showing that for any $\mathsf{HML}^{\mathsf{R}}$ formula $\mathcal{H}_1 = (X_1, X_1^0, \Delta_1)$, there exists another formula $\mathcal{H}_2 = (X_2, X_2^0, \Delta_2)$ with $[\![\mathcal{H}_1]\!] = [\![\mathcal{H}_2]\!]$ and such that for any $x \in X_2$, $\Delta_2(x)$ is of the form

$$\Delta_2(x) = \bigwedge_{N \in \Diamond(x)} \Big(\bigvee_{(a,y) \in N} \langle a \rangle y \Big) \wedge \bigwedge_{a \in \Sigma} [a] \Big(\bigvee_{y \in \Box^a(x)} y \Big),$$

for finite sets $\Diamond(x) \subseteq 2^{\Sigma \times X_2}$ and, for each $a \in \Sigma$, $\Box^a(x) \subseteq X_2$. This may be seen as generalizing the characteristic formulae of $\mathsf{HML}^{\mathsf{R}}$: the first part of the conjunction in $\Delta_2(x)$ specifies all labels which must be available, and the second part, which properties must be satisfied after each label's occurrence.

A *refinement* [14] of two $\mathsf{HML}^{\mathsf{R}}$ formulae $\mathcal{H}_1 = (X_1, X_1^0, \Delta_1)$ and $\mathcal{H}_2 = (X_2, X_2^0, \Delta_2)$ in normal form is a relation $R \subseteq X_1 \times X_2$ such that for every $x_1^0 \in X_1^0$ there exists $x_2^0 \in X_2^0$ for which $(x_1^0, x_2^0) \in R$, and for any $(x_1, x_2) \in R$,

(1) for all $N_2 \in \Diamond_2(x_2)$ there is $N_1 \in \Diamond_1(x_1)$ such that for each $(a, y_1) \in N_1$, there exists $(a, y_2) \in N_2$ with $(y_1, y_2) \in R$;
(2) for all $a \in \Sigma$ and every $y_1 \in \Box_1^a(x_1)$, there is $y_2 \in \Box_2^a(x_2)$ for which $(y_1, y_2) \in R$.

Note how this corresponds to the intuition for the normal form above.

Writing $\mathcal{H}_1 \leq \mathcal{H}_2$ whenever there exists a refinement as above, and denoting by $\chi(\mathcal{M})$ the characteristic formula of $\mathcal{M} \in \mathsf{LTS}$ introduced in the previous example, it can be shown [14] that $(\mathsf{HML}^{\mathsf{R}}, \chi, \leq)$ is a specification theory for LTS under bisimulation. This also implies that the refinement semantics of $\mathsf{HML}^{\mathsf{R}}$ agrees with the standard fixed-point semantics [3, 42]. □

Example 10. [14] exposes structural translations between $\mathsf{HML}^{\mathsf{R}}$ and two other specification formalism: a generalization of the *disjunctive modal transition systems* (DMTS) introduced in [44] to multiple initial states, and a non-deterministic

version of the *acceptance automata* (AA) of [19, 46]. This yields two other specification theories for LTS under bisimulation, one DMTS-based and one based on (non-deterministic) acceptance automata. □

Example 11. [31] introduces DMTS-based specification theories for $(\mathsf{LTS}, \cong)$, where $\cong$ is any equivalence in van Glabbeek's linear-time–branching-time spectrum [49]. Using the translations mentioned in the previous example, these also give rise to $\mathsf{HML}^{\mathsf{R}}$-based specification theories, and to specification theories based on acceptance automata, for all those equivalences. □

5 Logical Operations on Specifications

Behavioral specifications typically come equipped with logical operations of conjunction and disjunction. Recall that $\equiv$ is defined as $\leq \cap \geq$.

Definition 12. *A specification theory* $(\mathsf{Spec}, \chi, \leq)$ *for* $(\mathsf{Mod}, \sim)$ *is* logical *if* $(\mathsf{Spec}, \leq)$ *forms a bounded distributive lattice up to* $\equiv$.

The above implies that Spec admits commutative and associative binary operations $\vee$ of least upper bound and $\wedge$ of greatest lower bound: disjunction and conjunction. It also entails that there is a bottom specification $\mathbf{ff} \in \mathsf{Spec}$, satisfying $[\![\mathbf{ff}]\!] = \emptyset$, and a top specification $\mathbf{tt} \in \mathsf{Spec}$, satisfying $[\![\mathbf{tt}]\!] = \mathsf{Mod}$. We sum up the properties of these operations:

$$\mathcal{S}_1 \vee \mathcal{S}_2 \leq \mathcal{S}_3 \quad \text{iff} \quad \mathcal{S}_1 \leq \mathcal{S}_3 \text{ and } \mathcal{S}_2 \leq \mathcal{S}_3 \tag{1}$$

$$\mathcal{S}_1 \leq \mathcal{S}_2 \wedge \mathcal{S}_3 \quad \text{iff} \quad \mathcal{S}_1 \leq \mathcal{S}_2 \text{ and } \mathcal{S}_1 \leq \mathcal{S}_3 \tag{2}$$

$$\mathcal{S}_1 \wedge (\mathcal{S}_2 \vee \mathcal{S}_3) \equiv (\mathcal{S}_1 \wedge \mathcal{S}_2) \vee (\mathcal{S}_1 \wedge \mathcal{S}_3)$$

$$\mathcal{S}_1 \vee (\mathcal{S}_2 \wedge \mathcal{S}_3) \equiv (\mathcal{S}_1 \vee \mathcal{S}_2) \wedge (\mathcal{S}_1 \vee \mathcal{S}_3)$$

$$\mathbf{ff} \wedge \mathcal{S} \equiv \mathbf{ff} \qquad \mathbf{tt} \wedge \mathcal{S} \equiv \mathcal{S}$$

$$\mathbf{ff} \vee \mathcal{S} \equiv \mathcal{S} \qquad \mathbf{tt} \vee \mathcal{S} \equiv \mathbf{tt}$$

Note that the properties of least upper bound and greatest lower bound in (1) and (2) above *define* $\vee$ and $\wedge$ uniquely: they are universal properties.

Example 13. Hennessy-Milner logic has disjunction and conjunction as part of the syntax, and [14] shows that on the specification theory $(\mathsf{HML}^{\mathsf{R}}, \chi, \leq)$ from previous examples these are operations as above. The disjunction of two $\mathsf{HML}^{\mathsf{R}}$ formulae in normal form is again in normal form; for conjunction it may be defined directly on normal forms as follows:

Let $\mathcal{H}_1 = (X_1, X_1^0, \Delta_1)$ and $\mathcal{H}_2 = (X_2, X_2^0, \Delta_2)$ be $\mathsf{HML}^{\mathsf{R}}$ formulae in normal form and define $\mathcal{H} = (X_1 \times X_2, X_1^0 \times X_1^1, \Delta)$ by $\Box^a((x_1, x_2)) = \Box_1^a(x_1) \wedge \Box_2^a(x_2)$ for every $a \in \Sigma$ and $(x_1, x_2) \in X$ and

$$\begin{aligned}\Diamond((x_1, x_2)) = \big\{\{(a, (y_1, y_2)) \mid (a, y_1) \in N_1, y_2 \in \Box_2^a(x_2)\} \;\big|\; N_1 \in \Diamond_2(x_1)\big\} \\ \cup \big\{\{(a, (y_1, y_2)) \mid (a, y_2) \in N_2, y_1 \in \Box_1^a(x_1)\} \;\big|\; N_2 \in \Diamond_2(x_2)\big\},\end{aligned}$$

then $\mathcal{H} \equiv \mathcal{H}_1 \wedge \mathcal{H}_2$ [14].

Hence the three specification theories for $(\mathsf{Mod}, \sim)$ of [14]: $\mathsf{HML}^{\mathsf{R}}$, DMTS, and AA, are all logical. □

As a variation, some specification theories only admit conjunction and no disjunction, thus forming a *bounded meet-semilattice*. We call such specification theories *semi-logical*.

6 Structural Operations on Specifications

Many behavioral specifications also admit structural operations of *composition*, denoted $\|$, and *quotient*, denoted $/$, in order to enable compositional design and verification.

Definition 14. *A* compositional specification theory *is a specification theory* $(\mathsf{Spec}, \chi, \leq)$ *for* $(\mathsf{Mod}, \sim)$ *together with an operation* $\|$ *on* Spec *such that* $(\mathsf{Spec}, \|, \leq)$ *forms a commutative partially ordered semigroup up to* $\equiv$.

That is to say that the operation $\|$ is commutative and associative and additionally satisfies the following monotonicity law:

$$\mathcal{S}_1 \leq \mathcal{S}_2 \Longrightarrow \mathcal{S}_1 \| \mathcal{S}_3 \leq \mathcal{S}_2 \| \mathcal{S}_3$$

Contrary to the logical operations $\vee$ and $\wedge$, $\|$ is *not* defined uniquely; indeed a specification theory may admit many different composition operations.

Corollary 15 (Independent implementability). *If* $(\mathsf{Spec}, \|, \chi, \leq)$ *is compositional, then* $\mathcal{S}_1 \leq \mathcal{S}_3$ *and* $\mathcal{S}_2 \leq \mathcal{S}_4$ *imply* $\mathcal{S}_1 \| \mathcal{S}_2 \leq \mathcal{S}_3 \| \mathcal{S}_4$.

Proof. By monotonicity, $\mathcal{S}_1 \| \mathcal{S}_2 \leq \mathcal{S}_3 \| \mathcal{S}_2 \leq \mathcal{S}_3 \| \mathcal{S}_4$. □

Note that independent implementability also *implies* the monotonicity law above.

If $(\mathsf{Spec}, \|, \chi, \leq)$ is *compositional and logical*, then it is called a *lattice-ordered semigroup* (up to $\equiv$) as an algebraic structure; more precisely a bounded distributive lattice-ordered commutative semigroup. This entails that composition distributes over disjunction:

$$\mathcal{S}_1 \| (\mathcal{S}_2 \vee \mathcal{S}_2) \equiv \mathcal{S}_1 \| \mathcal{S}_2 \vee \mathcal{S}_1 \| \mathcal{S}_3$$

Note that composition does *not* necessarily distributed over *conjunction*.

If composition $\|$ also admits a unit $\mathbb{1} \in \mathsf{Spec}$ (up to $\equiv$), *i.e.* such that $\mathcal{S} \| \mathbb{1} \equiv \mathcal{S}$ for all $\mathcal{S} \in \mathsf{Spec}$, then $(\mathsf{Spec}, \|, \chi, \leq)$ is said to be *unital*, and "semigroup" is replaced by "monoid" above.

Definition 16. *A compositional specification theory* $(\mathsf{Spec}, \|, \chi, \leq)$ *for* $(\mathsf{Mod}, \sim)$ *is* complete *if* $(\mathsf{Spec}, \|, \leq)$ *forms a* residuated *partially ordered commutative semigroup up to* $\equiv$.

That is, the operation $\|$ admits a *residual* $/$, in our context called *quotient*, satisfying the following property:

$$\mathcal{S}_1 \| \mathcal{S}_2 \leq \mathcal{S}_3 \iff \mathcal{S}_2 \leq \mathcal{S}_3 / \mathcal{S}_1 \tag{3}$$

This property is again universal, so that $/$ is uniquely defined by $\|$.

If $(\mathsf{Spec}, \|, \chi, \leq)$ is also unital, then it forms a *residuated poset* up to $\equiv$. In that case, the following holds for all $\mathcal{S}_1, \mathcal{S}_2 \in \mathsf{Spec}$:

$$\mathcal{S}_1 \| (\mathbb{1} / \mathcal{S}_2) \leq \mathcal{S}_1 / \mathcal{S}_2$$

We refer to [37] for a survey on residuated posets and the residuated lattices we will encounter in a moment; we only highlight a few properties here.

Lemma 17 ([37]). *The following hold in any complete compositional specification theory:*

$$\begin{aligned} \mathcal{S}_1 \| (\mathcal{S}_2 / \mathcal{S}_3) &\leq (\mathcal{S}_1 \| \mathcal{S}_2) / \mathcal{S}_3 & \mathcal{S}_1 / \mathcal{S}_2 &\leq (\mathcal{S}_1 \| \mathcal{S}_3) / (\mathcal{S}_2 \| \mathcal{S}_3) \\ (\mathcal{S}_1 / \mathcal{S}_2) \| (\mathcal{S}_2 / \mathcal{S}_3) &\leq \mathcal{S}_1 / \mathcal{S}_3 & (\mathcal{S}_1 / \mathcal{S}_2) / \mathcal{S}_3 &\equiv (\mathcal{S}_1 / \mathcal{S}_3) / \mathcal{S}_2 \\ \mathcal{S}_1 / (\mathcal{S}_2 \| \mathcal{S}_3) &\equiv (\mathcal{S}_1 / \mathcal{S}_2) / \mathcal{S}_3 & \mathcal{S} \| (\mathcal{S} / \mathcal{S}) &\equiv \mathcal{S} \\ (\mathcal{S} / \mathcal{S}) \| (\mathcal{S} / \mathcal{S}) &\equiv \mathcal{S} / \mathcal{S} \end{aligned}$$

If $(\mathsf{Spec}, \|, \mathbb{1}, \chi, \leq)$ is *complete compositional and logical*, then it is called a *residuated lattice-ordered semigroup* (up to $\equiv$); more precisely a bounded distributive residuated lattice-ordered commutative semigroup. Distributivity of composition over disjunction now follows from residuation, and also the quotient is well-behaved with respect to the logical operations:

$$(\mathcal{S}_1 \wedge \mathcal{S}_2) / \mathcal{S}_3 \equiv \mathcal{S}_1 / \mathcal{S}_3 \wedge \mathcal{S}_2 / \mathcal{S}_3 \qquad \mathcal{S}_1 / (\mathcal{S}_2 \vee \mathcal{S}_3) \equiv \mathcal{S}_1 / \mathcal{S}_2 \wedge \mathcal{S}_1 / \mathcal{S}_3$$

Additionally, composition and quotient interact with the bottom and top elements as follows:

$$\mathcal{S} \| \mathbf{ff} \equiv \mathbf{ff} \qquad \mathcal{S} / \mathbf{ff} \equiv \mathbf{tt} \qquad \mathbf{tt} / \mathcal{S} \equiv \mathbf{tt}$$

Finally, if $(\mathsf{Spec}, \|, \mathbb{1}, \chi, \leq)$ is complete compositional, unital, and logical, then it is called a *residuated lattice*. We sum up the different algebraic structures we have encountered in Fig. 1.

Example 18. In [14] it is shown that the specification theory $(\mathsf{HML}^{\mathsf{R}}, \chi, \leq)$, and thus also the specification theories based on DMTS and AA, are unital complete compositional when enriched with CSP-style composition $\|$. (In [27] this is generalized to other types of composition.)

The composition $\mathcal{H}_1 \| \mathcal{H}_2$ is defined by translation between $\mathsf{HML}^{\mathsf{R}}$ and AA. Also quotient is defined through AA, and it is shown in [14] that due to these translations, composition may incur an exponential blow-up and quotient a double-exponential blow-up. □

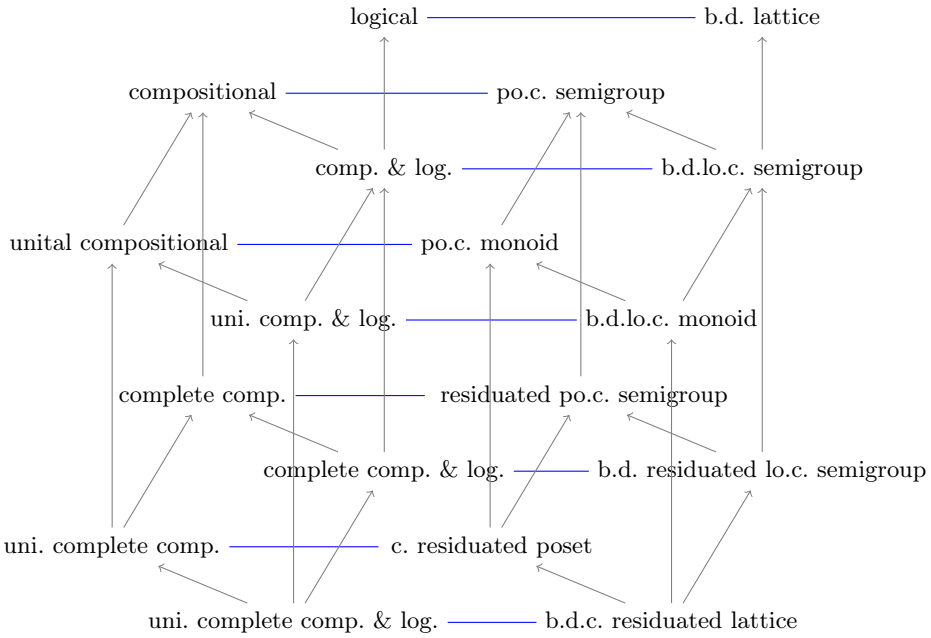

Fig. 1. Spectrum of specification theories and the corresponding algebraic structures. Abbreviations: b.—bounded; d.—distributive; c.—commutative; po.—partially ordered; lo.—lattice-ordered

7 Specification Theories for Real-Time and Probabilistic Systems

We quickly survey a few different specification theories for real-time and probabilistic systems.

7.1 Modal event-clock specifications

Modal event-clock specifications (MECS) were introduced in [15]. They form a specification theory for event-clock automata (ECA) [5], a determinizable subclass of timed automata [4], under timed bisimilarity. Models and specifications are assume to be deterministic, thus $\mathcal{S}_1 \leq \mathcal{S}_2$ iff $[\![\mathcal{S}_1]\!] \subseteq [\![\mathcal{S}_2]\!]$ in this case.

In [15] it is shown that MECS admit a conjunction, thus forming a meet-semilattice up to $\equiv$. The authors also introduce composition and quotient; but computation of quotient incurs an exponential blow-up. Altogether, MECS form a complete compositional semi-logical specification theory: a bounded residuated semilattice-ordered commutative semigroup.

Table 1. Algebraic taxonomy of some specification theories. Abbreviations: L—logical; C—compositional; Q—complete

Specifications	Models	L	C	Q	Notes
$\mathsf{HML}^{\mathsf{R}}$, DMTS, AA	LTS, bisim.	✓	✓	✓	[14]; bisimulation
$\mathsf{HML}^{\mathsf{R}}$, DMTS, AA	LTS, any	✗	✗	✗	[31]; any equivalence in LTBT spectrum [49]
DMTS	LTS, fail./div.	≈	✓	✗	[17]; failure/divergence equivalence; no disjunction
MECS	ECA, t.bisim.	≈	✓	✓	[15]; timed bisim.; no disjunction
TIOA	TIOA, t.bisim.	≈	✓	≈	[20]; no disjunction; weak quotient
IMC	PA, p.bisim.	✗	✗	✗	[25]; probabilistic bisim.
APA	PA, p.bisim.	≈	✓	✗	[24]; no disjunction

7.2 Timed input/output automata

[20, 21] introduce a specification theory based on a variant of the timed input/output automata (TIOA) of [38, 39]. Both models and specifications are TIOA which are action-deterministic and input-enabled; but models are further restricted using conditions of output urgency and independent progress. The equivalence on models being specified is timed bisimilarity.

In [20] it is shown that TIOA admit a conjunction, so they form a meet-semilattice up to $\equiv$. The paper also introduces a composition operation and a quotient, but the quotient is only shown to satisfy the property that

$$\mathcal{S}_1 \| \mathcal{M} \leq \mathcal{S}_3 \iff \mathcal{M} \leq \mathcal{S}_3 / \mathcal{S}_1$$

for all specifications $\mathcal{S}_1, \mathcal{S}_3$ and all *models* $\mathcal{M}$, which is strictly weaker than (3). With this caveat, TIOA form a complete compositional semi-logical specification theory: a bounded residuated semilattice-ordered commutative semigroup.

7.3 Abstract probabilistic automata

Abstract probabilistic automata (APA), introduced in [23,24], form a specification theory for probabilistic automata (PA) [48] under probabilistic bisimilarity. They build on earlier models of interval Markov chains (IMC) [25], see also [7, 26] for a related line of work.

In [24] it is shown that APA admit a conjunction, but that IMC do not. Also a composition is introduced in [24], and it is shown that composing two APA with interval constraints (an IMC) may yield an APA with *polynomial* constraints (not an IMC); but APA with polynomial constraints are closed under composition. APA form a compositional semi-logical specification theory: a bounded semilattice-ordered commutative semigroup.

Table 1 sums up the algebraic properties of the different specification theories we have surveyed here, plus the specification theory for failure/divergence semantics based on DMTS from [17].

References

1. Luca Aceto, Ignacio Fábregas, David de Frutos-Escrig, Anna Ingólfsdóttir, and Miguel Palomino. On the specification of modal systems. *Sci. Comput. Program.*, 78(12):2468–2487, 2013.
2. Luca Aceto, Wan Fokkink, Rob J. van Glabbeek, and Anna Ingólfsdóttir. Nested semantics over finite trees are equationally hard. *Inf. Comput.*, 191(2):203–232, 2004.
3. Luca Aceto, Anna Ingólfsdóttir, Kim G. Larsen, and Jiří Srba. *Reactive Systems*. Cambridge Univ. Press, 2007.
4. Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
5. Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theor. Comput. Sci.*, 211(1-2):253–273, 1999.
6. Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wąsowski. 20 years of modal and mixed specifications. *Bull. EATCS*, 95:94–129, 2008.
7. Anicet Bart, Benoît Delahaye, Paulin Fournier, Didier Lime, Eric Monfroy, and Charlotte Truchet. Reachability in parametric interval Markov chains using constraints. *Theor. Comput. Sci.*, 747:48–74, 2018.
8. Sebastian S. Bauer, Alexandre David, Rolf Hennicker, Kim G. Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wąsowski. Moving from specifications to contracts in component-based design. In Juan de Lara and Andrea Zisman, editors, *FASE*, volume 7212 of *Lect. Notes Comput. Sci.*, pages 43–58. Springer-Verlag, 2012.
9. Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus Thrane. Quantitative refinement for weighted modal transition systems. In Filip Murlak and Piotr Sankowski, editors, *MFCS*, volume 6907 of *Lect. Notes Comput. Sci.*, pages 60–71. Springer-Verlag, 2011.
10. Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus Thrane. Weighted modal transition systems. *Form. Meth. Syst. Design*, 42(2):193–220, 2013.
11. Sebastian S. Bauer, Line Juhl, Kim G. Larsen, Axel Legay, and Jiří Srba. Extending modal transition systems with structured labels. *Math. Struct. Comput. Sci.*, 22(4):581–617, 2012.
12. Nikola Beneš, Ivana Černá, and Jan Křetínský. Modal transition systems: Composition and LTL model checking. In Tevfik Bultan and Pao-Ann Hsiung, editors, *ATVA*, volume 6996 of *Lect. Notes Comput. Sci.*, pages 228–242. Springer-Verlag, 2011.
13. Nikola Beneš, Benoît Delahaye, Uli Fahrenberg, Jan Křetínský, and Axel Legay. Hennessy-Milner logic with greatest fixed points as a complete behavioural specification theory. In Pedro R. D'Argenio and Hernán C. Melgratti, editors, *CONCUR*, volume 8052 of *Lect. Notes Comput. Sci.*, pages 76–90. Springer-Verlag, 2013.
14. Nikola Beneš, Uli Fahrenberg, Jan Křetínský, Axel Legay, and Louis-Marie Traonouez. Logical vs. behavioural specifications. *Inf. Comput.*, 271:104487, 2020.
15. Nathalie Bertrand, Axel Legay, Sophie Pinchinat, and Jean-Baptiste Raclet. Modal event-clock specifications for timed component-based design. *Sci. Comput. Program.*, 77(12):1212–1234, 2012.
16. Stephen D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *J. ACM*, 31(3):560–599, 1984.

17. Ferenc Bujtor, Lev Sorokin, and Walter Vogler. Testing preorders for dMTS. *ACM Trans. Embedded Comput. Syst.*, 16(2):41:1–41:28, 2017.
18. Ferenc Bujtor and Walter Vogler. Failure semantics for modal transition systems. *ACM Trans. Embedded Comput. Syst.*, 14(4):67, 2015.
19. Benoît Caillaud and Jean-Baptiste Raclet. Ensuring reachability by design. In Abhik Roychoudhury and Meenakshi D'Souza, editors, *ICTAC*, volume 7521 of *Lect. Notes Comput. Sci.*, pages 213–227. Springer-Verlag, 2012.
20. Alexandre David, Kim G. Larsen, Axel Legay, Ulrik Nyman, Louis-Marie Traonouez, and Andrzej Wąsowski. Real-time specifications. *Softw. Tools Technol. Transf.*, 17(1):17–45, 2015.
21. Alexandre David, Kim Guldstrand Larsen, Axel Legay, Mikael H. Møller, Ulrik Nyman, Anders P. Ravn, Arne Skou, and Andrzej Wąsowski. Compositional verification of real-time systems using Ecdar. *Softw. Tools Technol. Transf.*, 14(6):703–720, 2012.
22. Benoît Delahaye, Uli Fahrenberg, Kim G. Larsen, and Axel Legay. Refinement and difference for probabilistic automata. In Kaustubh R. Joshi, Markus Siegle, Mariëlle Stoelinga, and Pedro R. D'Argenio, editors, *QEST*, volume 8054 of *Lect. Notes Comput. Sci.*, pages 22–38. Springer-Verlag, 2013.
23. Benoît Delahaye, Uli Fahrenberg, Kim G. Larsen, and Axel Legay. Refinement and difference for probabilistic automata. *Logical Methods Comput. Sci.*, 10(3), 2014.
24. Benoît Delahaye, Joost-Pieter Katoen, Kim G. Larsen, Axel Legay, Mikkel L. Pedersen, Falak Sher, and Andrzej Wasowski. Abstract probabilistic automata. *Inf. Comput.*, 232:66–116, 2013.
25. Benoît Delahaye, Kim G. Larsen, Axel Legay, Mikkel L. Pedersen, and Andrzej Wąsowski. Consistency and refinement for interval Markov chains. *Log. Algebraic Program.*, 81(3):209–226, 2012.
26. Benoît Delahaye, Didier Lime, and Laure Petrucci. Parameter synthesis for parametric interval Markov chains. In Barbara Jobstmann and K. Rustan M. Leino, editors, *VMCAI*, volume 9583 of *Lect. Notes Comput. Sci.*, pages 372–390. Springer-Verlag, 2016.
27. Uli Fahrenberg, Jan Křetínský, Axel Legay, and Louis-Marie Traonouez. Compositionality for quantitative specifications. *Soft Comput.*, 22(4):1139–1158, 2018.
28. Uli Fahrenberg and Axel Legay. A robust specification theory for modal event-clock automata. In Sebastian S. Bauer and Jean-Baptiste Raclet, editors, *FIT*, volume 87 of *Electr. Proc. Theor. Comput. Sci.*, pages 5–16, 2012.
29. Uli Fahrenberg and Axel Legay. General quantitative specification theories with modal transition systems. *Acta Inf.*, 51(5):261–295, 2014.
30. Uli Fahrenberg and Axel Legay. A linear-time-branching-time spectrum of behavioral specification theories. In Bernhard Steffen, Christel Baier, Mark van den Brand, Johann Eder, Mike Hinchey, and Tiziana Margaria, editors, *SOFSEM*, volume 10139 of *Lect. Notes Comput. Sci.*, pages 49–61. Springer-Verlag, 2017.
31. Uli Fahrenberg and Axel Legay. A linear-time-branching-time spectrum for behavioral specification theories. *J. Log. Algebraic Methods Program.*, 110, 2020.
32. Uli Fahrenberg, Axel Legay, and Louis-Marie Traonouez. Structural refinement for the modal nu-calculus. In Gabriel Ciobanu and Dominique Méry, editors, *ICTAC*, volume 8687 of *Lect. Notes Comput. Sci.*, pages 169–187. Springer-Verlag, 2014.
33. Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
34. Jan Friso Groote and Frits W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Inf. Comput.*, 100(2):202–260, 1992.
35. Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.

36. C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.
37. Peter Jipsen and Constantine Tsinakis. A survey of residuated lattices. In *Ordered algebraic structures*, volume 7, pages 19–56. Kluwer Acad. Publ., 2002.
38. Dilsun Kirli Kaynar, Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. Timed I/O automata: A mathematical framework for modeling and analyzing real-time systems. In *RTSS*, pages 166–177. IEEE Computer Society, 2003.
39. Dilsun Kirli Kaynar, Nancy A. Lynch, Roberto Segala, and Frits W. Vaandrager. *The Theory of Timed I/O Automata.* Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, second edition, 2010.
40. Kim G. Larsen. Modal specifications. In Joseph Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lect. Notes Comput. Sci.*, pages 232–246. Springer-Verlag, 1989.
41. Kim G. Larsen. Ideal specification formalism = expressivity + compositionality + decidability + testability + ... In Jos C. M. Baeten and Jan Willem Klop, editors, *CONCUR*, volume 458 of *Lect. Notes Comput. Sci.*, pages 33–56. Springer-Verlag, 1990.
42. Kim G. Larsen. Proof systems for satisfiability in Hennessy-Milner logic with recursion. *Theor. Comput. Sci.*, 72(2&3):265–288, 1990.
43. Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. In *POPL*, pages 344–352. ACM Press, 1989.
44. Kim G. Larsen and Liu Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117. IEEE Computer Society, 1990.
45. Amir Pnueli. Linear and branching structures in the semantics and logics of reactive systems. In Wilfried Brauer, editor, *ICALP*, volume 194 of *Lect. Notes Comput. Sci.*, pages 15–32. Springer-Verlag, 1985.
46. Jean-Baptiste Raclet. Residual for component specifications. *Electr. Notes Theor. Comput. Sci.*, 215:93–110, 2008.
47. Jean-Baptiste Raclet, Eric Badouel, Albert Benveniste, Benoît Caillaud, Axel Legay, and Roberto Passerone. A modal interface theory for component-based design. *Fund. Inf.*, 108(1-2), 2011.
48. Roberto Segala and Nancy A. Lynch. Probabilistic simulations for probabilistic processes. *Nord. J. Comput.*, 2(2):250–273, 1995.
49. Rob J. van Glabbeek. The linear time—branching time spectrum I. In Jan A. Bergstra, Alban Ponse, and Scott A. Smolka, editors, *Handbook of Process Algebra*, Chapter 1, pages 3–99. Elsevier, 2001.
50. Walter Vogler. Failures semantics and deadlocking of modular Petri nets. *Acta Inf.*, 26(4):333–348, 1989.
51. Walter Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, volume 625 of *Lect. Notes Comput. Sci.* Springer-Verlag, 1992.