Rule Formats for Distributivity*

Luca Aceto[1], Matteo Cimini[1], Anna Ingolfsdottir[1],
MohammadReza Mousavi[2], and Michel A. Reniers[3]

[1] ICE-TCS, School of Computer Science, Reykjavik University,
Menntavegur 1, IS 101 Reykjavik, Iceland
[2] Department of Computer Science, Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands
[3] Department of Mechanical Engineering, Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

Abstract. This paper proposes rule formats for Structural Operational Semantics guaranteeing that certain binary operators are left distributive with respect to a set of binary operators. Examples of left-distributivity laws from the literature are shown to be instances of the provided formats. Some conditions ensuring the impossibility of the validity of the left-distributivity law are also offered.

1 Introduction

The syntax of a programming or specification language defines the collection of syntactically correct expressions, and its core is typically described formally using some variation on the notion of context-free grammar. The semantics of a language associates a 'meaning' to each syntactically correct expression.

Over the last three decades, Structural Operational Semantics (SOS), see, e.g., [9, 27, 30, 31], has proven to be a powerful way to specify the semantics of programming and specification languages. In this approach to semantics, languages can be given a clear behaviour in terms of states and transitions, where the collection of transitions is specified by means of a set of syntax-driven inference rules. This behavioural description of the semantics of a language essentially tells one how the expressions in the language under definition behave when run on an idealized abstract machine.

Designers of languages often have expected algebraic properties of language constructs in mind when defining a language. For example, one expects that a sequential composition operator be associative and, in the field of process algebra [12, 17, 22, 23], operators such as nondeterministic and parallel composition

* The work of Aceto, Cimini and Ingolfsdottir has been partially supported by the projects 'New Developments in Operational Semantics' (nr. 080039021) and 'Metatheory of Algebraic Process Theories' (nr. 100014021) of the Icelandic Research Fund. The work on the paper was partly carried out while Luca Aceto held an Abel Extraordinary Chair at Universidad Complutense de Madrid, Spain, supported by the NILS Mobility Project.

are often meant to be commutative and associative with respect to bisimilarity [29]. Once the semantics of a language has been given in terms of state transitions, a natural question to ask is whether the intended algebraic properties do hold modulo the notion of behavioural equivalence or preorder of interest. The typical approach to answer this question is to perform an *a posteriori verification*: based on the semantics in terms of state transitions, one proves the validity of the desired algebraic laws, which describe semantic properties of the various operators in the language. An alternative approach is to ensure the validity of algebraic properties *by design*, using the so called *SOS rule formats* [11]. In this approach, one gives *syntactic templates* for the inference rules used in defining the operational semantics for certain operators that guarantee the validity of the desired laws by design. Not surprisingly, the definition of rule formats is based on finding a reasonably good trade-off between generality and ease of application. On the one hand, one strives to define a rule format that can capture as many examples from the literature as possible, including ones that may arise in the future. On the other, the rule format should be as easy to apply as possible and, preferably, the syntactic constraints of the format should be algorithmically checkable.

The literature on SOS provides rule formats for basic algebraic properties of operators such as commutativity [26], associativity [19], idempotence [1] and the existence of unit and zero elements [4, 10]. The main advantage of this approach is that one is able to verify the desired property by syntactic checks that can be mechanized. Moreover, it is interesting to use rule formats for establishing semantic properties since the results so obtained apply to a broad class of languages. Apart from providing one with an insight as to the semantic nature of algebraic properties and its link to the syntax of SOS rules, rule formats like those presented in the above-mentioned references may serve as a guideline for language designers who want to ensure, a priori, that the constructs under design enjoy certain basic algebraic properties.

In the present paper, we develop two rule formats guaranteeing that certain binary operators are left distributive with respect to others modulo bisimilarity. A binary operator $\circledast$ is *left distributive* with respect to a binary operator $\oplus$, modulo some notion of behavioural equivalence, whenever the following equation holds

$$(x \oplus y) \circledast z = (x \circledast z) \oplus (y \circledast z).$$

A classic example of left-distributivity law within the realm of process algebra is

$$(x + y) \mathbin{\|\!\!\lfloor} z = (x \mathbin{\|\!\!\lfloor} z) + (y \mathbin{\|\!\!\lfloor} z),$$

where '$+$' and '$\|\!\!\lfloor$' stand for nondeterministic choice and left merge, respectively, from [12, 17, 23]. (The reader may find many other examples in the main body of this paper.) Distributivity laws like the aforementioned one play a crucial role in (ground-)complete axiomatizations of behavioural equivalences over fragments of process algebras (see, e.g., the above-mentioned references and [2, 6, 7]), and their lack of validity with respect to choice-like operators is often the key to the nonexistence of finite (in)equational axiomatizations of behavioural semantics—see, for instance, [5, 8, 24, 25].

The first rule format we present is the simplest of the two, but suffices to handle many examples from the literature. The second rule format has more complex syntactic conditions and can handle left-distributivity laws that are outside the scope of the former format. In both rule formats, for the sake of simplicity, the $\oplus$ operator 'behaves like' some form of nondeterministic choice operator. Both rule formats are based on syntactic conditions that are decidable over finite language specifications.

We provide a wealth of examples showing that the validity of several left-distributivity laws from the literature on process algebras can be proved using the two rule formats. Moreover, in Section 6 we argue that the two rule formats can be applied just as well to show left-distributivity laws involving *unary* operators.

We also offer some impossibility results concerning the validity of the left-distributivity law. Unlike previous results about rule formats for algebraic properties, these theorems allow one to recognize when the left-distributivity law is guaranteed *not* to hold. When designing operational specifications for operators that are intended to satisfy a left-distributivity law, a language designer might also benefit from considering these kinds of negative results. To our knowledge this type of result does not have any precursor in the field of rule formats. Hitherto, all rule formats aimed at providing sufficient conditions for establishing semantic properties, whereas the above-mentioned results are the first ones that offer *necessary syntactic conditions* for some semantic property to hold.

Roadmap of the paper The paper is organized as follows. Section 2 reviews some standard definitions from the theory of SOS that will be used in the remainder of this study. Section 3 presents our two rule formats guaranteeing that a binary operator $\circledast$ is left-distributive with respect to a binary operator $\oplus$ modulo bisimilarity. The first rule format and some examples of its application are presented in 3.2 In Section 3.3, we introduce the second rule format, which extends the first rule format and can treat more examples. In order to ease its application, we simplify the checks in the second rule format in Section 4 and summarize the simplifications in a tabular form. Examples that can be handled using the second rule format (even by using the simplified checks in Section 4) are offered in Section 5. We apply the two rule formats to show left-distributivity laws involving unary operators in Section 6. Some impossibility results concerning the validity of the left-distributivity law are offered in Section 7. We conclude the paper with a discussion of its contributions and of lines for future research in Section 8.

2 Preliminaries

In this section we recall some standard definitions from the theory of SOS. We refer the readers to, e.g., [9] and [27] for more information.

2.1 Transition system specifications and bisimilarity

Definition 1 (Signatures, terms and substitutions) *We let V denote an infinite set of variables and use $x, x', x_i, y, y', y_i, \ldots$ to range over elements of V. A* signature *Σ is a set of function symbols, each with a fixed arity. We call these symbols* operators *and usually represent them by $f, g, \ldots$. An operator with arity zero is called a* constant. *We define the set $\mathbb{T}(\Sigma)$ of* terms *over Σ as the smallest set satisfying the following constraints.*

- *A variable $x \in V$ is a term.*
- *If $f \in \Sigma$ has arity n and $t_1, \ldots, t_n$ are terms, then $f(t_1, \ldots, t_n)$ is a term.*

We use s, t, u, possibly subscripted and/or superscripted, to range over terms. We write $t_1 \equiv t_2$ if t_1 and t_2 are syntactically equal. The function $\mathrm{vars} : \mathbb{T}(\Sigma) \to 2^V$ *gives the set of variables appearing in a term. The set $\mathbb{C}(\Sigma) \subseteq \mathbb{T}(\Sigma)$ is the set of* closed terms, *i.e., terms that contain no variables. We use $p, q, p', p_i, \ldots$ to range over closed terms. A* substitution *σ is a function of type $V \to \mathbb{T}(\Sigma)$. We extend the domain of substitutions to terms homomorphically and write $\sigma(t)$ for the result of applying the substitution σ to the term t. If the range of a substitution is included in $\mathbb{C}(\Sigma)$, we say that it is a* closed substitution. *For a substitution σ, a sequence $x_1, \ldots, x_n$ of distinct variables and a sequence $t_1, \ldots, t_n$ of terms, we write*

$$\sigma[x_1 \mapsto t_1, \ldots, x_n \mapsto t_n]$$

for the substitution that maps each x_i to t_i, $1 \leq i \leq n$, and each variable $x \notin \{x_1, \ldots, x_n\}$ to $\sigma(x)$.

Definition 2 (Transition system specification) *A* transition system specification *(TSS) is a triple $(\Sigma, \mathcal{L}, D)$ where*

- *Σ is a signature.*
- *$\mathcal{L}$ is a set of* labels *(or* actions*) ranged over by a, b, l. If $l \in \mathcal{L}$ and $t, t' \in \mathbb{T}(\Sigma)$, we say that $t \xrightarrow{l} t'$ is a* positive transition formula *and $t \xrightarrow{l}\!\!\!\!\!/\;\;$ is a* negative transition formula. *Such formulae are called t-*testing. *A transition formula (or just formula), typically denoted by ϕ or ψ, is either a negative transition formula or a positive one.*
- *D is a set of* deduction rules, *i.e., tuples of the form (Φ, ϕ) where Φ is a set of formulae and ϕ is a positive formula. We call the formulae contained in Φ the* premises *of the rule and ϕ the* conclusion.

We write $\mathrm{vars}(\Phi)$ *to denote the set of variables appearing in a set of formulae Φ, and* $\mathrm{vars}(r)$ *to denote the set of variables appearing in a deduction rule r. A deduction rule is t-*testing, *or* tests *t, if one of its premises is t-testing. We say that a formula or a deduction rule is* closed *if all of its terms are closed. Substitutions are also extended to formulae and sets of formulae in the natural way. For a rule r and a substitution σ, the rule $\sigma(r)$ is called a substitution instance of r. A set of positive closed formulae is called a* transition relation.

We often refer to a positive transition formula $t \xrightarrow{l} t'$ as a *transition* with t being its *source*, l its *label*, and t' its *target*. A deduction rule (Φ, ϕ) is typically written as $\frac{\Phi}{\phi}$. For the sake of consistency with SOS specifications of specific operators in the literature, in examples we use $\frac{\phi_1 \ldots \phi_n}{\phi}$ in lieu of $\frac{\{\phi_1, \ldots, \phi_n\}}{\phi}$.

An *axiom* is a deduction rule with an empty set of premises. We write $\frac{}{\phi}$ for an axiom with ϕ as its conclusion, and often abbreviate this notation to ϕ when this causes no confusion.

Definition 3 *Given a rule d of the form*

$$\frac{\Phi}{f(t_1, \ldots, t_n) \xrightarrow{a} t},$$

we say that

- *d is f-*defining*, and write* $\mathrm{op}(d) = f$*,*
- *d is a-*emitting*,*
- $\mathrm{toc}(d) = t$*, the* t*arget* o*f the* c*onclusion of d, and*
- $\mathrm{hyps}(d) = \Phi$*, the set of premises of d.*

We also denote by $D(f, a)$ the set of a-emitting and f-defining rules in a set of deduction rules D.

Example 1 (Choice operators). The choice operator from [23] is defined by the following rules, where a ranges over the set of actions.

$$(chl_a)\ \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} \qquad (chr_a)\ \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$$

For each action a, the rules (chl_a) and (chr_a) are a-emitting and $+$-defining. For rule (chl_a), we have that $\mathrm{toc}(chl_a) = x'$ and $\mathrm{hyps}(chl_a) = \{x \xrightarrow{a} x'\}$.

The left choice operator $+_l$ is defined by the rules chl_a (there is one such rule for each action a). Symmetrically, the right choice operator $+_r$ is defined by the rules chr_a. (Again, there is one such rule for each action a.)

The meaning of a TSS is defined by the following notion of least three-valued stable model. To define this notion, we need two auxiliary definitions, namely provable transition rules and contradiction, which are given below.

Definition 4 (Provable transition rules) *A closed deduction rule is called a* transition rule *when it is of the form $\frac{N}{\phi}$ with N a set of* negative formulae*. A TSS $\mathcal{T}$* proves *$\frac{N}{\phi}$, denoted by $\mathcal{T} \vdash \frac{N}{\phi}$, when there is a well-founded upwardly branching tree with closed formulae as nodes and of which*

- *the root is labelled by ϕ;*
- *if a node is labelled by ψ and the labels of the nodes directly above it form the set K then:*

- *ψ is a negative formula and $\psi \in N$, or*
- *ψ is a positive formula and $\frac{K}{\psi}$ is a substitution instance of a deduction rule in $\mathcal{T}$.*

We often write $\mathcal{T} \vdash \phi$ in lieu of $\mathcal{T} \vdash \frac{\emptyset}{\phi}$.

Definition 5 (Contradiction and consistency) *The formula $t \xrightarrow{l} t'$ is said to* contradict *$t \stackrel{l}{\nrightarrow}$, and vice versa. For two sets Φ and Ψ of formulae, Φ* contradicts *Ψ when there is a $\phi \in \Phi$ that contradicts a $\psi \in \Psi$. We write $\Phi \vDash \Psi$, read 'Φ is consistent with Ψ', when Φ does not contradict Ψ.*

It immediately follows from the above definition that contradiction and consistency are symmetric relations on (sets of) formulae. We now have all the necessary ingredients to define the semantics of TSSs in terms of three-valued stable models [32].

Definition 6 (Three-valued stable model) *A pair (C, U) of disjoint sets of positive closed transition formulae is called a* three-valued stable model *for a TSS $\mathcal{T}$ when the following conditions hold:*

- *for each $\phi \in C$, there is a set N of negative formulae such that $\mathcal{T} \vdash \frac{N}{\phi}$ and $C \cup U \vDash N$, and*
- *for each $\phi \in U$, there is a set N of negative formulae such that $\mathcal{T} \vdash \frac{N}{\phi}$ and $C \vDash N$.*

C stands for Certainly *and U for* Unknown*; the third value is determined by the formulae not in $C \cup U$. The* least *three-valued stable model is a three-valued stable model that is the least one with respect to the (information-theoretic) ordering on pairs of sets of formulae defined as $(C, U) \leq (C', U')$ iff $C \subseteq C'$ and $U' \subseteq U$. We say that $\mathcal{T}$ is* complete *when for its least three-valued stable model it holds that $U = \emptyset$. In a complete TSS, we say that a closed substitution σ satisfies a set of formulae Φ if $\sigma(\phi) \in C$, for each positive formula $\phi \in \Phi$, and $C \vDash \{\sigma(\phi)\}$, for each negative formula $\phi \in \Phi$. If a TSS is complete, we often also write $p \xrightarrow{l} p'$ in lieu of $(p \xrightarrow{l} p') \in C$, and $p \stackrel{l}{\nrightarrow}$ when there is no p' such that $p \xrightarrow{l} p'$.*

In what follows, we shall tacitly restrict ourselves to considering only complete TSSs.

Definition 7 (Bisimulation and bisimilarity [23, 29]) *Let $\mathcal{T}$ be a transition system specification with signature Σ and label set $\mathcal{L}$. A relation $\mathcal{R} \subseteq \mathbb{C}(\Sigma) \times \mathbb{C}(\Sigma)$ is a* bisimulation relation *if and only if $\mathcal{R}$ is symmetric and, for all $p_0, p_1, p_0' \in \mathbb{C}(\Sigma)$ and $l \in \mathcal{L}$,*

$$(p_0 \,\mathcal{R}\, p_1 \wedge \mathcal{T} \vdash p_0 \xrightarrow{l} p_0') \Rightarrow \exists p_1' \in \mathbb{C}(\Sigma).\ (\mathcal{T} \vdash p_1 \xrightarrow{l} p_1' \wedge p_0' \,\mathcal{R}\, p_1').$$

Two terms $p_0, p_1 \in \mathbb{C}(\Sigma)$ are called bisimilar*, denoted by $p_0 \leftrightarrow p_1$, when there exists a bisimulation relation $\mathcal{R}$ such that $p_0 \,\mathcal{R}\, p_1$.*

Bisimilarity is extended to open terms by requiring that $s, t \in \mathbb{T}(\Sigma)$ are bisimilar when $\sigma(s) \leftrightarrow \sigma(t)$ for each closed substitution $\sigma : V \to \mathbb{C}(\Sigma)$.

3 The left-distributivity rule formats

In this section, we present two rule formats guaranteeing that a binary operator $\circledast$ is left-distributive with respect to a binary operator $\oplus$ modulo bisimilarity. The first rule format is the simplest of the two, but suffices to handle many examples from the literature. The second rule format has more complex syntactic conditions and can handle left-distributivity laws that are outside the scope of the former format.

Definition 8 (Left-distributivity law) *We say that a binary operator $\circledast$ is* left-distributive *with respect to a binary operator $\oplus$ (modulo bisimilarity) if the following equality holds:*

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z). \quad (1)$$

For all closed terms p, q, r, proving the algebraic law (1) involves two proof obligations:

- **Firability:** ensuring that $(p \oplus q) \circledast r \xrightarrow{a}$ if, and only if, $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a}$, for each action a;
- **Matching conclusions:** ensuring that, for each closed term p_1, if $(p \oplus q) \circledast r \xrightarrow{a} p_1$, then there exists some closed term p_2 such that
  - $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} p_2$ and
  - $p_1 \leftrightarrows p_2$,

  and vice versa.

Logically, the 'firability condition' is implied by the 'matching-conclusion condition'. However, since the two rule formats we shall present in what follows use the same idea to guarantee the former condition, and differ in how they guarantee the existence of matching conclusions up to bisimilarity, we prefer to consider the two conditions separately. To our mind, this also leads to a clearer presentation of the ideas underlying the rule formats. In what follows, we first explain how we achieve the 'firability condition', and then we discuss how the two different rule formats guarantee the 'matching-conclusion condition'.

3.1 The firability condition

We begin by introducing the conditions on sets of rules for two binary operators $\circledast$ and $\oplus$ that we shall use to guarantee the firability condition for them. First of all, we present syntactic constraints on the rules for those operators that we shall use throughout the remainder of the paper.

Definition 9 *We say that a deduction rule is of* the form (R1) *when it has the structure*

$$(R1)\ \frac{(\emptyset \text{ or } \{x \xrightarrow{a} x'\}) \ \cup \ \Phi_y}{x \circledast y \xrightarrow{a} t},$$

where

– *the variables x, x', y are pairwise distinct, and*
– *Φ_y is a (possibly empty) set of (positive or negative) y-testing formulae such that $x, x' \notin \mathrm{vars}(\Phi_y)$.*

The above notation should be read as a short-hand for two possible types of rules, viz.

$$\frac{\Phi_y}{x \circledast y \xrightarrow{a} t} \quad \textit{and} \quad \frac{\{x \xrightarrow{a} x'\} \ \cup \ \Phi_y}{x \circledast y \xrightarrow{a} t}.$$

A deduction rule is of the form (R2) *when it has the structure*

$$(R2)\ \frac{(\{x \xrightarrow{a} x'\}\ \textit{or}\ \{y \xrightarrow{a} y'\}\ \textit{or}\ \{x \xrightarrow{a} x', y \xrightarrow{a} y'\})}{x \oplus y \xrightarrow{a} t},$$

where the variables x, x', y, y' are pairwise distinct Again, the above notation should be read as a short-hand for three possible types of rules, viz.

$$\frac{\{x \xrightarrow{a} x'\}}{x \oplus y \xrightarrow{a} t} \qquad \frac{\{y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} t} \qquad \frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} t}.$$

A rule of the form $(R1)$ or $(R2)$ is non-left-inheriting *if $x \notin \mathrm{vars}(t)$, that is, if x does not appear in the target of the conclusion of the rule. An operation f specified by rules of the form $(R1)$ or $(R2)$ is* non-left-inheriting *if so are all of the f-defining rules.*

Definition 10 (Firability constraint) *Given a TSS T, let $\circledast$ and $\oplus$ be binary operators in the signature of T. For each action a, we write* $\mathrm{Fire}(\circledast, \oplus, a)$ *whenever the following conditions are met:*

– *if $D(\circledast, a) \neq \emptyset$ then $D(\oplus, a) \neq \emptyset$,*
– *each $d \in D(\circledast, a)$ is of the form $(R1)$, and*
– *each $d \in D(\oplus, a)$ is of the form $(R2)$.*

Remark 1. Note that the first constraint in the definition of $\mathrm{Fire}(\circledast, \oplus, a)$ is asymmetric, as it only requires that if there is a $\circledast$-defining a-emitting rule, then there should also be some $\circledast$-defining a-emitting rule. As will become clear from Examples 12–14, amongst others, this is necessary to obtain a widely applicable rule format for left distributivity.

Example 2. Recall the choice operators $+$, $+_l$ and $+_r$ presented in Example 1. As our readers can easily check, $\mathrm{Fire}(f, g, a)$ holds for each action a and for all $f, g \in \{+, +_l, +_r\}$.

The firability constraint in Definition 10 is sufficient to guarantee the aforementioned firability condition.

Theorem 1 (Firability Theorem). *Given a TSS T, let $\circledast$ and $\oplus$ be binary operators from the signature of T. Suppose that* $\mathrm{Fire}(\circledast, \oplus, a)$ *holds for some action a. Then,*

$$(p \oplus q) \circledast r \xrightarrow{a} \text{ if, and only if, } (p \circledast r) \oplus (q \circledast r) \xrightarrow{a},$$

for all closed terms p, q, r.

Proof. See Appendix A. □

The import of Theorem 1 is that, when proving the validity of (1), we can guarantee the firability condition for action a just by showing that $\mathrm{Fire}(\circledast, \oplus, a)$ holds. Theorem 1 underlies the soundness of both the rule formats we present in what follows.

The reader will have already noticed that the rule form ($R1$) does not place any restriction on tests for the variable y. This is possible because the second argument of the terms $(p \oplus q) \circledast r$, $p \circledast r$ and $q \circledast r$ is always the same, i.e. the term r. This means that, for each $\circledast$-defining rule, the same tests performed on the second argument on one side of (1) are performed on the other. Roughly speaking, one side of (1) may fire as much as the other does, insofar the second argument is concerned.

3.2 The matching-conclusion condition

Theorem 1 tells us that any rule format, whose constraints imply condition $\mathrm{Fire}(\circledast, \oplus, a)$ for each action a guarantees the validity of (1) provided that the matching-conclusion condition is met. Intuitively, in order to guarantee syntactically that the matching-conclusion condition is satisfied, the targets of the conclusions of $\circledast$-defining and $\oplus$-defining rules should 'match' when those operators are used in the specific contexts of the left- and the right-hand sides of (1). In what follows, we shall examine two different ways of ensuring the above-mentioned 'match' of the targets of the conclusions of $\circledast$-defining and $\oplus$-defining rules. The first relies on assuming that the targets of the conclusions of $\oplus$-defining rules are target variables of rules of the form ($R2$). The resulting rule format, which we present in Section 3.2, is based on easily checkable syntactic constraints and covers a large number of left-distributivity laws from the literature. However, there are some examples of left-distributivity axioms that cannot be shown using that format. In order to be able to deal with those examples, and others that might be presented in the literature in the future, in Section 3.3 we propose a more complex rule format in which the 'match' of the targets of the conclusions of $\circledast$-defining and $\oplus$-defining rules is performed by means of a powerful 'compliance relation'.

The first rule format The first rule format that we present deals with examples of left distributivity with respect to operators whose semantics is given by rules of the form ($R2$) that, like those for the choice operators we mentioned in Example 1, have target variables of premises as targets of their the conclusions. The following definition presents the syntactic constraints of the rule format.

Definition 11 (First rule format) *Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators in the signature of T. We say that the rules for $\circledast$ and $\oplus$ are in the* first rule format for left distributivity *if the following conditions are met:*

1. $\mathrm{Fire}(\circledast, \oplus, a)$ *holds for each action* a*,*
2. $\circledast$ *is non-left-inheriting,*
3. *each* $\oplus$*-defining rule has a target variable of one of its premises as target of its conclusion and*
4. *for each action* a*, either there is no* a*-emitting and* $\oplus$*-defining rule that tests both* x *and* y*, or if some* a*-emitting and* $\circledast$*-defining rule tests its left argument* x *then so do all* a*-emitting and* $\circledast$*-defining rules.*

Theorem 2 (Left distributivity over choice-like operators). *Let* T *be a TSS, and let* $\circledast$ *and* $\oplus$ *be binary operators in the signature of* T*. Assume that the rules for* $\circledast$ *and* $\oplus$ *are in the* first rule format for left distributivity*. Then*

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z).$$

Proof. We show the following two claims, where p, q, r, s are arbitrary closed terms and a is any action:

1. If $(p \oplus q) \circledast r \xrightarrow{a} s$ then $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$.
2. If $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$ then $(p \oplus q) \circledast r \xrightarrow{a} s$.

In the proof of the former claim, we use the first condition in Definition 10. This condition is not used in the proof of the latter claim. On the other hand, the proof of the latter statement uses condition 4 in Definition 11, which is not used in the proof of the former claim. The full proof may be found in Appendix B. □

Remark 2. Condition 4 in Definition 11 is necessary for the soundness of the rule format for left distributivity proved in the above theorem. To see this, consider the operations $\oplus$ and $\circledast$ with rules

$$\frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} x'} \qquad \frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \circledast y \xrightarrow{a} x' \circledast y} \qquad \frac{\{y \xrightarrow{a} y'\}}{x \circledast y \xrightarrow{a} y'}.$$

The above rules satisfy all the conditions in Definition 11 apart from 4. Now, let a be a constant with rule $a \xrightarrow{a} \mathbf{0}$, where $\mathbf{0}$ is a constant with no rules. As our readers can easily check,

$$(a \circledast a) \oplus (\mathbf{0} \circledast a) \not\leftrightarrows (a \oplus \mathbf{0}) \circledast a.$$

Indeed, the term $(a \circledast a) \oplus (\mathbf{0} \circledast a)$ can perform a sequence of two a-labelled transitions, whereas $(a \oplus \mathbf{0}) \circledast a$ cannot because $a \oplus \mathbf{0}$ affords no transitions.

Examples of application of the first rule format Theorem 2 provides us with a simple, yet rather powerful, syntactic condition in order to infer left-distributivity laws for operators like $+$ and $+_l$. Many of the common left-distributivity laws are automatically derived from Theorem 2, as witnessed by the examples we now proceed to discuss.

Example 3 (Left merge and interleaving parallel composition). The operational semantics of the classic left-merge and interleaving parallel composition operators [12, 16, 17, 23] is given by the rules below.

$$\frac{x \xrightarrow{a} x'}{x \mathbin{\|\!\!\lfloor} y \xrightarrow{a} x' \parallel y} \qquad \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'}$$

Note that the rules for the left-merge operator $\mathbin{\|\!\!\lfloor}$ and those for any of $+$, $+_l$ and $+_r$ satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following laws.

$$(x + y) \mathbin{\|\!\!\lfloor} z \leftrightarroweq (x \mathbin{\|\!\!\lfloor} z) + (y \mathbin{\|\!\!\lfloor} z)$$
$$(x +_l y) \mathbin{\|\!\!\lfloor} z \leftrightarroweq (x \mathbin{\|\!\!\lfloor} z) +_l (y \mathbin{\|\!\!\lfloor} z)$$
$$(x +_r y) \mathbin{\|\!\!\lfloor} z \leftrightarroweq (x \mathbin{\|\!\!\lfloor} z) +_r (y \mathbin{\|\!\!\lfloor} z)$$

Observe that the equalities

$$(x +_l y) \parallel z \leftrightarroweq (x \parallel z) +_l (y \parallel z) \text{ and}$$
$$(x +_r y) \parallel z \leftrightarroweq (x \parallel z) +_r (y \parallel z)$$

are sound. However, their soundness *cannot* be shown using Theorem 2, since the parallel composition operator $\parallel$ does not satisfy condition 2 in Definition 11. Indeed, x occurs in the target of the conclusion of the second rule for $\parallel$.

Example 4 (Synchronous parallel composition). Consider the synchronous parallel composition from CSP [22, 21][4] specified by the rules below, where a ranges over the set of actions:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{a} y'}{x \parallel_s y \xrightarrow{a} x' \parallel_s y'}.$$

Note that the rules for the synchronous parallel composition operator and those for any of $+$, $+_l$ and $+_r$ satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following laws.

$$(x + y) \parallel_s z \leftrightarroweq (x \parallel_s z) + (y \parallel_s z)$$
$$(x +_l y) \parallel_s z \leftrightarroweq (x \parallel_s z) +_l (y \parallel_s z)$$
$$(x +_r y) \parallel_s z \leftrightarroweq (x \parallel_s z) +_r (y \parallel_s z)$$

Example 5 (Join and '/' operators). Consider the join operator $\bowtie$ from [15] and the 'hourglass' operator / from [2] specified by the rules below, where a, b range over the set of actions:

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{a} y'}{x \bowtie y \xrightarrow{a} x' \mp y'} \qquad \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y'}{x/y \xrightarrow{a} x'/y'},$$

[4] In [22], Hoare uses the symbol $\parallel$ to denote the synchronous parallel composition operator. Here we use that symbol for parallel composition.

where $\mp$ denotes the delayed choice operator from [15]. (The operational specification of the delayed choice operator is immaterial for the analysis of this example.) The above rules and those for any of $+$, $+_l$ and $+_r$ satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following laws, where $\circledast \in \{\bowtie, /\}$.

$$(x + y) \circledast z \leftrightarrows (x \circledast z) + (y \circledast z)$$
$$(x +_l y) \circledast z \leftrightarrows (x \circledast z) +_l (y \circledast z)$$
$$(x +_r y) \circledast z \leftrightarrows (x \circledast z) +_r (y \circledast z)$$

Example 6 (Disrupt). Consider the following disrupt operator $\blacktriangleright$ [13, 18] with rules

$$\frac{x \xrightarrow{a} x'}{x \blacktriangleright y \xrightarrow{a} x' \blacktriangleright y} \qquad \frac{y \xrightarrow{a} y'}{x \blacktriangleright y \xrightarrow{a} y'} \ .$$

The above rules and those for any of $+$, $+_l$ and $+_r$ satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following laws.

$$(x + y) \blacktriangleright z \leftrightarrows (x \blacktriangleright z) + (y \blacktriangleright z)$$
$$(x +_l y) \blacktriangleright z \leftrightarrows (x \blacktriangleright z) +_l (y \blacktriangleright z)$$
$$(x +_r y) \blacktriangleright z \leftrightarrows (x \blacktriangleright z) +_r (y \blacktriangleright z)$$

Example 7 (Unless operator). The unless operator $\triangleleft$ from [14] and the operator Δ from [2, page 23] are specified by the rules

$$\frac{x \xrightarrow{a} x' \quad y \stackrel{b}{\nrightarrow} \text{ for } a < b}{x \triangleleft y \xrightarrow{a} x'} \qquad \frac{x \xrightarrow{a} x' \quad y \stackrel{b}{\nrightarrow} \text{ for } a < b}{x \, \Delta \, y \xrightarrow{a} \theta(x')},$$

where $<$ is an irreflexive partial order over the set of actions and θ denotes the priority operator from [14]. (The operational specification of the priority operator is immaterial for the analysis of this example.) The above rules and those for any of $+$, $+_l$ and $+_r$ satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following laws, where $\circledast \in \{\triangleleft, \Delta\}$.

$$(x + y) \circledast z \leftrightarrows (x \circledast z) + (y \circledast z)$$
$$(x +_l y) \circledast z \leftrightarrows (x \circledast z) +_l (y \circledast z)$$
$$(x +_r y) \circledast z \leftrightarrows (x \circledast z) +_r (y \circledast z)$$

Example 8 (Interplay between the choice operators). Consider the choice operators $+$, $+_l$ and $+_r$ from Example 1. The rules for any of the nine combinations of those operators satisfy the constraints of the first rule format for left distributivity. Therefore, Theorem 2 yields the validity of the following law, where $\oplus, \circledast \in \{+, +_l, +_r\}$.

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z)$$

For example, as an instance of that family of equalities, we obtain the following 'self left-distributivity law' for any $\oplus \in \{+, +_l, +_r\}$:

$$(x \oplus y) \oplus z \leftrightarrows (x \oplus z) \oplus (y \oplus z).$$

As we will see in Section 6, our first rule format for left distributivity can also be used to derive left-distributivity laws involving unary $\circledast$ operators.

3.3 The second left-distributivity format

As witnessed by the above-mentioned examples, the rule format introduced in Definition 11 can handle many of the common left-distributivity laws from the literature. However, as we mentioned in Example 3, that rule format is *not* general enough to prove the validity of, e.g., the left-distributivity law

$$(x +_l y) \parallel z \leftrightarrows (x \parallel z) +_l (y \parallel z).$$

It is instructive to see why the equality

$$(p +_l q) \parallel r \leftrightarrows (p \parallel r) +_l (q \parallel r)$$

holds for all p, q, r. The terms that can be reached from $(p +_l q) \parallel r$ via an a-labelled transition have one of the two following forms:

- $p' \parallel r$, for some p' such that $p \xrightarrow{a} p'$ or
- $(p +_l q) \parallel r'$, for some r' such that $r \xrightarrow{a} r'$.

On the other hand, the terms that can be reached from $(p \parallel r) +_l (q \parallel r)$ via an a-labelled transition are of the form

- $p' \parallel r$, for some p' such that $p \xrightarrow{a} p'$ or
- $p \parallel r'$, for some r' such that $r \xrightarrow{a} r'$.

The first of those possible forms is identical to the first form of a possible derivative of $(p +_l q) \parallel r$. However, the second form—viz. $p \parallel r'$, for some r' such that $r \xrightarrow{a} r'$—matches $(p +_l q) \parallel r'$ only up to one application of the equation

$$x +_l y = x,$$

which is sound modulo bisimilarity, from left to right. This rewriting can be performed in the context of $\parallel$ since the rules for the interleaving parallel composition operator given in Example 3 are in de Simone format [20], which is one of the congruence formats for bisimilarity—see, for instance, the survey articles [9, 27].

The above discussion motivates the development of a generalization of the rule format we presented in Definition 11. The main idea behind this more powerful rule format is to weaken the constraints for ensuring the 'matching-conclusion condition', so that terms that are targets of transitions from $(p \oplus q) \circledast r$ and

$(p \circledast r) \oplus (q \circledast r)$ need only be equal up to the application of some equation, whose validity modulo bisimilarity can be justified 'syntactically', in a context consisting of operations that preserve bisimilarity. Of course, the resulting definition of the rule format depends on the set of equations that one is allowed to use. Indeed, one can obtain more powerful rule formats by simply extending the collection of allowed equations. Therefore, what we now present can be seen as a template for rule formats guaranteeing the validity of left-distributivity equations of the form (1). Our definition of the second rule format is based on a rewriting relation over terms that is sufficient to handle the examples from the literature we have met so far. The rewriting relation we present below can, however, be easily strengthened by adding more rewritings, provided their soundness with respect to bisimilarity can be 'justified syntactically'. (See the paragraphs after Definition 12 and Remark 4 for a brief discussion of extensions of the proposed rule format.)

Definition 12 (The rewriting relation $\rightsquigarrow$) *Let $T = (\Sigma, \mathcal{L}, D)$ be a TSS.*

1. *The relation $\rightsquigarrow$ is the least binary relation over $\mathbb{T}(\Sigma)$ that satisfies the following clauses, where we use $t \leftrightsquigarrow t'$ as a short-hand for $t \rightsquigarrow t'$ and $t' \rightsquigarrow t$:*
   - $t \rightsquigarrow t$,
   - $f(t,t) \leftrightsquigarrow t$, *if T is in idempotence format with respect to f from [1],*
   - $C[t] \rightsquigarrow C[t']$, *if $t \rightsquigarrow t'$ and T is in a congruence format for $\leftrightarrow$,*
   - $t_1 +_l t_2 \rightsquigarrow t_1$, *if $+_l \in \Sigma$,*
   - $t_1 +_r t_2 \rightsquigarrow t_2$, *if $+_r \in \Sigma$, and*
2. *Let $\circledast$ and $\oplus$ be two binary operations in Σ. We write $t \downarrow_{\circledast,\oplus} u$ if, and only, if there are some t' and u' such that $t \rightsquigarrow t'$, $u \rightsquigarrow u'$, and $t' = u'$ can be proved by possibly using one application of axiom*

$$(x \oplus y) \circledast z = (x \circledast z) \oplus (y \circledast z)$$

   at the top level—that is, either $t' \equiv u'$ or $t' \equiv (t_1 \oplus t_2) \circledast t_3$ and $u' = (t_1 \circledast t_3) \oplus (t_2 \circledast t_3)$, for some t_1, t_2, t_3.

Lemma 1. *Let $T = (\Sigma, \mathcal{L}, D)$ be a TSS. If $t \rightsquigarrow t'$ then $t \leftrightarrow t'$, for all $t, t' \in \mathbb{T}(\Sigma)$.*

Proof. By induction on the definition of $\rightsquigarrow$. The soundness of the rewrite rules

- $f(t,t) \leftrightsquigarrow t$, if T is in idempotence format with respect to f from [1], and
- $C[t] \rightsquigarrow C[t']$, if $t \rightsquigarrow t'$ and T is in a congruence format for $\leftrightarrow$,

is guaranteed by results in [1] and in the classic theory of structural operational semantics. □

In order to check whether a rewriting rule preserves bisimilarity, in all cases apart from the the first, the above definition relies on existing rule formats guaranteeing the validity of algebraic laws modulo bisimilarity, see [11], or on equations whose soundness with respect to bisimilarity is easy to check, such as

$$x +_l y = x \qquad \text{and} \qquad x +_r y = y.$$

This choice allows us to achieve an expressive and extensible rule format while retaining its syntactic nature. For instance, one may easily extend the rewriting relation $\rightsquigarrow$ with the following two clauses:

– $f(t_1, t_2) \leftrightsquigarrow f(t_2, t_1)$, if T is in the commutativity rule format with respect to f from [26], and
– $f(t, f(t', t'')) \leftrightsquigarrow f(f(t, t'), t'')$, if T is in the associativity rule format with respect to f from [19].

While proving the soundness of a left-distributivity law of the form

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z),$$

the validity of equivalences of the form

$$(t \oplus t') \circledast t'' = (t \circledast t'') \oplus (t' \circledast t'')$$

will be guaranteed by coinduction.

In Definition 13 to follow, which is the key ingredient in the definition of our second rule format for left distributivity, we shall use the relation $\downarrow_{\circledast,\oplus}$ to describe when a $\circledast$-defining rule d_1 is 'distributivity compliant' to a $\oplus$-defining rule d_2. The intuitive idea is that this will hold when those two rules can be combined to derive transitions from terms of the form $(p \oplus q) \circledast r$ and $(p \circledast r) \oplus (q \circledast r)$ that 'match' up to bisimilarity. Since the definition of distributivity compliance is quite technical, we find it useful to explain, by means of examples, the intuition behind it. For the sake of consistency and clarity, in the examples to follow, we shall use the same naming convention for substitutions that will be employed in Definition 13.

Suppose that the transition $(p \oplus q) \circledast r \xrightarrow{a} s$ is proved using rule d_1 and rule d_2. Assume, furthermore, that

$$(d_1)\ \frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y_1, y \xrightarrow{b} y_2\}}{x \circledast y \xrightarrow{a} t}$$

and that d_2 tests only one of its arguments, say

$$(d_2)\ \frac{\{x \xrightarrow{a} x'\}}{x \oplus y \xrightarrow{a} t'}.$$

Then $s = \sigma_1(t)$, where

$$\sigma_1 = [x \mapsto p \oplus q, y \mapsto r, x' \mapsto \sigma_2'(t'), y_1 \mapsto r_1, y_2 \mapsto r_2]$$
$$\sigma_2' = [x \mapsto p, y \mapsto q, x' \mapsto p']$$

and $p \xrightarrow{a} p'$, $r \xrightarrow{a} r_1$ and $r \xrightarrow{b} r_2$.

As highlighted by the proof of Theorem 1, rules d_2 and d_1 can be used to derive a transition $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} \sigma_2(t')$, where

$$\sigma_2 = [x \mapsto p \circledast r, y \mapsto q \circledast r, x' \mapsto \sigma_{1x}(t)]$$
$$\sigma_{1x} = [x \mapsto p, y \mapsto r, x' \mapsto p', y_1 \mapsto r_1, y_2 \mapsto r_2].$$

The transition $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} \sigma_2(t')$ will be deemed to 'match' $(p \oplus q) \circledast r \xrightarrow{a} s = \sigma_1(t)$ provided that

$$\sigma_1(t) \downarrow_{\circledast,\oplus} \sigma_2(t').$$

This will give a syntactically checkable guarantee that $\sigma_1(t) \leftrightarroweq \sigma_2(t')$ holds.

Assume now that d_2 tests both its arguments, say

$$(d_2)\ \frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} t'},$$

and that the transition $(p \oplus q) \circledast r \xrightarrow{a} s$ is proved using rule d_1 and rule d_2. Then $s = \sigma_1(t)$, where

$$\begin{aligned}\sigma_1 &= [x \mapsto p \oplus q, y \mapsto r, x' \mapsto \sigma_2'(t'), y_1 \mapsto r_1, y_2 \mapsto r_2]\\ \sigma_2' &= [x \mapsto p, y \mapsto q, x' \mapsto p', y' \mapsto q']\end{aligned}$$

and $p \xrightarrow{a} p'$, $q \xrightarrow{a} q'$, $r \xrightarrow{a} r_1$ and $r \xrightarrow{b} r_2$.

Let

$$(d_3)\ \frac{\{x \xrightarrow{a} x', y \stackrel{a}{\nrightarrow}, y \xrightarrow{c} y'\}}{x \circledast y \xrightarrow{a} t''}.$$

Again, as highlighted by the proof of Theorem 1, rules d_2, d_1 and d_3 can be used to derive a transition $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} \sigma_{2x}(t')$, where

$$\begin{aligned}\sigma_{2x} &= [x \mapsto p \circledast r, y \mapsto q \circledast r, x' \mapsto \sigma_{1x}(t), y' \mapsto \sigma_{1y}'(t'')]\\ \sigma_{1y}' &= [x \mapsto q, y \mapsto r, x' \mapsto q', y' \mapsto r'],\end{aligned}$$

and $p \circledast r \xrightarrow{a} \sigma_{1x}(t)$, $q \circledast r \xrightarrow{a} \sigma_{1y}'(t'')$, $q \xrightarrow{a} q'$ and $r \xrightarrow{c} r'$.

The transition $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} \sigma_{2x}(t')$ will be deemed to 'match' $(p \oplus q) \circledast r \xrightarrow{a} s = \sigma_1(t)$ provided that

$$\sigma_1(t) \downarrow_{\circledast,\oplus} \sigma_{2x}(t').$$

Again, this will give a syntactically checkable guarantee that $\sigma_1(t) \leftrightarroweq \sigma_{2x}(t')$ holds. Note that, in this case, we also need to check this matching condition when the roles of rules d_1 and d_3 are swapped, since rule d_3 might be used to satisfy the x-testing premise of d_2 and rule d_1 might be used to satisfy the y-testing premise of that rule. In that case, our proof obligation is to show that

$$\sigma_1(t) \downarrow_{\circledast,\oplus} \sigma_{2y}(t'),$$

where

$$\begin{aligned}\sigma_{2y} &= [x \mapsto p \circledast r, y \mapsto q \circledast r, x' \mapsto \sigma_{1x}'(t''), y' \mapsto \sigma_{1y}(t)]\\ \sigma_{1x}' &= [x \mapsto p, y \mapsto r, x' \mapsto p', y' \mapsto r']\\ \sigma_{1y} &= [x \mapsto q, y \mapsto r, x' \mapsto q', y_1 \mapsto r_1, y_2 \mapsto r_2].\end{aligned}$$

Definition 13 (Distributivity compliance up to $\rightsquigarrow$) *Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators in the signature of T. Let d_1 be a $\circledast$-defining rule in T and d_2 be a $\oplus$-defining rule in T. We say that d_1 is* distributivity compliant *to d_2 up to $\rightsquigarrow$, and we write it $d_1 \stackrel{\rightsquigarrow}{\sim} d_2$, whenever*

1. *rule d_1 is of the form (R1) and rule d_2 is of the form (R2),*
2. *the collection of positive y-testing premises in d_1 is of the form $\{y \xrightarrow{a_i} y_i \mid i \in I\}$, for some index set I, where all the variables are pairwise distinct, and*
3. *one of the following two cases applies:*
   (a) *d_2 has premises $\{x \xrightarrow{a} x'\}$ or $\{y \xrightarrow{a} y'\}$, and*

$$\sigma_1(\mathrm{toc}(d_1)) \downarrow_{\circledast,\oplus} \sigma_2(\mathrm{toc}(d_2)),$$

   or

   (b) *d_2 has premises $\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}$ and, for each rule $d_3 \in D(\circledast, a)$,*
      – *the collection of positive y-testing premises in d_3 is of the form $\{y \xrightarrow{a_j} y_j \mid j \in J\}$, for some index set J, where all the variables are pairwise distinct,*
      – $\sigma_1(\mathrm{toc}(d_1)) \downarrow_{\circledast,\oplus} \sigma_{2x}(\mathrm{toc}(d_2))$ *and*
      – $\sigma_1(\mathrm{toc}(d_1)) \downarrow_{\circledast,\oplus} \sigma_{2y}(\mathrm{toc}(d_2))$*,*

   where the substitutions σ_1, σ_{1x}, σ_{1y}, σ_2, σ_{2x} and σ_{2y} are defined as follows, with p, q, p', q', r, r', and all the variables in $\{r_i \mid i \in I\} \cup \{r_j \mid j \in J\}$ being fresh and pairwise distinct variables.

   – $\sigma_1 = [x \mapsto p \oplus q,\ y \mapsto r,\ x' \mapsto \sigma_2'(\mathrm{toc}(d_2)),\ y_i \mapsto r_i\ (i \in I)]$.
   – $\sigma_2 = [x \mapsto p \circledast r,\ y \mapsto q \circledast r,\ x' \mapsto \sigma_{1x}(\mathrm{toc}(d_1)),\ y' \mapsto \sigma_{1y}(\mathrm{toc}(d_1))]$.
   – $\sigma_2' = [x \mapsto p,\ y \mapsto q,\ x' \mapsto p',\ y' \mapsto q']$.
   – $\sigma_{1x} = [x \mapsto p,\ y \mapsto r,\ x' \mapsto p',\ y_i \mapsto r_i\ (i \in I)]$.
   – $\sigma_{1x}' = [x \mapsto p,\ y \mapsto r,\ x' \mapsto p',\ y_j \mapsto r_j\ (j \in J)]$.
   – $\sigma_{1y} = [x \mapsto q,\ y \mapsto r,\ x' \mapsto q',\ y_i \mapsto r_i\ (i \in I)]$.
   – $\sigma_{1y}' = [x \mapsto q,\ y \mapsto r,\ x' \mapsto q',\ y_j \mapsto r_j\ (j \in J)]$.
   – $\sigma_{2x} = [x \mapsto p \circledast r,\ y \mapsto q \circledast r,\ x' \mapsto \sigma_{1x}(\mathrm{toc}(d_1)),\ y' \mapsto \sigma_{1y}'(\mathrm{toc}(d_3))]$.
   – $\sigma_{2y} = [x \mapsto p \circledast r,\ y \mapsto q \circledast r,\ x' \mapsto \sigma_{1x}'(\mathrm{toc}(d_3)),\ y' \mapsto \sigma_{1y}(\mathrm{toc}(d_1))]$.

The reader should notice that, in order not to complicate the definition further by a more refined case distinction, in condition 3a of Definition 13, the substitution σ_2 is defined for both x' and y', even if in that case only one of them appears in rule d_2.

The following result is straightforward.

Theorem 3 (Decidability of $\stackrel{\rightsquigarrow}{\sim}$). *Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators in the signature of T. Let d_1 be a $\circledast$-defining rule in T and d_2 be a $\oplus$-defining rule in T. The problem of determining whether $d_1 \stackrel{\rightsquigarrow}{\sim} d_2$ holds is decidable.*

Remark 3. Note that $\stackrel{\rightsquigarrow}{\sim}$ performs only one rewriting step on both the terms. Clearly, extending Definition 13 in order to consider any finite amount of rewriting steps would not jeopardize Theorem 3.

We now have all the necessary ingredients to define our second rule format for left distributivity.

Definition 14 (Second left-distributivity format) *A TSS T is in the* second left-distributivity format *for a binary operator $\circledast$ with respect to a binary operator $\oplus$ whenever, for each action a,*

1. $\mathrm{Fire}(\circledast, \oplus, a)$*, and*
2. *if $D(\circledast, a) \neq \emptyset$ then $d_1 \tilde{\rightsquigarrow} d_2$, for each $d_1 \in D(\circledast, a)$ and for each $d_2 \in D(\oplus, a)$.*

We are now ready to formulate the two main theorems of the paper.

Theorem 4 (Soundness of the second left-distributivity format). *Let T be a TSS. If T is in the second left-distributivity format for $\circledast$ with respect to $\oplus$ then*

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z).$$

Proof. A proof of this result may be found in Appendix C. □

Remark 4. The above theorem holds true for any notion of distributivity compliance up to rewriting that is based on a rewriting relation $\rightsquigarrow$ over terms that has the following properties:

- $\rightsquigarrow \subseteq \leftrightarrows$ and
- $\rightsquigarrow$ is decidable.

The latter requirement is not necessary for the soundness of the format. However, it is highly desirable from the point of view of applications. Indeed, in order to obtain a *bona fide* rule format, the relation $\rightsquigarrow$ should be defined by using rules whose applicability can be checked syntactically, for instance using extant rule format for operational semantics. The proposal we presented in Definition 12 fits this requirement.

The following result is straightforward, but important from the point of view of applications. In its statement, we use $Range(f)$ to stand for the set of actions a for which there exists an a-emitting f-defining rule.

Theorem 5 (Decidability of the second rule format). *Let T be a TSS, and let $\circledast$ and $\oplus$ be two binary operators from the signature of T. Assume that $Range(\circledast)$ is finite, and that $D(\circledast, a) \cup D(\oplus, a)$ is finite for each $a \in Range(\circledast)$. Then it is decidable whether T is in the second left-distributivity format for $\circledast$ with respect to $\oplus$.*

The import of Theorems 4 and 5 is that, when establishing that an operator $\circledast$ is left distributive with respect to an operator $\oplus$, it is sufficient to check whether the SOS specification for those operators meets the conditions of the format of Definition 14, which can be done effectively when the TSS under study is finite.

4 Analyzing the distributivity compliance

In this section, we reduce the analysis of distributive compliance $\overset{\rightsquigarrow}{\sim}$ to a syntactic check on the targets of the conclusions of the $\circledast$- and $\oplus$-defining rules. By analyzing different possible syntactic shapes for terms, we check which pairs of shapes can be related using the distributivity-compliance relation. This analysis is useful in order to avoid many of the substitutions involved in Definition 13, and, as witnessed by some of the examples in Section 5, to avoid all of them in many cases.

Table 1 summarizes our results. Even though the offered list is not exhaustive, which, at first sight, seems a challenging task to achieve, we believe Table 1 offers enough cases to avoid substitutions completely in most cases.

Table 1. Analysis of the distributivity-compliance pairs

	toc(d_1)	toc(d_2)	result	further requirements
1	$x' \circledast y$	x	$p \circledast r$	
2	$x' \circledast y$	y	$q \circledast r$	
3	x	$x' \oplus y'$	$p \oplus q$	$D(\circledast, a) = \{d_1\}$
4	x'	$x' \oplus y'$	$p' \oplus q'$	$D(\circledast, a) = \{d_1\}$
5	$x \circledast t$	$x' \oplus y'$	$(p \oplus q) \circledast \sigma(t)$	$D(\circledast, a) = \{d_1\}$, $x, x' \notin \text{vars}(t)$
6	$x' \circledast t$	$x' \oplus y'$	$(p' \oplus q') \circledast \sigma(t)$	$D(\circledast, a) = \{d_1\}$, $x, x' \notin \text{vars}(t)$
7	t	$x' \oplus y'$	$\sigma(t)$	$\oplus$ idempotent, $D(\circledast, a) = \{d_1\}$, $x, x' \notin \text{vars}(t)$
8	t	x'	$\sigma'(t)$	Condition 4 of Definition 11, $x \notin \text{vars}(t)$
9	t	y'	$\sigma'(t)$	Condition 4 of Definition 11, $x \notin \text{vars}(t)$

with $\sigma = [y \mapsto r,\, y_i \mapsto r_i \ (i \in I)]$ and $\sigma' = [y \mapsto r,\, x' \mapsto p',\, y_i \mapsto r_i \ (i \in I)]$

In Table 1, x and y are considered as the variables for the first and second argument, respectively, for both $\circledast$- and $\oplus$-defining rules. When the variable x' is mentioned, implicitly the considered rule has a premise $x \xrightarrow{a} x'$ (for a-emitting rules). Similarly, when the variable y' is mentioned, implicitly the rule considered has a premise $y \xrightarrow{a} y'$. The term t stands for a generic open term from the signature, and, following Definition 13, p, q and r are hypothetical closed terms applied to the distributivity equation in this way: $(p \oplus q) \circledast r \leftrightarrows (p \circledast r) \oplus (q \circledast r)$. The symbols p', q', and r_i, are considered as targets of possible transitions possible transitions from p, q and r.

Table 1 is to be read as follows. First of all, $d_1 \in D(\circledast, a)$ and $d_2 \in D(\oplus, a)$, for some action a. In each row, the first column (column toc(d_1)) specifies the form of the target of the conclusion of the $\circledast$-defining rule d_1 (e.g., x in case of row 3), and the second column (column toc(d_2)) specifies the form of the target of the conclusion of the $\oplus$-defining rule d_2 (e.g., $x' \oplus y'$ in case of row 3). If the conditions in the column *further requirements* are satisfied (e.g., in row 3, d_1 is the only $\circledast$-defining and a-emitting rule), then the result of the transition of terms $(p \oplus q) \circledast r$ and $(p \circledast r) \oplus (q \circledast r)$ is specified by the term given in column *result* (e.g., $p \oplus q$ in row 3). In rows 5–6, the stated result is up to one application of the left-distributivity equation (1). The requirement $\oplus$ *idempotent* means that the operator $\oplus$ can be proved idempotent, e.g., by means of the rule format offered in [1].

The reader may want to notice that the first rule format of Section 3.2 is partly based on the analysis which leads to rows 8 and 9.

Theorem 6 (Soundness of Table 1). *Let T be a TSS. Let $\circledast$ and $\oplus$ be binary operations in the signature of T satisfying*

1. $\mathrm{Fire}(\circledast, \oplus, a)$*, and*
2. *if $D(\circledast, a) \neq \emptyset$ then for each $d_1 \in D(\circledast, a)$ and for each $d_2 \in D(\oplus, a)$, the rules d_1 and d_2 match a row in Table 1.*

It holds that:

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z).$$

Proof. The proof of the theorem goes by a straightforward check of the conditions of Definition 13 on the combination specified in each row. For example, we discuss the case of row 7 in some detail below.

Applying the substitutions, we can see that on the left side of the distributivity equation $(p \oplus q) \circledast r \leftrightarrows (p \circledast r) \oplus (q \circledast r)$, we can prove the transition $(p \oplus q) \circledast r \xrightarrow{a} v$, with $v = t[x \mapsto p \oplus q,\ y \mapsto r,\ x' \mapsto (x' \oplus y')[x \mapsto p,\ y \mapsto q,\ x' \mapsto p',\ y' \mapsto q'],\ y_i \mapsto r_i\ (i \in I)]$, and thus

$$v = t[x \mapsto p \oplus q, y \mapsto r, x' \mapsto p' \oplus q', y_i \mapsto r_i\ (i \in I)].$$

On the right side of the distributivity equation, we can prove the transition $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} v'$, with $v' = (x' \oplus y')[x \mapsto p \circledast r,\ y \mapsto q \circledast r,\ x' \mapsto t[x \mapsto p,\ y \mapsto r,\ x' \mapsto p',\ y_i \mapsto r_i\ (i \in I)]),\ y' \mapsto t[x \mapsto q,\ y \mapsto r,\ x' \mapsto q',\ y_i \mapsto r_i\ (i \in I)]$, and thus $v' = v'_1 \oplus v'_2$, where

$$v'_1 = t[x \mapsto p, y \mapsto r, x' \mapsto p', y_i \mapsto r_i\ (i \in I)] \qquad \text{and}$$
$$v'_2 = t[x \mapsto q, y \mapsto r, x' \mapsto q', y_i \mapsto r_i\ (i \in I)].$$

From the column *further requirements* of row 7, we know that the variables x and x' do not appear in t, leading the two terms to be $v = t[y \mapsto r, y_i \mapsto r_i\ (i \in I)]$ and $v' = v \oplus v$. Since, as a further requirement, the operator $\oplus$ is idempotent with respect to bisimilarity, i.e., $x \oplus x \leftrightarrows x$, we can conclude that

$$v' \downarrow_{\circledast,\oplus} v = t[y \mapsto r, y_i \mapsto r_i\ (i \in I)],$$

where $t[y \mapsto r, y_i \mapsto r_i\ (i \in I)]$ is the term stated in the column *result* of row 7. ◻

5 Examples

In what follows, we apply the rule format provided in Section 3.3 in order to check some examples of left-distributivity laws whose validity cannot be inferred using Theorem 2.

Example 9 (Interleaving parallel composition and left choice). As we remarked in Example 3, the equality

$$(x +_l y) \parallel z \leftrightarrows (x \parallel z) +_l (y \parallel z)$$

is sound. However, its soundness cannot be shown using Theorem 2, since the parallel composition operator $\parallel$ does not satisfy condition 2 in Definition 11. Indeed, x occurs in the target of the conclusion of the second rule for $\parallel$.

On the other hand, the validity of the above law can be shown by applying the rule format from Definition 14. Indeed, we observe that

- the targets of the conclusions of the pair of rules

$$(par_0)\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad (lc_0)\ \frac{x \xrightarrow{a} x'}{x +_l y \xrightarrow{a} x'},$$

  when instantiated as required in Definition 13, both become $p' \parallel r$, and
- the targets of the conclusions of the pair of rules

$$(par_1)\ \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} \qquad (lc_1)\ \frac{x \xrightarrow{a} x'}{x +_l y \xrightarrow{a} x'},$$

  when instantiated as required in Definition 13, become$(p +_l q) \parallel r'$ and $p \parallel r'$, with $(p +_l q) \parallel r' \rightsquigarrow p \parallel r'$.

Example 10 (Unit-delay operator and the choice operator from ATP). Consider any TSS T containing the unit-delay operator $\lfloor\ \rfloor$ and the choice operator $+^*$ from ATP [28][5] and for which the transition relation $\xrightarrow{\chi}$ is deterministic. (The distinguished symbol χ denotes the passage of one unit of time.) The semantics of those operators is defined by the following rules, where $a \neq \chi$.

$$(ud_a)\ \frac{x \xrightarrow{a} x'}{\lfloor x \rfloor(y) \xrightarrow{a} x'} \qquad (ud_\chi)\ \frac{}{\lfloor x \rfloor(y) \xrightarrow{\chi} y}$$

$$(extChl_a)\ \frac{x \xrightarrow{a} x'}{x +^* y \xrightarrow{a} x'} \qquad (extChr_a)\ \frac{y \xrightarrow{a} y'}{x +^* y \xrightarrow{a} y'}$$

[5] In [28], the symbol of this operator is $\oplus$, whose use we prefer to avoid in this paper for the sake of clarity.

$$(extTime)\ \frac{x \xrightarrow{\chi} x' \quad y \xrightarrow{\chi} y'}{x +^* y \xrightarrow{\chi} x' +^* y'}$$

We claim that T is in the second left-distributivity format for $\lfloor\ \rfloor$ with respect to $+^*$. Indeed, we observe that

- the targets of the conclusions of the pair of rules $(ud_a, extChl_a)$ when instantiated as required in Definition 13, both become p',
- the targets of the conclusions of the pair of rules $(ud_a, extChr_a)$ when instantiated as required in Definition 13, both become q', and
- the targets of the conclusions of the pair of rules $(ud_\chi, extTime)$ when instantiated as required in Definition 13, become r and $r +^* r$, with $r +^* r \rightsquigarrow r$ because T is in idempotence format with respect to $+^*$, as argued in [1, Example 9].

The well-known law

$$\lfloor x +^* y \rfloor(z) \Leftrightarrow \lfloor x \rfloor(z) +^* \lfloor y \rfloor(z)$$

thus follows from Theorem 4.

Table 1 can be used to match the targets of the conclusions as follows: the combination of ud_a and $extChl_a$ follows from row 8, the combination of ud_a and $extChr_a$ follows from row 9, and finally the combination of ud_χ and $extTime$ follows from row 7.

Example 11 (Timed left merge and the choice operator from ATP). Consider the TSS for ATP with the timed extension of the left-merge operator from Example 3 specified by the following rules, where $a \neq \chi$:

$$(merge_a)\frac{x \xrightarrow{a} x'}{x \mathbin{\lfloor\!\lfloor} y \xrightarrow{a} x' \parallel y} \qquad (merge_\chi)\frac{x \xrightarrow{\chi} x' \quad y \xrightarrow{\chi} y'}{x \mathbin{\lfloor\!\lfloor} y \xrightarrow{\chi} x' \mathbin{\lfloor\!\lfloor} y'}.$$

We claim that this TSS is in the second left-distributivity format for $\lfloor\!\lfloor$ with respect to $+^*$. We limit ourselves to checking that the targets of the conclusions of the second rule for $\lfloor\!\lfloor$ and rule $extTime$ match when instantiated as required in Definition 13. This follows because, in all cases, the resulting terms yield an instance of the equality

$$(p' +^* q') \mathbin{\lfloor\!\lfloor} r' = (p' \mathbin{\lfloor\!\lfloor} r') +^* (q' \mathbin{\lfloor\!\lfloor} r').$$

The law

$$(x +^* y) \mathbin{\lfloor\!\lfloor} z = (x \mathbin{\lfloor\!\lfloor} z) +^* (y \mathbin{\lfloor\!\lfloor} z)$$

thus follows from Theorem 4.

Checking the conditions of the second rule format can be simplified by using the syntactic checks of Table 1, as follows: the combination $merge_a$, $extChl_a$ follows from row 8, the combination $merge_a$, $extChr_a$ follows from row 9 and the combination $merge_\chi$, $extTime$ follows from row 6.

6 Examples of left-distributivity laws involving unary operators

In this section we apply the rule formats from Section 3 in order to prove left-distributivity laws involving unary operators from the literature. In order to do so, we turn unary operators into binary operators that simply ignore their right argument.

We begin with three examples that can be dealt with using Theorem 2.

Example 12 (Encapsulation and choice). Consider the classic unary encapsulation operators ∂_H from ACP [12], where $H \subseteq \mathcal{L}$, with rules

$$\frac{x \xrightarrow{a} x'}{\partial_H(x) \xrightarrow{a} \partial_H(x')} \quad a \notin H.$$

It is well known that

$$\partial_H(x + y) \leftrightarrows \partial_H(x) + \partial_H(y), \tag{2}$$

where $+$ is the choice operator from Example 1.

We shall now argue that the validity of this equation can be shown using Theorem 2. To this end, we turn the encapsulation operators into binary operators that ignore their second argument. The above rules therefore become

$$\frac{x \xrightarrow{a} x'}{\partial_H(x, y) \xrightarrow{a} \partial_H(x', y)} \quad a \notin H.$$

Note that the rules for ∂_H and $+$ are in the first rule format for left distributivity from Definition 11. In particular, $\mathrm{Fire}(\partial_H, +, a)$ holds for each action a, because if there is an a-emitting rule for ∂_H then there is also an a-emitting rule for $+$. (Note that the converse only holds if $H = \emptyset$. This explains the asymmetric nature of the constraint $\mathrm{Fire}(\circledast, \oplus, a)$.) Therefore Theorem 2 yields the validity of the left-distributivity law

$$\partial_H(x + y, z) \leftrightarrows \partial_H(x, z) + \partial_H(y, z),$$

from which the soundness of (2) follows immediately.

Example 13 (Match operator and choice). Consider the unary match operators $[a = b]$ from the π-calculus [33][6], where $a, b \in \mathcal{L}$, with rules

$$\frac{x \xrightarrow{c} x'}{[a = b](x) \xrightarrow{c} x'} \quad \text{if } a = b,$$

where $c \in \mathcal{L}$.

[6] Note that in the π-calculus a and b in the formula $[a = b]p$ are names and *not* labels.

It is well known that

$$[a = b](x + y) \leftrightarrows [a = b](x) + [a = b](y), \tag{3}$$

where $+$ is the choice operator from Example 1.

We shall now argue that the validity of this equation can be shown using Theorem 2. To this end, as above, we turn the match operators into binary operators that ignore their second argument. The above rules therefore become

$$\frac{x \xrightarrow{c} x'}{[a = b](x, y) \xrightarrow{c} x'} \quad \text{if } a = b.$$

Note that the rules for $[a = b]$ and $+$ are in the first rule format for left distributivity from Definition 11. Therefore Theorem 2 yields the validity of the left-distributivity law

$$[a = b](x + y, z) \leftrightarrows [a = b](x, z) + [a = b](y, z),$$

from which the soundness of (3) follows immediately.

Example 14 (Projection operator and choice). Consider the unary projection operators π_n from ACP [12, 16], where $n \geq 0$, with rules

$$\frac{x \xrightarrow{a} x'}{\pi_{n+1}(x) \xrightarrow{a} \pi_n(x')} \quad a \in \mathcal{L}.$$

It is well known that

$$\pi_n(x + y) \leftrightarrows \pi_n(x) + \pi_n(y), \tag{4}$$

where $+$ is the choice operator from Example 1.

We shall now argue that the validity of this equation can be shown using Theorem 2. Again, we turn the projection operators into binary operators that ignore their second argument. The above rules therefore become

$$\frac{x \xrightarrow{a} x'}{\pi_{n+1}(x, y) \xrightarrow{a} \pi_n(x', y)} \quad a \in \mathcal{L}.$$

Note that the rules for π_n and $+$ are in the first rule format for left distributivity from Definition 11. Therefore Theorem 2 yields the validity of the left-distributivity law

$$\pi_n(x + y, z) \leftrightarrows \pi_n(x, z) + \pi_n(y, z),$$

from which the soundness of (4) follows immediately.

Example 15 (Prefix operator and synchronous parallel operator). Consider any TSS T containing the synchronous parallel operator $\|_s$ from Example 4 and containing the following binary version of the prefix operator from CCS [23], where a ranges over a set of actions $\mathcal{L}$:

$$pref_a = \frac{}{a.(x,y) \xrightarrow{a} x}.$$

We claim that T is in the second left-distributivity format for the prefix operator with respect to $\|_s$. Let us pick an action a. Then the targets of the conclusions of $pref_a$ and of

$$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{a} y'}{x \|_s y \xrightarrow{a} x' \|_s y'},$$

which is the only a-emitting rule for $\|_s$, both yield the term $p \|_s q$ when instantiated as required in Definition 13. Therefore, Theorem 4 yields the validity of the law

$$a.(x \|_s y, z) \leftrightarrows a.(x, z) \|_s a.(y, z).$$

Turning the prefix operator back to its unary version, we obtain the soundness of the following equality:

$$a.(x \|_s y) \leftrightarrows a.x \|_s a.y.$$

Row 3 in Table 1 can be used to match the targets of the conclusions of the synchronous parallel composition and the prefix operators.

Example 16 (Unit-delay operator and choice operator). Consider any TSS T that includes the choice operator $+^*$ from Example 10 and the following binary versions of the unit-delay operator:

$$delay_1 = \frac{}{(1)(x,y) \xrightarrow{\chi} x}.$$

We claim that T is in the second left-distributivity format for (1) with respect to $+^*$. To see this, it suffices to observe that the targets of the conclusions of the χ-emitting rules for those two operators, when instantiated as required in Definition 13, both yield the term $p +^* q$. Therefore, Theorem 4 yields the validity of the law

$$(1)(x +^* y, z) \leftrightarrows (1)(x, z) +^* (1)(y, z).$$

Turning the unit-delay operator back to its unary version, we obtain the well-known law

$$(1)(x +^* y) \leftrightarrows (1)(x) +^* (1)(y).$$

Row 3 in Table 1 can be used to match the targets of the conclusions of the delay rules for the unit-delay and choice operators.

7 Impossibility results

In this section we provide some impossibility results concerning the validity of the left-distributivity law. Unlike previous results about rule formats for algebraic properties, such as those surveyed in [11], we offer theorems to recognize when the left-distributivity law is guaranteed *not* to hold. When designing operational specifications for operators that are intended to satisfy a left-distributivity law, a language designer might also benefit from considering these kinds of negative results.

7.1 Left-inheriting operators

Our first negative result will concern a kind of left-inheriting operator, which we call strong left-inheriting and we now proceed to define.

Definition 15 (Forwarder operators) *Let* $\overrightarrow{k} = (k_1, k_2, \ldots, k_\ell)$*, where* $1 \leq \ell \leq n$ *and* $1 \leq k_1 < k_2 < \ldots < k_\ell \leq n$*. An operator* f *of arity* n *is a* $\overrightarrow{k}$-forwarder *if the following conditions hold for each action* a *and for all closed terms* $p_1, \ldots, p_n$*:*

- *if* $f(p_1 \ldots, p_{k_1}, \ldots, p_{k_2}, \ldots, p_{k_\ell}, \ldots, p_n) \xrightarrow{a}$ *then there is some* $1 \leq i \leq \ell$ *such that* $p_{k_i} \xrightarrow{a}$ *and*
- *for each* $1 \leq i \leq \ell$*, if* $p_{k_i} \xrightarrow{a}$ *then* $f(p_1 \ldots, p_{k_1}, \ldots, p_{k_2}, \ldots, p_{k_\ell}, \ldots, p_n) \xrightarrow{a}$*.*

Syntactic conditions to guarantee that an operator is a $\overrightarrow{k}$-forwarder can be given. However, this is beyond the scope of the present paper.

Example 17. As the reader can easily check, the left-merge operator $\lfloor\!\lfloor$ from Example 3 and the replication operator ! given by the rules below

$$\frac{x \xrightarrow{a} x'}{!x \xrightarrow{a} x' \parallel !x} \quad (a \in \mathcal{L}),$$

where $\parallel$ is the interleaving parallel composition operator from Example 3, are (1)-forwarders. On the other hand, the interleaving parallel composition operator and the choice operator + from Example 1 are $(1, 2)$-forwarders.

Definition 16 (Forwarder contexts) *The grammar for* forwarder contexts for a variable x *is*

$$F[x] ::= x \mid f(x_1, \ldots, x_{i-1}, F[x], x_{i+1}, \ldots, x_n),$$

where f *is an* n*-ary operator,* $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ *are variables,* $F[x]$ *appears as the* i*th argument of* f*, and* f *is* $\overrightarrow{k}$*-forwarder with* i *appearing in* $\overrightarrow{k}$*.*

Lemma 2. *Assume that* $F[x]$ *is a forwarder context for a variable* x*. Then, for each closed substitution* σ *and for each action* a*, the following statements hold:*

1. *if* $\sigma(x) \xrightarrow{a}$ *then* $\sigma(F[x]) \xrightarrow{a}$;
2. *if* $\sigma(F[x]) \xrightarrow{a}$ *then there is some* $y \in \mathrm{vars}(F[x])$ *such that* $\sigma(y) \xrightarrow{a}$.

Proof. Both claims can be shown by structural induction on $F[x]$. □

Definition 17 (Strong left-inheriting operators) *Given a TSS* T*, let* $\circledast$ *be a binary operator from the signature of* T*. We say that* $\circledast$ *is* strong left-inheriting with respect to an action a *whenever each* a*-emitting* $\circledast$*-defining rule* d *has the form*

$$\frac{\Phi_x \cup \Phi_y}{x \circledast y \xrightarrow{a} F[x]},$$

where

- Φ_x *and* Φ_y *are sets of* x*-testing and* y*-testing formulae, respectively, whose subsets of positive premises are finite,*
- *no two formulae in* $\Phi_x \cup \Phi_y$ *contradict each other,*
- *each positive formula in* $\Phi_x \cup \Phi_y$ *has the form* $z \xrightarrow{b} z'$ *for some action* b *and variable* z'*,*
- *the variables* x*,* y *and the targets of the positive formulae in* $\Phi_x \cup \Phi_y$ *are all distinct, and*
- $F[x]$ *is a forwarder context for* x *with* $\mathrm{vars}(F[x]) \subseteq \mathrm{vars}(\Phi_x \cup \Phi_y) \cup \{x\}$.

Intuitively, not only does a strong left-inheriting operator inherit its left argument; it also makes sure that the inherited term may affect the next step of computation.

Theorem 7 (Impossibility Theorem: strong left-inheriting operators). *Given a TSS* T*, let* $\circledast$ *be a binary operator in the signature of* T*. Assume that*

- *the set of actions is infinite,*
- *the signature of* T *contains the inaction constant from Remark 2, the prefix operators from CCS (see Example 15) and the choice operator from Example 1,*
- $\circledast$ *is a strong left-inheriting operator with respect to some action* $a \in \mathcal{L}$*, and*
- *there is some* a*-emitting and* $\circledast$*-defining rule.*

Then

$$(x + y) \circledast z \not\leftrightarrow (x \circledast z) + (y \circledast z).$$

The proof of Theorem 7 relies on the fact that, when $(p + q) \circledast r \xrightarrow{a} s_1$ for some action a and closed terms p, q, r and s_1, the term s_1 has both the initial capabilities of p and q because s_1 has some occurrence of the term $p + q$ in a forwarder context, and $+$ is itself a $(1, 2)$-forwarder. On the other hand, if $(p \circledast r) + (q \circledast r) \xrightarrow{a} s_2$, for some s_2, then s_2 is never able to have both of the initial capabilities of p and q simultaneously, since $+$ performs a choice.

Using Theorem 7, we obtain, for instance, that:

– $(x + y) \parallel z \not\Leftrightarrow (x \parallel z) + (y \parallel z)$
– $a.(x + y) \not\Leftrightarrow (a.x) + (a.y)$
– $!(x + y) \not\Leftrightarrow (!x) + (!y)$

For the last two cases, in order to apply the above-mentioned theorem, one needs to consider the binary version of the action prefixing operator from Example 15 and the binary version of the replication operator, which ignores its second argument and can be defined along the lines we followed in the examples in Section 6.

7.2 The use of negative premises

We now present two results that rely on the use of negative premises in rules.

Definition 18 (Always Moving Operators) *Given a TSS T, we say that an operator f from the signature of T with arity n is* always moving for action a *whenever $f(\overrightarrow{p}) \xrightarrow{a}$, for each n-tuple of closed terms $\overrightarrow{p}$.*

For example, an n-ary operator f, with $n \geq 1$, is always moving for action a when the set of rules $D(f, a)$ contains

– either some rule d with $\text{hyps}(d) = \emptyset$,
– or rules d_1, d_2 with $\text{hyps}(d_1) = \{x_1 \xrightarrow{a} x_1'\}$ and $\text{hyps}(d_2) = \{x_1 \overset{a}{\nrightarrow}\}$.

An example of operator that is always moving for action a is the prefixing operator $a.\_$.

Remark 5. It is possible to find syntactic conditions on the set of rules for some operator f guaranteeing that f is always moving. For instance, the decidable logic of initial transition formulae offered in [3], which is able to reason about firability of GSOS rules, can be used in order to check whether operators are always moving. The development of rule formats for always-moving operators is, however, orthogonal to the gist of this paper and therefore we do not address it here.

Theorem 8. *Given a TSS T, let $\circledast$ and $\oplus$ be binary operators in the signature of T. Assume that*

1. *the signature of T contains at least one constant,*
2. *$a \in \mathcal{L}$,*
3. *$\circledast$ is always moving for action a, and*
4. *the set of premises of each a-emitting and $\oplus$-defining rule contains either $x \overset{a}{\nrightarrow}$ or $y \overset{a}{\nrightarrow}$.*

Then

$$(x \oplus y) \circledast z \not\Leftrightarrow (x \circledast z) \oplus (y \circledast z),$$

and any triple of closed terms witnesses the above inequivalence.

Proof. Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators of the signature of T. Let p, q and r be arbitrary closed terms, which exist since the signature of T contains at least one constant.

Since $\circledast$ is always moving for action a, we have that $(p \oplus q) \circledast r \xrightarrow{a}$, $(p \circledast r) \xrightarrow{a}$ and $(q \circledast r) \xrightarrow{a}$. As each a-emitting and $\oplus$-defining rule d is, by assumption, such that $x \stackrel{a}{\nrightarrow} \in \mathrm{hyps}(d)$ or $y \stackrel{a}{\nrightarrow} \in \mathrm{hyps}(d)$, none of those rules can be used to prove an a-labelled transition for $(p \circledast r) \oplus (q \circledast r)$. It follows that

$$(p \oplus q) \circledast r \not\leftrightarrows (p \circledast r) \oplus (q \circledast r),$$

as required. □

In what follows we offer a result that ensures the invalidity of the distributivity law when negative premises appear in $\circledast$-defining rules.

Theorem 9. *Let T be a TSS whose signature contains a binary operator $\circledast$, the inaction constant $\mathbf{0}$, the prefix operators from CCS and the choice operator. Assume that there is some action a such that the only a-emitting $\circledast$-defining rule in T has the form*

$$(d)\ \frac{\Phi_x \cup \Phi_y}{x \circledast y \xrightarrow{a} t},$$

where

- *Φ_x and Φ_y are sets of x-testing and y-testing formulae, respectively, whose subsets of positive premises are finite,*
- *no two formulae in $\Phi_x \cup \Phi_y$ contradict each other,*
- *each positive formula in $\Phi_x \cup \Phi_y$ has the form $z \xrightarrow{b} z'$ for some action b and variable z',*
- *the variables x, y and the targets of the positive formulae in $\Phi_x \cup \Phi_y$ are all distinct, and*
- *$\{x \stackrel{b}{\nrightarrow} \mid b \in L\} \subseteq \Phi_x$, for some non-empty set of actions L.*

Then

$$(x + y) \circledast z \not\leftrightarrows (x \circledast z) + (y \circledast z).$$

Proof. Let $\{x \xrightarrow{a_i} x_i \mid i \in I\}$ and $\{y \xrightarrow{b_j} y_j \mid j \in J\}$, where I and J are finite index sets, be the collections of positive premises in Φ_x and Φ_y, respectively. Define

$$p = \sum_{i \in I} a_i.\mathbf{0} \quad \text{and}$$
$$r = \sum_{j \in J} b_j.\mathbf{0}.$$

By the assumption of the theorem, the closed substitution σ mapping x to p, y to r and all the other variables to $\mathbf{0}$ satisfies the premises of d. Therefore, we have that

$$p \circledast r \xrightarrow{a} \sigma(t).$$

Let $q = b.\mathbf{0}$ for some $b \in L$. Then,

$$(p \circledast r) + (q \circledast r) \xrightarrow{a} \sigma(t).$$

On the other hand, the term $(p + q) \circledast r$ does not afford an a-labelled transition because $p + q \xrightarrow{b} \mathbf{0}$ and therefore no closed substitution mapping x to $p + q$ can satisfy the premises of d, which is the only a-emitting $\circledast$-defining rule in T. This means that

$$(p + q) \circledast r \not\leftrightarrow (p \circledast r) + (q \circledast r),$$

and the claim follows. □

Example 18. Let $>$ be an irreflexive partial order over $\mathcal{L}$. The priority operator Θ from [14] is specified by the following rules:

$$\frac{x \xrightarrow{a} x', \quad x \stackrel{b}{\nrightarrow} \ (\forall b > a)}{\Theta(x) \xrightarrow{a} \Theta(x')} \quad (a \in \mathcal{L}).$$

The binary version of that operator can be defined following the lines presented in the examples in Section 6. Theorem 9, when applied to the binary version of Θ, yields the well-known fact that, when $>$ is a non-trivial partial order,

$$\Theta(x + y) \not\leftrightarrow \Theta(x) + \Theta(y).$$

Indeed, if $>$ is non-trivial, then there are actions a and b with $a < b$. The single a-emitting rule for the binary version of Θ has a negative premise of the form $x \stackrel{b}{\nrightarrow}$, and therefore Theorem 9 is applicable to derive the above inequivalence.

8 Conclusions

In this paper we have provided two rule formats guaranteeing that certain binary operators are left distributive with respect to choice-like operators. As witnessed by the wealth of examples we discussed in the main body of this study, the rule formats are general enough to cover relevant examples from the literature. In particular, they can also be applied to establish the validity of left-distributivity laws involving unary operators. This can be achieved by simply considering unary operators as binary operators that ignore their second argument.

We have also offered conditions that allow one to recognize the invalidity of the left-distributivity law in the context of left-inheriting operators and in the presence of negative premises. Such conditions can be applied to well-known examples of *invalid* left-distributivity laws.

The research presented in this article opens several interesting lines for future investigation. First of all, our rule formats can be easily adapted to obtain rule formats guaranteeing the validity of right-distributivity laws of the form

$$x \circledast (y \oplus z) = (x \circledast y) \oplus (x \circledast z).$$

The rule formats we have presented should also be extended in order to handle examples of distributivity laws where $\oplus$ is not 'choice-like'. It would also be interesting to see whether one can relax the syntactic constraints of the rule formats presented in this paper substantially, while preserving their soundness and ease of application.

Last, but not least, we intend to find further 'impossibility theorems' along the lines of those we presented in Section 7.

This future work will lead to a better understanding of the semantic nature of distributivity properties and of its links to the syntax of SOS rules.

References

1. L. Aceto, A. Birgisson, A. Ingolfsdottir, M. R. Mousavi, and M. A. Reniers. Rule formats for determinism and idempotence. In F. Arbab and M. Sirjani, editors, *Fundamentals of Software Engineering, Third IPM International Conference, FSEN 2009, Kish Island, Iran, April 15-17, 2009, Revised Selected Papers*, volume 5961 of *Lecture Notes in Computer Science*, pages 146–161. Springer-Verlag, 2010.
2. L. Aceto, B. Bloom, and F. W. Vaandrager. Turning SOS rules into equations. *Inf. Comput.*, 111(1):1–52, 1994.
3. L. Aceto, M. Cimini, and A. Ingolfsdottir. A bisimulation-based method for proving the validity of equations in GSOS languages. In *Proceedings of Structural Operational Semantics 2009, August 31, 2009, Bologna (Italy)*, volume 18 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–16, 2010.
4. L. Aceto, M. Cimini, A. Ingolfsdottir, M. Mousavi, and M. A. Reniers. On rule formats for zero and unit elements. In *Proceedings of the 26th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXVI)*, volume 265 of *Electronic Notes in Theoretical Computer Science*, pages 145–160, Ottawa, Canada, 2010. Elsevier B.V., The Netherlands.
5. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. CCS with Hennessy's merge has no finite equational axiomatization. *Theoretical Comput. Sci.*, 330(3):377–405, 2005.
6. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. Finite equational bases in process algebra: Results and open questions. In *Processes, Terms and Cycles*, volume 3838 of *Lecture Notes in Computer Science*, pages 338–367. Springer-Verlag, 2005.
7. L. Aceto, W. Fokkink, A. Ingolfsdottir, and B. Luttik. A finite equational base for CCS with left merge and communication merge. *ACM Trans. Comput. Log.*, 10(1), 2009.
8. L. Aceto, W. Fokkink, A. Ingolfsdottir, and S. Nain. Bisimilarity is not finitely based over BPA with interrupt. *Theoretical Comput. Sci.*, 366(1–2):60–81, 2006.
9. L. Aceto, W. Fokkink, and C. Verhoef. Structural operational semantics. In *Handbook of Process Algebra*, pages 197–292. Elsevier, 2001.
10. L. Aceto, A. Ingolfsdottir, M. Mousavi, and M. A. Reniers. Rule formats for unit elements. In J. van Leeuwen, A. Muscholl, D. Peleg, J. Pokorný, and B. Rumpe, editors, *SOFSEM 2010, 36th Conference on Current Trends in Theory and Practice of Computer Science, Špindleruv Mlýn, Czech Republic, January 23-29, 2010. Proceedings*, volume 5901 of *Lecture Notes in Computer Science*, pages 141–152. Springer-Verlag, 2010.

11. L. Aceto, A. Ingolfsdottir, M. R. Mousavi, and M. A. Reniers. Algebraic properties for free! *Bulletin of the European Association for Theoretical Computer Science*, 99:81–104, Oct. 2009. Columns: Concurrency.
12. J. Baeten, T. Basten, and M. Reniers. *Process Algebra: Equational Theories of Communicating Processes*, volume 50 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2009.
13. J. Baeten and J. Bergstra. Mode transfer in process algebra. Technical Report Report CSR 00–01, Eindhoven University of Technology, 2000.
14. J. Baeten, J. Bergstra, and J. W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, IX(2):127–168, 1986.
15. J. Baeten and S. Mauw. Delayed choice: An operator for joining Message Sequence Charts. In D. Hogrefe and S. Leue, editors, *Formal Description Techniques VII, Proceedings of the 7th IFIP WG6.1 International Conference on Formal Description Techniques, Berne, Switzerland, 1994*, volume 6 of *IFIP Conference Proceedings*, pages 340–354. Chapman & Hall, 1995.
16. J. Bergstra and J. W. Klop. Fixed point semantics in process algebras. Report IW 206, Mathematisch Centrum, Amsterdam, 1982.
17. J. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1/3):109–137, 1984.
18. E. Brinksma. A tutorial on LOTOS. In *Proceedings of the IFIP WG6.1 Fifth International Conference on Protocol Specification, Testing and Verification V*, pages 171–194, Amsterdam, The Netherlands, The Netherlands, 1985. North-Holland Publishing Co.
19. S. Cranen, M. Mousavi, and M. A. Reniers. A rule format for associativity. In F. van Breugel and M. Chechik, editors, *Proceedings of the 19th International Conference on Concurrency Theory (CONCUR'08)*, volume 5201 of *Lecture Notes in Computer Science*, pages 447–461, Toronto, Canada, 2008. Springer-Verlag, Berlin, Germany.
20. R. de Simone. Higher-level synchronising devices in Meije-SCCS. *Theoretical Computer Science*, 37:245–267, 1985.
21. C. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, 1978.
22. C. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, Englewood Cliffs, 1985.
23. R. Milner. *Communication and Concurrency*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
24. F. Moller. The importance of the left merge operator in process algebras. In M. Paterson, editor, *Proceedings* 17^{th} *ICALP,* Warwick, volume 443 of *Lecture Notes in Computer Science*, pages 752–764. Springer-Verlag, July 1990.
25. F. Moller. The nonexistence of finite axiomatisations for CCS congruences. In *Proceedings* 5^{th} *Annual Symposium on Logic in Computer Science,* Philadelphia, USA, pages 142–153. IEEE Computer Society Press, 1990.
26. M. Mousavi, M. Reniers, and J. F. Groote. A syntactic commutativity format for SOS. *Information Processing Letters*, 93:217–223, Mar. 2005.
27. M. R. Mousavi, M. A. Reniers, and J. F. Groote. SOS formats and meta-theory: 20 years after. *Theor. Comput. Sci.*, 373(3):238–272, 2007.
28. X. Nicollin and J. Sifakis. The algebra of timed processes, ATP: Theory and application. *Information and Computation*, 114(1):131–178, 1994.
29. D. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI-Conference on Theoretical Computer Science*, pages 167–183, London, UK, 1981. Springer-Verlag.

30. G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.
31. G. D. Plotkin. A structural approach to operational semantics. *J. Log. Algebr. Program.*, 60-61:17–139, 2004.
32. T. Przymusinski. The well-founded semantics coincides with the three-valued stable semantics. *Fundamenta Informaticae*, 13(4):445–463, 1990.
33. D. Sangiorgi and D. Walker. *The π-calculus: a Theory of Mobile Processes.* Cambridge University Press, 2001.

A Proof of Theorem 1

Instead of proving Theorem 1 we prove a stronger theorem. In what follows, when we say $(p \oplus q) \circledast r \xrightarrow{a}$ using rules d_1 and d_2, the considered transition is provable by the $\circledast$-defining rule d_1, possibly using the $\oplus$-defining rule d_2 to prove a transition $(p \oplus q) \xrightarrow{a} p'$ satisfying the set $\Phi_x(d_1)$ of x-testing premises in d_1. We say $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a}$ using rules d_2, d_1 and d_3, with the straightforward analogous meaning, using d_1 to prove a transition from $(p \circledast r)$ satisfying $\Phi_x(d_2)$ and d_3 to prove a transition from $(q \circledast r)$ satisfying $\Phi_y(d_2)$.

Theorem 10. *Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators in the signature of T. Suppose that* $\mathrm{Fire}(\circledast, \oplus, a)$, *for some actions a. Then, for all closed terms p, q, and r,*

- *if $(p \oplus q) \circledast r \xrightarrow{a}$ using rules d_1 and d_2 then $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a}$ using rules d_2, d_1 and d_1.*
- *$(p \circledast r) \oplus (q \circledast r) \xrightarrow{a}$ using rules d_2, d_1 and d_3 then $(p \oplus q) \circledast r \xrightarrow{a}$ using rules d_1 or d_3, and d_2.*

It is easy to see that Theorem 10 implies Theorem 1.

Theorem 10 can be proved along the lines of Theorem 2 and we therefore omit the details.

B Proof of Theorem 2

Let T be a TSS, and let $\circledast$ and $\oplus$ be binary operators in the signature of T. Assume that the rules for $\circledast$ and $\oplus$ are in the first rule format for left distributivity. We show the following two claims, where p, q, r, s are arbitrary closed terms and a is any action:

1. If $(p \oplus q) \circledast r \xrightarrow{a} s$ then $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$.
2. If $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$ then $(p \oplus q) \circledast r \xrightarrow{a} s$.

We consider each of the above claims in turn.

1. Assume that $(p \oplus q) \circledast r \xrightarrow{a} s$. We shall prove that $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$.

Since $(p \oplus q) \circledast r \xrightarrow{a} s$ and Fire$(\circledast, \oplus, a)$ holds, there are a rule d_1 of the form

$$\frac{(\emptyset \text{ or } \{x \xrightarrow{a} x'\}) \ \cup \ \Phi_y}{x \circledast y \xrightarrow{a} t}$$

and a closed substitution σ such that

- $\sigma(x) = p \oplus q$,
- $\sigma(y) = r$,
- $\sigma(t) = s$ and
- σ satisfies the premises of d_1.

We shall argue that $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$ by considering two cases, depending on whether d_1 has a premise of the form $x \xrightarrow{a} x'$.

(a) Case: d_1 has no x-testing premise. In this case, rule d_1 can be used to infer that $p \circledast r \xrightarrow{a} s$ and $q \circledast r \xrightarrow{a} s$ both hold. Indeed, recall that $x \notin \text{vars}(\Phi_y)$ by the constraints of the rule form $(R1)$ and $x \notin \text{vars}(t)$ by constraint 2 in Definition 11. Therefore, the closed substitution $\sigma[x \mapsto p]$ satisfies the premises of d_1 and is such that

$$\sigma[x \mapsto p](x \circledast y \xrightarrow{a} t) = p \circledast r \xrightarrow{a} s.$$

A similar reasoning using the closed substitution $\sigma[x \mapsto q]$ shows that $q \circledast r \xrightarrow{a} s$ is also provable using d_1 as claimed. The first condition in Definition 10 yields the existence of some rule $d_2 \in D(\oplus, a)$ of the form

$$\frac{(\{x \xrightarrow{a} x'\} \text{ or } \{y \xrightarrow{a} y'\} \text{ or } \{x \xrightarrow{a} x', y \xrightarrow{a} y'\})}{x \oplus y \xrightarrow{a} t}.$$

By constraint 3, d_2 has a target variable of one of its premises as target of its conclusion. Therefore, regardless of the set of premises of d_2, we can instantiate that rule using any closed substitution mapping x to $p \circledast r$, y to $q \circledast r$ and both x' and y' to s to infer that

$$(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s,$$

as required.

(b) Case: d_1 has a premise of the form $x \xrightarrow{a} x'$. In this case, as σ satisfies the premises of d_1, we have that

$$\sigma(x) = p \oplus q \xrightarrow{a} \sigma(x').$$

The above transition can be proved using a rule $d_2 \in D(\oplus, a)$ of the form

$$\frac{(\{x \xrightarrow{a} x'\} \text{ or } \{y \xrightarrow{a} y'\} \text{ or } \{x \xrightarrow{a} x', y \xrightarrow{a} y'\})}{x \oplus y \xrightarrow{a} t'},$$

where, by constraint 3, $t' = x'$ or $t' = y'$. Assume, without loss of generality, that $t' = y'$. Then $y \xrightarrow{a} y'$ is a premise of rule d_2 and

$$q \xrightarrow{a} \sigma(x').$$

So, instantiating rule d_1 above using $\sigma[x \mapsto q]$, we have that

$$\sigma[x \mapsto q](x \circledast y) = q \circledast r \xrightarrow{a} \sigma[x \mapsto q](t) = \sigma(t) = s.$$

(Recall that $x \notin \text{vars}(t)$ by constraint 2 in Definition 11.) If d_2 does not have any x-testing premise then the above transition can be used to satisfy its premise and we can infer

$$(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s,$$

as required. Assume therefore that d_2 has $x \xrightarrow{a} x'$ as a premise, and therefore has the form

$$\frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} y'}.$$

Since the transition $p \oplus q \xrightarrow{a} \sigma(x')$ is proved using d_2, there is some p' such that $p \xrightarrow{a} p'$. Recall that, by the assumptions for this case of the proof,

$$d_1 = \frac{\{x \xrightarrow{a} x'\} \cup \Phi_y}{x \circledast y \xrightarrow{a} t}.$$

Then the substitution $\sigma[x \mapsto p, x' \mapsto p']$ satisfies the premises of d_1, and we can deduce that

$$\sigma[x \mapsto p, x' \mapsto p'](x \circledast y) = p \circledast r \xrightarrow{a} \sigma[x \mapsto p, x' \mapsto p'](t) = \sigma[x' \mapsto p'](t).$$

Using rule d_2 and any substitution that maps x to $p \circledast r$, x' to $\sigma[x' \mapsto p'](t)$, y to $q \circledast r$ and y' to s, we may conclude that

$$(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s,$$

as required.

2. Assume that $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$. We shall prove that $(p \oplus q) \circledast r \xrightarrow{a} s$. Since $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s$ and $\text{Fire}(\circledast, \oplus, a)$ holds, there are a rule d_2 of the form

$$\frac{(\{x \xrightarrow{a} x'\} \text{ or } \{y \xrightarrow{a} y'\} \text{ or } \{x \xrightarrow{a} x', y \xrightarrow{a} y'\})}{x \oplus y \xrightarrow{a} t},$$

where, by constraint 3, $t = x'$ or $t = y'$, and a closed substitution σ such that
   - $\sigma(x) = p \circledast r$,
   - $\sigma(y) = q \circledast r$,
   - $\sigma(t) = s$ and
   - σ satisfies the premises of d_2.

Assume, without loss of generality, that $t = x'$. Therefore $x \xrightarrow{a} x'$ is a premise of d_2 and

$$\sigma(x) = p \circledast r \xrightarrow{a} s = \sigma(x').$$

Since $p \circledast r \xrightarrow{a} s$, there are some rule

$$d_1 = \frac{(\emptyset \text{ or } \{x \xrightarrow{a} x'\}) \ \cup \ \Phi_y}{x \circledast y \xrightarrow{a} t'}$$

and a closed substitution σ' such that

- $\sigma'(x) = p$,
- $\sigma'(y) = r$,
- $\sigma'(t') = s$ and
- σ' satisfies the premises of d_1.

We shall argue that $(p \oplus q) \circledast r \xrightarrow{a} s$ by considering two cases, depending on whether d_1 has a premise of the form $x \xrightarrow{a} x'$.

(a) Case: d_1 has no x-testing premise.
Consider the substitution $\sigma'[x \mapsto p \oplus q]$. Since $x \notin \text{vars}(\Phi_y)$ and σ' satisfies the premises of d_1, it follows that $\sigma'[x \mapsto p \oplus q]$ also satisfies Φ_y. Therefore, we can instantiate rule d_1 with $\sigma'[x \mapsto p \oplus q]$ to infer that

$$\sigma'[x \mapsto p \oplus q](x \circledast y) = (p \oplus q) \circledast r \xrightarrow{a} \sigma'[x \mapsto p \oplus q](t') = \sigma'(t') = s,$$

as required. (Recall that $\circledast$ is non-left-inheriting by condition 2 in Definition 11.)

(b) Case: d_1 has a premise of the form $x \xrightarrow{a} x'$. Then,

$$d_1 = \frac{\{x \xrightarrow{a} x'\} \ \cup \ \Phi_y}{x \circledast y \xrightarrow{a} t'}.$$

As σ' satisfies the premises of d_1, we have that

$$\sigma'(x) = p \xrightarrow{a} \sigma'(x').$$

If $x \xrightarrow{a} x'$ is the only premise of rule d_2, then we can use that rule and the above transition to infer that

$$p \oplus q \xrightarrow{a} \sigma'(x').$$

Consider now the closed substitution $\sigma'[x \mapsto p \oplus q]$. This substitution satisfies the premises of rule d_1, because so does σ' and $x \notin \text{vars}(\Phi_y)$. Therefore, instantiating rule d_1 with $\sigma'[x \mapsto p \oplus q]$, we may derive the transition

$$(p \oplus q) \circledast r \xrightarrow{a} \sigma'[x \mapsto p \oplus q](t') = \sigma'(t') = s,$$

as required.

Assume now that $x \xrightarrow{a} x'$ is *not* the only premise of rule d_2. Then, because of the assumptions of this case,

$$d_2 = \frac{\{x \xrightarrow{a} x', y \xrightarrow{a} y'\}}{x \oplus y \xrightarrow{a} x'}.$$

Recall that we used the above rule and the closed substitution σ to prove the transition

$$(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} s.$$

Therefore we have that

$$\sigma(y) = q \circledast r \xrightarrow{a} \sigma(y').$$

Using condition 4 in Definition 11 and the form of the rules d_1 and d_2, this means that there are a rule

$$d_3 = \frac{\{x \xrightarrow{a} x'\} \ \cup \ \Phi'_y}{x \circledast y \xrightarrow{a} t''}$$

and a closed substitution $\hat{\sigma}$ such that

- $\hat{\sigma}(x) = q \xrightarrow{a} \hat{\sigma}(x')$,
- $\hat{\sigma}(y) = r$,
- $\hat{\sigma}(t'') = \sigma(y')$ and
- $\hat{\sigma}$ satisfies Φ'_y.

Using rule d_2 with premises $p \xrightarrow{a} \sigma'(x')$ and $q \xrightarrow{a} \hat{\sigma}(x')$, we obtain that

$$p \oplus q \xrightarrow{a} \sigma'(x').$$

Finally, instantiating rule d_1 with the closed substitution $\sigma'[x \mapsto p \oplus q]$, we infer the transition

$$\sigma'[x \mapsto p \oplus q](x \circledast y) = (p \oplus q) \circledast r \xrightarrow{a} \sigma'[x \mapsto p \oplus q](t') = \sigma'(t') = s,$$

as required.

This completes the proof.

C Proof of Theorem 4

Let $T = (\Sigma, \mathcal{L}, D)$ be a TSS. Assume that T is in the second left-distributivity format for $\circledast$ with respect to $\oplus$. We shall prove that

$$(x \oplus y) \circledast z \leftrightarrows (x \circledast z) \oplus (y \circledast z).$$

To this end, it suffices to show that the relation

$$\mathcal{R} = \{((p \oplus q) \circledast r, (p \circledast r) \oplus (q \circledast r)) \mid p, q, r \in \mathbb{C}(\Sigma)\} \cup \leftrightarrows$$

is a bisimulation.

Let us pick an action a and closed terms p, q and r. We now prove the following two claims:

1. If $(p \oplus q) \circledast r \xrightarrow{a} v_1$ then $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} v_2$, for some v_2 such that $v_1 \mathcal{R} v_2$.
2. If $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} v_2$ then $(p \oplus q) \circledast r \xrightarrow{a} v_1$, for some v_1 such that $v_1 \leftrightarrows v_2$.

We consider these two claims separately.

1. Assume that $(p \oplus q) \circledast r \xrightarrow{a} v_1$ for some closed term v_1. This means that $(p \oplus q) \circledast r \xrightarrow{a}$ using rules d_1 and d_2, for some $\circledast$-defining rule d_1 and some $\oplus$-defining rule d_2.
   By Theorem 10, $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} v_2$, for some closed term v_2, using rules d_2, d_1 and d_1. We shall now show that $v_1 \mathcal{R} v_2$.
   As T is in the second left-distributivity format for $\circledast$ with respect to $\oplus$, we have that $d_1 \overset{\rightsquigarrow}{\sim} d_2$. We distinguish two cases depending on whether the set of premises of d_2 is a singleton.
   - Case: hyps$(d_2) = \{x \xrightarrow{a} x'\}$ or hyps$(d_2) = \{y \xrightarrow{a} y'\}$. In both of the cases, the term v_1 is formed by exactly the substitutions of condition 3a in Definition 13, when the variable p' is used as a term such that $p \xrightarrow{a} p'$, similarly q' for q, and each r_i for y_i, $i \in I$. Thus, $v_1 = \sigma_1(concl(d_1))$ and, for the same reasons, $v_2 = \sigma_2(\text{toc}(()d_2))$. Now, by the definition of $\overset{\rightsquigarrow}{\sim}$, we have that $v_1 \rightharpoonup v_1'$ and $v_2 \rightharpoonup v_2'$, for some v_1' and v_2' with $v_1' = v_2'$, by possibly using one application of axiom

     $$(x \oplus y) \circledast z = (x \circledast z) \oplus (y \circledast z)$$

     at the top level. Since $v_1 \leftrightarrows v_1'$ and that $v_2 \leftrightarrows v_2'$ hold by Lemma 1, by possibly using the transitivity of bisimilarity, we may conclude that $v_1 \mathcal{R} v_2$, as required.
   - Case: hyps$(d_2) = \{x \xrightarrow{a} x', y \xrightarrow{a} y'\}$. In this case, by condition 3b in Definition 13, the bisimilarity proven in the previous case is guaranteed for all the possible pairs of $\circledast$-defining rules, and this also includes the case when the two premises of rule d_2 are both satisfied using rule d_1.
2. Assume that $(p \circledast r) \oplus (q \circledast r) \xrightarrow{a} v_2$ for some closed term v_2. This transition can be proved using rules d_2, d_1, d_3, for some $\oplus$-defining rule d_2 and some $\circledast$-defining rules d_1 and d_3.
   By Theorem 10, $(p \oplus q) \circledast r \xrightarrow{a} v_1$, for some closed term v_1, using rules d_1 or d_3 and d_2. We now argue that $v_1 \mathcal{R} v_2$. By condition 3b in Definition 13, reasoning as above, $v_1 \mathcal{R} v_2$ is guaranteed for all the possible pairs of $\circledast$-defining rules, including the case when the transition $(p \oplus q) \circledast r \xrightarrow{a} v_1$ is proved using d_1 and d_2 or using d_3 and d_2.

This completes the proof.

D Proof of Theorem 7

Let T be a TSS and let $\circledast$ be a binary operator of the signature of T. Assume the hypotheses of Theorem 7.

Let us pick an a-emitting and $\circledast$-defining rule d. By the hypotheses of the theorem, d has the form

$$\frac{\Phi_x \cup \Phi_y}{x \circledast y \xrightarrow{a} F[x]},$$

where

- Φ_x and Φ_y are sets of x-testing and y-testing formulae, respectively,
- no two formulae in $\Phi_x \cup \Phi_y$ contradict each other,
- each positive formula in $\Phi_x \cup \Phi_y$ has the form $z \xrightarrow{b} z'$ for some action b and variable z',
- the variables x, y and the targets of the positive formulae in $\Phi_x \cup \Phi_y$ are all distinct, and
- $F[x]$ a forwarder context for x with $\text{vars}(F[x]) \subseteq \text{vars}(\Phi_x \cup \Phi_y) \cup \{x\}$.

Since the signature of T contains the inaction, the prefix operators and the choice operator, and no two formulae in $\Phi_x \cup \Phi_y$ contradict each other, it is easy to construct three terms p, q, and r such that

1. p 'satisfies' Φ_x,
2. if $x \stackrel{b}{\nrightarrow} \in \Phi_x$, then $q \stackrel{b}{\nrightarrow}$,
3. r 'satisfies' Φ_y,
4. $p \xrightarrow{b}$, $q \stackrel{b}{\nrightarrow}$ and $r \stackrel{b}{\nrightarrow}$, for some action b,
5. $q \xrightarrow{c}$, $p \stackrel{c}{\nrightarrow}$ and $r \stackrel{c}{\nrightarrow}$, for some action c, and
6. the depth of p and r is one—that is, for all action b and c, and closed terms p' and r', if $p \xrightarrow{b} p'$ then $p' \stackrel{c}{\nrightarrow}$, and if $r \xrightarrow{b} r'$ then $r' \stackrel{c}{\nrightarrow}$.

Conditions 4 and 5 can be met because, by assumption, the set of actions is infinite.

We claim that

$$(p + q) \circledast r \not\Leftrightarrow (p \circledast r) + (q \circledast r).$$

To see this, observe that, since $+$ is a $(1,2)$-forwarder operator, due to conditions 1 and 2, $p + q$ 'satisfies' Φ_x. By condition 3, the rule d fires with some closed substitution σ mapping x to $p+q$ and y to r. Thus $(p+q) \circledast r \xrightarrow{a} \sigma(F[x])$. By conditions 4–5 and Lemma 2, we have that $\sigma(F[x]) \xrightarrow{b}$ and $\sigma(F[x]) \xrightarrow{c}$.

Assume now that $(p \circledast r) + (q \circledast r) \xrightarrow{a} s$, for some s. We will now argue that $\sigma(F[x]) \not\Leftrightarrow v$, proving our claim that

$$(p + q) \circledast r \not\Leftrightarrow (p \circledast r) + (q \circledast r).$$

Indeed, suppose that $p \circledast r \xrightarrow{a} s$. Since $\circledast$ is strong left-inheriting with respect to an action a, we have that there are an a-emitting $\circledast$-defining rule of the form

$$\frac{\Phi'_x \cup \Phi'_y}{x \circledast y \xrightarrow{a} F'[x]},$$

satisfying the requirements in Definition 17 and a closed substitution σ' such that $s = \sigma'(F'[x])$. By conditions 5 and 6, using Lemma 2 we have that $s \stackrel{c}{\nrightarrow}$. Therefore $\sigma(F[x]) \not\trianglelefteq s$.

If $q \circledast r \xrightarrow{a} s$ then, reasoning in similar fashion using conditions 4 and 6 as well as Lemma 2, we infer that $s \stackrel{b}{\nrightarrow}$. Therefore $\sigma(F[x]) \not\trianglelefteq s$, and we are done.