OPEN ACCESS
OPUS AUGSBURG

Universität Augsburg
Universitätsbibliothek

Modal algebra and Petri nets

Han-Hing Dang, Bernhard Möller

Angaben zur Veröffentlichung / Publication details:

Dang, Han-Hing, and Bernhard Möller. 2015. "Modal algebra and Petri nets." *Acta Informatica* 52 (2-3): 109–32. https://doi.org/10.1007/s00236-015-0216-3.

Nutzungsbedingungen / Terms of use: **licgercopyright**

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Noname manuscript No.
(will be inserted by the editor)

Modal Algebra and Petri Nets

Han-Hing Dang · Bernhard Möller

Walter Vogler and Bernhard Möller have now for about 22 years been working at the Institute for Informatics at the University of Augsburg in close neighbourhood. Both were and are very much interested in formal semantics, although in different areas: Walter in concurrent, Bernhard in sequential systems. Still there were many debates whether or how their fields of work could have concrete common touchpoints, in particular, since both use algebraic concepts to a smaller or larger extent. Since in recent years Bernhard also started looking at the concurrent side, we felt that Walter's upcoming jubilee was the right point in time to try and construct a bridge (or at least a gangplank) between the two fields. Therefore it is our great pleasure to dedicate this paper to Walter Vogler on the occasion of his 60th birthday, also with sincere thanks for his friendship and the pleasant collaboration. May he continue to throw out his nets to bring in an ample harvest of impressive results!

Received: date / Accepted: date

Abstract We use the by now well established setting of modal semirings to derive a modal algebra for Petri nets. It is based on a relation-algebraic calculus for separation logic that enables calculations of properties in a pointfree fashion and at an abstract level. Basically, we start from an earlier logical approach to Petri nets that in particular uses modal box and diamond operators for stating properties about the state space of such a net. We provide relational translations of the logical formulas which further allow the characterisation of general behaviour of transitions in an algebraic fashion. From the relational structure an algebra for frequently used properties of Petri nets is derived. In particular, we give connections to typical used assertion classes of separation logic. Moreover, we demonstrate applicability of the algebraic approach by calculations concerning a standard example of a mutex net.

Keywords modal algebra · mutex · Petri net · relations · separation algebra

1 Introduction

The formalism of Petri nets has been a major research topic for many decades and has a large variety of applications. As a particular case of such nets there are so-called Signal Transition Graphs to which Walter Vogler has contributed

This research was partially funded by the DFG project *MO 690/9-1* AlgSep — *Algebraic Calculi for Separation Logic.*

Han-Hing Dang · Bernhard Möller
Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany,
E-mail: {h.dang,moeller}@informatik.uni-augsburg.de

a multitude of papers (e.g. [42,24]). In earlier papers he has been involved with questions of equivalence and refinement of Petri nets (e.g. [41,18]). The present paper constructs an algebraic framework for some of the basic aspects of Petri nets, such as transitions, markings, reachability and fairness. It is based on the by now well established theory of modal semirings [12,13] as well as on earlier logical approaches to Petri nets [15,35,36], where, in particular, connections to separation logic [38] have been introduced. Separation logic was originally introduced to facilitate reasoning about data structures involving pointers in a Hoare logic style. In [35,36] the logical approach to Petri nets developed in [15] is reconsidered and extended with modal operators used to state properties about reachable markings within such nets.

The goal of the present work is to develop from that approach a general modal algebraic structure that allows abstract reasoning about reachability within Petri nets. As a starting point, we use a general relational approach to separation logic developed in [9,10].

The paper is structured as follows. In Section 2 we define Petri nets. In Section 3 we present a logic for them that allows reasoning with modal formulas about reachability of markings. Section 4 introduces the basics of separation algebras and a relational semantics for commands over them. In Section 5 we specialise that semantics to Petri nets viewed as a separation algebra and extend it to the logic of Section 3. In Section 6 this is abstracted to give a Petri net algebra based on well-known algebraic concepts. In Section 7 we enrich that algebra by the notions of tests and modal operators. Moreover, we provide useful consequences of the algebraic laws that, in particular, allow pointfree proofs of frequently used inference rules in Petri net logic. In Section 8 we state properties of nets in an algebraic fashion. In Sections 9 and 10 we show how to express safety, fairness and liveness algebraically and illustrate this with concrete calculations for a mutex net. Finally, we discuss some related work in Section 11 and conclude with a summary and an outlook on future work in Section 12.

2 Petri Nets

We repeat the basic notions of Petri nets as given in [15].

Definition 2.1

- A *Petri net* is a structure $\mathcal{N} = (P, T, pre(\_), post(\_))$. The set P consists of *places* and is disjoint from the set T of *transitions*.
- A *marking* is a function $M : P \to \mathbb{N}$, i.e., a mapping from places to natural numbers, assigning a number of *tokens* to each place. The set of all markings is denoted by $\mathcal{M}$.
- $pre(\_)$ and $post(\_)$ are functions of type $T \to \mathcal{M}$. For a transition t, the marking $pre(t)$ represents the number of tokens on each place required to enable *firing* t, while $post(t)$ denotes the number of tokens that t emits to each place once it fires.
- The addition of markings M, N is given by $(M + N)(p) =_{df} M(p) + N(p)$ for any place p, and $[]$ denotes the empty marking, i.e., $[](p) = 0$ for any $p \in P$. Moreover, for a place $p \in P$ we define the *singleton marking* M_p with $M_p(p) = 1$ and $M_p(q) = 0$ for all other places $q \neq p$.

- The order $\preceq$ on $\mathcal{M}$ is the pointwise extension of $\leq$ on $\mathbb{N}$, i.e., $M \preceq N \Leftrightarrow_{df} \forall\, p \in P : M(p) \leq N(p)$.

Definition 2.2 The behaviour of transitions t described above can be formalised by the following *firing* relation $[t\rangle$ between markings:

$$M[t\rangle N \Leftrightarrow_{df} \exists\, M' \in \mathcal{M} :\ M = \mathit{pre}(t) + M' \wedge N = \mathit{post}(t) + M'\,. \qquad (1)$$

This induces the *one-step reachability* relation given by

$$M \rightsquigarrow N \Leftrightarrow_{df} \exists\, t \in T :\ M[t\rangle N\,.$$

Finally, we call N *reachable* from M if $M \rightsquigarrow^* N$, where $\rightsquigarrow^*$ is the reflexive and transitive closure of $\rightsquigarrow$.

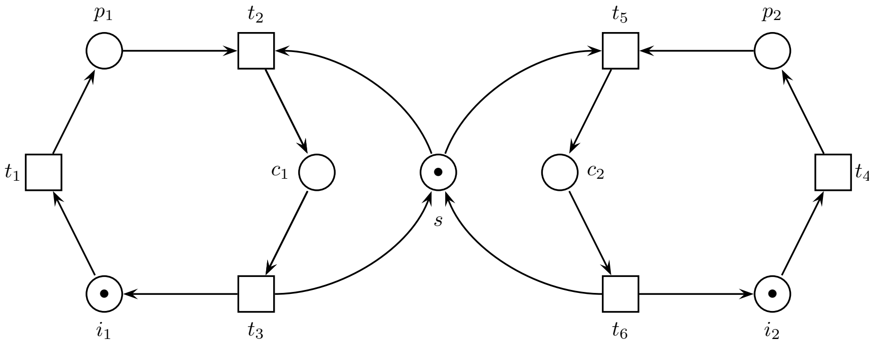

Fig. 1 An example of a mutex net.

As a standard example, consider the net depicted in Figure 1. It illustrates two processes Pro_1 and Pro_2 that are synchronised by a semaphore represented by the place s (cf. [25]). Both processes are separated graphically from each other by the semaphore, i.e., Pro_1 denotes the left subnet and Pro_2 the right one. The components of the whole net are given by $P = \{p_1, p_2, c_1, c_2, i_1, i_2, s\}$ and $T = \{t_i \mid i \in \{1, \ldots, 6\}\}$ where

- p_i denotes a state where Pro_i is pending, i.e., Pro_i is waiting for the semaphore to be available for entering its critical section;
- i_i corresponds to an idle state where Pro_i does nothing;
- c_i represents the critical section of each process.

The semaphore s works in the following way: The transitions t_2 or t_5 can only fire if a token is available on s and process i is in its pending state, i.e., p_i is marked. By firing t_2 or t_5 a token of s and p_i is consumed by the respective transition and a further token is produced in c_i which means that Pro_i is in its critical section. Hence, for the case of t_2 we have $\mathit{pre}(t_2) = M_{p_1} + M_s$ and $\mathit{post}(t_2) = M_{c_1}$. Analogous markings can be given for the transition t_5 .

Then we have $(pre(t_2) + M_{p_1})(p_1) = 2$, $(pre(t_2) + M_{p_1})(s) = 1$ and $(pre(t_2) + M_{p_1})(p) = 0$ on the remaining places p. It is not difficult to see that transition t_2 satisfies $(pre(t_2) + M_{p_1})\ [t_2\rangle\ (post(t_2) + M_{p_1})$ by choosing $M' = M_{p_1}$ in Definition 2.2. In particular, by setting $M' = []$ we also have $pre(t_2)\ [t_2\rangle\ post(t_2)$ since $[]$ is neutral w.r.t. $+$ on markings and $(pre(t_2) + M_p)\ [t_2\rangle\ (post(t_2) + M_p)$ for any other marking $M_p \in \mathcal{M}$. For this behaviour of transitions in Petri nets we will later provide an algebraic formalisation that reflects the described observation in a simple and abstract fashion.

3 A Logic for Petri Nets

We continue by giving the syntax and semantics of a logic, presented in [35,36], for characterising states, i.e., markings, and reachability conditions in Petri nets using modal operators. The purpose of the logic is to define formulas that characterise sets of markings in a given fixed Petri net. The syntax is as follows:

$$\begin{aligned} A ::= {} & \pi \mid \mathsf{false} \mid \mathsf{true} \mid \neg A \mid A \vee A \mid A \wedge A \mid \\ & A * A \mid A \mathrel{-\!\!*} A \mid I \mid \\ & \Box_+ A \mid \Box_- A \mid \Diamond_+ A \mid \Diamond_- A\,. \end{aligned}$$

The base case assertions π are taken from a set of atomic formulas. Frequently used examples are $pre(t)$ or $post(t)$ that simply characterise the corresponding markings w.r.t. a transition t or any $p \in P$ which logically denotes the singleton marking M_p.

The remaining syntactic constructs in the first row are the same as in classical logic. In the second row the assertions are built from operators that are well-known in separation logic, i.e., *separating conjunction* $*$ and *separating implication* $\mathrel{-\!\!*}$.

Intuitively, the former corresponds to the sum of markings in the following sense: e.g., $p * q$ means a singleton marking of each of the places p and q if they are different, while by $p * p$ one would characterise that p carries two tokens (separating conjunction is not idempotent). By contrast, the standard conjunction $p \wedge q$ expresses that both p and q are marked, regardless of whether they are the same or not. Note that $p \wedge q$ might be unsatisfiable if p, q make assumptions about different places. As an example consider $p = c_1$ which asserts a marking where exactly one token on the place c_1 is available and nothing anywhere else. Similarly, $q = c_2$ asserts exactly one token in c_2 and nothing elsewhere. Hence c_1 and c_2 can not hold at the same time and therefore $c_1 \wedge c_2$ is unsatisfiable. Finally, $p * \mathsf{true}$ requires at least p to be marked.

Separating implication is the upper adjoint of separating conjunction, satisfying for any assertions P, Q, R the relationship

$$(P * Q) \to R \;\Leftrightarrow\; Q \to (P \mathrel{-\!\!*} R)\ .$$

Basically, the formula $P \mathrel{-\!\!*} R$ characterises all markings M such that whenever a marking N satisfying P is added to M then $N + M$ will satisfy R.

The special assertion I denotes the empty marking and is the unit of $*$.

Finally, we deal with the modal operators in the third row above. As an example we again consider the net of Figure 1. The formula $\Box_+(s * i_1 * i_2)$ characterises all markings M where every marking reachable from M is always only singly-marked in the semaphore s and the idle states. Symmetrically, $\Box_-(s * i_1 * i_2)$ denotes the markings M where every marking leading to M has to satisfy $s * i_1 * i_2$. The diamond operators are the De Morgan duals of the box ones, i.e., $\Diamond\varphi$ is equivalent to $\neg\Box\neg\varphi$, and hence are existential quantifiers about markings, whereas the boxes act as universal quantifiers.

In the formal semantics of the modal operators we follow the approach of [35, 36] which overcomes the drawbacks of the restrictive intuitionistic logic in [15] and yields a more flexible and expressive logic for reasoning about reachability conditions within a Petri net. Formally, a Kripke semantics for the logic is given as follows. We assume a valuation function i that assigns to each atomic formula π the set of markings for which this formula is true.

$$\begin{array}{lll}
M \models \pi & \Leftrightarrow_{df} & M \in i(\pi)\,, \\
M \models I & \Leftrightarrow_{df} & M = [\,]\,, \\
M \models \mathsf{false} & \Leftrightarrow_{df} & \text{false}\,, \\
M \models A \to B & \Leftrightarrow_{df} & M \models A \text{ implies } M \models B\,, \\
M \models A * B & \Leftrightarrow_{df} & \exists\, N_1, N_2 \in \mathcal{M} : M = N_1 + N_2 \text{ and } N_1 \models A \text{ and } N_2 \models B\,, \\
M \models A \mathrel{-\!\!*} B & \Leftrightarrow_{df} & \forall\, N \in \mathcal{M} : N \models A \text{ implies } N + M \models B\,, \\
M \models \Box_+ A & \Leftrightarrow_{df} & \forall\, N : M \rightsquigarrow^* N \text{ implies } N \models A\,, \\
M \models \Box_- A & \Leftrightarrow_{df} & \forall\, N : N \rightsquigarrow^* M \text{ implies } N \models A\,.
\end{array}$$

All remaining well-known connectives of classical and modal logic can be defined as follows:

$$\begin{array}{rlrl}
\neg A & \Leftrightarrow_{df}\ A \to \mathsf{false}\,, & \mathsf{true} & \Leftrightarrow_{df}\ \neg\mathsf{false}\,, \\
A \vee B & \Leftrightarrow_{df}\ \neg A \to B\,, & A \wedge B & \Leftrightarrow_{df}\ \neg(\neg A \vee \neg B)\,, \\
\Diamond_+ A & \Leftrightarrow_{df}\ \neg\Box_+\neg A\,, & \Diamond_- A & \Leftrightarrow_{df}\ \neg\Box_-\neg A\,.
\end{array}$$

4 Separation Algebras and Commands

In Definition 2.2 we have seen that every transition t induces a relation $[t\rangle$ between markings which then was lifted to the relation $\rightsquigarrow$. This is the motivation for tying in Petri nets and their logic with the well established area of relational program semantics. We use the general relational approach of [7, 10] which also comprises the above-mentioned $*$ operator of separation logic (SL). It is built using the concept of separation algebras [4] that provides a general way to characterise the structure and properties of abstract resources.

In Petri nets the resources are the markings. The firing rule involves splitting (or *separating*) the token supply on a place, which is why separation logic is relevant to the area of Petri nets. The converse *combination* operator is the sum of markings. An algebraic abstraction of this is the following notion.

Definition 4.1

1. A *partial monoid* is a structure $(\Sigma, \bullet, u)$ with a set Σ of *states* (e.g., markings of Petri net places), a partial combination operator $\bullet : \Sigma \times \Sigma \to \Sigma$ and an element $e \in \Sigma$ such that the following properties hold:
   - e is the neutral element w.r.t. $\bullet$, i.e., for all $\sigma \in \Sigma$ we have $e \bullet \sigma = \sigma = \sigma \bullet e$.
   - $\bullet$ is associative, i.e., for all $\rho, \sigma, \tau \in \Sigma$ we have $(\rho \bullet \sigma) \bullet \tau = \rho \bullet (\sigma \bullet \tau)$.

Here an equation $t_1 = t_2$ between terms t_1, t_2 means that both terms are defined and equal or both terms are undefined.

2. A partial monoid is *cancellative* if $\sigma_1 \bullet \tau = \sigma_2 \bullet \tau \;\Rightarrow\; \sigma_1 = \sigma_2$ for all $\sigma_1, \sigma_2, \tau \in \Sigma$.
3. A *separation algebra* is a partial monoid in which $\bullet$ is commutative and cancellative. It induces a *combinability* relation $\#$ defined by

$$\sigma_0 \,\#\, \sigma_1 \;\Leftrightarrow_{df}\; \sigma_0 \bullet \sigma_1 \text{ is defined .}$$

   In the following, when writing $\sigma \bullet \tau$ for states σ, τ we will implicitly assume $\sigma \,\#\, \tau$.

For a given Petri net the structure $(\mathcal{M}, +, [])$ forms a separation algebra in which the combination operator $\bullet$ is total, i.e, $M \# M'$ holds for all states $M, M' \in \mathcal{M}$. Splitting and combining markings was already part of the semantic definition of the separating conjunction operator $*$ in Section 3. The absence of a proper combinability relation means that there exist no bounds on the capacity of the places, i.e., we are considering unbounded Petri nets.

The operator $\bullet$ is the basis for defining the central connective *separating conjunction* of SL, see below. It allows splitting a resource, e.g., a program state, into disjoint parts about which one can assert separate properties conjunctively. In the case where states are markings, $\bullet$ is just pointwise sum, which is even a total operator.

Definition 4.2 Assume a separation algebra $(\Sigma, \bullet, u)$. A *command* is a relation $R \subseteq \Sigma \times \Sigma$. Relational composition of commands is denoted by ; . Its unit $\mathsf{skip} =_{df} \{(\sigma, \sigma) \,|\, \sigma \in \Sigma\}$ is the identity relation, while the universal relation is denoted by $\top$. A *test* is a command p with $p \subseteq \mathsf{skip}$, i.e., $p = \{(\sigma, \sigma) \,|\, \sigma \in S\}$ for some $S \subseteq \Sigma$. Hence tests are in one-to-one correspondence with subsets of Σ and will be used as an algebraic representation of such subsets. We denote tests by $p, q, r...$ in the sequel. The relative complement of a test p w.r.t. skip is denoted by $\neg p$. As particular tests we define $\mathsf{emp} =_{df} \{(u, u)\}$ that characterises the empty state u , $\ulcorner R =_{df} \{(\sigma, \sigma) \,|\, \exists\, \tau \in \Sigma : \sigma \, R \, \tau\}$ that represents the domain of a command R and dually $R^{\urcorner}$ that denotes the codomain of R, defined analogously. The former is characterised by the universal property

$$\ulcorner R \subseteq q \;\Leftrightarrow\; R \subseteq q \,;R \tag{2}$$

for all tests q. In particular, $R \subseteq \ulcorner R \,;R$ and hence $R = \ulcorner R \,;R$. Moreover, we have $\ulcorner R = (R \,;\top) \cap \mathsf{skip}$ and for relations R, S we have $\ulcorner(R \,;S) = \ulcorner(R \,;\ulcorner S)$. A characterisation for codomain can be given symmetrically.

Note that tests form a Boolean algebra with skip as its greatest and $\emptyset$ as its least element w.r.t. $\subseteq$. Moreover, on tests $\cup$ coincides with join and ; with meet. In particular, tests are idempotent and commute under composition, i.e., $p\,;p = p$ and $p\,;q = q\,;p$.

Using domain and codomain we can define forward and backward modal diamond and box operators. They are given for a command R and test q as follows:

$$\begin{array}{ll} |R\rangle q =_{df} \ulcorner(R\,;q)\,, & |R]q =_{df} \neg\ulcorner(R\,;\neg q)\,, \\ \langle R|q =_{df} (q\,;R)^{\urcorner}\,, & [R|q =_{df} \neg(\neg q\,;R)^{\urcorner}\,. \end{array} \tag{3}$$

Hence $|R\rangle q$ characterises those states for which there exists an execution of R that ends in a state in (the subset represented by) q while $|R]q$ characterises those states for which all executions of R will end in a state in q. The dual statements hold for the backward modal operators.

Next we introduce a relational operator $*$ on commands that corresponds to the separating conjunction of SL. It connects the actions of two commands by "running" them on separate portions of the overall program state; this is indeed expressed by a logical conjunction. Formally,

$$\sigma\ (R * S)\ \tau \ \Leftrightarrow_{df}\ \exists \sigma_1, \sigma_2, \tau_1, \tau_2 : \ \sigma = \sigma_1 \bullet \sigma_2 \ \wedge\ \tau = \tau_1 \bullet \tau_2 \ \wedge\ \sigma_1 \# \sigma_2 \ \wedge\ \tau_1 \# \tau_2 \wedge \\ \sigma_1\ R\ \tau_1 \ \wedge\ \sigma_2\ S\ \tau_2\,.$$

Hence, separated composition of commands can be interpreted as their parallel execution on combinable portions of states [9,10], i.e., $\sigma\,(R * S)\,\tau$ iff σ can be split into states σ_1, σ_2 on which R and S can act and produce results τ_1, τ_2 that are again combinable to $\tau = \tau_1 \bullet \tau_2$. Another interpretation of $R * S$ is that it provides a possibility to characterise the structure of commands, i.e., their behaviour on parts of a state. We will later give a characterisation of a general behaviour of transitions in Petri nets.

Note that for tests p, q the command $p * q$ is also a test and, in particular,

$$\mathsf{skip} * \mathsf{skip} = \mathsf{skip}\ . \tag{4}$$

Additionally, $*$ is associative and commutative and has emp as its unit. Moreover, it distributes through arbitrary unions from both sides.

5 A Command Semantics for Petri Nets

Using the definitions for Petri nets of the previous sections we can now give a denotational model for such nets based on the relational structure from Section 5. Later on we will abstract from the concrete relational setting to a modal Kleene algebraic approach.

Definition 5.1 Given a fixed Petri net, a *net command* is a relation $R \subseteq \mathcal{M} \times \mathcal{M}$, considering markings as states. The set of all net commands is denoted by $\mathcal{C}$. We assign to each formula A the test command

$$[\![A]\!] =_{df} \{(M, M) \,|\, M \models A\}\ ,$$

where validity $\models$ is as in the Kripke semantics of Section 3.

As is well known, by this the logical operators $\vee$ and $\wedge$ correspond to $\cup$ and $\cap$, respectively. Moreover, since $M_1 \# M_2 \Leftrightarrow \mathsf{true}$, we have

$$\begin{aligned}[\![A * B]\!] &= \{(M_1 + M_2, M_1 + M_2) \,|\, M_1 \models A, M_2 \models B\} \\ &= \{(M_1 + M_2, M_1 + M_2) \,|\, M_1 \in [\![A]\!], M_2 \in [\![B]\!]\} \\ &= \{(M_1 + M_2, M_1 + M_2) \,|\, M_1 \in [\![A]\!], M_2 \in [\![B]\!]\,, M_1 \# M_2\} \\ &= [\![A]\!] * [\![B]\!]\,.\end{aligned}$$

A direct consequence of the definition of $*$ is that the reverse exchange law [10], which provides an interplay of relation composition ; and separating conjunction $*$, holds unconditionally in the relational setting of this section. The law reads for any net commands R_1, R_2, S_1, S_2 over a separation algebra with a total combination operator as follows:

$$(R_1 \,;\, R_2) * (S_1 \,;\, S_2) \,\subseteq\, (R_1 * S_1) \,;\, (R_2 * S_2)\,. \tag{5}$$

By interpreting the operator ; as sequential composition and $*$ as interference-free concurrent composition of transitions in Petri nets, one sees that the parallel execution of the sequentially composed transitions $R_1 \,;\, R_2$ and $S_1 \,;\, S_2$ can be reordered to the sequential execution of the parallely composed transitions $R_1 * S_1$ and $R_2 * S_2$. Moreover, in the case of an underlying total separation algebra, the domain and codomain operators distribute over $*$, i.e, we have for relations R, S

$$\ulcorner(R * S) = \ulcorner R * \ulcorner S \quad \text{and} \quad (R * S)^\urcorner = R^\urcorner * S^\urcorner\,. \tag{6}$$

Proofs of (5) and (6) can be found in [10]. We mention that in general only the $\subseteq$ - directions hold for arbitrary separation algebras. An example of a partial separation algebra that does not satisfy the above (in)equations for any relations is given by *safe* Petri nets $\mathcal{N}$ which satisfy for all $M \in \mathcal{M}$ and $p \in P$ the inequation $M(p) \leq 1$ (cf. [4]). In that separation algebra we have $M \# M' \Leftrightarrow \forall\, p \in P : (M + M')(p) \leq 1$. In this work we mainly consider unbounded Petri nets; bounded ones can be handled by imposing a corresponding safety condition. This will be shown in an example later.

Finally, we immediately infer that the modal $\Box_+$ operator can be interpreted as a forward box operator $|\_]$ (cf. Equation (3)) in the relational structure using the defined abstractions:

$$\begin{aligned}
[\![\Box_+ A]\!] &= \{(M,M) \,|\, \forall\, N : M \leadsto^* N \,\Rightarrow\, N \models A\} \\
&= \{(M,M) \,|\, \forall\, N : (M,N) \in\, \leadsto^* \Rightarrow\, (N,N) \in [\![A]\!]\} \\
&= \{(M,M) \,|\, \neg(\exists\, N : (M,N) \in\, \leadsto^* \,\wedge (N,N) \notin [\![A]\!])\} \\
&= \neg\{(M,M) \,|\, \exists\, N : (M,N) \in\, \leadsto^* \,\wedge (N,N) \in \neg[\![A]\!]\} \\
&= \neg\{(M,M) \,|\, \exists\, N : (M,N) \in (\leadsto^* \,;\neg[\![A]\!])\} \\
&= \neg(\leadsto^* \,;\neg[\![A]\!])^\urcorner \\
&= |\leadsto^*][\![A]\!]\,,
\end{aligned}$$

where $|\leadsto^*]$ is the forward box operator associated with the transition relation $\leadsto^*$.

Clearly, by analogous calculations we immediately get

$$[\![\Box_- A]\!] = [\leadsto^*|[\![A]\!]\,, \quad [\![\Diamond_+ A]\!] = |\leadsto^*\rangle[\![A]\!]\,, \quad \text{and} \quad [\![\Diamond_- A]\!] = \langle\leadsto^*|[\![A]\!]\,.$$

Now we turn to a pointfree treatment of transitions.

Definition 5.2 For a transition t we define the semantics $[\![\, t \,]\!]$ to be the one-step reachability relation $[t\rangle$ considered as a net command, i.e.,

$$[\![\, t \,]\!] \;=_{df}\; \{(M,N) \,|\, M[t\rangle N\} = [t\rangle\;.$$

As mentioned in Section 2 after Figure 1, the transitions come with a special behaviour.

Definition 5.3 A net command R is *local* [10] if it satisfies $R * \mathsf{skip} = R$.

Intuitively, this characterises the ability of R to perform its task with a possibly smaller substate of the overall state.

Theorem 5.4 *For every transition t the relational abstraction $[\![\, t \,]\!]$ is local.*

Applying the above intuitive explanation of local commands, we obtain that if t can fire on some marking M then it will also be able to fire on any larger marking $N \succeq M$. Conversely, any execution of an enabled transition t starting from a marking N can be tracked back to a possibly smaller marking $M \preceq N$.

Proof First, we show the $(\supseteq)$ part of the locality equation. By neutrality of emp and isotony,

$$[\![\, t \,]\!] = [\![\, t \,]\!] * \mathsf{emp} \subseteq [\![\, t \,]\!] * \mathsf{skip}\,.$$

For the converse $(\subseteq)$ we calculate

$$\begin{array}{ll}
& (M,N) \in [\![\, t \,]\!] * \mathsf{skip} \\
\Leftrightarrow & \quad \{\![\text{ definition of } * \ ]\!\} \\
& \exists\, M_1, M_2, N_1, N_2 : (M_1, N_1) \in [\![\, t \,]\!] \wedge (M_2, N_2) \in \mathsf{skip} \wedge M = M_1 + M_2 \wedge \\
& N = N_1 + N_2 \\
\Leftrightarrow & \quad \{\![\text{ definition of } \mathsf{skip} \ ]\!\} \\
& \exists\, M_1, M_2, N_1, N_2 : (M_1, N_1) \in [\![\, t \,]\!] \wedge M_2 = N_2 \wedge M = M_1 + M_2 \wedge \\
& N = N_1 + N_2 \\
\Rightarrow & \quad \{\![\text{ logic }]\!\} \\
& \exists\, M_1, M_2, N_1 : (M_1, N_1) \in [\![\, t \,]\!] \wedge M = M_1 + M_2 \wedge N = N_1 + M_2 \\
\Leftrightarrow & \quad \{\![\text{ definition of } [\![\, t \,]\!] \ ]\!\} \\
& \exists\, M_1, M_2, M', N_1 : M_1 = \mathit{pre}(t) + M' \wedge N_1 = \mathit{post}(t) + M' \wedge \\
& M = M_1 + M_2 \wedge N = N_1 + M_2 \\
\Rightarrow & \quad \{\![\text{ logic }]\!\} \\
& \exists\, M_2, M' : M = \mathit{pre}(t) + M' + M_2 \wedge N = \mathit{post}(t) + M' + M_2 \\
\Rightarrow & \quad \{\![\text{ setting } M'' = M' + M_2 \ ]\!\} \\
& \exists\, M'' : M = \mathit{pre}(t) + M'' \wedge N = \mathit{post}(t) + M'' \\
\Leftrightarrow & \quad \{\![\text{ definition of } [\![\, t \,]\!] \ ]\!\} \\
& (M,N) \in [\![\, t \,]\!]\,.
\end{array}$$

□

We will show later in a more abstract setting that locality of a net command lifts to its reflexive and transitive closure.

6 Applying Algebra to Petri Nets

In this section we abstract the net command semantics algebraically to elements of special algebraic structures known as a quantales; these, in turn are special cases of idempotent semirings.

Definition 6.1

1. An *idempotent semiring* is a structure $(A, +, 0, \cdot, 1)$ such that $(A, +, 0)$ is a commutative monoid with idempotent addition, that is, $a + a = a$ for all $a \in A$, $(A, \cdot, 1)$ is a monoid, multiplication distributes over addition, that is, for all $a, b, c \in A$,

   $$a \cdot (b + c) = a \cdot b + a \cdot c \qquad \text{and} \qquad (a + b) \cdot c = a \cdot c + b \cdot c \ ,$$

   and 0 is a left and right annihilator for multiplication, that is, for all $a \in A$,

   $$a \cdot 0 = 0 = 0 \cdot a \ .$$

2. Every idempotent semiring is partially ordered by

   $$a \leq b \Leftrightarrow_{df} a + b = b \ .$$

   Then $+$ and $\cdot$ are isotone w.r.t. $\leq$ and 0 is the least element. Moreover, $a + b$ is the supremum of $a, b \in A$.
3. A semiring is *Boolean* if it has a complement operator $\overline{} : A \to A$ and satisfies Huntington's axiom:

   $$x = \overline{\overline{x} + y} + \overline{\overline{x} + \overline{y}} \ .$$

   In this case one defines the meet operator as

   $$x \sqcap y =_{df} \overline{\overline{x} + \overline{y}} \ .$$

4. A *Kleene algebra* is a structure $(A, +, \cdot, ^*, 0, 1)$ such that $(A, +, \cdot, 0, 1)$ is an idempotent semiring and the star operator * satisfies the unfold and induction laws

   $$1 + a \cdot a^* \leq a^* \ , \qquad 1 + a^* \cdot a \leq a^* \ , \tag{7}$$
   $$c + a \cdot b \leq b \Rightarrow a^* \cdot c \leq b \ , \qquad c + b \cdot a \leq b \Rightarrow c \cdot a^* \leq b \ . \tag{8}$$

   The star here should not be confused with the separation operator $*$ above.
5. An idempotent semiring $(A, +, 0, \cdot, 1)$ is called a *quantale* [34,39] or *standard Kleene algebra* [5] if $\leq$ induces a complete lattice on A and multiplication distributes over arbitrary suprema. The infimum and the supremum of a subset $B \subseteq A$ are denoted by $\bigsqcap B$ and $\bigsqcup B$, respectively. Their binary variants are $a \sqcap b$ and $a \sqcup b$ (the latter coinciding with $a + b$). Every quantale can be made into a Kleene algebra by defining $a^* =_{df} \mu x \, . \, 1 + a \cdot x$, where μ is the least fixed point operator.
6. A *bi-semiring* is a structure $(A, +, 0, \cdot, 1, *, u)$ such that both $(A, +, 0, \cdot, 1)$ and $(A, +, 0, *, u)$ are idempotent semirings with commutative $*$. Note that u here is the abstraction of the command `emp` and not the neutral element of a separation algebra as in Definition 4.1.
7. A *concurrent net semiring* is a bi-semiring in which the operators are connected by the following additional axioms:

   $$1 * 1 \leq 1 \ , \tag{9}$$
   $$(a \cdot b) * (c \cdot d) \leq (a * c) \cdot (b * d) \ . \tag{r-exchange}$$

   The second of these is the generalised form of the reverse exchange law (cf. Equation 5). The first one was stated in equational form in (4); in fact the direction $\geq$ follows from the reverse exchange axiom.

8. A *concurrent net quantale* is a concurrent net semiring in which both component semirings are quantales.

In a concurrent net quantale one could also define an iteration operator w.r.t. to the $*$ composition operator, but we will not need that here.

Our concurrent net semirings/quantales are quite similar to the concurrent semirings/Kleene algebras of [20,19]. The main difference is that our exchange axiom is order-reverse to the one in those structures, which reflects a bias towards relation-like models.

Commands provide a model in the following way.

Theorem 6.2 *The structure* $(\mathcal{C}, \cup, \emptyset, ;, \mathsf{skip}, *, \mathsf{emp})$ *forms a Boolean concurrent net quantale.*

Proof The quantale property of $(\mathcal{C}, \cup, \emptyset, ;, \mathsf{skip})$ is well known (e.g. [40]). The semiring property of $(\mathcal{C}, \cup, \emptyset, *, \mathsf{emp})$ has been shown in [10]. Since there $*$ is defined in terms of ; as

$$R * S = \lhd \,;\, (R \times S) \,;\, \rhd \ ,$$

$\lhd$ and $\rhd$ are constant relations and ; and $\times$ distribute over arbitrary unions, the quantale property of $(\mathcal{C}, \cup, \emptyset, *, \mathsf{emp})$ also holds. The remaining axioms of concurrent net semirings were again shown in [10], as mentioned in Section 2. □

Lemma 6.3 *In every concurrent net semiring the following inequations hold.*

1. $u \le 1$.
2. $a \le a * 1$.

Proof

1. By neutrality of u w.r.t. $*$, neutrality of 1 w.r.t. $\cdot$, reverse exchange and neutrality of u w.r.t. $*$ again,

$$u = u * u = (u \cdot 1) * (1 \cdot u) \le (u * 1) \cdot (1 * u) = 1 \cdot 1 = 1 \ .$$

2. By neutrality of u w.r.t. $*$, Part 1 and isotony,

$$a = a * u \le a * 1 \ .$$

□

We distinguish some special properties.

Definition 6.4 An element a of a concurrent net semiring is called

reflexive	if $1 \le a$,	(reflexivity)
transitive	if $a \cdot a \le a$,	(transitivity)
local	if $a * 1 \le a$.	(locality)

Since the semiring elements are abstractions of relations, we call a reflexive and transitive element a *preorder*. A preorder with locality is called a *transition element*.

By our axioms, 1 is a transition element.

Lemma 6.5 *For a local element a and arbitrary elements b, c the reverse small exchange laws hold:*

1. $b * (c \cdot a) \leq (b * c) \cdot a$,
2. $b * (a \cdot c) \leq a \cdot (b * c)$.

Proof We only give a proof of the first result since the second can be proved analogously. Using neutrality of 1, the reverse exchange law and locality of a, we calculate:

$$b * (c \cdot a) = (b \cdot 1) * (c \cdot a) \leq (b * c) \cdot (1 * a) = (b * c) \cdot a \,.$$

□

As a first application of our algebraic structures we deal with the behaviour of local elements under iteration.

Lemma 6.6 *If an element a of a concurrent net quantale is local then so is a^*.*

Proof We use the principle of least-fixed-point sub-fusion (e.g. [1]): Let $f, g, h : L \to L$ be isotone functions on a complete lattice $(L, \leq)$ with least element 0. Suppose that g is continuous, i.e., preserves suprema of non-empty chains, and assume $g(0) \leq \mu h$. Then

$$g \circ h \leq f \circ g \;\Rightarrow\; g(\mu h) \leq \mu f \;. \tag{10}$$

This allows fusing the application of g into the recursion described by h.

To prove our claim we need to show $a^* * 1 \leq a^*$. We set $f(x) = h(x) = 1 + a \cdot x$ and $g(x) = x * 1$. Then $\mu f = \mu h = a^*$ and our claim is shown if the premises of least-fixed-point sub-fusion are satisfied. By the global assumption g is continuous. Moreover, $g(0) = 0 \leq \mu h$. So it remains to check $g \circ h \leq f \circ g$. We calculate:

$$\begin{array}{ll}
 & g(h(x)) \\
= & \quad \{\!\![\text{ definitions }]\!\!\} \\
 & (1 + a \cdot x) * 1 \\
= & \quad \{\!\![\text{ distributivity of } *]\!\!\} \\
 & 1 * 1 + (a \cdot x) * 1 \\
\leq & \quad \{\!\![\text{ locality of 1 (Equation (9)) }]\!\!\} \\
 & 1 + (a \cdot x) * 1 \\
\leq & \quad \{\!\![\text{ small reverse exchange, since } a \text{ is assumed as local }]\!\!\} \\
 & 1 + a \cdot (x * 1) \\
= & \quad \{\!\![\text{ definitions }]\!\!\} \\
 & f(g(x)) \;.
\end{array}$$

□

Apart from this, the iteration operator of Kleene algebra is used in calculating the transition elements corresponding to concrete nets. However, we will have no need to refer to it in our further proofs, since they all just deploy reflexivity, transitivity and sometimes locality.

7 Tests and Modal Semirings

Next we introduce some further algebraic concepts that abstract tests and the domain/codomain operators $\ulcorner$ and $\urcorner$ as presented in Definition 4.2 and give some useful consequences.

Definition 7.1 A *test* [28,26] in an idempotent semiring A is an element $p \leq 1$ that has a complement relative to 1, i.e., an element $\neg p$ that satisfies $p + \neg p = 1$ and $p \cdot \neg p = 0 = \neg p \cdot p$. The set of tests of A is denoted by $\mathsf{test}(A)$. Test implication is defined by $p \to q \ =_{df}\ \neg p + q$.

It is not hard to show that $\mathsf{test}(A)$ forms a Boolean subalgebra in which $+$ coincides with the binary supremum $\sqcup$ and $\cdot$ with the binary infimum $\sqcap$. We always have $0, 1 \in \mathsf{test}(A)$, with 0 corresponding to the predicate false and 1 to true. In a Boolean semiring, every element $p \leq 1$ is a test with relative complement $\neg p = \overline{p} \sqcap 1$ (e.g. [13]).

Definition 7.2

- A *modal semiring* is a structure $(A, +, 0, \cdot, 1, \ulcorner, \urcorner)$ where $(A, +, 0, \cdot, 1)$ is an idempotent semiring and the operators $\ulcorner, \urcorner : A \to \mathsf{test}(A)$ satisfy the following axioms for arbitrary element a and test p:

$$\begin{array}{lll} a \leq {\ulcorner a} \cdot a\ , & {\ulcorner (p \cdot a)} \leq p\ , & {\ulcorner (a \cdot b)} \ = \ {\ulcorner (a \cdot {\ulcorner b})}\ , \\ a \leq a \cdot {a \urcorner}\ , & {(a \cdot p) \urcorner} \leq p\ , & {(a \cdot b) \urcorner} \ = \ {({a \urcorner} \cdot b) \urcorner}\ . \end{array}$$

- A *modal concurrent net semiring* is a structure $(A, +, 0, \cdot, 1, *, u, \ulcorner, \urcorner)$ in which $(A, +, 0, \cdot, 1, \ulcorner, \urcorner)$ is a modal semiring such that the $*$-distributivity laws for domain and codomain (cf. Equation 6) hold:

$$\ulcorner (a * b) \geq \ulcorner a * \ulcorner b \qquad \text{and} \qquad (a * b)\urcorner \geq a\urcorner * b\urcorner \tag{11}$$

  Since, by the above axioms, (r-exchange) and the above axioms again,

$$a * b \leq (\ulcorner a \cdot a) * (\ulcorner b \cdot b) \leq (\ulcorner a * \ulcorner b) \cdot (a * b) \ \Rightarrow\ \ulcorner (a * b) \leq \ulcorner a * \ulcorner b\ ,$$

  and symmetrically for $\urcorner$, these inequations strengthen to equalities by antisymmetry of $\leq$.
- A *modal concurrent net quantale* is a *modal concurrent net semiring* that forms a concurrent net quantale.
- In any of the above modal structures forward and backward diamond and box operators can be defined as in Equation (3):

$$\begin{array}{ll} |a\rangle p \ =_{df}\ \ulcorner (a \cdot p)\ , & \langle a| p \ =_{df}\ (p \cdot a)\urcorner\ , \\ |a] p \ =_{df}\ \neg |a\rangle \neg p\ , & [a| p \ =_{df}\ \neg \langle a| \neg p\ . \end{array}$$

Forward diamond $|a\rangle$ and backward diamond $|a\rangle$ respectively correspond to the preimage and image operators as discussed after (3) for binary relations. The notation is a combination of standard modal notation and transition relation notation: if one writes $\overset{a}{\mapsto} q$ for the set of all predecessors of q-states under transition relation a and omits the horizontal line so that a "drops to the bottom" one obtains $|a\rangle q$. Symmetrically $p \overset{a}{\mapsto}$ denotes the set of all successors of p-states under a; to make the modal operators compose more easily we flip sides and write $\langle a| q$. The result

$|a]p$ of applying the forward box $|a]$ to a test p is another test that represents the set of all states from which every transition under a leads inevitably into the subset represented by the test p. An analogous interpretation can be given for the backward box. When the direction of the operators does not matter we will write $\langle a\rangle$ and $[a]$ for them.

By Theorem 6.2 and the results mentioned in Section 2 we have the following result.

Theorem 7.3 *The structure* $(\mathcal{C}, \cup, \emptyset, ;, \mathsf{skip}, *, \mathsf{emp}, {}^{\ulcorner}, {}^{\urcorner})$ *forms a modal Boolean concurrent net quantale.*

The modal operators satisfy a rich set of laws; for proofs see [13,33,32,30].

First, De Morgan duality gives the *swapping rules*

$$|a\rangle p \leq |b]q \Leftrightarrow |b\rangle\neg q \leq |a]\neg p\ , \qquad \langle a|p \leq [b|q \Leftrightarrow \langle b|\neg q \leq [a|\neg p\ . \tag{12}$$

They correspond to the Schröder rules of relation algebra.

Strictness of $\cdot$ w.r.t. 0 and De Morgan yield what is known as axiom (M) of modal logic:

$$\langle a\rangle 0 \,=\, 0\ , \qquad [a]1 \,=\, 1\ . \tag{M}$$

By distributivity, the modalities are homomorphic w.r.t. $+$:

$$\langle a+b\rangle p \,=\, \langle a\rangle p + \langle b\rangle p\ , \qquad [a+b]p \,=\, ([a]p)\cdot([b]p)\ . \tag{13}$$

Hence box is antitone and diamond is isotone in the first argument:

$$a \leq b \;\Rightarrow\; \langle a\rangle p \leq \langle b\rangle p \;\wedge\; [a]p \geq [b]p\ . \tag{14}$$

Moreover, both box and diamond are isotone in their second arguments:

$$p \leq q \;\Rightarrow\; \langle a\rangle p \leq \langle a\rangle q \;\wedge\; [a]p \leq [a]q\ . \tag{15}$$

Isotony entails interactions of the operators with subtraction and implication, since every additive endofunction f and every multiplicative endofunction g on a Boolean algebra satisfy, for all elements p and q,

$$f(p) - f(q) \leq f(p-q), \qquad g(p \to q) \leq g(p) \to g(q). \tag{16}$$

Instantiating g with the box operators we obtain the *normality* laws, also known as axiom (K) of modal logic:

$$[a](p \to q) \leq [a](p) \to [a](q)\ . \tag{K}$$

For tests p the forward and backward modalities behave as follows:

$$\langle p\rangle q = p \cdot q\ , \qquad [p]q = p \to q\ . \tag{17}$$

Hence, $\langle 1\rangle = [1]$ is the identity function on tests. Moreover, $\langle 0\rangle p = 0$ and $[0]p = 1$.

Moreover, the modal operators are homomorphic w.r.t. $\cdot$ as well:

$$\langle a\cdot b\rangle p = |a\rangle|b\rangle p\ , \qquad [a\cdot b]p = |a]|b]p\ . \tag{18}$$

We have the exchange laws

$$p \leq |a]q \Leftrightarrow \langle a|p \leq q\ , \qquad p \leq [a|q \Leftrightarrow |a\rangle p \leq q\ , \tag{19}$$

which establish Galois connections between diamonds and boxes.

The Galois connections have interesting consequences. In particular, diamonds (boxes) commute with all existing suprema (infima) of the test algebra.

The modal structures allow abstract and pointfree proofs of a large set of inference rules of the logical approach to Petri nets given in [35,36]. Moreover, they avoid tedious inductions over transition sequences, which is a gain for manual proving as well as for partially automated proof support.

We start with a proof rule stating an interplay between the diamond operator and separating conjunction, i.e.,

$$\frac{}{\Diamond(A * \Diamond B) \vdash \Diamond(A * B)}\,. \qquad (\textsc{Monotonicity})$$

Note that this law holds for both past and future diamond operators $\Diamond_-$ and $\Diamond_+$. Intuitively, in the case of $\Diamond_+$ this rule states that if there is a reachable marking M for which one part satisfies A and a further distinct part that ensures that B is reachable, then one can also reach from M a marking for which A and B hold on distinct parts. The entailment operator $\vdash$ can be interpreted as the partial order $\leq$ of a quantale or, in the case of the concrete relational structure, by the subset inclusion order.

In the abstract setting of a modal concurrent net quantale the above rule can be translated, for a transitive and local element a and tests p, q, into

$$\langle a\rangle(p * \langle a\rangle q) \leq \langle a\rangle(p * q)\,. \qquad (20)$$

We only give a proof for the backward diamond, since the forward case is analogous. By definition of $\langle a|$, p being a test, distributivity of $\_^{\urcorner}$ over $*$, modality, Lemma 6.5, transitivity of a with isotony of $^{\urcorner}$, and definition of $\langle a|$ again:

$$\begin{array}{cl}
 & \langle a|(p * \langle a|q) \\
= & ((p * (q \cdot a)^{\urcorner}) \cdot a)^{\urcorner} \\
= & ((p^{\urcorner} * (q \cdot a)^{\urcorner}) \cdot a)^{\urcorner} \\
= & ((p * (q \cdot a))^{\urcorner} \cdot a)^{\urcorner} \\
= & ((p * (q \cdot a)) \cdot a)^{\urcorner} \\
\leq & ((p * q) \cdot a \cdot a)^{\urcorner} \\
\leq & ((p * q) \cdot a)^{\urcorner} \\
= & \langle a|(p * q)\,.
\end{array}$$

As evidence of adequacy of our algebraic semantics we provide some further validity proofs of the inference rules given in [35,36]. Since they do not mention the $*$ operator, they do not need a concurrent net semiring; a modal semiring is sufficient. Therefore these rules and their proofs apply to a much wider class of structures and temporal logics such as LTL, CTL/CTL* (cf. [14]) or STL [27]. It has been shown in [31] how to give quantale semantics for some of these logics.

Lemma 7.4 *The following proof rules are valid, where $\vdash A$ is short for* true $\vdash A$*:*

$$(\text{R1})\ \frac{\vdash \neg\neg A}{\vdash A}\,, \quad (\text{R2})\ \frac{}{\Box A \vdash A}\,, \quad (\text{R3})\ \frac{\vdash A}{\vdash \Box A}\,, \quad (\text{R4})\ \frac{}{\Box A \vdash \Box\Box A}\,,$$

$$(\text{R5})\ \frac{\Diamond_+ A \vdash B}{A \vdash \Box_- B}\,, \quad (\text{R6})\ \frac{\Diamond_- A \vdash B}{A \vdash \Box_+ B}\,.$$

Proof For the proof it is sufficient to assume that the underlying element a is a preorder, i.e., reflexive and transitive; locality is not needed.

(R1) translates for test p to $1 \leq \neg\neg p \;\Rightarrow\; 1 \leq p$ which is clear, since tests form a Boolean algebra.

For (R2) – (R6) we give all calculations in terms of $|\_]$ and $|\_\rangle$, since the proofs for $[\_|$ and $\langle\_|$ are analogous.

For (R2) we calculate by reflexivity of a, anti-disjunctivity of $|\_]$ in its first argument and isotony $|a]p = |1+a]p = |1]p \cdot |a]p = p \cdot |a]p \leq p$.

Rule (R3) translates into $1 \leq p \;\Rightarrow\; 1 \leq |a]p$. For this we have $1 = |a]1 \leq |a]p$ by (M), isotony and the assumption.

(R4) means $|a]p \leq |a \cdot a]p$, which holds by transitivity of a and antitony of $|\_]$ in its first argument, while (R5) and (R6) follow from the exchange laws in (19).

□

We conclude with an exchange property between diamond and separating conjunction.

Lemma 7.5 *For all elements a, b and tests p, q we have $(\langle a\rangle p) * (\langle b\rangle q) \leq \langle a * b\rangle(p * q)$.*

Proof We show the property for the forward diamond; for the backward one it is symmetric.

$$\begin{array}{ll}
 & (|a\rangle p) * (|b\rangle q) \\
= & \quad \{\![\text{ definition of diamond }]\!\} \\
 & \ulcorner(a \cdot p) * \ulcorner(b \cdot q) \\
= & \quad \{\![\text{ by (11) }]\!\} \\
 & \ulcorner((a \cdot p) * (b \cdot q)) \\
\leq & \quad \{\![\text{ by (r-exchange) and isotony of domain }]\!\} \\
 & \ulcorner((a * b) \cdot (p * q)) \\
= & \quad \{\![\text{ definition of diamond }]\!\} \\
 & |a * b\rangle(p * q) \;.
\end{array}$$

□

8 Further Properties and Characterisations

As further ingredients for the algebraic setting we continue with some pointfree characterisations that describe special classes of assertions in separation logic. Since they have been given in abstract algebraic terms in [7], they can easily be interpreted also in the particular application of the present paper. We start with so-called *intuitionistic* assertions which are closely related to assertions used in the early paper [15]. These authors additionally assumed a downward closure condition w.r.t. reachability on markings, i.e., for assertions A and markings M, N:

$$(N \vdash A \wedge M \leadsto^* N) \Rightarrow M \vdash A\,.$$

This restriction was replaced in [35,36] by the use of modal operators, which makes formulas more expressive than using just standard intuitionistic logic. In the algebraic setting such a closure condition could be stated as $|a\rangle p \leq p$.

In separation logic, local tests play an important role; for historical reasons they are called *intuitionistic assertions* there. Such a test p shows the behaviour that if p holds for some state σ then it is also valid for any larger state $\tau \succeq \sigma$.

A concrete example w.r.t. the running mutex example in Figure 1 can be given by the test $s * 1$ which describes markings where at least the place s is marked. In the concrete case of relations the test $s * 1$ coincides with $\{(M, M) \mid M(s) \geq 1, p \neq s \Rightarrow M(p) \geq 0\}$. Hence, local tests allow an imprecise description of states in the sense that parts of the states may be arbitrary.

Next we consider another class of tests that, contrary to local tests, describe a set of marked places in a precise fashion, i.e., states in which no part can be arbitrary. An algebraic characterisation can be given as follows [4].

Definition 8.1 A test p is called *precise* iff for all $q, r \in \mathsf{test}(S)$

$$p * q \sqcap p * r \;\leq\; p * (q \sqcap r)\,.$$

We formulate the property using $\sqcap$ rather than $\cdot$, since then a generalisation to infinite sets of tests is possible, see (21) below.

Obviously, the above inequation can be strengthened to an equation by isotony of $\sqcap$ and $*$. Moreover, it was shown in [7] that precise tests are closed under $*$. In the above form the property is also called *determinacy*, as known from relation algebras (e.g., [11]). An example of such a test is $c_1 * s * c_2$ which characterises a state where exactly one token is available in the places c_1, c_2 *and* s, whereas the test $c_1 + s + c_2$ is not precise, since it describes states where a token is available in c_1, c_2 *or* s.

Since in a Boolean quantale the test algebra is complete, it is also possible to extend Definition 8.1 to distributivity over arbitrary non-empty infima, like in [4], i.e.,

$$X \neq \emptyset \Rightarrow \bigsqcap\{p * q \mid q \in X\} \leq p \cdot \bigsqcap X\,. \tag{21}$$

We state a useful property of precise tests.

Lemma 8.2 *p is precise iff $p * \neg q \leq \neg(p * q)$ for all tests q.*

A proof can be found in [11]. This lemma gives a characterisation of preciseness using test negation. For precise tests it is therefore possible to state an interaction of separating conjunction and Boolean test negation.

Corollary 8.3 *For precise test p and arbitrary test q we have*

$$\langle a\rangle(p * \neg q) \leq \neg[a](p * q) \quad \textit{and} \quad p * \langle a\rangle\neg q \leq \neg(p * [a]q)\,.$$

Proof First, $\langle a\rangle(p * \neg q) \leq \langle a\rangle\neg(p * q) = \neg\neg\langle a\rangle\neg(p * q) = \neg[a](p * q)$ which follows from Lemma 8.2, Boolean algebra and diamond/box duality. Second, $p * \langle a\rangle\neg q = p * \neg[a]q \leq \neg(p * [a]q)$ holds by Lemma 8.2 and diamond/box duality. □

9 Treating Safety

In the following three sections we show how safety, liveness and fairness can be dealt with in our algebraic setting. For illustration we use again the mutex example (cf. [25]).

First, as a further ingredient of our algebraic approach we introduce a characteristic inequation for the particular test $\neg u$ that represents all non-empty markings. The Petri net model of the modal algebra satisfies the inequation

$$\neg u * \neg u \leq \neg u \tag{non-emp}$$

which means that any composition $M + N$ of non-empty markings M and N is non-empty again. An immediate consequence of this is the following.

Lemma 9.1 $\neg u * 1 = \neg u$ *and* $u \leq \neg(\neg u * \neg u)$.

Proof First, by Boolean algebra, distributivity, neutrality and (non-emp) with the definition of $\leq$,

$$\neg u * 1 = \neg u * (u + \neg u) = \neg u * u + \neg u * \neg u = \neg u + \neg u * \neg u = \neg u \,.$$

The second inequation follows from (non-emp) by contraposition. □

Hence, assuming (non-emp), the test $\neg u$ becomes local. In particular, the empty marking [] is contained in the test $\neg(\neg u * \neg u)$.

In the literature, the test $\neg(\neg e \circ \neg e)$, where $\circ$ denotes a multiplicative operator and e its neutral element, has also been used for the multiplicative operator $\cdot$ in the concrete context of temporal logics [43,31,21], where it is called step . While it is interpreted there by progress in time, it provides a spatial resource interpretation for the application of Petri nets by choosing $*$ and u for $\circ$ and e. Intuitively, the element $\neg(\neg u * \neg u)$ represents the set of all singleton markings M_s for places s together with the empty marking [] . The former are the atoms in the set of markings w.r.t. the submarking order $\preceq$.

Therefore we define $\mathsf{single\_mark} =_{df} \neg u \sqcap \neg(\neg u * \neg u)$ and abstractly characterise sets of such states by tests p with

$$p \leq \mathsf{single\_mark} \,.$$

If we further assume p to be a precise test, it represents in an abstract fashion a state where only a single token is available in one place, which we denote again by p. We write $\mathcal{M}_{sp}$ for the set of such tests and further assume, for arbitrary tests $p \in \mathcal{M}_{sp}$,

$$\neg(p * 1) * \neg(p * 1) \leq \neg(p * 1) \ , \tag{22}$$

$$\mathsf{single\_mark} \leq p + \neg(p * 1) \ . \tag{23}$$

Inequation (22) states that if the place p is not marked in disjoint parts of a state then it is not marked in the whole state. Inequation (23) expresses that in any $\mathsf{single\_mark}$ state every place p is either marked with a single token or unmarked.

As an application of this we can formulate a condition when a net $\mathcal{N}$ is *safe*, i.e., every place p of the net contains at most one token. For this we assume an initial marking of a net $\mathcal{N}$ that we denote by p_0 and a preorder a that abstracts the reflexive transitive closure of the firing relation of the net and define

$$\mathcal{N} \text{ is } \textit{safe} \;\Leftrightarrow_{df}\; \langle a|p_0 \;\le\; \bigsqcap_{q \in \mathcal{M}_{sp}} \neg(q * q * 1)\,.$$

By $\langle a|p_0$ we only consider markings reachable from the initial marking p_0. For every particular test $q \in \mathcal{M}_{sp}$, the composed test $\neg(q * q * 1)$ in the right-hand side excludes any occurrences of two or more markings in the place q. Taking the infimum over all tests $q \in \mathcal{M}_{sp}$ corresponds to a universal quantification.

For an example we consider again the mutex net of Figure 1. We use the initial marking $p_0 =_{df} i_1 * s * i_2$ shown in that figure. Now, the test representing all reachable markings is given by

$$\begin{aligned}\langle a|p_0 = {} & i_1 * s * i_2 + p_1 * s * i_2 + c_1 * i_2 + i_1 * s * p_2 + \\ & i_1 * c_2 + p_1 * s * p_2 + c_1 * p_2 + p_1 * c_2\,.\end{aligned} \tag{24}$$

The states in that sum and the possible transitions between them are depicted in the reachability graph [25] of Figure 2.

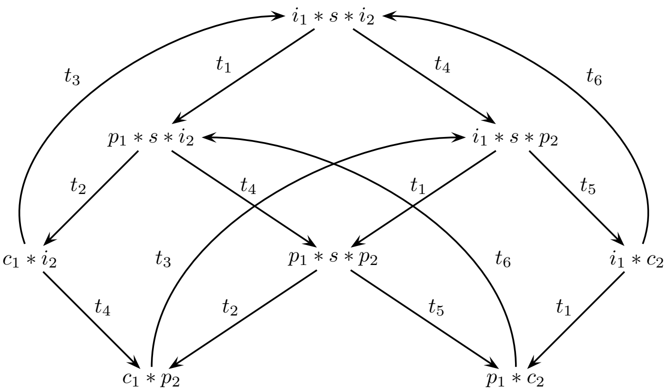

Fig. 2 The reachability graph of the mutex example.

For the mutex net we have $\mathcal{M}_{sp} = \{s, i_1, p_1, c_1, i_2, p_2, c_2\}$. This yields

$$\begin{aligned}\bigsqcap_{q \in \mathcal{M}_{sp}} \neg(q * q * 1) = {} & \neg(s * s * 1) \sqcap \neg(i_1 * i_1 * 1) \sqcap \neg(p_1 * p_1 * 1) \sqcap \\ & \neg(c_1 * c_1 * 1) \sqcap \neg(i_2 * i_2 * 1) \sqcap \neg(p_2 * p_2 * 1) \sqcap \\ & \neg(c_2 * c_2 * 1).\end{aligned}$$

To show safety of the mutex net we use the supremum property of $+$, viz., $p + q \le r \Leftrightarrow p \le r \wedge q \le r$ and the dual one for infima $\sqcap$, viz., $p \le q \sqcap r \Leftrightarrow p \le q \wedge p \le r$ for arbitrary tests p, q, r. Using this we can individually show that each part of the sum in Equation (24) needs to be included in all parts of the meet. We exemplify a part of the calculation showing $i_1 * s * i_2 \le \neg(s * s * 1)$:

$$i_1 * s * i_2 \sqcap s * s * 1 \le s * (i_1 * i_2 \sqcap s * 1)\;.$$

Now, $i_1 * i_2 \leq \neg(s * 1)$, since from Equation (23) with $p = s$, we can infer by distributivity, Boolean algebra and isotony that $\neg s \sqcap \mathsf{single\_mark} \leq \neg(s*1)$. Moreover, $i_1, i_2 \leq \neg s$ and $i_1, i_2 \leq \mathsf{single\_mark}$ imply $i_1, i_2 \leq \neg(s * 1)$. Hence, by Equation (22) and isotony we have $i_1 * i_2 \leq \neg(s * 1) * \neg(s * 1) \leq \neg(s * 1)$. Therefore, $i_1 * s * i_2 \sqcap s * s * 1 \leq 0$ which is equivalent to $i_1 * s * i_2 \leq \neg(s * s * 1)$ by contraposition.

As a further frequently used property of Petri nets we can extend the formulation for safeness to an arbitrary bound k on the places of a net:

$$\mathcal{N} \text{ is } \textit{k-bounded} \quad \Leftrightarrow_{df} \quad \langle a|p_0 \leq \bigsqcap_{q \,\in\, \mathcal{M}_{sp}} \neg(q^{k+1} * 1)\,.$$

Here $q^{k+1} = \underbrace{q * \cdots * q}_{k+1}$, i.e., a $(k+1)$-fold iteration of separating conjunction, and hence $q^{k+1} * 1$ is the set of all markings where q carries *at least* $k+1$ tokens.

For a further proof of correct behaviour in the mutex net (cf. [25]) consider Figure 3. The process Pro_1 enters its critical section by firing transition t_2. Note that Pro_2 cannot enter its critical section even when p_2 is marked, since there is no token available on s. When Pro_1 leaves its critical section, transition t_3 fires and leaves tokens in the places i_1 and s. Note that the whole scenario can symmetrically be applied to Pro_2. In any such state either c_1, c_2 or s is marked, i.e., we cannot reach a state where more than two of these places are marked. This can also be seen in the reachability graph of Figure 2. This behaviour represents an invariant of the net and is required to guarantee that both processes do not enter their critical sections at the same time. Formally, a test p is an *invariant* of an element a if $p \leq |a]p$.

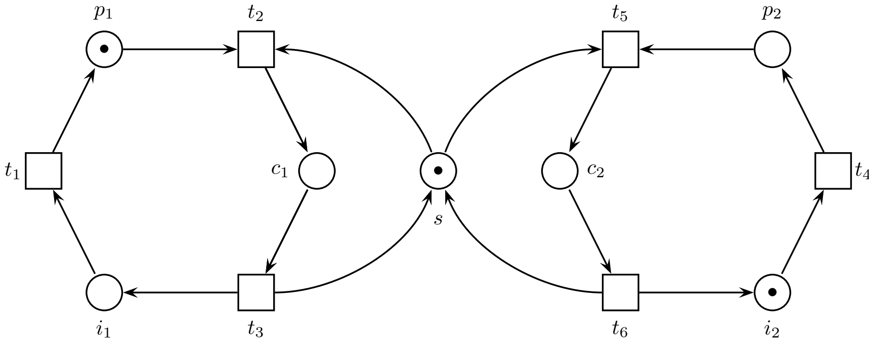

Fig. 3 The process Pro_1 is its pending state and the semaphore s is available.

In a logical fashion we can describe the invariant, that either c_1, c_2 or s is marked, e.g., by the assertion $\Box_+(c_1 \vee s \vee c_2)$ which means that for all states of net only one of the mentioned places is singly-marked. Algebraically this is stated by the inequation

$$\langle a|p_0 \leq |a](c_1 + s + c_2) \tag{25}$$

for a transition element a and initial state p_0 . By the explanation of the modal operators after Definition 7.1, this means that all states reachable from p_0 have

a-transitions only to states where c_1, c_2 or s are marked, and hence satisfy the invariant.

Next we give a proof showing that we cannot reach a state where both c_1 and c_2 are marked.

Lemma 9.2 $|a](c_1 + s + c_2) \leq |a]\neg(c_1 * c_2 * 1)$. *Informally, states which have a-transitions only to states where c_1, c_2 or s are marked are guaranteed to have no a-transitions to states in which at least c_1 and c_2 are marked.*

Proof First, we know that $c_i \leq \neg u$ and $s, c_i \leq \neg(\neg u * \neg u)$. This implies by isotony of $*$ and Lemma 9.1 that $c_1 * c_2 * 1 \leq \neg u * \neg u * 1 = \neg u * \neg u$. By contraposition this is equivalent to $\neg(\neg u * \neg u) \leq \neg(c_1 * c_2 * 1)$. Now, since by assumption $s, c_i \leq \neg(\neg u * \neg u)$, we have that $s, c_i \leq \neg(c_1 * c_2 * 1)$.

By this we can easily infer from isotony of box in its second argument and idempotence of $+$ that $|a](c_1 + s + c_2) \leq |a]\neg(c_1 * c_2 * 1)$. □

Again it is an easy task to infer from this lemma and Equation (25) that

$$\langle a|p_0 \leq |a]\neg(c_1 * c_2 * 1) \ ,$$

meaning that in no reachable state both processes are in their critical sections.

10 Liveness and Fairness

For describing progress and fairness of particular transitions in the mutex net we need a further temporal concept which is called the *leadsto* operator (e.g. [27,37]) and defined for formulas A, B by $A \rhd B = \Box_+(A \rightarrow \Diamond_+ B)$. The corresponding algebraic formulation of the leadsto operator for tests p, q and preorder a is

$$p \rhd q =_{df} |a](p \rightarrow |a\rangle q) \,.$$

This operator again only needs a modal semiring to be well defined; the $*$ operator does not appear in it. Therefore our laws concerning it and their proofs carry over to LTL, $\mathsf{CTL}/\mathsf{CTL}^*$ and STL etc.

We stipulate that $\rhd$ binds weaker than $\cdot$ and $*$.

Before proving properties of $\rhd$ we present two auxiliary results.

Lemma 10.1 *The implication operator $\rightarrow$ on tests (Def. 7.1) is reflexive and transitive and satisfies an exchange law, i.e., for all tests p, q, r, s we have*

$$\begin{gathered} 1 \leq p \rightarrow p \ , \\ (p \rightarrow q) \cdot (q \rightarrow r) \leq (p \rightarrow r) \ , \\ (p \rightarrow q) \cdot (r \rightarrow s) \leq (p + r) \rightarrow (q + s) \,. \end{gathered}$$

Proof Reflexivity is obvious. For transitivity we calculate

$$\begin{aligned} (p \rightarrow q) \cdot (q \rightarrow r) &= (\neg p + q) \cdot (\neg q + r) = \neg p \cdot q + \neg p \cdot r + q \cdot \neg q + \neg q \cdot r \\ &\leq \neg p + \neg p + r = p \rightarrow r \end{aligned}$$

which follows from the definition of $\rightarrow$, distributivity, Boolean algebra and isotony. Using a similar argumentation we infer

$$\begin{aligned} (p \rightarrow q) \cdot (r \rightarrow s) &= (\neg p + q) \cdot (\neg r + s) = \neg p \cdot \neg r + q \cdot \neg r + \neg p \cdot s + q \cdot s \\ &\leq \neg(p + r) + q + s + q = (p + r) \rightarrow (q + s) \,. \end{aligned}$$

□

The following proof principle (see e.g. [27]) will be handy in the next lemma.

Lemma 10.2 *Starting in a state σ that is guaranteed to reach a state p while maintaining q guarantees that from σ a state in $p \cdot q$ can be reached. Formally,*

$$\langle a \rangle p \cdot [a]q \leq \langle a \rangle (p \cdot q) \ .$$

Proof $\langle a \rangle p \cdot [a]q \leq \langle a \rangle (p \cdot q)$

$$\begin{aligned}
\Leftrightarrow \quad & \{\![\text{ tests form a Boolean algebra }]\!\} \\
& \langle a \rangle p \leq \neg [a]q + \langle a \rangle (p \cdot q) \\
\Leftrightarrow \quad & \{\![\text{ def. box }]\!\} \\
& \langle a \rangle p \leq \langle a \rangle (\neg q) + \langle a \rangle (p \cdot q) \\
\Leftrightarrow \quad & \{\![\text{ additivity of diamond }]\!\} \\
& \langle a \rangle p \leq \langle a \rangle (\neg q + p \cdot q) \\
\Leftarrow \quad & \{\![\text{ isotony of diamond }]\!\} \\
& p \leq \neg q + p \cdot q \\
\Leftrightarrow \quad & \{\![\text{ tests form a Boolean algebra }]\!\} \\
& p \cdot q \leq p \cdot q \\
\Leftrightarrow \quad & \{\![\text{ reflexivity of } \leq]\!\} \\
& \mathsf{TRUE} \ .
\end{aligned}$$

□

Lemma 10.3 *The operator $\triangleright$ is reflexive and transitive and satisfies an exchange law, i.e., for all tests p, q, r, s we have*

$$\begin{gathered}
1 \leq p \triangleright p \ , \\
(p \triangleright q) \cdot (q \triangleright r) \leq (p \triangleright r) \ , \\
(p \triangleright q) \cdot (r \triangleright s) \leq (p + r) \triangleright (q + s) \, .
\end{gathered}$$

Proof Reflexivity follows from reflexivity of a, isotony of diamond and (M) via $|a\rangle(\neg p + |a\rangle p) \geq |a](\neg p + p) = |a]1 = 1$. The exchange law follows immediately from (13) and Lemma 10.1.

Transitivity can be shown as follows.

$$\begin{aligned}
& |a](p \rightarrow |a\rangle q) \cdot |a](q \rightarrow |a\rangle r) \\
= \quad & \{\![\ a \text{ a preorder, hence } a = a \cdot a \text{, modality }]\!\} \\
& |a](p \rightarrow |a\rangle q) \cdot |a]|a](q \rightarrow |a\rangle r) \\
= \quad & \{\![\text{ conjunctivity of diamond }]\!\} \\
& |a]((p \rightarrow |a\rangle q) \cdot |a](q \rightarrow |a\rangle r)) \\
= \quad & \{\![\text{ def. } \rightarrow]\!\} \\
& |a]((\neg p + |a\rangle q) \cdot |a](q \rightarrow |a\rangle r)) \\
= \quad & \{\![\text{ distributivity }]\!\} \\
& |a]((\neg p \cdot |a](q \rightarrow |a\rangle r) + |a\rangle q \cdot |a](q \rightarrow |a\rangle r)) \\
\leq \quad & \{\![\ |a](p \rightarrow |a\rangle q) \leq 1, \text{ Lemma 10.2 }]\!\} \\
& |a](\neg p + |a\rangle(q \cdot (q \rightarrow |a\rangle r))) \\
= \quad & \{\![\text{ Boolean algebra }]\!\} \\
& |a](\neg p + |a\rangle(q \cdot |a\rangle r))
\end{aligned}$$

$$
\begin{array}{ll}
\le & \{\![\; q \le 1, \text{isotony of diamond} \;]\!\} \\
 & |a](\neg p + |a\rangle|a\rangle r) \\
= & \{\![\; a \text{ a preorder} \;]\!\} \\
 & |a](\neg p + |a\rangle r) \\
= & \{\![\; \text{def.} \to \;]\!\} \\
 & |a](p \to |a\rangle r)\ .
\end{array}
$$

□

Using $\triangleright$ we are now able to algebraically state the property that the net is fair in the sense that whenever Pro_i has requested the semaphore s, i.e., is pending, it will eventually enter its critical section. This is formalised by

$$\langle a|p_0 \;\le\; (p_i * 1) \triangleright (c_i * 1)\,. \tag{26}$$

Note that we use local tests to ensure that some place is marked, while the remaining part of the state can be characterised imprecisely, since we do not need to impose any further restriction on it. For a proof of Equation (26) we need some further assumptions (cf. [25]) about the behaviour of the mutex net. First, we need to state that whenever one of the places c_i representing the critical sections is marked then also the semaphore will eventually become marked again. This means that neither Pro_1 nor Pro_2 will stay in its critical section forever. Using the $\triangleright$ operator we describe this behaviour algebraically as follows:

$$\langle a|p_0 \le (c_i * 1) \triangleright (s * 1)\,. \tag{27}$$

Next, we infer from the invariant in (25), isotony, definition of $\to$, reflexivity of a and the definition of $\triangleright$,

$$
\begin{aligned}
\langle a|p_0 &\le |a](c_1 + s + c_2) \\
&\le |a](\neg(p_i * 1) + c_1 * 1 + s * 1 + c_2 * 1) \\
&= |a]((p_i * 1) \to (c_1 * 1 + s * 1 + c_2 * 1)) \\
&\le |a]((p_i * 1) \to |a\rangle(c_1 * 1 + s * 1 + c_2 * 1)) \\
&= (p_i * 1) \triangleright (c_1 * 1 + s * 1 + c_2 * 1)\,.
\end{aligned}
$$

Moreover, by idempotence of test w.r.t. $\cdot$, disjunctivity, reflexivity of $\triangleright$ in Corollary 10.3 and the assumptions in (27) we have

$$
\begin{aligned}
\langle a|p_0 &\le (c_1 * 1 \triangleright s * 1) \cdot (c_2 * 1 \triangleright s * 1) \cdot (s * 1 \triangleright s * 1) \\
&= (c_1 * 1 + c_2 * 1 + s * 1) \triangleright (s * 1)\,.
\end{aligned}
$$

In sum, we can further infer from transitivity of $\triangleright$ that

$$\langle a|p_0 \le (p_i * 1) \triangleright (s * 1)\,, \tag{28}$$

which means that any reachable state where at least p_i is marked will lead to a state where s is marked so that the corresponding process is able to enter its critical region.

As a final ingredient to prove (26) we need to additionally assume further behaviour for the mutex net. For this we state that whenever the semaphore s becomes marked then the transition t_2, respectively t_5, will eventually fire and therefore produce a token on c_i . For transition t_2 this is formalized by

$$\langle a|p_0 \le |a](|a\rangle(s * 1) \to |a \cdot t_2\rangle(c_1 * 1))\,. \tag{29}$$

A similar formula can be given for t_5. Note that $a \cdot t_2$ states that finally t_2 will fire, yielding a state that contains at least a token on c_1. From (29) we obtain by antitony of $\to$ in its first argument, reflexivity of a, isotony and transitivity of a, that

$$\langle a|p_0 \leq |a]((s * 1) \to |a\rangle(c_1 * 1)) = (s * 1) \rhd (c_1 * 1)\,. \tag{30}$$

Finally, transitivity of $\rhd$ and Equation (28) show the goal $\langle a|p_0 \leq (p_i * 1) \rhd (c_i * 1)$.

11 Related Work

The concept of locality for transitions in Petri nets has already been discussed in other papers (e.g. [17]). However, algebraic treatments yielding simple and pointfree characterisations have not been widely investigated. A similar abstract approach that builds a formal model for Petri nets based on predicate transformers, i.e., mappings between sets of states, that also introduces a notion of locality, can be found in [44].

A further work where also a Petri net algebra is developed can be found in [3]. That approach basically uses a process algebraic approach to such nets that is called the *Petri Box* calculus. In particular, an abstract approach called the *Box Algebra* is discussed of which the Petri Box calculus and other process algebras can be seen as instances. Compared to that work we rather focus on general algebraic structures involving especially modal operators for reasoning about the concrete application of Petri nets.

In [2,16], a relation-algebraic approach to Petri nets is considered that introduces relational formulas for frequently used properties such as enabledness or liveness in such nets. This allows in particular mechanised reasoning and visualisations by the graphical system RelView. Compared to the present approach formulas become quite complex and difficult to read. Moreover, they are not as general as the formulas provided in the present paper. Mechanisation or automation can be obtained in parts for the first-order fragment of the algebra with theorem proving tools like Prover9/Mace4 [29]. Such tools have already been successfully instantiated for the algebras used here (e.g. [22,23,6]).

12 Conclusion and Outlook

We have shown that algebraic structures like modal concurrent net quantales can be used for abstract reasoning about the behaviour of Petri nets. In particular, we have been able to avoid any inductive arguments about transition sequences in favour of just invoking transitivity, and presented several pointfree formulas that allowed algebraic correctness proofs of inference rules given in the logic of [35,36]. Additionally, we demonstrated practicality of the approach within the example of a standard mutex net in calculating with safety, liveness and fairness properties.

As future work, it would be interesting to investigate concrete connections to the work on relational system support to the analysis of Petri nets in [2,16]. This might yield wider applicability for the present algebraic approach and, in turn, might help to facilitate the relational approach there.

A further interesting topic concerns so-called Signal Transition Graphs (e.g. [42]) which are central to a large part of Walter Vogler's papers. Since such graphs are basically Petri nets, we hope that our modal Petri net algebra can be also applied to such nets.

Finally, it is worth investigating whether the algebraic structures we have introduced can be used for automated proofs using PROVER9/MACE4 or similar systems, along the lines of earlier case studies (e.g. [22,23,8]).

References

1. Aarts, C., Backhouse, R., Boiten, E., Doornbos, H., van Gasteren, N., van Geldrop, R., Hoogendijk, P., Voermans, E., van der Woude, J.: Fixed-point calculus. Information Processing Letters **53**(3), 131–136 (1995)
2. Berghammer, R., von Karger, B., Ulke, C.: Relation-algebraic analysis of Petri nets with RELVIEW. In: T. Margaria, B. Steffen (eds.) Tools and Algorithms for the Construction and Analysis of Systems, *LNCS*, vol. 1055, pp. 49–69. Springer (1996)
3. Best, E., Devillers, R., Koutny, M.: Petri Net Algebra. Monographs in Theoretical Computer Science. Springer (2001). 378 pages
4. Calcagno, C., O'Hearn, P.W., Yang, H.: Local Action and Abstract Separation Logic. In: Proc. of the 22nd Symposium on Logic in Computer Science, pp. 366–378. IEEE Press (2007)
5. Conway, J.H.: Regular Algebra and Finite Machines. Chapman & Hall (1971)
6. Dang, H.H., Höfner, P.: First-order theorem prover evaluation w.r.t. relation- and Kleene algebra. In: R. Berghammer, B. Möller, G. Struth (eds.) Relations and Kleene Algebra in Computer Science — PhD Programme at RelMiCS 10/AKA 05, no. 2008-04 in Technical Report, pp. 48–52. Institut für Informatik, Universität Augsburg (2008)
7. Dang, H.H., Höfner, P., Möller, B.: Algebraic Separation Logic. Journal of Logic and Algebraic Programming **80**(6), 221–247 (2011)
8. Dang, H.H., Möller, B.: Simplifying pointer Kleene algebra. In: P. Höfner, A. McIver, G. Struth (eds.) ATE, *CEUR Workshop Proceedings*, vol. 760, pp. 20–29. CEUR-WS.org (2011)
9. Dang, H.H., Möller, B.: Reverse Exchange for Concurrency and Local Reasoning. In: J. Gibbons, P. Nogueira (eds.) 11th Intl. Conf. on Mathematics of Program Construction, *LNCS*, vol. 7342, pp. 177–197. Springer (2012)
10. Dang, H.H., Möller, B.: Concurrency and Local Reasoning under Reverse Exchange. Science of Computer Programming **85, Part B**, 204–223 (2014). Special Issue on Mathematics of Program Construction 2012
11. Desharnais, J., Möller, B.: Characterizing determinacy in Kleene algebra. Information Sciences **139**, 253–273 (2001)
12. Desharnais, J., Möller, B., Struth, G.: Modal Kleene algebra and applications — A survey. Journal of Relational Methods in Computer Science **1**, 93–131 (2004)
13. Desharnais, J., Möller, B., Struth, G.: Kleene algebra with domain. ACM Transactions on Computational Logic **7**(4), 798–833 (2006)
14. Emerson, E.: Temporal and modal logic, pp. 995–1072. Elsevier (1991)
15. Engberg, U., Winskel, G.: Petri Nets as Models of Linear Logic. In: A. Arnold (ed.) Proceedings of the 15th Colloquium on Trees in Algebra and Programming (CAAP '90), *LNCS*, vol. 431, pp. 147–161. Springer (1990)
16. Fronk, A., Kehden, B.: State space analysis of Petri nets with relation-algebraic methods. Journal of Symbolic Computation **44**(1), 15–47 (2009)
17. Girault, C., Valk, R.: Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications. Springer (2003)
18. Gold, R., Vogler, W.: Quality criteria for partial order semantics of place/transition-nets with capacities. Fundam. Inform. **17**(3), 187–209 (1992)
19. Hoare, C.A.R., Hussain, A., Möller, B., O'Hearn, P., Petersen, R., Struth, G.: On Locality and the Exchange Law for Concurrent Processes. In: J.P. Katoen, B. König (eds.) CONCUR 2011, *LNCS*, vol. 6901, pp. 250–264. Springer (2011)
20. Hoare, C.A.R., Möller, B., Struth, G., Wehrman, I.: Concurrent Kleene Algebra and its Foundations. Journal of Logic and Algebraic Programming **80**(6), 266–296 (2011)
21. Höfner, P.: Algebraic Calculi for Hybrid Systems. Ph.D. thesis, Universitt Augsburg (2009)

22. Höfner, P., Struth, G.: Automated reasoning in Kleene algebra. In: F. Pfenning (ed.) Automated Deduction — CADE-21, *Lecture Notes in Artificial Intelligence*, vol. 4603, pp. 279–294. Springer (2007)
23. Höfner, P., Struth, G.: On automating the calculus of relations. In: A. Armando, P. Baumgartner, G. Dowek (eds.) Automated Reasoning (IJCAR 2008), *LNCS*, vol. 5159, pp. 50–66. Springer (2008)
24. Khomenko, V., Schäfer, M., Vogler, W., Wollowski, R.: STG decomposition strategies in combination with unfolding. Acta Inf. **46**(6), 433–474 (2009)
25. Kindler, E., Walter, R.: Mutex needs fairness. Information Processing Letters **62**(1), 31–39 (1997)
26. Kozen, D.: Kleene algebra with tests. ACM Transactions on Programming Languages and Systems **19**(3), 427–443 (1997)
27. Lamport, L.: The Temporal Logic of Actions. ACM Transactions on Programming Languages and Systems **16**(3), 872–923 (1994)
28. Manes, E., Benson, D.: The inverse semigroup of a sum-ordered semiring. Semigroup Forum **31**, 129–152 (1985)
29. McCune, W.: Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9`
30. Möller, B.: Modal knowledge and game semirings. Comput. J. **56**(1), 53–69 (2013)
31. Möller, B., Höfner, P., Struth, G.: Quantales and temporal logics. In: M. Johnson, V. Vene (eds.) Algebraic Methodology and Software Technology, *LNCS*, vol. 4019, pp. 263–277. Springer (2006)
32. Möller, B., Struth, G.: Algebras of modal operators and partial correctness. Theoretical Computer Science **351**(2), 221–239 (2006)
33. Möller, B., Struth, G.: WP is WLP. In: W. MacCaull, M. Winter, I. Düntsch (eds.) Relational Methods in Computer Science, *Lecture Notes in Computer Science*, vol. 3929, pp. 200–211. Springer (2006)
34. Mulvey, C.: &. Rendiconti del Circolo Matematico di Palermo **12**(2), 99–104 (1986)
35. Pym, D.J.: The Semantics and Proof Theory of the Logic of Bunched Implications. No. 26 in Applied Logic Series. Kluwer (2002). Errata and remarks (Pym 2008) maintained at http://homepages.abdn.ac.uk/d.j.pym/pages/BI-monograph-errata.pdf
36. Pym, D.J., O'Hearn, P.W., Yang, H.: Possible Worlds and Resources: The Semantics of BI. Theoretical Computer Science **315**(1), 257–305 (2004)
37. Reisig, W.: Elements of distributed algorithms: modeling and analysis with Petri nets. Springer (1998)
38. Reynolds, J.C.: Separation logic: A logic for shared mutable data structures. In: Proc. of the 17th Annual IEEE Symposium on Logic in Computer Science, pp. 55–74. IEEE Computer Society (2002)
39. Rosenthal, K.: Quantales and their Applications, *Pitman Research Notes in Mathematics Series*, vol. 234. Longman Scientific & Technical (1990)
40. Schmidt, G., Ströhlein, T.: Relations and Graphs: Discrete Mathematics for Computer Scientists. Springer (1993)
41. Vogler, W.: Executions: A new partial-order semantics of petri nets. Theor. Comput. Sci. **91**(2), 205–238 (1991)
42. Vogler, W., Wollowski, R.: Decomposition in Asynchronous Circuit Design. In: J. Cortadella, A. Yakovlev, G. Rozenberg (eds.) Concurrency and Hardware Design, *LNCS*, vol. 2549, pp. 152–190. Springer (2002)
43. Von Karger, B.: Temporal Algebra. Mathematical Structures in Computer Science **8**(3), 277–320 (1998)
44. Yang, H., O'Hearn, P.W.: A Semantic Basis for Local Reasoning. In: M. Nielsen, U. Engberg (eds.) Foundations of Software Science and Computation Structures, Proceedings FOSSACS 2002, *LNCS*, vol. 2303, pp. 402–416. Springer (2002)