On the Expressiveness of Spatial Constraint Systems*

Michell Guzmán[1] and Frank D. Valencia[2]

1 **INRIA-LIX École Polytechnique, Paris, France**
guzman@lix.polytechnique.fr

2 **CNRS-LIX École Polytechnique, Paris, France**
frank.valencia@lix.polytechnique.fr

Abstract

In this paper we shall report on our progress using spatial constraint system as an abstract representation of modal and epistemic behaviour. First we shall give an introduction as well as the background to our work. Then, we present our preliminary results on the representation of modal behaviour by using spatial constraint systems. Then, we present our ongoing work on the characterization of the epistemic notion of knowledge. Finally, we discuss about the future work of our research.

1998 ACM Subject Classification F.4.1 Mathematical Logic

Keywords and phrases Epistemic logic, Modal logic, Constraint systems, Concurrent constraint programming.

Digital Object Identifier 10.4230/OASIcs.ICLP.2016.16

1 Introduction

Epistemic, mobile and spatial behaviour are common practice in today's distributed systems. The intrinsic *epistemic* nature of these systems arises from social behaviour. Most people are familiar with digital systems where users share their *beliefs*, *opinions* and even intentional *lies* (hoaxes). Also, systems modeling decision behaviour must account for those decisions dependance on the results of interactions with others within some social context. Spatial and mobile behaviour is exhibited by applications and data moving across (possibly nested) spaces defined by, for example, friend circles, groups, and shared folders. We therefore believe that a solid understanding of the notion of *space* and *spatial mobility* as well as the flow of epistemic information is relevant in many models of today's distributed systems.

Constraint systems (cs's) provide the basic domains and operations for the semantic foundations of the family of *formal declarative models* from concurrency theory known as *concurrent constraint programming (ccp)* process calculi [15]. *Spatial constraint systems* [9] (scs) are algebraic structures that extend cs for reasoning about basic spatial and epistemic behaviour such as *extrusion* and *belief*. Both spatial and epistemic assertions can be viewed as specific

* This work was partially supported by INRIA-CORDI.

Technical Communications of the 32nd International Conference on Logic Programming (ICLP 2016).
Editors: Manuel Carro, Andy King, Neda Saeedloei, and Marina De Vos; Article No. 16; pp. 16:1–16:13
Open Access Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

modalities. Other modalities can be used for assertions about time, knowledge and other concepts used in the specification and verification of concurrent systems.

The main goal of this PhD project is the study of the expressiveness of spatial constraint systems in the broader perspective of modal behaviour. In this summary, we shall show that spatial constraint systems are sufficiently robust to capture other modalities and to derive new results for modal logic. We shall also discuss our future work on extending constraint systems to express fundamental epistemic behaviour such as *knowledge* and *distributed knowledge*.

This summary is structured as follows: In Section 2 we give some background. In Section 3 we present our results with applications to modal logic. In Sections 4 and 5 we describe ongoing work and future work for the remaining part of this PhD project. The results in this summary have been recently published as [7, 6].

2 Background

In this section we recall the notion of basic constraint system [3] and the more recent notion of spatial constraint system [9]. We presuppose basic knowledge of order theory and modal logic [1, 14, 5, 2].

The concurrent constraint programming model of computation [15] is parametric in a *constraint system* (cs) specifying the structure and interdependencies of the partial information that computational agents can ask of and post in a *shared store*. This information is represented as *assertions* traditionally referred to as *constraints*.

Constraint systems can be formalized as *complete algebraic lattices* [3][1]. pThe elements of the lattice, the *constraints*, represent (partial) information. A constraint c can be viewed as an *assertion* (or a *proposition*). The lattice order $\sqsubseteq$ is meant to capture entailment of information: $c \sqsubseteq d$, alternatively written $d \sqsupseteq c$, means that the assertion d represents as much information as c. Thus, we may think of $c \sqsubseteq d$ as saying that d *entails* c or that c can be *derived* from d. The *least upper bound (lub)* operator $\sqcup$ represents join of information; $c \sqcup d$, the least element in the underlying lattice above c and d. Thus, $c \sqcup d$ can be seen as an assertion stating that both c and d hold. The top element represents the lub of all, possibly inconsistent, information, hence it is referred to as *false*. The bottom element *true* represents the empty information.

▶ **Definition 1** (Constraint Systems [3]). A constraint system (cs) $\mathbf{C}$ is a complete algebraic lattice $(Con, \sqsubseteq)$. The elements of Con are called *constraints*. The symbols $\sqcup$, *true* and *false* will represent the least upper bound (lub) operation, the bottom, and the top element of $\mathbf{C}$, respectively.

We shall use the following notions and notations from order theory.

▶ **Notation 1** (Lattices and Limit Preservation). *Let $\mathbf{C}$ be a partially ordered set (poset) $(Con, \sqsubseteq)$. We shall use $\bigsqcup S$ to denote the least upper bound (lub) (or* supremum *or* join*) of the elements in S, and $\bigsqcap S$ is the greatest lower bound (glb) (*infimum *or* meet*) of the elements in S. We say that $\mathbf{C}$ is a* complete lattice *iff each subset of Con has a supremum*

[1] An alternative syntactic characterization of cs, akin to Scott information systems, is given in [15].

and an infimum in Con*. A non-empty set* $S \subseteq Con$ *is* directed *iff every* finite *subset of* S *has an upper bound in* S*. Also,* $c \in Con$ *is* compact *iff for any directed subset* D *of* Con*,* $c \sqsubseteq \bigsqcup D$ *implies* $c \sqsubseteq d$ *for some* $d \in D$*. A complete lattice* $\mathbf{C}$ *is said to be* algebraic *iff for each* $c \in Con$*, the set of compact elements below it forms a directed set and the lub of this directed set is* c*. A self-map on* Con *is a function* $f : Con \to Con$*. Let* $(Con, \sqsubseteq)$ *be a complete lattice. The self-map* f *on* Con preserves *the supremum of a set* $S \subseteq Con$ *iff* $f(\bigsqcup S) = \bigsqcup\{f(c) \mid c \in S\}$*. The preservation of the infimum of a set is defined analogously. We say* f *preserves finite/infinite suprema iff it preserves the supremum of arbitrary finite/infinite sets. Preservation of finite/infinite infima is defined similarly.*

2.1 Spatial Constraint Systems

The authors of [9] extended the notion of cs to account for distributed and multi-agent scenarios where agents have their own space for their local information and performing their computations.

Intuitively, each agent i has a *space* function $[\cdot]_i$ from constraints to constraints. We can think of $[c]_i$ as an assertion stating that c is a piece of information residing *within a space attributed to agent* i. An alternative *epistemic logic* interpretation of $[c]_i$ is an assertion stating that agent i *believes* c or that c holds within the space of agent i (but it may not hold elsewhere). Similarly, $[[c]_j]_i$ is a hierarchical spatial specification stating that c holds within the local space the agent i attributes to agent j. Nesting of spaces can be of any depth. We can think of a constraint of the form $[c]_i \sqcup [d]_j$ as an assertion specifying that c and d hold within two *parallel/neighboring* spaces that belong to agents i and j, respectively.

▶ **Definition 2** (Spatial Constraint System [9]). An n-agent *spatial constraint system (n-scs)* $\mathbf{C}$ is a cs $(Con, \sqsubseteq)$ equipped with n self-maps $[\cdot]_1, \ldots, [\cdot]_n$ over its set of constraints Con such that:

S.1 $[true]_i = true$, and

S.2 $[c \sqcup d]_i = [c]_i \sqcup [d]_i$ for each $c, d \in Con$.

Axiom S.1 requires $[\cdot]_i$ to be strict map (i.e bottom preserving). Intuitively, it states that having an empty local space amounts to nothing. Axiom S.2 states that the information in a given space can be distributed. Notice that requiring S.1 and S.2 is equivalent to requiring that each $[\cdot]_i$ preserves *finite suprema*. Also, S.2 implies that $[\cdot]_i$ is monotonic: I.e., if $c \sqsupseteq d$ then $[c]_i \sqsupseteq [d]_i$.

2.2 Extrusion and utterance

We can also equip each agent i with an *extrusion* function $\uparrow_i : Con \to Con$. Intuitively, within a space context $[\cdot]_i$, the assertion $\uparrow_i c$ specifies that c must be posted outside of (or extruded from) agent i's space. This is captured by requiring the *extrusion* axiom $[\, \uparrow_i c\,]_i = c$. In other words, we view *extrusion/utterance* as the right inverse of *space/belief* (and thus space/belief as the left inverse of extrusion/utterance).

▶ **Definition 3** (Extrusion). Given an n-scs $(Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$, we say that $\uparrow_i$ is an extrusion function for the space $[\cdot]_i$ iff $\uparrow_i$ is a right inverse of $[\cdot]_i$, i.e., iff $[\, \uparrow_i c\,]_i = c$.

2.3 The Extrusion/Right Inverse Problem

A legitimate question is: Given space $[\cdot]_i$ can we derive an extrusion function $\uparrow_i$ for it ? From set theory we know that there is an extrusion function (i.e., a right inverse) $\uparrow_i$ for $[\cdot]_i$ iff $[\cdot]_i$ is *surjective*. Recall that the *pre-image* of $y \in Y$ under $f : X \to Y$ is the set $f^{-1}(y) = \{x \in X \mid y = f(x)\}$. Thus, $\uparrow_i$ can be defined as a function, called *choice* function, that maps each element c to some element from the pre-image of c under $[\cdot]_i$.

3 Preliminary Results

In this part of the summary we shall describe the work we have achieved so far. It is based on the paper [7] recently accepted for publication.

3.1 Modalities in Terms of Space

Modal logics [14] extend classical logic to include operators expressing modalities. Depending on the intended meaning of the modalities, a particular modal logic can be used to reason about space, knowledge, belief or time, among others. Although the notion of spatial constraint system is intended to give an algebraic account of spatial and epistemic assertions, we shall show that it is sufficiently robust to give an algebraic account of more general modal assertions.

The aim of this part of the summary is the study of the *extrusion problem* for a meaningful family of scs's that can be used as semantic structures for modal logics. They are called *Kripke spatial constraint systems* because its elements are *Kripke Structures* (KS's). KS's can be seen as transition systems with some additional structure on their states.

3.2 Constraint Frames and Normal Self Maps

Spatial constraint systems can be used, by building upon ideas from Geometric Logic and Heyting Algebras [16], as semantic structures for modal logic. We shall give an algebraic characterization of the concept of normal modality as maps preserving finite suprema.

First, recall that a *Heyting implication* $c \to d$ in our setting corresponds to the *weakest constraint* one needs to join c with to derive d: The greatest lower bound (glb) $\bigsqcap\{e \mid e \sqcup c \sqsupseteq d\}$. Similarly, the negation of a constraint c, written $\sim c$, can be seen as the *weakest constraint inconsistent* with c, i.e., the glb $\bigsqcap\{e \mid e \sqcup c \sqsupseteq \mathit{false}\} = c \to \mathit{false}$.

▶ **Definition 4** (Constraint Frames). A constraint system $(Con, \sqsubseteq)$ is said to be a *constraint frame* iff its joins distribute over arbitrary meets: More precisely, $c \sqcup \bigsqcap S = \bigsqcap\{c \sqcup e \mid e \in S\}$ for every $c \in Con$ and $S \subseteq Con$. Given a constraint frame $(Con, \sqsubseteq)$ and $c, d \in Con$, define Heyting implication $c \to d$ as $\bigsqcap\{e \in Con \mid c \sqcup e \sqsupseteq d\}$ and Heyting negation $\sim c$ as $c \to \mathit{false}$.

In modal logics one is often interested in *normal modal* operators. The formulae of a modal logic are those of propositional logic extended with modal operators. Roughly speaking, a modal logic operator m is normal iff (1) the formula $\mathrm{m}(\phi)$ is a theorem (i.e., true in all models for the underlying modal language) whenever the formula ϕ is a theorem, and (2) the implication formula $\mathrm{m}(\phi \Rightarrow \psi) \Rightarrow (\mathrm{m}(\phi) \Rightarrow \mathrm{m}(\psi))$ is a theorem. Thus, using Heyting implication, we can express the normality condition in constraint frames as follows.

▶ **Definition 5** (Normal Maps). Let $(Con, \sqsubseteq)$ be a constraint frame. A self-map m on Con is said to be *normal* if (1) $m(true) = true$ and (2) $m(c \to d) \to (m(c) \to m(d)) = true$ for each $c, d \in Con$.

The next theorem basically states that Condition (2) in Definition 5 is equivalent to the seemingly simpler condition: $m(c \sqcup d) = m(c) \sqcup m(d)$.

▶ **Theorem 6** (Normality & Finite Suprema). *Let* $\mathbf{C}$ *be a constraint frame* $(Con, \sqsubseteq)$ *and let* f *be a self-map on* Con*. Then* f *is normal if and only if* f *preserves finite suprema.*

By applying the above theorem, we can conclude that space functions from constraint frames are indeed normal self-maps, since they preserve finite suprema.

3.3 Extrusion Problem for Kripke Constraint Systems

In this section we will study the extrusion/right inverse problem for a meaningful family of spatial constraint systems (scs's), the Kripke scs. In particular, we shall derive and give a *complete* characterization of normal extrusion functions as well as identify the *weakest* condition on the elements of the scs under which extrusion functions may exist. To illustrate the importance of this study, let us give some intuition first.

Kripke structures (KS) are a fundamental mathematical tool in logic and computer science. They can be seen as transition systems and they are used to give semantics to modal logics. Formally, a KS can be defined as follows.

▶ **Definition 7** (Kripke Structures). An n-agent Kripke Structure (KS) M over a set of atomic propositions Φ is a tuple $(S, \pi, \mathcal{R}_1, \ldots, \mathcal{R}_n)$ where S is a nonempty set of states, $\pi : S \to (\Phi \to \{0, 1\})$ is an interpretation associating with each state a truth assignment to the primitive propositions in Φ, and $\mathcal{R}_i$ is a binary relation on S. A *pointed KS* is a pair (M, s) where M is a KS and s, called the *actual world*, is a state of M. We write $s \xrightarrow{i}_M t$ to denote $(s, t) \in \mathcal{R}_i$. ◀

We now define the Kripke scs wrt a set $\mathcal{S}_n(\Phi)$ of pointed KS.

▶ **Definition 8** (Kripke Spatial Constraint Systems [9]). Let $\mathcal{S}_n(\Phi)$ be a non-empty set of n-agent Kripke structures over a set of primitive propositions Φ. We define the Kripke n-scs for $\mathcal{S}_n(\Phi)$ as $\mathbf{K}(\mathcal{S}_n(\Phi)) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ where $Con = \mathcal{P}(\Delta)$, $\sqsubseteq \, = \, \supseteq$, and

$$[c]_i \stackrel{\text{def}}{=} \{(M, s) \in \Delta \mid \triangleright_i(M, s) \subseteq c\}$$

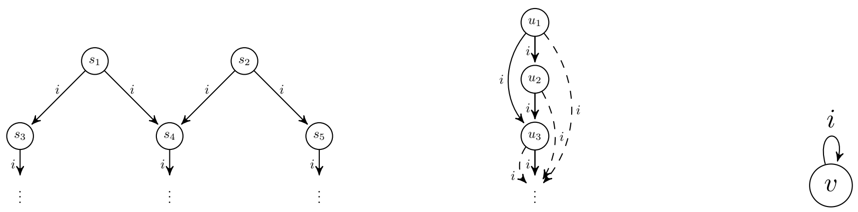
where Δ is the set of all pointed Kripke structures (M, s) such that $M \in \mathcal{S}_n(\Phi)$ and $\triangleright_i(M, s) = \{(M, t) \mid s \xrightarrow{i}_M t\}$ denotes the pointed KS reachable from (M, s).

The structure $\mathbf{K}(\mathcal{S}_n(\Phi)) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ is a complete algebraic lattice given by a powerset ordered by reversed inclusion $\supseteq$. The join $\sqcup$ is set intersection, the meet $\sqcap$ is set union, the top element *false* is the empty set $\emptyset$, and bottom *true* is the set Δ of all pointed Kripke structures (M, s) with $M \in \mathcal{S}_n(\Phi)$. Notice that $\mathbf{K}(\mathcal{S}_n(\Phi))$ is a frame since meets are unions and joins are intersections so the distributive requirement is satisfied. Furthermore, each $[\cdot]_i$ preserves arbitrary suprema (intersection) and thus, from Theorem 6 it is a normal self-map.

3.4 Existence of Right Inverses

We shall now address the question of whether a given Kripke constraint system can be extended with extrusion functions. We shall identify a sufficient and necessary condition on accessibility relations for the existence of an extrusion function $\uparrow_i$ given the space $[\cdot]_i$.

▶ **Definition 9** (Determinacy and Unique-Determinacy)**.** Let S and $\mathcal{R}$ be the set of states and an accessibility relation of a KS M, respectively. Given $s, t \in S$, we say that s *determines* t wrt $\mathcal{R}$ if $(s,t) \in \mathcal{R}$. We say that s *uniquely determines* t wrt $\mathcal{R}$ if s is the *only state* in S that determines t wrt $\mathcal{R}$. A state $s \in S$ is said to be *determinant* wrt $\mathcal{R}$ if it uniquely determines some state in S wrt $\mathcal{R}$. Furthermore, $\mathcal{R}$ is *determinant-complete* if every state in S is determinant wrt $\mathcal{R}$. ◀



(i) M_1 **(ii)** M_2 **(iii)** M_3

Figure 1 Accessibility relations for an agent i. In each sub-figure we omit the corresponding KS M_k from the edges and draw $s \xrightarrow{i} t$ whenever $s \xrightarrow{i}_{M_k} t$.

▶ **Example 10.** Figure 1 illustrates some determinant-complete accessibility relations. Figures 1.(i) and 1.(iii) are determinant-complete accessibility relations. Figure 1.(ii) shows a non determinant-complete accessibility relation (the transitive closure of an infinite line structure).

▶ **Notation 2.** *We write* $s \overset{i}{\twoheadrightarrow}_M t$ *for* s uniquely determines t *wrt* $\xrightarrow{i}_M$.

The following theorem provides a complete characterization, in terms of classes of KS, of the existence of right inverses for space functions.

▶ **Theorem 11** (Completeness)**.** *Let* $[\cdot]_i$ *be a spatial function of a Kripke scs* $\mathbf{K}(\mathcal{S})$*. Then* $[\cdot]_i$ *has a right inverse iff for every* $M \in \mathcal{S}$ *the accessibility relation* $\xrightarrow{i}_M$ *is determinant-complete.*

Henceforth we use $\mathcal{M}^{\mathsf{D}}$ to denote the class of KS's whose accessibility relations are determinant-complete. It follows from Theorem 11 that $\mathcal{S} = \mathcal{M}^{\mathsf{D}}$ is the largest class for which space functions of a Kripke scs $\mathbf{K}(\mathcal{S})$ have right inverses.

3.5 Right Inverse Constructions

Let $\mathbf{K}(\mathcal{S}) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be a Kripke scs. The Axiom of Choice and Theorem 11 tell us that each $[\cdot]_i$ has a right inverse (extrusion function) if and only if $\mathcal{S} \subseteq \mathcal{M}^{\mathsf{D}}$. We are interested, however, in explicit constructions of the right inverses.

Since any Kripke scs space function preserves arbitrary suprema, we obtain the following canonical greatest right-inverse construction. Recall that the pre-image of c under $[\cdot]_i$ is given by the set $[c]_i^{-1} = \{d \mid c = [d]_i\}$.

▶ **Definition 12** (Max Right Inverse). Let a Kripke scs $\mathbf{K}(\mathcal{S}) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be defined over $\mathcal{S} \subseteq \mathcal{M}^{\mathtt{D}}$. We define $\uparrow_i^{\mathtt{M}}$ as the following self-map on Con: $\uparrow_i^{\mathtt{M}} : c \mapsto \bigsqcup [c]_i^{-1}$.

Then $\uparrow_i^{\mathtt{M}}$ is a right inverse for $[\cdot]_i$, and from its definition it is clear that $\uparrow_i^{\mathtt{M}}$ is the greatest right inverse of $[\cdot]_i$ wrt $\sqsubseteq$. However, $\uparrow_i^{\mathtt{M}}$ is not necessarily *normal* in the sense of Definition 5.

In what follows we shall identify right inverse constructions that are normal.

3.6 Normal Right Inverses

The following central lemma provides distinctive properties of any normal right-inverse.

▶ **Lemma 13.** *Let $\mathbf{K}(\mathcal{S}) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be the Kripke scs over $\mathcal{S} \subseteq \mathcal{M}^{D}$. Suppose that f is a normal right-inverse of $[\cdot]_i$. Then for every $M \in \mathcal{S}$, $c \in Con$: (i) $\triangleright_i(M,s) \subseteq f(c)$ if $(M,s) \in c$, (ii) $\{(M,t)\} \subseteq f(c)$ if t is multiply determined wrt $\xrightarrow{i}_M$, and (iii) $true \subseteq f(true)$.*

The above property tell us what sets should necessarily be included in every $f(c)$ if f is to be both normal and a right-inverse of $[\cdot]_i$.

In fact, the least self-map f wrt $\subseteq$, i.e., the greatest one wrt the lattice order $\sqsubseteq$, satisfying Conditions 1, 2 and 3 in Lemma 13 is indeed a normal right-inverse. We call such a function the *max normal right-inverse* $\uparrow_i^{\mathtt{MN}}$ and it is given below.

▶ **Definition 14** (Max Normal-Right Inverse). Let $\mathbf{K}(\mathcal{S}) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be a Kripke scs over $\mathcal{S} \subseteq \mathcal{M}^{\mathtt{D}}$. We define the *max normal right-inverse* for agent i, $\uparrow_i^{\mathtt{MN}}$ as the following self-map on Con:

$$\uparrow_i^{\mathtt{MN}}(c) \stackrel{\mathtt{def}}{=} \begin{cases} -true & \text{if } c = true \\ -\{(M,t) \mid t \text{ is determined wrt } \xrightarrow{i}_M \ \& \\ \forall s : s \overset{i}{\rightarrowtriangle}_M t, (M,s) \in c\} \end{cases} \tag{1}$$

Notice that $\uparrow_i^{\mathtt{MN}}(c)$ excludes indetermined states (i.e. a state t such that for every $s \in S, (s,t) \notin \mathcal{R}$.) if $c \neq true$. It turns out that we can add them and obtain a more succinct normal right-inverse:

▶ **Definition 15** (Normal Right-Inverse). Let $\mathbf{K}(\mathcal{S}) = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be a Kripke scs over $\mathcal{S} \subseteq \mathcal{M}^{\mathtt{D}}$. Define $\uparrow_i^{\mathtt{N}} : Con \to Con$ as $\uparrow_i^{\mathtt{N}}(c) \stackrel{\mathtt{def}}{=} \{(M,t) \mid \forall s : s \overset{i}{\rightarrowtriangle}_M t, (M,s) \in c\}$.

Clearly $\uparrow_i^{\mathtt{N}}(c)$ includes every (M,t) such that t is indetermined wrt $\xrightarrow{i}_M$.

3.7 Applications

In this section we will illustrate and briefly discuss the results obtained in the previous section in the context of modal logic.

We can interpret modal formulae as constraints in a given Kripke scs $\mathbf{C} = \mathbf{K}(\mathcal{S}_n(\Phi))$.

▶ **Definition 16** (Kripke Constraint Interpretation). Let $\mathbf{C}$ be a Kripke scs $\mathbf{K}(\mathcal{S}_n(\Phi))$. Given a modal formula ϕ in the modal language $\mathcal{L}_n(\Phi)$, its interpretation in the Kripke scs $\mathbf{C}$ is the constraint $\mathbf{C}[\![\phi]\!]$ inductively defined as follows: $\mathbf{C}[\![p]\!] = \{(M,s) \mid \pi_M(s)(p) = 1\}$, $\mathbf{C}[\![\phi \wedge \psi]\!] = \mathbf{C}[\![\phi]\!] \sqcup \mathbf{C}[\![\psi]\!]$, $\mathbf{C}[\![\neg\phi]\!] = \sim \mathbf{C}[\![\phi]\!]$, $\mathbf{C}[\![\Box_i\phi]\!] = [\ \mathbf{C}[\![\phi]\!]\ ]_i$.

To illustrate our results in the previous sections, we fix a modal language $\mathcal{L}_n(\Phi)$ (whose formulae are) interpreted in an arbitrary Kripke scs $\mathbf{C} = \mathbf{K}(\mathcal{S}_n(\Phi))$. Suppose we wish to extend it with modalities $\Box_i^{-1}$, called reverse modalities also interpreted over the same set of KS's $\mathcal{S}_n(\Phi)$ and satisfying some minimal requirement. The language is given by the following grammar.

▶ **Definition 17** (Modal Language with Reverse Modalities)**.** Let Φ be a set of primitive propositions. The modal language $\mathcal{L}_n^{+r}(\Phi)$ is given by the following grammar: $\phi, \psi, \ldots := p \mid \phi \wedge \psi \mid \neg\phi \mid \Box_i\phi \mid \Box_i^{-1}\phi$ where $p \in \Phi$ and $i \in \{1, \ldots, n\}$.

The minimal semantic requirement for each $\Box_i^{-1}$ is that:

$$\Box_i \Box_i^{-1}\phi \;\Leftrightarrow\; \phi \quad \text{valid in } \mathcal{S}_n(\Phi). \tag{2}$$

We then say that $\Box_i^{-1}$ is a *right-inverse* modality for $\Box_i$.

Since $\mathbf{C}[\![\Box_i\phi]\!] = [\ \mathbf{C}[\![\phi]\!]\ ]_i$, we can derive semantic interpretations for $\Box_i^{-1}\phi$ by using a right inverse $\uparrow_i$ for $[\cdot]_i$ in Definition 16. Assuming that such a right inverse exists, we can interpret the reverse modality in $\mathbf{C}$ as

$$\mathbf{C}[\![\Box_i^{-1}\phi]\!] = \uparrow_i(\ \mathbf{C}[\![\phi]\!]\ ). \tag{3}$$

We can choose $\uparrow_i$ in Equation (3) from the set $\{\uparrow_i^{\mathtt{N}}, \uparrow_i^{\mathtt{MN}}, \uparrow_i^{\mathtt{M}}\}$ of right-inverses given in Section 3.5.

3.7.1 Temporal Operators

We conclude this section with a brief discussion on some right-inverse linear-time modalities. Let us suppose that $n = 2$ in our modal language $\mathcal{L}_n(\Phi)$ under consideration (thus interpreted in Kripke scs $\mathbf{C} = \mathbf{K}(\mathcal{S}_2(\Phi))$. Assume further that the intended meaning of the two modalities $\Box_1$ and $\Box_2$ are the *next* operator ($\bigcirc$) and the *henceforth/always* operator ($\Box$), respectively, in a *linear-time* temporal logic. To obtain the intended meaning we take $\mathcal{S}_2(\Phi)$ to be the largest set such that: If $M \in \mathcal{S}_2(\Phi)$, M is a 2-agent KS where $\xrightarrow{1}_M$ is isomorphic to the successor relation on the natural numbers and $\xrightarrow{2}_M$ is the reflexive and transitive closure of $\xrightarrow{1}_M$. The relation $\xrightarrow{1}_M$ is intended to capture the linear flow of time. Intuitively, $s \xrightarrow{1}_M t$ means t is the only next state for s. Similarly, $s \xrightarrow{2}_M t$ for $s \neq t$ is intended to capture the fact that t is one of the infinitely many future states for s.

Let us first consider the next operator $\Box_1 = \bigcirc$. Notice that $\xrightarrow{1}_M$ is determinant-complete. If we apply Equation (3) with $\uparrow_1 = \uparrow_1^{\mathtt{M}}$, we obtain $\Box_1^{-1} = \ominus$, a past modality known in the literature as *strong* previous operator [13]. If we take $\uparrow_i$ to be the normal right inverse $\uparrow_i^{\mathtt{N}}$, we obtain $\Box_1^{-1} = \widetilde{\ominus}$, the past modality known as *weak* previous operator [13]. Notice that the only difference between the two operators is that, if s is an indetermined/initial state wrt $\xrightarrow{1}_M$ then $(M, s) \not\models \ominus\, \phi$ and $(M, s) \models \widetilde{\ominus}\, \phi$ for any ϕ. It follows that $\ominus$ is not a normal operator, since $\ominus\mathrm{T}$ is not valid in $\mathcal{S}_2(\Phi)$ but T is.

Let us now consider the always operator $\Box_2 = \Box$. Notice that $\xrightarrow{2}_M$ is not determinant-complete: Take any increasing chain $s_0 \xrightarrow{1}_M s_1 \xrightarrow{1}_M \ldots$ The state s_1 is not determinant because for every s_j such that $s_1 \xrightarrow{2}_M s_j$ we also have $s_0 \xrightarrow{2}_M s_j$. Theorem 11 tells us that there is no right-inverse $\uparrow_2$ of $[\cdot]_i$ that can give us an operator $\Box_2^{-1}$ satisfying Equation (2).

4 Ongoing Work

4.1 Knowledge in Terms of Space

In this section we show our current work on using spatial constraint systems to express the epistemic concept of knowledge by using the following notion of global information:

▶ **Definition 18** (Global Information). Let $\mathcal{C}$ be an n-scs with space functions $[\cdot]_1, \ldots, [\cdot]_n$ and G be a non-empty subset of $\{1, \ldots, n\}$. *Group-spaces* $[\cdot]_G$ and *global* information $[\![\cdot]\!]_G$ of G in $\mathcal{C}$ are defined as:

$$[c]_G \stackrel{\text{def}}{=} \bigsqcup_{i \in G} [c]_i \quad \text{and} \quad [\![c]\!]_G \stackrel{\text{def}}{=} \bigsqcup_{j=0}^{\infty} [c]_G^j \tag{4}$$

where $[c]_G^0 \stackrel{\text{def}}{=} c$ and $[c]_G^{k+1} \stackrel{\text{def}}{=} [[c]_G^k]_G$.

The constraint $[c]_G$ means that c holds in the spaces of agents in G. The constraint $[\![c]\!]_G$ entails $[[\ldots[c]_{i_m}\ldots]_{i_2}]_{i_1}$ for any $i_1, i_2, \ldots, i_m \in G$. Thus, it realizes the intuition that c holds *globally* wrt G: c holds in each nested space involving only the agents in G. In particular, if G is the set of all agents, $[\![c]\!]_G$ means that c holds *everywhere*. From the epistemic point of view $[\![c]\!]_G$ is related to the notion of common-knowledge of c [5].

4.2 Knowledge Constraint System

In [9] the authors extended the notion of spatial constraint system to account for *knowledge*. In this summary we shall refer to the extended notion in [9] as *S4 constraint systems* since it is meant to capture the epistemic logic for knowledge S4. Roughly speaking, one may wish to use $[c]_i$ to represent not only some information c that agent i has but rather a *fact* that he knows. The domain theoretical nature of constraint systems allows for a rather simple and elegant characterization of knowledge by requiring space functions to be *Kuratowski closure operators* [10]: i.e., monotone, extensive and idempotent maps preserving bottom and lubs.

▶ **Definition 19** (Knowledge Constraint System [9]). An n-agent *S4 constraint system* (n-*s4cs*) $\mathbf{C}$ is an n-scs whose space functions $[\cdot]_1, \ldots, [\cdot]_n$ are also *closure operators*. Thus, in addition to S.1 and S.2 in Definition 2, each $[\cdot]_i$ also satisfies: (EP.1) $[c]_i \sqsupseteq c$ and (EP.2) $[[c]_i]_i = [c]_i$.

Intuitively, in an n-s4cs, $[c]_i$ states that the agent i has knowledge of c in its store $[\cdot]_i$. The axiom EP.1 says that if agent i knows c then c must hold, hence $[c]_i$ has at least as much information as c. The epistemic principle that an agent i is aware of its own knowledge (*the agent knows what he knows*) is realized by EP.2. Also, the epistemic assumption that agents are *idealized reasoners* follows from the monotonicity of space functions, i.e., for a consequence c of d ($d \sqsupseteq c$), then if d is known to agent i, so is c, $[d]_i \sqsupseteq [c]_i$.

In [9] the authors use the notion of *Kuratowski closure operators* $[c]_i$ to capture knowledge. In what follows we show an alternative interpretation of knowledge as the global construct $[\![c]\!]_G$ in Definition 18.

4.3 Knowledge as Global Information

Let $\mathcal{C} = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be a *spatial* constraint system. From Definition 18 we obtain the following equation:

$$\llbracket c \rrbracket_{\{i\}} = c \sqcup [c]_i \sqcup [c]_i^2 \sqcup [c]_i^3 \sqcup \ldots = \bigsqcup_{j=0}^{\infty} [c]_i^j \tag{5}$$

For simplicity, we shall use $\llbracket \cdot \rrbracket_i$ as an abbreviation of $\llbracket \cdot \rrbracket_{\{i\}}$. We shall demonstrate that $\llbracket c \rrbracket_i$ can also be used to represent the knowledge of c by agent i.

We will show that the global function $\llbracket c \rrbracket_i$ is in fact a Kuratowski closure operator and thus satisfies the epistemic axioms EP.1 and EP.2 above: It is easy to see that $\llbracket c \rrbracket_i$ satisfies $\llbracket c \rrbracket_i \sqsupseteq c$ (EP.1). Under certain natural assumptions we shall see that it also satisfies $\llbracket \llbracket c \rrbracket_i \rrbracket_i = \llbracket c \rrbracket_i$ (EP.2). Furthermore, we can combine knowledge with our belief interpretation of space functions: clearly, $\llbracket c \rrbracket_i \sqsupseteq [c]_i$ holds for any c. This reflects the epistemic principle that *whatever is known is also believed* [8].

We now show that any *spatial constraint system* with continuous space functions (i.e. functions preserving lubs of any directed set) $[\cdot]_1, \ldots, [\cdot]_n$ induces an s4cs with space functions $\llbracket \cdot \rrbracket_1, \cdots \llbracket \cdot \rrbracket_n$.

▶ **Definition 20.** Given an scs $\mathcal{C} = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$, we use $\mathcal{C}^*$ to denote the tuple $(Con, \sqsubseteq, \llbracket \cdot \rrbracket_1, \ldots, \llbracket \cdot \rrbracket_n)$.

One can show that $\mathcal{C}^*$ is also a spatial constraint system. Besides, it is an s4cs as stated next.

▶ **Theorem 21.** *Let $\mathcal{C} = (Con, \sqsubseteq, [\cdot]_1, \ldots, [\cdot]_n)$ be a spatial constraint system. If $[\cdot]_1, \ldots, [\cdot]_n$ are continuous functions, then $\mathcal{C}^*$ is an n-agent s4cs.*

We shall now prove that S4 can also be captured using the global interpretation of space.

From now on $\mathbf{C}$ denotes the Kripke constraint system $\mathbf{K}(\mathcal{M})$ (Definition 8), where $\mathcal{M}$ represent a set of non-empty set of n-agent Kripke strutures. Notice that constraints in $\mathbf{C}$, and consequently also in $\mathbf{C}^*$, are sets of *unrestricted* (pointed) Kripke structures. Although $\mathbf{C}$ is not an S4cs, from the above theorem, its induced scs $\mathbf{C}^*$ is. Also, we can give in $\mathbf{C}^*$ a sound and complete compositional interpretation of S4 formulae.

The compositional interpretation of modal formulae in our constraint system $\mathbf{C}^*$ is similar to the one introduced in 16 except for the interpretation of the $\Box_i \phi$ modality.

Notice that $\Box_i \phi$ is interpreted in terms of the global operation. Since $\mathbf{C}^*$ is a power-set ordered by reversed inclusion, the lub is given by set intersection. Thus, from Equation (5)

$$\mathbf{C}^*\llbracket \Box_i \phi \rrbracket = \llbracket\ \mathbf{C}^*\llbracket \phi \rrbracket\ \rrbracket_i = \bigsqcup_{j=0}^{\omega} [\mathbf{C}^*\llbracket \phi \rrbracket]_i^j = \bigcap_{j=0}^{\omega} [\mathbf{C}^*\llbracket \phi \rrbracket]_i^j \tag{6}$$

In particular, from Theorem 21 and Axiom EP.2, $\mathbf{C}^*\llbracket \Box_i \phi \rrbracket = \mathbf{C}^*\llbracket \Box_i(\Box_i \phi) \rrbracket$ follows as an S4-knowledge modality; i.e., if agent i knows ϕ he knows that he knows it.

We conclude this section with the following theorem stating the correctness wrt validity of the interpretation of knowledge as as global operator.

▶ **Theorem 22.** $\mathbf{C}^*[\![\phi]\!] = true$ *if and only if* ϕ *is S4-valid.*

5 Future Work

As future work we are planning to specify the epistemic notion of *Distributed Knowledge* (DK) [5] as well as a computational notion of *process* in our algebraic structures.

5.1 Distributed Knowledge in Terms of Space

Informally, DK says that, if a given agent i has $c \to d$ in his space and an agent j has c in her space, then if we were to communicate with each other we could have d in their space though individually neither i nor j has d. This could be an important concept for distributed systems, e.g. to predict unwanted behavior in a system upon potential communication among agents.

Using [5] and our notion of Heyting implication in Definition 4 we could extend scs with DK as follows.

▶ **Definition 23.** Let $\mathcal{C} = (Con, \subseteq, [\cdot]_1, \ldots, [\cdot]_n)$. Let $G \subseteq \{1, 2, \ldots, n\}$ be a non-empty subset of agents. Distributed knowledge of G is a self-map $\mathcal{D}_G : Con \to Con$ satisfying the next axioms:

1. $\mathcal{D}_G(true) = true$
2. $\mathcal{D}_G(c \sqcup d) = \mathcal{D}_G(c) \sqcup \mathcal{D}_G(d)$
3. $\mathcal{D}_G(c) = [c]_i$ *if* $G = \{i\}$
4. $\mathcal{D}_{G'}(c) \sqsupseteq \mathcal{D}_G(d)$ *if* $G' \subseteq G$

Intuitively $\mathcal{D}_G(c)$ means that G has DK of c. The first condition says that any G has DK of *true*. The second condition says that, if G has DK of two pieces of information c and d, then G has DK of their join. The third condition tells us that an agent has DK of what he knows. Finally, the fourth condition says that the larger the subgroup, the greater its DK.

In previous paragraphs we argued that if agent i has $c \to d$ and an agent j has c then they would have DK of d ($\mathcal{D}_{\{i,j\}}(d)$). Indeed, from the above axioms and the properties of space, one can prove that $[c \to d]_i \sqcup [c]_j \sqsupseteq \mathcal{D}_{\{i,j\}}(d)$.

As future work we would like to give an explicit spatial construction that characterizes $\mathcal{D}_G(c)$.

5.2 Processes as Constraint Systems

Concurrent constraint programming (ccp) calculi are a well-known family of process algebras from concurrency theory [15, 12, 4, 11]. Computational processes from ccp can be seen as closure operators over an underlying constraint system $\mathcal{C} = (Con, \sqsubseteq)$. A closure operator f over $\mathcal{C} = (Con, \sqsubseteq)$ is a monotonic self map on Con such that $f(c) \sqsupseteq c$ and $f(f(c)) = f(c)$.

It is well known that closure operators form themselves a complete lattice. Thus, ccp processes can be interpreted as elements of the cs $\mathcal{C}^+ = (Con^+, \sqsubseteq)$ where Con^+ is the set of closure operators over Con ordered wrt $\sqsubseteq$ (recall that $f \sqsubseteq g$ iff $f(c) \sqsubseteq g(c)$ for every $c \in Con$.)

We plan to use the space and extrusion functions from spatial constraint systems to give a declarative semantics to the corresponding spatial, time and extrusion constructs in ccp-based process algebras. More importantly, we plan to use the notion of distributed knowledge to derive a corresponding notion in ccp-process algebras. To our knowledge this will be the first time that distributed knowledge is used in the context of process calculi.

References

1 Samson Abramsky and Achim Jung. Domain theory. *Handbook of logic in computer science*, pages 1–77, 1994.

2 Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, 1st edition, 2002.

3 Frank S. Boer, Alessandra Di Pierro, and Catuscia Palamidessi. Nondeterminism and infinite computations in constraint programming. *Theoretical Computer Science*, pages 37–78, 1995.

4 Alessandra Di Pierro, Catuscia Palamidessi, and Frank S. Boer. An algebraic perspective of constraint logic programming. *Journal of Logic and Computation*, pages 1–38, 1997.

5 Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Y Vardi. *Reasoning about knowledge*. MIT press Cambridge, 4th edition, 1995.

6 M. Guzman, S. Haar, S. Perchy, C. Rueda, and F. Valencia. Belief, knowledge, lies and other utterances in an algebra for space and extrusion. *Journal of Logical and Algebraic Methods in Programming*, 2016.

7 M. Guzman, S. Perchy, C. Rueda, and F. Valencia. Deriving extrusion on constraint systems from concurrent constraint programming process calculi. In *ICTAC 2016*, 2016.

8 Jaakko Hintikka. *Knowledge and belief*. Cornell Univeristy Press, 1962.

9 Sophia Knight, Catuscia Palamidessi, Prakash Panangaden, and Frank D Valencia. Spatial and epistemic modalities in constraint-based process calculi. In *CONCUR 2012*, pages 317–332. Springer, 2012.

10 John Charles Chenoweth McKinsey and Alfred Tarski. The algebra of topology. *Annals of mathematics*, pages 141–191, 1944.

11 Nax P Mendler, Prakash Panangaden, Philip J Scott, and RAG Seely. A logical view of concurrent constraint programming. *Nordic Journal of Computing*, pages 181–220, 1995.

12 Prakash Panangaden, Vijay Saraswat, Philip J Scott, and RAG Seely. A hyperdoctrinal view of concurrent constraint programming. In *Workshop of Semantics: Foundations and Applications, REX*, pages 457–476. Springer, 1993.

13 Amir Pnueli and Zohar Manna. *The temporal logic of reactive and concurrent systems*. Springer, 1992.

14 Sally Popkorn. *First steps in modal logic*. Cambridge University Press, 1st edition, 1994.

15 Vijay A Saraswat, Martin Rinard, and Prakash Panangaden. Semantic foundations of concurrent constraint programming. In *POPL'91*, pages 333–352, 1991.

16 Steven Vickers. *Topology via logic.* Cambridge University Press, 1st edition, 1996.