Exploring the Bidimensional Space: A Dynamic Logic Point of View

Philippe Balbiani
IRIT-CNRS
Toulouse University, France
Philippe.Balbiani@irit.fr

David Fernández-Duque
IRIT, Toulouse University
France
David.Fernandez@irit.fr

Emiliano Lorini
IRIT-CNRS
Toulouse University, France
Emiliano.Lorini@irit.fr

ABSTRACT

We present a family of logics for reasoning about agents’ positions and movements in the plane which have several potential applications in the area of multi-agent systems, such as multi-agent planning and robotics. The most general logic includes (i) atomic formulas for representing the truth of a fact or the presence of an agent at a certain position of the plane, (ii) atomic programs corresponding to the four basic orientations in the plane (up, down, left, right) as well as the four program constructs of propositional dynamic logic PDL (sequential composition, nondeterministic composition, iteration, and test). As this logic is not computably enumerable, we study some interesting decidable and axiomatizable fragments of it. We also present a decidable extension of its iteration-free fragment by special programs representing movements of agents in the plane.

Keywords
Dynamic Logic; Spatial Logic; Epistemic Logic

1. INTRODUCTION

Most of existing logics for multi-agent systems (MAS) including multi-agent epistemic logic [9], multi-agent variants of propositional dynamic logic [16] and logics of action and strategic reasoning such as ATL [1], Coalition Logic CL [17] and STIT [6] are “ungrounded” in the sense that their formal semantics are based on abstract primitive notions such as the concept of Kripke model or the concept of possible world (or state). As a result, there is no direct connection between these abstract concepts and the concrete environment in which the agents interact. This kind of grounding problem of logics for MAS becomes particularly relevant for robotic applications. Since robots are situated in spatial environments, in order to make logics for MAS useful for robotics, their semantics have to be grounded on space. Specifically, a formal semantics is required that provides an explicit representation of the space in which the robots’ actions and perceptions are situated. Some initial steps into the direction of grounding

formal semantics of logics for MAS on space have done in the recent years. Among them, we should mention logics of multi-agent knowledge in both one-dimensional space and two-dimensional space [11, 4], spatio-temporal logics such as constraint LTL applied to model 2D grid environments [3], multi-robot task logic based on monadic second-order logic [2] and logics of robot localization [5]. The present paper shares with these approaches the idea that in order to make existing logics of MAS useful for MAS applications such as multi-agent planning and robotics, their semantics should provide an explicit representation of the agents’ environment.

The main motivation of the present work is to provide a logical framework whose language and semantics are, at the same time, simple and sufficiently general to describe (i) the properties of the spatial environment in which several agents can move, and (ii) the consequences of the agents’ movements on such a spatial environment. To meet this objective, we have decided to exploit the language of propositional dynamic logic PDL as a general formalism for representing actions of agents and their effects, and to interpret this language on a simple formal semantics of the two-dimensional (2D) space. The reason why we decided to start from the 2D space is that its representation already presents some interesting conceptual aspects as well as some difficulties with respect to the computational properties of the resulting logic. We believe that, before studying action in the 3D space and, more generally, action in n-dimensional spaces (with n > 2), a comprehensive logical theory of action in the 2D space is required.

More concretely, this paper presents a family of logics for reasoning about agents’ positions and movements in the plane. The most general logic, called Dynamic Logic of Space DL-S*, is presented in Section 2. DL-S* includes (i) atomic formulas for representing the truth of a given fact (atomic facts) or the presence of a given agent at a certain position of the plane (positional atoms), (ii) atomic programs corresponding to the four basic orientations in the plane (up, down, left, right) as well as the four program constructs of PDL (sequential composition, nondeterministic composition, iteration, and test). The logic is proved to be non-computably enumerable (non-c.e.) and its satisfiability problem undecidable (Section 3), while its model-checking problem is proved to be decidable in deterministic polynomial time (Section 4). Given the negative properties of DL-S*, we decided to study some interesting decidable and axiomatizable fragments of it. This includes the iteration-free fragment of DL-S* (Section 5) as well as CL-P*, a frag-

Copyright © 2017, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.
ment that only allows iteration of atomic programs (e.g.,
the action of moving an indefinite number of times to the
right) and has no propositional variables aside from position-
tional atoms (Section 7). In view of its similarity to well-
known non-c.e. products of modal logics, the decidability re-

tail for $\text{CL-P}^\ast$ is quite remarkable. As the logic $\text{DL-S}^\ast$ only
provides a static representation of the 2D space, in Section
[7] we present a decidable extension of its iteration-free frag-
ment by special programs representing movements of agents
in the plane. Conclusion of the paper (Section 8) presents
perspectives of future research including integration of an
epistemic component in the logic as well as of the concept
of coalitional capability in the sense of [14].

2. SPACE

$\text{DL-S}^\ast$ (Dynamic Logic of Space) is a dynamic logic which
consists of: (i) formulas representing the agents’ positions
and the truth of facts in the different positions of the bidi-

menionality space, and (ii) programs allowing to move from
one position to another position of the bidimensional space.

2.1 Syntax

Assume a countable set of atomic propositions $\text{Atm} =
\{p, q, \ldots \}$ and a finite set of agents $\text{Agt} = \{1, \ldots , n\}.
The language of $\text{DL-S}^\ast$, denoted by $\mathcal{L}_{\text{DL-S}^\ast}(\text{Atm}, \text{Agt})$, is
defined by the following grammar in Backus-Naur Form:

\[
\begin{align*}
\alpha &::= \uparrow | \downarrow | \implies | \alpha \cup \alpha’ | \alpha^\ast | \alpha^* | \alpha’ | \alpha^\perp | (\alpha) \\
\varphi &::= p | h_i | \neg \varphi | \varphi \land \psi | \alpha \varphi
\end{align*}
\]

where $p$ ranges over $\text{Atm}$ and $i$ ranges over $\text{Agt}$. Other
Boolean constructions $\land, \lor, \rightarrow$, and $\iff$ are defined from
$p, \neg, \land$ in the standard way. Instances of $\alpha$ are called
spatial programs. When there is no risk of confusion we will
omit parameters and simply write $\mathcal{L}_{\text{DL-S}^\ast}$. The modal degree
of a formula $\varphi \in \mathcal{L}_{\text{DL-S}^\ast}$ (in symbols $\deg(\varphi)$) is defined in
the standard way as the nesting depth of modal operators
in $\varphi$. Let $\llbracket \varphi \rrbracket$ denote the size of $\varphi$. For all (negate-
or positive) integers $x$, let $[\llbracket \uparrow \rrbracket]^x$ be the modality consist-
ing $x$ consecutive $[\uparrow]$ when $x \leq 0$, otherwise let $[\llbracket \uparrow \rrbracket]^x$ be
the modality consistency of $x$ consecutive $[\uparrow]$. Similarly for $[\rightarrow]^x$.
The formula $h_i$ is read “the agent $i$ is here”, whereas $\alpha \varphi$ has to be read “$\varphi$ is true in the position that is reachable
from the current position through the program $\alpha$”.

We also consider sublanguages of $\mathcal{L}_{\text{DL-S}^\ast}$. Let $P$ be a
set atomic propositions, $I$ a set of agents and $A$ a set of
spatial programs. $\mathcal{L}_{\text{DL-S}^\ast}(P, I, A)$ denotes the restriction of
$\mathcal{L}_{\text{DL-S}^\ast}(\text{P}, I)$ to programs from $A$.

2.2 Semantics

The main notion in semantics is given by the following
concept of spatial model.

Definition 1 (Spatial model (SM)). A spatial model is a tuple
$M = (P, V)$ where:

- $P : \text{Agt} \rightarrow \mathbb{Z} \times \mathbb{Z}$;
- $V : \mathbb{Z} \times \mathbb{Z} \rightarrow 2^{Atm}$.

The set of all spatial models is denoted by $\mathcal{M}$.

For every $(x, y) \in \mathbb{Z} \times \mathbb{Z}$, $P(i) = (x, y)$ means that the
agent $i$ is in the position $(x, y)$, whereas $p \in V(x, y)$ means
that $p$ is true at the position $(x, y)$. For every $x \in \mathbb{Z}$, $\text{succ}(x)$
denotes the direct successor of $x$ (i.e., $x + 1$), while $\text{prec}(x)$
denotes the direct predecessor of $x$ (i.e., $x - 1$).

Formulas are evaluated with respect to a spatial model
$M$ and a spatial position $(x, y)$. Below, if $R, S$ are binary
relations, $R^\ast$ denotes the transitive, reflexive closure of $R$
and $R \circ S$ the composition of $R$ and $S$.

Definition 2 ($R_\alpha$ and truth conditions). Let $M =
(P, V)$ be a spatial model. For all spatial programs $\alpha$ and for
all formulas $\varphi$, the binary relation $R_\alpha$ on $\mathbb{Z} \times \mathbb{Z}$ and the
truth conditions of $\varphi$ in $M$ are defined by parallel induction as follows:

\[
\begin{align*}
R_\emptyset &= \{(x, y) : x = y = \text{succ}(y)\} \\
R_\uparrow &= \{(x, y) : x = y = \text{prec}(y)\} \\
R_\rightarrow &= \{(x, y) : x = \text{succ}(x) \land y = y\} \\
R_\ast &= \{(x, y) : x = \text{prec}(x) \land y = y\} \\
R_\alpha^\ast &= (R_\alpha)^+ \\
R_\varphi &= \{(x, y) : M, (x, y) \models \varphi\}
\end{align*}
\]

\[
\begin{align*}
M, (x, y) \models p & \iff p \in V(x, y) \\
M, (x, y) \models h_i & \iff P(i) = (x, y) \\
M, (x, y) \models \neg \varphi & \iff M, (x, y) \not\models \varphi \\
M, (x, y) \models \varphi \land \psi & \iff M, (x, y) \models \varphi \text{ and } M, (x, y) \models \psi \\
M, (x, y) \models [\alpha] \varphi & \iff \forall (x’, y’) \in \mathbb{Z} \times \mathbb{Z} : (if (x, y)R_\alpha(x’, y’)
\text{ then } M, (x’, y’) \models \varphi) \\
\end{align*}
\]

When $(x, y)R_\alpha(x’, y’)$, we will say that position $(x’, y’)$ is
accessible from position $(x, y)$ by program $\alpha$.

Remark that formulas like $h_i$ behave like nominals in hy-
brid logics, i.e. their truth sets are singletons.

We say that $\varphi \in \mathcal{L}_{\text{DL-S}^\ast}$ is valid, denoted by $\models \varphi$, if and
only if, for every spatial model $M$ and position $(x, y)$, we have
$M, (x, y) \models \varphi$. We say that formula $\varphi \in \mathcal{L}_{\text{DL-S}^\ast}$ is
satisfiable if and only if $\neg \varphi$ is not valid.

Before going into more technical details let us present an
example illustrating the expressive power of the logic $\text{DL-S}^\ast$.

Example 1. Two robots, called Ann and Bob, are located
respectively at position $(0, 0)$ and $(2, 0)$. Suppose there is a
danger between the two robots at position $(1, 0)$. This means
Ann is located at $(0, 0)$ and has a danger in his right-hand
side, while Bob is located at $(2, 0)$ and has a danger in his
left-hand side, that is, $M, (0, 0) \models h_\text{Ann} \land [\rightarrow] \text{danger}$
and $M, (2, 0) \models h_\text{Bob} \land [\leftarrow] \text{danger}$.

2.3 Bisimulation

The essential tool we will use to establish our decidability
results is the notion of bounded bisimulation.

Definition 3. Fix a set $P$ of atomic propositions, a set
$I$ of agents and a set $A$ of spatial programs. Given spatial
models $M_1 = (P_1, V_1)$ and $M_2 = (P_2, V_2)$, $n < \omega$, we define
a binary relation $(M_1, \cdot) =_n (M_2, \cdot) \subseteq \mathbb{Z}^2 \times \mathbb{Z}^2$ by induction
on $n$ as follows.

We set $(M_1, \bar{x}) =_n (M_2, \bar{y})$ if

1 To be more precise, we should define one relation $R_\alpha^\ast$ per
spatial model $M$. However, we omit the superscript $M$ since
it is clear from the context.
1. for every \( i \in I \), \( \bar{x} = \mathcal{P}_i(i) \) if and only if \( \bar{y} = \mathcal{P}_i(i) \);
2. for every \( p \in P \), \( \bar{x} \in \mathcal{V}_1(p) \) if and only if \( \bar{y} \in \mathcal{V}_2(p) \);
3. if \( n > 0 \), then for every \( \alpha \in A \),

\[
\text{Forth}_\alpha \text{ Whenever } \bar{x}R_\alpha \bar{x}, \text{ there is } \bar{y}' \text{ such that } \bar{y}'R_\alpha \bar{y} \text{ and } (M_1, \bar{x}) \models \neg \alpha_{n-1} (M_2, \bar{y}) \text{, and}
\]

\[
\text{Back}_\alpha \text{ Whenever } \bar{y}R_\alpha \bar{y}', \text{ there is } \bar{x}' \text{ such that } \bar{x}R_\alpha \bar{x}' \text{ and } (M_1, \bar{x}') \models \neg \alpha_{n-1} (M_2, \bar{y}) \text{.}
\]

We may just write \( \bar{x} \models \alpha \) instead of \( (M, \bar{x}) \models \alpha \). The following is then standard:

**Lemma 1.** Fix a set \( P \) of atomic propositions, a set \( I \) of agents and a set of programs \( A \). If \( M_1 = (\mathcal{P}_1, \mathcal{V}_1) \) and \( M_2 = (\mathcal{P}_2, \mathcal{V}_2) \) are spatial models and \( \varphi \in \mathcal{L}_{\text{DL-S}}(P, I, A) \) has modal degree at most \( n \), then whenever \((M_1, \bar{x}) \models \alpha (M_2, \bar{y}) \), it follows that \( M_1, \bar{x} \models \varphi \) if and only if \( M_2, \bar{y} \models \varphi \).

### 3. Undecidability

This section presents results about undecidability for the satisfiability problem of \( \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \)-formulas. Products of linear logics are logics with two (or more) modalities, interpreted over structures very similar to spatial models. Their formulas are equivalent to \( \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \)-formulas over the class of all spatial models and are often undecidable \cite{10, 12, 15}. This suggests that the satisfiability problem of formulas in \( \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \), as well as some proper fragments, is undecidable as well. The idea is to allow actions only along the horizontal and vertical axes, which following \cite{13} we call the ‘compass directions’. To be precise, we define the language of compass logic of space by \( \mathcal{L}_{\text{CLS}}(\text{Atm}, \text{Agt}) = \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}, C) \), where

\[
C = \{ \mathbf{\uparrow}, \mathbf{\downarrow}, \Rightarrow, \Leftarrow, \mathbf{\uparrow}^*, \mathbf{\downarrow}^*, \Rightarrow^*, \Leftarrow^* \}.
\]

As before, we may omit the parameters \( \text{Atm}, \text{Agt} \) when this does not lead to confusion. By \( \mathcal{L}_{\text{CLS}} \) (the language of compass logic of facts) we denote the special case where \( \text{Agt} = \emptyset \), and similarly \( \mathcal{L}_{\text{CLS}} \) (the language of compass logic of positions) denotes the case where \( \text{Atm} = \emptyset \).

We start with the following undecidability result for the satisfiability problem of the latter.

**Theorem 1.** The set of valid formulas of \( \mathcal{L}_{\text{CLS}} \) is not computably enumerable.

**Proof.** This follows from Theorem 5.38 in \cite{10}, which states (in their notation) that \( \mathcal{P} \text{TL}_{\text{CL-F}} \times \mathcal{P} \text{TL}_{\text{CL-F}} \) is not c.e. But this is a notational variant of a fragment of CL-F∗, where \( \mathcal{O}_1 \approx [\Rightarrow] \), \( \mathcal{O}_2 \approx [\Rightarrow^*] \), \( \mathcal{O}_1 \approx [\Leftarrow] \), and \( \mathcal{O}_2 \approx [\Leftarrow^*] \).

We remark that we only need two of the four compass directions for this proof, provided they are perpendicular. As a corollary, we obtain undecidability of the larger logic.

**Corollary 1.** The set of valid formulas of \( \mathcal{L}_{\text{DL-S}} \) is not computably enumerable.

In order to study model-checking, we need a finite representation of spatial models. To this aim, we introduce the following definition of \textit{bounded spatial model of size } \( n \). For \((x, y) \in \mathbb{Z}^2 \), write \( \| (x, y) \| \leq n \) iff \( |x| \leq n \) and \( |y| \leq n \).

**Definition 4.** (Bounded spatial model (BSM)). Let \( n \) be a nonnegative integer. A spatial model \( M = (\mathcal{P}, \mathcal{V}) \) is said to be \( n \)-bounded iff for all \( i \in \text{Agt} \), \( |\mathcal{P}(i)| \leq n \) and for all \((x, y) \in \mathbb{Z}^2 \), if \( |(x, y)| \leq n \) then \( \mathcal{V}(x, y) = \emptyset \).

Observe that while the interpretations of variables are bounded, the frame itself is not; we still interpret formulas over \( \mathbb{Z}^2 \).

As for the class of all models, the restriction to bounded models gives rise to an undecidable set of valid formulas of \( \mathcal{L}_{\text{DL-S}} \)-formulas:

**Theorem 2.** The set of formulas of \( \mathcal{L}_{\text{CLS}} \) valid over the class of bounded spatial models is not computably enumerable.

**Sketch of Proof.** This essentially follows from Corollary 7.18 in \cite{10}, which in their notation states that \( \text{Log}(\{\mathbb{N}, \geq\} \times \mathbb{N}, \geq\} \) is not c.e. As above, this is a notational variant of a fragment of CL-F∗, where \( \mathcal{O}_1 \approx [\Rightarrow] \) and \( \mathcal{O}_2 \approx [\Rightarrow^*] \), although interpreted over frames of the form \( \{0, \ldots, n\} \times \{0, \ldots, m\} \). That it is not c.e. is obtained by reducing the halting problem for Turing machines to \( \text{Log}(\{\mathbb{N}, \geq\} \times \mathbb{N}, \geq\} \), representing finite computations as finite models. Minor adjustments of this construction can be used, instead, to represent finite computations as bounded models.

As before, the undecidability of the set of formulas of \( \mathcal{L}_{\text{DL-S}} \) valid over the class of bounded spatial models follows. There are different ways to get out of the undecidability of the satisfiability problem of \( \mathcal{L}_{\text{DL-S}} \)-formulas as highlighted by Corollary 4. One possibility is to consider the star-free fragment of \( \mathcal{L}_{\text{DL-S}} \). Another possibility is to study fragments of \( \mathcal{L}_{\text{DL-S}} \) that omit atomic propositions and allow only nominals. These two possibilities are explored, respectively, in Sections 5 and 6.

### 4. Model-checking

The model-checking problem for \( \mathcal{L}_{\text{CLS}}(\text{Atm}, \text{Agt}) \) is the following: let \( \varphi \in \mathcal{L}_{\text{CLS}}(\text{Atm}, \text{Agt}) \), let \( n \) be a nonnegative integer, let \( M \) be an \( n \)-bounded spatial model and let \((x, y) \in \mathbb{Z} \times \mathbb{Z} \), is it the case that \( (M, (x, y)) \models \varphi \)?

In this section we will show that the model-checking problem for \( \mathcal{L}_{\text{CLS}}(\text{Atm}, \text{Agt}) \) is in PTIME. We use techniques similar to those used for proving that, e.g., model-checking for ordinary modal logic or for CTL is also in PTIME \cite{8}, but there are some subtleties in dealing with the state-space being infinite (even if the valuations are bounded).

**Lemma 2.** Let \( n, d \) be nonnegative integers. Suppose that \((x, x', y, y') \in \mathbb{Z} \) are such that one of the following conditions holds:

- \( x > n + d + 1, x' = x - 1 \) and \( y' = y \),
- \( x < -n - d - 1, x' = x + 1 \) and \( y' = y \),
- \( y > n + d + 1, x = x \) and \( y' = y - 1 \),
- \( y < -n - d + 1, x' = x \) and \( y' = y + 1 \).

Then, for any \( n \)-bounded model \( M \), we have that

\[
(M, (x, y)) \models_d (M, (x', y')).
\]

**Proof.** Left to the reader.
Hence,

**Lemma 3.** Let \( \varphi \) be a \( \mathcal{L}_{\text{CL-S}}(\text{Atm}, \text{Agt}) \)-formula, \( n \) be a nonnegative integer and \( M \) be an \( n \)-bounded model. For all integers \( x, y \), we have:

- if \( x > n + \deg(\varphi) + 1 \) then \( M, (x, y) \models \varphi \) iff \( M, (x - 1, y) \models \varphi \),
- if \( x < -n - \deg(\varphi) - 1 \) then \( M, (x, y) \models \varphi \) iff \( M, (x + 1, y) \models \varphi \),
- if \( y > n + \deg(\varphi) + 1 \) then \( M, (x, y) \models \varphi \) iff \( M, (x, y - 1) \models \varphi \),
- if \( y < -n - \deg(\varphi) - 1 \) then \( M, (x, y) \models \varphi \) iff \( M, (x, y + 1) \models \varphi \).

**Proof.** By Lemmas 1 and 2. \( \square \)

Now, for all \( \mathcal{L}_{\text{CL-S}}(\text{Atm}, \text{Agt}) \)-formulas \( \varphi \) and for all integers \( z \), let \( z_{\varphi} \) be the integer defined by cases as follows:

**Case** \( |z| \leq n + \deg(\varphi) + 1 \): In that case, let \( z_{\varphi} = z \).

**Case** \( z < -n - \deg(\varphi) - 1 \): In that case, let \( z_{\varphi} = -n - \deg(\varphi) - 1 \).

**Case** \( z > n + \deg(\varphi) + 1 \): In that case, let \( z_{\varphi} = n + \deg(\psi) + 1 \).

The reader may easily verify that for all \( \mathcal{L}_{\text{CL-S}}(\text{Atm}, \text{Agt}) \)-formulas \( \varphi \) and for all integers \( z \), \( z_{\varphi} \leq n + \deg(\varphi) + 1 \). Now, given a \( \mathcal{L}_{\text{CL-S}}(\text{Atm}, \text{Agt}) \)-formula \( \varphi \), let \( (\varphi_1, \ldots, \varphi_N) \) be an enumeration of the set of all \( \varphi \)'s subformulas. Let us assume that for all \( a, b \in \{1, \ldots, N\} \), if \( \varphi_a \) is a strict subformula of \( \varphi_b \) then \( a < b \). For all \( a \in \{1, \ldots, N\} \) and for all \( (x, y) \in \mathbb{Z}^2 \), if \( |(x, y)| \leq n + \deg(\varphi_a) + 1 \) then we will associate a truth value \( tv(a, x, y) \) by case as follows:

**Case** \( \varphi_a = p \): In that case, let \( tv(a, x, y) = \langle(x, y) \in V(p) \rangle \).

**Case** \( \varphi_a = h_i \): In that case, let \( tv(a, x, y) = \langle x, y \rangle = P(i) \).

**Case** \( \varphi_a = \bot \): In that case, let \( tv(a, x, y) = \bot \).

**Case** \( \varphi_a = \neg \psi \): Let \( b \in \{1, \ldots, a\} \) be such that \( \psi = \varphi_b \). Remind that \( b < a \). In that case, if \( tv(b, x, y) = \bot \) then let \( tv(a, x, y) = \top \) else let \( tv(a, x, y) = \top \).

**Case** \( \varphi_a = \psi \vee \chi \): Let \( b, c \in \{1, \ldots, a\} \) be such that \( \psi = \varphi_b \) and \( \chi = \varphi_c \). Remind that \( b, c < a \). In that case, if \( tv(b, x, y) = \bot \) and \( tv(c, x, y) = \top \) then \( tv(a, x, y) = \top \) else let \( tv(a, x, y) = \top \).

**Case** \( \varphi_a = [\neg x] \psi \): Let \( b \in \{1, \ldots, a\} \) be such that \( \psi = \varphi_b \). Remind that \( b < a \). In that case, let \( tv(a, x, y) = tv(b, x + 1, y) \).

**Cases** \( \varphi_a = [\#] \psi \), \( \varphi_a = [\#^*] \psi \) and \( \varphi_a = [\#^*] \psi \): Similar to the previous case.

**Case** \( \varphi_a = [\#^*] \psi \): Let \( b \in \{1, \ldots, a\} \) be such that \( \psi = \varphi_b \). Remind that \( b < a \). In that case, if \( tv(b, x, y) = \bot \) for some integer \( z \geq x \) then \( tv(a, x, y) = \bot \) else let \( tv(a, x, y) = \top \).

**Cases** \( \varphi_a = [\#^*] \psi \) and \( \varphi_a = [\#^*] \psi \): Similar to the previous case.

Obviously, within a polynomial time with respect to \( |\varphi| \), one can deterministically compute the truth values \( tv(a, x, y) \) for \( a \in \{1, \ldots, N\} \) and \( (x, y) \in \mathbb{Z}^2 \) such that \( |(x, y)| \leq n + \deg(\varphi_a) + 1 \). Consequently,

**Theorem 3.** The model-checking problem for \( \mathcal{L}_{\text{CL-S}}(\text{Atm}, \text{Agt}) \) is decidable in deterministic polynomial time.

## 5. STAR-FREE FRAGMENTS

In this section and the next, we identify two decidable fragments. The first is obtained by restricting the language to \( \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \), as given by the following grammar:

\[
\begin{align*}
\alpha &::= \uparrow | \uparrow \Rightarrow | \leftarrow \alpha; \alpha' | \alpha \lor \alpha' | ? \varphi \\

\varphi &::= p | h_i | \neg \varphi | \varphi \land \psi | [\alpha] \varphi
\end{align*}
\]

We denote the set of valid formulas of \( \mathcal{L}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \) by \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \). The second is the fragment \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \) by \( \mathcal{D}_{\text{DL-S}} \). The corresponding set of valid formulas will be denoted \( \mathcal{D}_{\text{DL-S}}^0 \). Note that \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \) can be reduced to the fragment \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \):

**Lemma 4.** Every formula \( \varphi \in \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \) is equivalent to some \( \varphi^0 \in \mathcal{D}_{\text{DL-S}}^0(\text{Atm}, \text{Agt}) \).

**Proof.** It suffices to observe that the following are valid:

\[
\begin{align*}
[\alpha; \alpha'] &\models \psi \Leftrightarrow [\alpha] [\alpha'] \psi \\
[\alpha \lor \alpha'] &\models \psi \Leftrightarrow [\alpha] \psi \lor [\alpha'] \psi \\
[\theta] \psi &\models \psi \Leftrightarrow [\theta \rightarrow \psi] \\
\end{align*}
\]

With these validities, any formula of \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \) can be recursively reduced to an equivalent formula in the language \( \mathcal{D}_{\text{DL-S}}(\text{Atm}, \text{Agt}) \). \( \square \)

Our decidability proof will be based on a small model property, obtained by truncating a larger model. Fix a natural number \( n \). Given a model \( M = (P, V) \), we define \( M \models n = (P \upharpoonright n, V \upharpoonright n) \).

**Theorem 4.** For all \( \overrightarrow{x} \in \mathbb{Z}^2 \), if \( |\overrightarrow{x}| \leq m \leq n \), then \( (M, \overrightarrow{x}) \equiv_{n-m} (M \upharpoonright n, \overrightarrow{x}) \).

**Proof.** The proof proceeds by a standard induction on \( m \). The atoms and position clauses are trivial since \( x \leq n \) and the values of atomic propositions is not changed. For the inductive case, consider (for example) \( \alpha = \Rightarrow \). Then, if \( \overrightarrow{x} = (x_0, x_1) \), \( \forall \overrightarrow{x} \models \psi \) if and only if \( \psi = (x_0 + 1, x_1) \). Clearly \( |\overrightarrow{y}| \leq m + 1 \), so that by the induction hypothesis, \( (M, \overrightarrow{x}) \equiv_{n-m-1} (M \upharpoonright n, \overrightarrow{y}) \), as needed. \( \square \)

With this we obtain our first decidability result.
THEOREM 4. The logics DL-S\(^0\), DL-S are decidable. In particular, DL-S\(^0\) is in NP.

PROOF. Since DL-S can be reduced to DL-S\(^0\), it suffices to show that the latter is decidable. Suppose that \(\varphi\) is satisfied on some model \(M\); without loss of generality, we can assume that \(\varphi\) is satisfied on the origin. Let \(n\) be the modal degree of \(\varphi\). By Lemma 5 \((M, 0) \simeq_n (M \upharpoonright n, 0)\), so by Lemma 6 \(\varphi\) is also satisfied on \((M \upharpoonright n, 0)\). It follows that \(\varphi\) is satisfiable if and only if it is satisfiable on the class of models such that \(\mathcal{P}\) and \(\mathcal{V}\) are both \((n + 1)\)-bounded, so it remains to enumerate all such models and check whether any of them satisfy \(\varphi\). Note that the size of any \((n + 1)\)-bounded model is \(O(n^2 \times \text{size}(\varphi))\), so the complexity bound for DL-S\(^0\) follows.

Observe that it does not follow from our techniques that DL-S is in NP, since the reduction procedure is not polynomial.

Now, our aim in this section will be to completely axiomatize DL-S\(^0\). In this respect, we need the following axioms and inference rules, besides all tautologies of propositional calculus:

- All axioms and inference rules saying that \([\varphi], [\psi], [\Rightarrow]\) and \([-\Rightarrow]\) are normal modalities,
- \([a] \varphi \leftrightarrow (a) \varphi\) for each \(a \in \{\uparrow, \downarrow, \Rightarrow, \Leftarrow\}\),
- \(\varphi \rightarrow [\uparrow](\psi)\varphi\) and \(\varphi \rightarrow [\downarrow](\check{\psi})\varphi\),
- \(\varphi \rightarrow [\Rightarrow](\psi)\varphi\) and \(\varphi \rightarrow [\Leftarrow](\check{\psi})\varphi\),
- \([\alpha_1, \alpha_2] \varphi \leftrightarrow [\alpha_2]([\alpha_1] \varphi)\) for each \(\alpha_1, \alpha_2 \in \{\uparrow, \downarrow, \Rightarrow, \Leftarrow\}\),
- \(h_i \rightarrow [\uparrow]^x \Rightarrow [\Rightarrow]^{y - h_i}\) for all integers \(x, y\) such that \(x \neq 0\) or \(y \neq 0\).

We will say that a formula \(\varphi \in \mathcal{L}_{DL-S}(Atm, \mathcal{A}gt)\) is derivable if it belongs to the least set of \(\mathcal{L}_{DL-S}(Atm, \mathcal{A}gt)\)-formulas containing the above axioms and closed under the above inference rules.

THEOREM 5. Let \(\varphi\) be an \(\mathcal{L}_{DL-S}(Atm, \mathcal{A}gt)\)-formula. The following conditions are equivalent: (i) \(\varphi\) is derivable; (ii) \(\varphi\) is valid.

PROOF. (i)\(\Rightarrow\) (ii): It suffices to check that all axioms are valid and that all inference rules preserve validity.

(ii)\(\Rightarrow\) (i): Suppose \(\varphi\) is not derivable. Let \(d\) denote the modal degree of \(\varphi\). By Lindenbaum’s Lemma, let \(\Gamma\) be a maximal consistent set of formulas such that \(\varphi \notin \Gamma\). Remark that for all \(a \in \mathcal{A}gt\), there exists at most one pair \((x, y)\) of (negative or positive) integers such that \([\uparrow]^x \Rightarrow [\Rightarrow]^{y}\) \(h_i \in \Gamma\). Let \(\mathcal{A}gt(\Gamma)\) be the set of all \(a \in \mathcal{A}gt\) such that \([\uparrow]^x \Rightarrow [\Rightarrow]^{y}\) \(h_i \in \Gamma\) for some pair \((x, y)\) of integers such that \(|(x, y)| \leq d\). Let \(M = (\mathcal{P}, \mathcal{V})\) be the spatial model defined as follows:

- For all \(a \in \mathcal{A}gt\), if \(a \in \mathcal{A}gt(\Gamma)\) then let \(\mathcal{P}(i)\) be the unique pair \((x, y)\) of integers such that \([\uparrow]^x \Rightarrow [\Rightarrow]^{y}\) \(h_i \in \Gamma\), else let \(\mathcal{P}(i)\) be \((d + 1, 0)\),
- For all pairs \((x, y)\) of integers, if \(|(x, y)| \leq d\) then let \(\mathcal{V}(x, y) = \{p \in \mathcal{A}tm : [\Rightarrow]^{x}[\uparrow]^{y}p \in \Gamma\}\), else let \(\mathcal{V}(x, y) = \emptyset\).

The reader may easily prove by induction on \(\psi\) that if \(\psi\) is a subformula of \(\varphi\) then for all pairs \((x, y)\) such that \(|(x, y)| \leq \text{deg}(\varphi) - \text{deg}(\psi)\), \(M, (x, y) \models \psi\) iff \([\uparrow]^x[\check{\uparrow}]^y\psi \in \Gamma\). Since \(\varphi \notin \Gamma\), therefore \(M, (0, 0) \not\models \varphi\). Thus, \(\varphi\) is not valid.

6. COMPASS LOGIC OF POSITIONS

Next we consider the fragment \(\mathcal{L}_{CL-P}\), defined in Section 3. Since there are no atomic propositions, models are somewhat simpler.

DEFINITION 5. A position model is a function \(\mathcal{P}: \mathcal{A}gt \rightarrow \mathbb{Z} \times \mathbb{Z}\).

That is, a position model is just a spatial model without a valuation for atomic propositions. As we will show, position models do not need to have big ‘gaps’ if we only care about satisfiability of \(\mathcal{L}_{CL-P}\)-formulas. This will give us a small model property.

DEFINITION 6. Let \(\mathcal{P}\) be a position model. A vertical gap is a set \(G = [a, b] \times \mathbb{Z}\) such that for all \(i \in \mathcal{A}gt\), \(\mathcal{P}(i) \notin G\). If \((x, y) \in G\), we say that the depth of \((x, y)\) in \(G\) is \(\min(x - a, b - x)\), and \(G_m\) denotes the set of elements of depth at least \(m\); observe that \(G_0 = G\), and \(G_m\) is also a gap when non-empty. The removal of \(G\) is the function \(\rho\) given by \(\rho(x, y) = (x', y)\) where \(x' = x\) if \(x \leq a\), \(x' = \min(a, x - (b - a))\) otherwise.

A horizontal gap is defined analogously, but is of the form \(\mathbb{Z} \times [a, b]\). The depth and the removal are defined analogously as well.

In this section we use \(\simeq_n\) for \(n\)-bisimilarity with respect to all basic relations of \(\mathcal{L}_{CL-P}\).

LEMMA 6. Let \(G = [a, b] \times \mathbb{Z}\) be a vertical gap and \(\mathcal{P}\) be a position model. Then, if \((x, y), (x', y) \in G_m\), it follows that \(\mathcal{P}(x, y) \simeq_m \mathcal{P}(x', y)\).

The analogous claim holds for horizontal gaps.

PROOF. We proceed by induction on \(m\). The atomic clauses are straightforward since, if \((x, y), (x', y) \in G_0 = G\), then they satisfy no atoms.

For the other clauses, assume the claim inductively for \(m\), and suppose that \((x, y), (x', y) \in G_{m+1}\). Any ‘vertical’ program \((\check{\uparrow}, \check{\downarrow}, \check{\Rightarrow}, \check{\Leftarrow})\) stays within \(G_{m+1} \subseteq G_m\) so we may immediately apply the induction hypothesis. For example, if \((u, v)R_{x, y}(x, y)\), then \(u = x\) and \(v \geq y\); hence, \((x', y)R_{x, y}(x', v)\) and by the induction hypothesis, \((u, v) = (x, v) \equiv_m (x', v)\). The ‘back’ clauses and the rest of the vertical programs are entirely symmetrical.

Next consider a ‘horizontal’ program: \(\Leftarrow, \Rightarrow, \Leftarrow^*, \Rightarrow^*\). By symmetry, we will only consider the ‘forth’ clauses of the ‘right’ programs. We have that \(R_{x, y}\) is a function; specifically, \(R_{x, y}(x, y) = (x + 1, y)\). Observe that \((x + 1, y) \in G_m\), and similarly \((x' + 1, y) \in G_m\). But, by the induction hypothesis, \((x + 1, y) \equiv_m (x', y)\), as needed.

Now suppose that \((x, y)R_{x, y}(u, v)\), so that \(u \geq x\) and \(v = y\). We consider two cases. If also \(u \geq x'\), then we also have \((x', y)R_{x, y}(u, v)\), and we may use the same witness. Otherwise, \(x \leq u < x'\), which means that \((u, v) \in G_{m+1} \subseteq G_m\) so by the induction hypothesis \((x, v) \equiv_m (x', y)\). But also, \((x', y)R_{x, y}(x', v)\), and we can use it as our witness.

As mentioned, the other clauses are entirely symmetrical and left to the reader. By induction on \(m\), the claim follows.

The analogous claim for horizontal gaps is also entirely analogous.
Proof. We prove by induction on \( k \leq m \), that \( \mathcal{P}, \bar{x} \vDash_k \rho \mathcal{P}, \rho(x) \). For \( k = 0 \) this is clear, since if \( (x, y) \in G_m \), no nominal occurs on \( (x, y) \) or on \( \rho(x, y) = (a, y) \). Otherwise, \( (x, y) = \mathcal{P}(i) \) if and only if \( \rho(x, y) = \rho \mathcal{P}(i) \).

Now, assume the claim for \( k \), and let \( \rho(x, y) = (x', y') \). The ‘forth’ clauses for \( \alpha \in \{ R^*_a, R^*_b, R^*_c, R^*_d \} \) follow by observing that if \( (x, y) R_a(u, v) \), then \( \rho(x, y) R_a \rho(u, v) \); for example, if \( \alpha = \downarrow \), then we must have \( u = x \) and \( v = y - 1 \), and since \( \rho \) fixes the \( y \) coordinate we have that if \( (x, y) = (x', y') \), then \( \rho(u, v) = (x', y - 1) \), as needed. Similarly, for the ‘back’ clause, if \( \rho(x, y) = (x', y') \) and \( (x', y') R_a(u, v) \), we must have \( u = x' \) and we can readily observe that \( (x, y) R_a(u, v) \) and \( \rho(x, v) = (x', v) \), so that by the induction hypothesis, \( \mathcal{P}, (x, y) \vDash_k \rho \mathcal{P}, \rho(x', v) \).

Next we look at \( \alpha \in \{ \leftarrow, \Rightarrow, \leftarrow^*, \Rightarrow^* \} \). First, we check the ‘forth’ clauses. If \( (x, y) R_a(u, v) \), then \( u = x - 1 \) and \( v = y \). If \( x \notin (a, b] \), then it readily follows that \( \rho(x, y) R_a \rho(u, v) \), and we may use the induction hypothesis. If instead \( x \in (a, b] \), then \( \rho(u, v) = \rho(x, y) = (a, y) \). However, \( R_a(a, y) = (a - 1, y) \in G_{m-1} \), so by the induction hypothesis and Lemma 6

\[
\rho \mathcal{P}, (a - 1, y) \vDash_k \mathcal{P}, (a - 1, y) \vDash_k \mathcal{P}, (x - 1, y),
\]
as needed. For \( \alpha = \leftarrow^* \), suppose \( (x, y) R_a \rho^*(u, v) \). Then, \( y = v \), and since \( \rho \) is non-decreasing on the first component, we also have \( \rho(x, y) R_a \rho(u, v) \). The cases for the ‘right’ programs are similar.

Finally, we check the ‘back’ clauses for the horizontal programs. Observe that \( R_a, R_a^* \) are functional, so the ‘forth’ and ‘back’ clauses are equivalent. Hence we consider only \( R_a^*, R_a^{**} \). If \( (x', y') R_a^*(u, v) \), then consider two cases. If \( u \leq a \), then \( \rho(u, v) = (u, y) \) and \( u \leq x' \leq x \), so we have that \( (x, y) R_a(u, v) \) and we may use the induction hypothesis on \( (u, y) \). If \( u > a \), then \( \rho(u + b - a, y) = (u, y) \), and we may use the induction hypothesis on \( (u + b - a, y) \). But note that, in this case, we must have that \( x = x' + b - a, \) so \( (x, y) R_a(u + b - a, y) \).

Finally, if \( (x', y') R_a^{**}(u, v) \), again consider two cases. If \( u < a \), then \( \rho(u, y) = (u, y) \) and \( u \geq x' = x \), so we have that \( (x, y) R_a(u, v) \) and we may use the induction hypothesis on \( (u, y) \). If \( u \geq a \), then \( \rho(u + b - a, y) = (u, y) \), and we may use the induction hypothesis on \( (u + b - a, y) \). Note that, in this case, \( x \leq x' + b - a \leq u + b - a, \) so \( (x, y) R_a(u + b - a, y) \), as needed.

The case for a horizontal gap is similar. \[\Box\]

Theorem 6. If \( \varphi \in \mathcal{L}_{\mathcal{C}_L, \mathcal{P}} \) is satisfiable, it is satisfiable on a position model where all coordinates of positions are bounded by \( 2(|\varphi| + 1)^2 \).

Proof. Assume that \( \varphi \) is satisfied on some position model \( \mathcal{P} \). Suppose that \( x_1 \leq \ldots \leq x_n \) are the \( x \)-coordinates of all positions of agents such that \( \mathcal{P} \) appears in \( \varphi \), together with the evaluation point, \((0,0)\) (note that \( n \leq |\varphi| + 1 \)). If for some \( i < n \) we have that \( x_{i+1} - x_i > 2(|\varphi| + 1) \), then \( G = (x_i, x_{i+1}) \times Z \) is a horizontal strip with \( G_{|\varphi|} \) having width at least two, so that its removal is not the identity.

Now, if the \( x_i \)'s are not bounded by \( 2(|\varphi| + 1)^2 \), note that such a gap must exist so we can remove it. After enough iterations, we can bound all \( x_i \)'s. Then we proceed to bound the vertical components analogously. \[\Box\]

Theorem 7. Satisfiability of \( \mathcal{L}_{\mathcal{C}_L, \mathcal{P}} \)-formulas is in \( NP \).

7. SPACE AND MOVEMENT

DL-S studied in the previous sections is a logic for representing static properties of the bidimensional space. Specifically, in DL-S, positions of agents in the space do not change. The aim of this section is to extend \( \mathbb{L}_{DL-S}(Atm, Agt) \) by programs describing the agents’ movements in the bidimensional space. We assume that agents act in a synchronous way (i.e., they act in parallel). We call the resulting language \( \mathbb{L}_{DL-SM}(Atm, Agt) \) and the resulting logic DL-SM (Dynamic Logic of Space and Moving).

7.1 Syntax

In \( \mathbb{L}_{DL-SM}(Atm, Agt) \), agent \( i \) is associated with her corresponding repertoire of actions \( Act_i = \{ \uparrow_i, \downarrow_i, \leftarrow_i, \Rightarrow_i, nil \} \). \( \uparrow_i \) is agent \( i \)'s action of moving up, \( \downarrow_i \) is agent \( i \)'s action of moving down, \( \leftarrow_i \) is agent \( i \)'s action of moving left, \( \Rightarrow_i \) is agent \( i \)'s action of moving right and \( nil \) is agent \( i \)'s action of doing nothing.

The set of joint actions is defined to be \( \Delta = \prod_{i \in Agt} Act_i \). Elements of \( \Delta \) are denoted by \( \delta, \delta', \ldots \). For every \( \delta \in \Delta, \delta_i \) denotes the element in \( \delta \) corresponding to agent \( i \).

Since the logic DL-S* is undecidable, we start from its decidable star-free fragment as the basis of our dynamic extension by programs describing the agents’ movements.

The language, denoted by \( \mathbb{L}_{DL-SM}(Atm, Agt) \), is defined by the following grammar in Backus-Naur Form:

\[
\begin{align*}
\alpha &::= \uparrow \mid \downarrow \mid \Rightarrow \mid \leftarrow \mid \alpha \mid \alpha' \mid \alpha \cup \alpha' \mid \varphi \\
\beta &::= \delta \mid \delta' \mid \delta \cup \delta' \mid \varphi \\
\varphi &::= p \mid \neg \varphi \mid \varphi \land \psi \mid [\alpha] \varphi \mid [\beta] \varphi
\end{align*}
\]

where \( p \) ranges over \( Atm \) and \( i \) ranges over \( Agt \). Instances of \( \beta \) are called movement programs.

7.2 Semantics

The semantics is a model update semantics as in the style of dynamic epistemic logic (DEL) [17].

Definition 7 \( (R_{\beta}^{(x,y)} \text{ and truth conditions}) \). Let \( M \in \mathbb{M} \) be a spatial model. For all movement programs \( \beta \), for all formula \( \varphi \) and for all positions \( (x, y) \), the binary relation \( R_\beta^{(x,y)} \) on \( \mathbb{M} \times \mathbb{M} \) and the truth conditions of \( \varphi \) in \( M \) are defined by parallel induction as follows. (We only give the truth condition for \( [\beta] \varphi \) as the truth conditions for the boolean constructs and for \( [\alpha] \varphi \) as in DL-S*):

\[
\begin{align*}
R_\delta^{(x,y)} &::= \{(M, M') : \forall \psi \in Agt, \mathcal{P}'(i) = \mathcal{P}'(i) \} \\
R_{\beta_1 \beta_2}^{(x,y)} &::= R_{\beta_1}^{(x,y)} \circ R_{\beta_2}^{(x,y)} \\
R_{\beta_1 \cup \beta_2}^{(x,y)} &::= R_{\beta_1}^{(x,y)} \cup R_{\beta_2}^{(x,y)} \\
R_{\varphi}^{(x,y)} &::= \{(M, M) : (x, y) \models \varphi \}
\end{align*}
\]

where:
the equivalences of Proposition 1 together with this allow to
the former.

Given the decidability result and the complete

7.3 Decidability and axiomatization

The aim of this section is to show how the satisfiability
problem of DL-SM can be reduced to the satisfiability problem
of DL-S. Given the decidability result and the complete
axiomatization for the latter of Section 5, this reduction will
provide a decidability result as well as an axiomatization for
the former.

Proposition 1. The following L_{DL-SM}(Atm, Agt)-formulas are valid:

\[ \alpha; \alpha' \varphi \leftrightarrow [\alpha][\alpha']\varphi \]  (1)
\[ \alpha \cup \alpha' \varphi \leftrightarrow ([\alpha]\varphi \land [\alpha']\varphi) \]  (2)
\[ [\beta; \beta'] \varphi \leftrightarrow [\beta][\beta']\varphi \]  (3)
\[ [\beta \cup \beta'] \varphi \leftrightarrow ([\beta]\varphi \land [\beta']\varphi) \]  (4)
\[ [\varphi] \varphi \leftrightarrow (\varphi \rightarrow \varphi) \]  (5)
\[ [\delta] p \leftrightarrow p \]  (6)
\[ [\delta] h_1 \leftrightarrow [F_i(\delta)] h_1 \]  (7)
\[ [\delta] \lnot \varphi \leftrightarrow [\lnot \delta] \varphi \]  (8)
\[ [\delta] (\varphi \land \psi) \leftrightarrow ([\delta] \varphi \land [\delta] \psi) \]  (9)
\[ [\delta][\alpha] \varphi \leftrightarrow [\alpha][\delta] \varphi \]  (10)

with \alpha \in \{ \top, \bot, \Rightarrow, \Leftarrow \} and where the function \( F_i \) is defined as follows:

\[ F_i(\delta) = \top \text{ if } \delta_i = \top_i \]
\[ F_i(\delta) = \bot \text{ if } \delta_i = \bot_i \]
\[ F_i(\delta) = \Rightarrow \text{ if } \delta_i = \Leftarrow \]
\[ F_i(\delta) = \Leftarrow \text{ if } \delta_i = \Rightarrow \]
\[ F_i(\delta) = \top \text{ if } \delta_i = \text{nil}_i \]

As the following rule of replacement of equivalents preserves validity (even if \( \psi \) appears inside modalities in \( \varphi \)):

\[ \psi_1 \leftrightarrow \psi_2 \]
\[ \varphi \leftrightarrow [\varphi](\psi_1 \lor \psi_2) \]  (11)

the equivalences of Proposition 1 together with this allow to
find for every L_{DL-SM}(Atm, Agt)-formula an equivalent formula of L_{DL-S}(Atm, Agt) studied in Section 5. Call the
mapping which iteratively applies the equivalences of Proposition 1 from the left to the right, starting from one of the
innermost modal operators. \( \text{red} \) pushes the dynamic operators \[ [\beta] \] inside the formula, and finally eliminates them when
facing an atomic formula. The mapping \( \text{red} \) is inductively defined by:

1. \( \text{red}(p) = p \)
2. \( \text{red}(h_i) = h_i \)
3. \( \text{red}(\lnot \varphi) = \lnot \text{red}(\varphi) \)
4. \( \text{red}(\varphi \lor \psi) = \text{red}(\varphi) \lor \text{red}(\psi) \)
5. \( \text{red}([[\alpha]\varphi] = [[\alpha]\text{red}(\varphi)] \) with \( \alpha \in \{ \top, \bot, \Rightarrow, \Leftarrow \} \)
6. \( \text{red}([[\alpha]\varphi] = [[\alpha]\text{red}(\varphi)] \) with \( \alpha \in \{ \top, \bot, \Rightarrow, \Leftarrow \} \)
7. \( \text{red}([[\alpha \cup \alpha']\varphi] = ([\alpha]\text{red}(\varphi) \land [\alpha']\text{red}(\varphi)) \)
8. \( \text{red}([[\alpha] \varphi] = \text{red}(\lnot (\varphi \land \psi)) \)
9. \( \text{red}([[\delta]p] = p \)
10. \( \text{red}([[\delta]h_i] = [F_i(\delta)] h_i \)
11. \( \text{red}([[\delta] \lnot \varphi] = [\lnot \delta] \varphi \)
12. \( \text{red}([[\delta] (\varphi \land \psi)] = [\lnot \delta] \varphi \land [\lnot \delta] \psi \)
13. \( \text{red}([[\delta][\alpha] \varphi] = [\alpha][\delta] \varphi \)
14. \( \text{red}([\beta; \beta']\varphi = [\beta][\beta']\varphi \)
15. \( \text{red}([\beta \cup \beta']\varphi = ([\beta][\beta']\varphi \land [\beta'][\beta]\varphi) \)

Proposition 2. Let \( \varphi \in L_{DL-SM}(Atm, Agt) \). Then, \( \varphi \leftrightarrow \text{red}(\varphi) \) is valid.

Decidability of the satisfiability problem of DL-SM follows straightforwardly from the decidability of the star-free
fragment L_{DL-S}(Atm, Agt) of DL-S* (Theorem 4). Indeed, \( \text{red} \) provides an effective procedure for reducing a formula \( \varphi \in L_{DL-SM}(Atm, Agt) \) into an equivalent formula \( \text{red}(\varphi) \in L_{DL-S}(Atm, Agt) \).

Theorem 8. The logic DL-SM is decidable.

Thanks to the completeness result for the star-free
fragment of DL-SM and the reduction axioms of Proposition 1 we can state the following theorem.

Theorem 9. The logic DL-SM is completely axiomatized by the axioms and rules of inference of the star-free fragment
of DL-SM given in Section 5, the valid formulas of Proposition 1 and the rule of replacement of equivalents.

8. PERSPECTIVES

Before concluding the paper, we discuss two perspectives
for the extension of the logic DL-S* and DL-SM by concepts
of perceptual knowledge and coalitional capability.

Perceptual knowledge.

DL-S* and DL-SM support reasoning about properties of
the 2D space as well as about positions and movements of
agents in the 2D space. However, an agent in the space does
not only move but also sees where other agents are, how
the space around her is, what other agents do, etc. More
generally, agents in the space have perceptual knowledge (i.e.,
knowledge based on what they see). We want to propose
here a simple extension of DL-S* and DL-SM by modal
operators of perceptual knowledge. Specifically, we consider
epistemic-like operators of type S^k_x describing what an agent
could see from her current position, if she had a range of a
vision of size \(k \in \mathbb{N}\). An agent’s range of vision of size \(k\) corresponds to the square centered at the agent’s position with side length equal to \(2 \times k\). We call the latter agent’s neighborhood of size \(k\).

In order to provide an interpretation of the operator \(S^i_k\), the following concept of indistinguishibility is required. Let \(i \in \text{Agt}\) and let \(M = (\mathcal{P}, V)\) and \(M' = (\mathcal{P}', V')\) be two spatial models. We say that \(M\) and \(M'\) are indistinguishable for agent \(i\) given her current position and her range of vision of size \(k\), denoted by \(M \sim^i_k M'\), if and only if:

\[
\forall (x, y) = V(x, y) \\
\mathcal{P}'(j) = \mathcal{P}(j)
\]

for all \((x, y) \in \mathbb{Z} \times \mathbb{Z}\) and for all \(j \in \text{Agt}\) such that \((x, y) \in \mathcal{D}(i, k)\) and \(\mathcal{P}(j) \in \mathcal{D}(i, k)\) with

\[
\mathcal{D}(i, k) = \{(x', y') : \mathcal{P}_x(i) - k \leq x' \leq \mathcal{P}_x(i) + k\}
\]

where \(\mathcal{P}_x(i)\) and \(\mathcal{P}_y(i)\) are, respectively, the \(x\)-coordinate and \(y\)-coordinate in \(\mathcal{P}(i)\).

This notion of indistinguishibility is essential to provide a notion of the following fundamental property:

\[
\text{S5 modality, as its interpretation requires to change the reference point from (} x, y \text{) to } \mathcal{P}(i) \text{ (although a different approach is possible, as in e.g. [18]). Nonetheless, } S^i_k \text{ satisfies KD45, namely:}
\]

\[
(S^i_k \varphi \land S^i_k (\varphi \rightarrow \psi)) \rightarrow S^i_k \psi \tag{12}
\]

\[
\neg(S^i_k \varphi \land S^i_k \neg \varphi) \tag{13}
\]

\[
S^i_k \varphi \rightarrow S^i_{k+1} \varphi \tag{14}
\]

\[
S^i_k \varphi \rightarrow S^i_k \neg \varphi \tag{15}
\]

\[
\varphi \land \neg \varphi \tag{16}
\]

as well as the following form of “local reflexivity” property:

\[
(h_i \land S^i_k \varphi) \rightarrow \varphi \tag{17}
\]

\(S^i_k\) satisfies additional principles that are proper to its spatial interpretation. For instance, let

\[
\mathcal{Pr}(k) = \{e^h \varphi^h : 0 \leq h \leq k\} \cup \{\neg e^h \varphi^h : 0 \leq h \leq k\} \cup \{e^h \psi^h : 0 \leq h \leq k\} \cup \{\neg e^h \psi^h : 0 \leq h \leq k\}
\]

be the set of spatial programs that allow to reach all and only those points in an agent’s neighborhood of size \(k\). Then, under the previous interpretation of the operator \(S^i_k\), the following formulas become valid for every \(\alpha \in \mathcal{Pr}(k)\):

\[
h_i \rightarrow ([\alpha]p \leftrightarrow S^i_k [\alpha]p) \tag{18}
\]

\[
h_i \rightarrow ([\alpha]h_i \leftrightarrow S^i_k [\alpha]h_i) \tag{19}
\]

This means that if an agent \(i\) has a range of vision of size \(k\), then she can perceive all facts that are true and all agents that are positioned in her neighborhood of size \(k\). The following formula is an example of instance of the previous validity:

\[
h_i \rightarrow ([\pi]h_j \leftrightarrow S^i_k [\pi]h_j) \tag{20}
\]

The latter means that if agent \(i\) has a range of vision of size \(1\) then, agent \(j\) is above her iff agent \(i\) perceives this.

We postpone to future work a study of the complexities of model-checking and of decidability for the extensions by epistemic operators \(S^i_k\) of the different logics presented in the paper.

**Coalitional capability.**

DL-SM provides an interesting basis for the development of a logic of coalitional capabilities in the two-dimensional space. We take the concept of ‘coalitional capability’ in the sense of Coalition Logic \(\text{CL} [14]\). Specifically, we say that coalition \(C \subseteq \text{Agt}\) has the capability of ensuring \(\varphi\), denoted by \(\langle C \rangle \varphi\), if and only if “there exists a joint action \(\delta_C\) of coalition \(C\) such that, by performing it, outcome \(\varphi\) will be ensured, no matter what the agents outside \(C\) decide to do”.

The extension of DL-SM by coalitional capability operators \(\langle C \rangle\) is rather simple, as the agents’ action repertoires only includes the four basic movements in the plane \(\{(h_1, \psi) \in \mathbb{R}^2 \text{ and } \varphi\}\). Following Section 5 for each coalition \(C \subseteq \text{Agt}\) we define its set of joint of actions \(\Delta_C = \bigcap_{c \in C} Act_c\) and denote elements of \(\Delta_C\) by \(\delta_C, \delta_{C'}\), . . . Then, the truth condition of the operator \(\langle C \rangle\) goes as follows: \(M, (x, y) \models \langle C \rangle \varphi\) if and only if \(\exists \delta_C \in \Delta_C\) such that

\[
\forall \delta'_{Agt \setminus C} \in \Delta_{Agt \setminus C} : M, (x, y) \models [\delta_C, \delta'_{Agt \setminus C}] \varphi.
\]

Since \(\delta_C\) and \(\delta'_{Agt \setminus C}\) are finite, \(\langle C \rangle \varphi\) is expressible in DL-SM but at the price of an exponential blowup in the size of \(\varphi\). It is easy to check that the operator \(\langle C \rangle\) satisfies the following basic principles of coalition logic [14]:

\[
\neg\langle C \rangle \bot \tag{21}
\]

\[
\langle C \rangle \top \tag{22}
\]

\[
\neg (\emptyset) \rightarrow \langle \text{Agt} \rangle \varphi \tag{23}
\]

\[
\langle C \rangle (\varphi \land \psi) \rightarrow [\langle C \rangle \varphi \land \langle C \rangle \psi] \tag{24}
\]

\[
\langle C_1 \cup C_2 \rangle (\varphi \land \psi) \rightarrow [\langle C_1 \rangle \varphi \land \langle C_2 \rangle \psi] \tag{25}
\]

\[
\varphi \leftrightarrow \psi \tag{26}
\]

\(\langle C \rangle\) satisfies additional principles that are proper to its spatial interpretation. For instance, it is easy to check that, under the previous interpretation, the following two formulas become valid:

\[
\neg \langle C \rangle h_i \text{ if } i \notin C \tag{27}
\]

\[
([\pi]h_i \lor [\neg]h_i) \lor ([\psi]h_i \lor ([\neg]h_i) \rightarrow [\langle i \rangle]h_i) \tag{28}
\]

The two validities captures the basic idea that an agent has exclusive control of her position in the sense that: (i) if coalition \(C\) does not include agent \(i\) then \(C\) cannot force \(i\) to be “here”, and (ii) agent \(i\) has the capability to move “here” if she is “around”.

We postpone to future work a more systematic analysis of the basic principles of the operator \(\langle C \rangle\) as well as a study of a strategic capability operator in the sense of ATL [1] based on the semantics of the logic DL-SM.
REFERENCES