Developing Bounded Reasoning *

Michał Walicki, Marc Bezem, Wojtek Szajnkenig
Department of Informatics, University of Bergen
PB. 7800, N-5020 Bergen, Norway
{*michal,bezem*}*@ii.uib.no*

Abstract. We introduce a three-tiered framewok for modelling and reasoning about agents who (i) can use possibly complete reasoning systems without any restrictions but who nevertheless are (ii) bounded in the sense that they never reach infinitely many results and, finally, who (iii) perform their reasoning in time. This last aspect does not concern so much the time it takes for agents to actually carry out their reasoning, as the time which can bring about external changes in the agents' states such as arriving of new information or discarding previously available information due to bounds of the agent's resources.

These three aspects are treated with the maximal possible degree of independence from each other. The treatment of layer (iii) can be combined with arbitrary logic at level (ii) which, in turn, can be combined with arbitrary agent logic at level (i). At the level (iii), we discuss briefly the duality (or rather, complementarity) of system descriptions based on actions and transitions, on the one hand, and states and their changes, on the other. We settle for the latter and present a simple language, for describing state changes, which is parameterized by an arbitrary language for describing properties of the states. The language can be viewed as a simple fragment of step logic, admitting however various extensions by appropriate choices of the underlying logic. Alternatively, it can be seen as a very specific fragment of temporal logic (with a variant of 'until' or 'chop' operator), and is interpreted over dense linear time. The reasoning system presented here is sound, as well as strongly complete and decidable (provided that so is the parameter logic for reasoning about single states). We give the main idea of the completeness proof and suggest a wide range of possible applications, which is a simple consequence of the parametric character of both the language and the reasoning system. We address then in more detail the case of non-omniscient rational agents, (ii). Their models are syntactic structures (sets of available formulae) similar in spirit, though not in the technical formulation, to the models used in other syntax oriented approaches and in active logic. We give a new sound, complete and decidable system for reasoning about such agents. Finally, we illustrate its extensions with the internal reasoning of agents, (i), by equipping them with some example logics.

* We gratefully acknowledge partial financial support from the Norwegian Research Council under the projects MoSIS and SHIP.

1. Introduction

The problem of modelling and reasoning about bounded agents interacting with each other and with their environment can be divided into three aspects or layers which, although related to each other, carry some degree of relative independence. At the lowest level, there are agents capable of internal reasoning. These agents are bounded and the bounds are not so much induced by their internal logic as by other factors, like limited resources or available time. Ideally, one would therefore like to achieve a model of boundedness which can be combined with arbitrary agent logics. Finally, at the highest level of abstraction, agents can be seen as operating in time, that is, changing their states. These changes result, to some extent, from their internal reasoning. However, and perhaps most significantly, changes can occur due to external influences, like receiving input from other agents, acquiring new information from observing the environment or discarding some pieces of information due to limited capacity of the storage.

The paper introduces a framework where these three layers are modelled in a relative independence from each other. At the highest level, we introduce sequence logic which allows one to reason about sequences of state changes. It is parameterized by an arbitrary underlying logic for reasoning about single states. Then, we introduce a logic of bounded agents in which one can reason about agents' knowledge expressing not only, as classical epistemic logics, its lower but also the upper bounds. This logic addresses *only* the aspect of boundedness and can be combined with arbitrary agent logic. At the lowest level, agents can thus be equipped with reasoning mechanisms depending on the actual applications and we give various examples of such instantiations. The main contribution of the paper is this separation of the three aspects and the means for modelling each of them. However, in actual applications, one often faces the need for these aspects to interact. We give a general pattern for modelling such interactions and give some examples. This issue is not, however, investigated in full detail and remains the main object for further research. The following two subsections sketch the main ideas of the two primary issues discussed in the paper: sequence logic and logic of bounded agents.

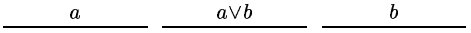
1.1. Sequence logic

In the description of reactive systems one has focused primarily on their capability to perform some specific *actions* (process algebras, labelled transition systems, CSP). For example, the famous vending machine can perform the actions of 'accepting a coin' and then 'dispense a coffee' an unspecified number of times. This is certainly a fruitful approach. However, one reason to be interested in actions (and maybe the only one) is that they change the *state* of the world, an agent's beliefs, or any other abstraction. A vending machine can be described equivalently as a device which can stay in a state of 'inactivity', from which it can pass to a state of 'having accepted a coin' and then to one in which it has 'dispensed a coffee'. The latter state may be identified with the state of inactivity if one, among other things, abstracts from the number of coins accepted and from the remaining amount of coffee. These views are in some sense dual, but we present an approach related to the second one, that is, we will describe systems in terms of evolving states. The states evolve over time and during consecutive time intervals certain specifications, that is, partial descriptions of the states, can be observed.

For instance, a description of a system might be as follows. At first an agent knows (or assumes) a. After an announcement he is no longer sure, and knows only $a \vee b$. Finally, after yet another event, he learns that b and retains this knowledge (for the rest of the time: here the scenario ends). This system is described as an expression consisting of a sequence of formulae, each partially describing the state(s), during three consecutive intervals:

$$a; a \vee b; b \tag{1}$$

The meaning of this expression could be described on a linear time scale as



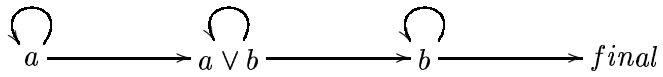
Here, lines represent the 'duration' of the state(s) satisfying the formula which annotates it. One further abstraction is that we view 'duration' as something qualitative but not quantitative. Thus all intervals have been given the same length. Throughout the paper we will depict intervals with different lengths, for convenience, not to suggest different durations.

Viewing the above as a description of the result of some interactions (the announcement and the other event), it is natural to ask for possible consequences, and for an entailment relation between such descriptions in general. For example, a system

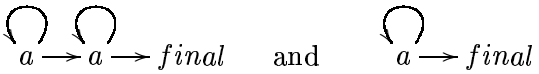
$$a \vee \neg b; a \vee b \tag{2}$$

can be viewed as a consequence of the previous one, with the last two intervals concatenated. Furthermore, $a \wedge b$ during a certain interval has both (1) and (2) as consequence by appropriately cutting the interval in three and two smaller ones, respectively.

Although the paper takes a logical approach, it is possible to interpret the systems above with the help of labelled transition systems. As a consequence of the focus on states instead of on actions, there is just one label. For example, the transition system corresponding to (1) is



In our setting, 'being in a state satisfying a specification' is assumed to last for some time, whereas the transitions are instantaneous. The loops in the transition system express this assumption that considered intervals can always be split into smaller ones, that is, the density of the linear ordering modelling the time domain. Transitivity enables the concatenation of intervals. Traces should include the states and are therefore taken to be terminating reduction sequences. Systems are equivalent if they have a trace in common. Equivalent systems have actually infinitely many traces in common, but the trace sets may be different. For example, $a;a$ and a, represented respectively as



are equivalent, but have different trace sets. The consequence relation between systems can also be expressed in terms of transition systems, see Section 2.2.3.

1.2. Bounded agents

Modelling bounded agents with classical epistemic logics, one encounters the problem of omniscience: every agent knows, at least, all tautologies. When the set of all tautologies is infinite, their explicit knowledge forces agents to store infinitely many pieces of information. To avoid the problem, the attempts have been made to limit the logic so that agents are simply *not able* to derive all tautologies. This, however, is hardly a solution to the problem, because one certainly wants the agents to *be able* to derive all possible tautologies (or consequences of their beliefs). One only does not want them to *actually* store all of them.

Our solution to this problem follows [1]. We model an agent's *explicit* knowledge as a finite set of "pieces of information" which the agent has available (say, in his memory). The associated logic allows one to reason about the bounds of agent's knowledge, that is, not only about what

agent knows but also about what he does not know. If $\triangle_A x$ expresses the fact that A knows (has stored) the (collection) x then $\neg \triangle_A y$ says that A has not stored y. However, since the universe of possible "pieces of information" is typically infinite (the set of formulae of some agent language), the fact that an agent knows *at most* some x is not expressible by a finite formula in usual epistemic logic. One introduces therefore an operator for expressing exactly this fact: $\bigtriangledown_A y$ means that A knows at most y. From such a statement, it follows that $\neg \triangle_A z$ for any $z \not\subseteq y$.

We will use the logic of boundedness as the underlying logic for sequence logic. The logic of boundedness, addressing only finitude of agents' storage, can be combined with arbitrary agent logic. In Section 5 we give a general pattern for obtaining such combinations and illustrate it by examples.

1.3. The structure of the paper

Section 2 introduces the language and semantics of sequence logic, giving examples of combinations with various underlying logics. Section 3 introduces a reasoning system and provides the answer to the main question about such systems and their descriptions: given a specification of the sequence of states an agent (or a group of agents) is required to pass through, like (1), what other decriptions will be passed through in all cases? In other words, given a language of finite sequences of state-formulae, we ask for an axiomatization which is strongly complete with respect to intervals of total, dense orderings.

The sequence logic is parameterized by an arbitrary underlying logic of state-formulae. For the sake of illustration we often use propositional logic, but in Section 4 we introduce a special logic of finite agents which allows explicit treatment not only of the lower bounds (as in usual modal logics) but also of the upper bounds on agents' knowledge. The language and semantics follow [1], but we give a new sound, complete and decidable reasoning system.

We show then in Section 5 how bounded agents can be equipped with internal reasoning mechanisms, and illustrate the interaction with sequence logic on a couple of simple examples.

Since our three-tiered approach addresses three quite different aspects, often treated independently from each other, detailed comparisons to alternative approaches might quickly require a whole new paper. We comment only briefly relations of each level to some works addressing the respective aspect, focusing only on the differences in the proposed solutions rather than on detailed analyses of alternative approaches. Subsection 2.2 comments relations of sequence logic to

action-based descriptions, linear-time temporal logic, interval logics and transition systems. Comparison of our bounded agents logic to other approaches to boundedness is summarized above in 1.2 and is not elaborated. Such a comparison can be found in [1, 2, 3], where model we are using was introduced and studied. In Subsection 5.4, we give a more detailed comparison of our handling of bounded reasoning agents to that offered by active logic. Section 6 summarizes the main points and suggests further development. Proofs omitted here can be obtained by contacting the first author or can be found in [24].

2. Sequence Logic: Language and Semantics

The logic is parameterized by an arbitrary underlying logic (u.l.) which one might want to use for describing the single states. Hence, the language is parameterized by the language of the underlying logic, U.

DEFINITION 3 (Sequence Language $\mathcal{SL}$). *The language $\mathcal{SL}$ – containing sequence formulae over a parameter language U – is given by the following grammar:*

$$\sigma := U \mid \top \mid \sigma;\sigma$$

$\top$ denotes tautology of the u.l. and is added if it is missing there.

In the sequel $\sigma, \sigma_1, \ldots$ denote (sequence) formulae of $\mathcal{SL}$; $f, f_1 ... g, g_1 ...$ – atomic formulae, i.e., those without ; occurring inside. The formulae of sequence logic logic will be simple sequents, i.e., have the form $\sigma_1 \Vdash \sigma_2$. We will denote sequents using $q, q', \ldots$. Complexity of a sequence formulae/sequent refers to the number of $\_;\_$ occurring in it.

The semantics is parameterized by the semantics of the u.l. Sequence formulae are evaluated over a total, dense ordering which is left-closed and right-open. Given such an ordering $\mathcal{O} = (O, <)$, its points are (mapped to) models of the u.l. An SL-structure is a function

$$r\colon O \to Mod(u.l.)$$

Left-closedness models the "beginning" (of a computation, or its part), and right-openness its possibly unbounded character. In general, we will consider also subintervals of a whole order $\mathcal{O}$. We denote by $[a, b)$ a *left-closed right-open interval*, $[a, b) = \{o \in O : a \leq o < b\}$, of a given order $\mathcal{O}$. We do not consider empty intervals at all, so the notation $[a, b)$ always implicitly means $a < b$. The satisfaction relation is defined, in general, for any such interval with satisfaction by the whole r, i.e., in the whole O, being a special case.

DEFINITION 4. *Satisfaction of an $\mathcal{SL}$-formula σ in an $\mathcal{SL}$-structure, written $[a,b) \models_{\mathcal{O},r} \sigma$, is defined as follows:*

1. $[a,b) \models_{\mathcal{O},r} \top$ *for all* $[a,b)$
2. $[a,b) \models_{\mathcal{O},r} f \iff \forall o \in O : a \leq o < b \Rightarrow r(o) \models_{u.l.} f \quad (f \in u.l.)$
3. $[a,b) \models_{\mathcal{O},r} \sigma_1;\sigma_2 \iff \exists o \in O : a < o < b \;\&\; [a,o) \models_{\mathcal{O},r} \sigma_1 \;\&\; [o,b) \models_{\mathcal{O},r} \sigma_2$

We skip the subscripts in the notation, assuming always given $\mathcal{O}$ and r. We will concentrate on the case where the interval is actually the whole O, writing $r \models \sigma$. This is justified by the following equivalence between (5) and (6). For a semantical entailment, we write:

$$\sigma_1 \models\!\models \sigma_2 \quad \text{iff} \quad \forall\mathcal{O}\forall r\forall[a,b) : [a,b) \models_{\mathcal{O},r} \sigma_1 \Rightarrow [a,b) \models_{\mathcal{O},r} \sigma_2 \qquad (5)$$

Equivalently, we can consider only whole orderings (and not all subintervals):

$$\sigma_1 \models\!\models \sigma_2 \quad \text{iff} \quad \forall\mathcal{O}\forall r : O \models_{\mathcal{O},r} \sigma_1 \Rightarrow O \models_{\mathcal{O},r} \sigma_2 \qquad (6)$$

It is an easy exercise to verify that $\_;\_$ is associative.

2.1. Cuts

An equivalent definition of satisfaction that an interval $[a,b)$ satisfies a sequence formula $f_1;...;f_n$ iff it is possible to cut the interval into n subintervals, each left-closed right-open and each satisfying the corresponding f_i. This is the content of the following definition.

DEFINITION 7. *An n-cut C of $[a,b)$ is a partition of $[a,b)$ into $n+1$ intervals*

$$[a,b)|_C = [a,o_1)[o_1,o_2)\ldots[o_i,o_{i+1})\ldots[o_n,b)$$

with $a < o_i < o_{i+1} < b$.

By $r|_\sigma$ we denote a cut of r which verifies σ, i.e., shows that $r \models \sigma$.[1]

Two cuts can be superimposed on each other yielding a (possibly) more refined cut.

EXAMPLE 8. *Consider two cuts of a given r, each verifying $\sigma_1 = f_1;f_2;f_3$, respectively, $\sigma_2 = g_1;g_2;g_3$. A possible situation is the following:*

$$\begin{array}{l} r|_{\sigma_1} \\ r|_{\sigma_2} \end{array} \qquad \left[\begin{array}{c|c|c|c} f_1 & f_2 & f_2 & f_3 \\ \hline g_1 & g_2 & g_3 & g_3 \end{array} \right)$$

[1] This is ambiguous, since there may be many different cuts of r all verifying σ.

The result of superimposing these two cuts is as shown below:

$$r|_{\sigma_1 \& \sigma_2}: \qquad [\ \frac{f_1}{g_1}\ |\ \frac{f_2}{g_2}\ |\ \frac{f_2}{g_3}\ |\ \frac{f_3}{g_3}\ )$$

The following definition formalizes the superposition of two cuts. It will be used only in situations where each cut verifies a respective sequence formula and, moreover, when we are considering the sequent $\sigma_1 \Vdash \sigma_2$. Hence, although the constructions are symmetric, we note the asymmetry $\sigma_1 \Vdash \sigma_2$ in the notation. The operation $Paths(\_)$ collects all possible ways of superimposing a cut verifying $\sigma_1 = f_1; ...; f_n$ and one verifying $\sigma_2 = g_1; ...; g_m$. In other words, whenever an interval satisfies both formulae, the superposition of the two cuts will satisfy some path in $Paths(\sigma_1 \Vdash \sigma_2)$, which is defined below. (We also have the opposite implication, see Lemma 11.)

DEFINITION 9. *For an arbitrary sequent q, we define $Paths(q)$ by induction on the complexity l of (number of $\_;\_$ in) q :*

$l = 0$*: i.e.,* $q = f \Vdash g$ – $Paths(f \Vdash g) := \{[f \vdash g]\}$.

$l > 0$: *a.* $q = f_1; f_2; \ldots; f_n \Vdash g$
$Paths(q) := \{[f_1 \vdash g] - [f_2 \vdash g] - \ldots - [f_n \vdash g]\}$

b. $q = f \Vdash g_1; g_2; \ldots; g_m$
$Paths(q) := \{[f \vdash g_1] - [f \vdash g_2] - \ldots - [f \vdash g_m]\}$

c. $q = f_1; f_2; \ldots; f_n \Vdash g_1; g_2; \ldots; g_m$
i. $Paths(q) := \{[f_1 \vdash g_1]\} - Paths(f_2; \ldots; f_n \Vdash g_2; \ldots; g_m) \cup$
ii. $\{[f_1 \vdash g_1]\} - Paths(f_1; f_2; \ldots; f_n \Vdash g_2; \ldots; g_m) \cup$
iii. $\{[f_1 \vdash g_1]\} - Paths(f_2; \ldots; f_n \Vdash g_1; g_2; \ldots; g_m)$

Point c. exhausts the possible ways of overlapping of subsequent intervals. Starting with f_1 and g_1, we have the three possibilities illustrated in Figure 1, each corresponding to one of the cases listed under c.

f_1	$f_2; ...; f_n$		f_1	$f_2; ...; f_n$		f_1	$f_2; ...; f_n$
g_1	$g_2; ...; g_m$		g_1	$g_2; ...; g_m$		g_1	$g_2; ...; g_m$
(i)			(ii)			(iii)	

Figure 1. Possible overlappings of initial intervals.

EXAMPLE 10. *$Paths(q)$, for q from Example 8, are the following:*

$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(1)*
$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(2)*
$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(3)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(4)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_1 \vdash g_3] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(5)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(6)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(7)*
$[f_1 \vdash g_1] - [f_1 \vdash g_2] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(8)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(9)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(10)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$, *(11)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$, *(12)*
$[f_1 \vdash g_1] - [f_2 \vdash g_1] - [f_3 \vdash g_1] - [f_3 \vdash g_2] - [f_3 \vdash g_3]$ *(13)*

The path in line (2) corresponds to the cut we obtained in Example 8.

A structure r satisfies a path if it satisfies the corresponding sequence formula. For instance, an r satisfies path (2) from the above example iff $r \models f_1 \wedge g_1; f_2 \wedge g_2; f_2 \wedge g_3; f_3 \wedge g_3$.

The following lemma states the observation from Example 8.

LEMMA 11. *For all r, σ_1, σ_2 we have that $r \models \sigma_1$ and $r \models \sigma_2$ if and only if there exists a $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ such that $r \models \pi$.*

2.2. Related systems

We comment briefly relations of sequence logic to action-based descriptions, 2.2.1, then linear-time temporal logic and interval logic, 2.2.2, and to transitions systems, 2.2.3.

2.2.1. *Action-based descriptions*

Actions are not part of sequence logic. This, however, is not necessarily a vice. The logic allows modelling different kinds of actions, which can be performed in different scenarios, in terms of their effects on the agents' states. We illustrate this flexibility on the example of communication between epistemic agents.

We want to model the action of sending a message m from (agent) A to B, $send(A, m, B)$. As the underlying logic, we use some variant of epistemic logic, with $\mathbf{K}_E(x)$ meaning that E knows (has available) x. Communication of m from A to B is modeled by a rewrite rule, which defines it in terms of the effects on the consequtive states:

$$send(A, m, B) \quad \rightsquigarrow \quad s_1 \ ; \ s_2$$

The most elementary model would simply let $s_1 = \mathbf{K}_A(m)$ and $s_2 = \mathbf{K}_A(m) \wedge \mathbf{K}_B(m)$. But one is typically interested in the security issues and models the environment as another agent E. Security of the communication channel means that E cannot eavesdrop on the communication between A and B. This corresponds to the situation where, starting in a state where A, but not E, knows m, we pass to a state where both A and B, but still not E, know m. The sending event is then rewritten to the following s_1, s_2:

$$s_1 = \mathbf{K}_A(m) \wedge \neg\mathbf{K}_E(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \neg\mathbf{K}_E(m) \wedge \mathbf{K}_B(m).$$

A reliable communication, i.e., one which is not only secure, but where A can also be sure that B obtains his message, is modeled by adding an additional conjunct to the resulting state of A:

$$s_1 = \mathbf{K}_A(m) \wedge \neg\mathbf{K}_E(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \neg\mathbf{K}_E(m) \wedge \mathbf{K}_B(m) \wedge \mathbf{K}_A(\mathbf{K}_B(m)).$$

An insecure channel makes it possible for E to eavesdrop on all messages. Such a worst case scenario gives the following state transition:

$$s_1 = \mathbf{K}_A(m)$$

$$s_2 = \mathbf{K}_A(m) \wedge \mathbf{K}_E(m) \wedge \mathbf{K}_B(m).$$

Further variations are possible, e.g., extending s_2 with $\mathbf{K}_A(\neg\mathbf{K}_E(m))$, stating that A knows that E has not intercepted the message, etc. Once a series of actions is rewritten as a series of their effects, σ_1, we can apply the logic $\mathcal{SEQ}$, Section 3, for deriving its consequences, by asking for σ_2 such that $\sigma_1 \Vdash \sigma_2$. We thus see how the flexibility of the proposed setting for handling a virtually unlimited variety of possible action types is achieved by *not* axiomatizing any actions, but merely by representing them in terms of their effects on the states.

2.2.2. *LTL and ITL*

The operator $\_;\_$ can be viewed as the until, $\mathbf{U}$, of temporal logic over linear time, LTL. Then, our language could be viewed as a very special subset of temporal logic, where a sequence formula

$$f_1; f_2; f_3; ...; f_n \quad \text{corresponds to} \quad f_1\mathbf{U}\Big(f_2\mathbf{U}(f_3\mathbf{U}...(f_{n-1}\mathbf{U}\Box f_n)...)\Big),$$

where the final f_n (and only it) appears always $\Box$, to remain true from then on.[2] Thus it is not surprising that we can express several temporal modalities, for instance:

[2] For convenience we have used an until-operator with semantics defined by $i \models \phi U\psi$ if $\exists k > i$ $(k \models \psi \wedge \forall j$ $(i \leq j < k \Rightarrow j \models \phi))$. This until-operator is definable

1. $r \models f$ iff f holds always in an $\mathcal{SL}$ structure r
2. $r \models \top; f$ iff f becomes eventually true and holds then forever
3. $r \models f; \top$ iff f holds initially for at least some time
4. $r \models f; \neg f$ iff f holds for some time, after which $\neg f$ holds forever

Example 2 above admits all the same r's as does 1 but, in addition, also all where f holds *almost* always, i.e., everywhere with the possible exception of some initial interval. Thus, $f \models \top; f$ but $\top; f \not\models f$. Dually, 3 also allows all models of 1 but also ones where f, holding initially, becomes false after some time, so $f \models f; \top$ but $f; \top \not\models f$. In 4, the requirement is for f to actually become false after some time, never to become true again.

We are not aware of an independent study of this particular fragment of LTL while, as we will try to show in the rest of this paper, it seems to merit closer attention. Restricted expressivity, as compared to full LTL, can possibly lead to improvement in complexity of various algorithms, but this issue is left for future investigation.

The semantics is based on points but, nevertheless, it is strongly interval-oriented. Alternatively to the above representation in LTL, one can almost identify _;_ with the chop operator, common in interval logics, ITL, e.g., [18, 16]. We have, however, only a very limited fragment of such logics and there are also some significant differences. For the first, sequence logic does not include "point-intervals" (as single points are sometimes called in the interval semantics.) More significantly, although the satisfaction relation is defined relatively to satisfaction at single points, $\models_{u.l.}$, a formula satisfied only at a single point is not satisfied by any interval. For instance, consider $[0, 2)$, where for all $x \in [0, 1) \cup (1, 2) : r(x) \models a$ while $r(1) \not\models a$. Then $[0, 2) \not\models a$ and $[0, 2) \not\models \neg a$. There are some subintervals satisfying a, e.g., $[0, 1) \models a$, but there is no subinterval of $[0, 2)$ satisfying $\neg a$. This is related also to the phenomenon illustrated by the following example.

EXAMPLE 12. *Assume that u.l. contains propositional disjunction. We may have that* $[a, b) \models f \vee g$, *while for every subinterval* $[c, d) \sqsubseteq_I [a, b) : [c, d) \not\models f$ *and* $[c, d) \not\models g$. [3] *Take, for instance,* $a = 0, b = 1$ *and* $[a, b)$ *to be all the rationals in* $[0, 1)$ *with the standard ordering. We distribute densely two models* $m_f \models f \wedge \neg g$ *and* $m_g \models \neg f \wedge g$, *for instance, as follows. Each* $o \in [0, 1)$ *equals* $\frac{n}{k}$ *for some relatively prime*

by $\phi \wedge (\phi U' \psi)$ with U' the until-operator from [7], and by $\phi \wedge X(\phi U'' \psi)$ with U'' the until-operator and X the next-operator from [23].

[3] $[c, d) \sqsubseteq_I [a, b)$ denotes that $[c, d)$ is a *subinterval* of $[a, b)$, i.e., an interval with $a \leq c < d \leq b$.

n, k with $n < k$. We let $r(\frac{n}{k}) = m_f$ if n is odd and $r(\frac{n}{k}) = m_g$ if n is even. Then, for any two distinct points with $r(o_1) = m_g = r(o_2)$, there is a point between them, $o_1 < o_3 < o_2$, with $r(o_3) = m_f$, and vice versa. Thus $[0,1) \models f \vee g$ but nowhere, i.e., in no subinterval $[c,d) \sqsubseteq_I [0,1)$, we have that $[c,d) \models f$ or $[c,d) \models g$.

Consequently, sequence logic does not satisfy some axioms typical for ITL, e.g.: [4]

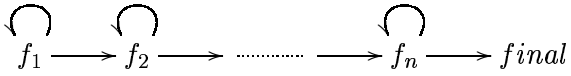
$$\begin{array}{rcl} f_0;(f_1 \vee f_2) & \models & (f_0;f_1) \vee (f_0;f_2) \\ (f_1 \vee f_2);f_3 & \models & (f_1;f_3) \vee (f_2;f_3). \end{array}$$

It is known that, unlike sequence logic, most ITL logics are (highly) undecidable, [22]. The undecidability results are established by various means depending on the specific variant considered. But for the restricted language of sequence logic, they can be traced back to the validity of these two axioms, as shown in [20].

In comparison to both LTL and ITL, sequence logic has two main advantages. For the first, it is fully parameterized by the underlying logic. The possibility of freely combining it with a variety of logics can provide, on the one hand, some generic results showing dependence on the purely temporal aspect and, hence, valid for arbitrary underlying logics. On the other hand, one can envision a wide scope of applications which are often dependent on the specific logic used for reasoning about particular states/agents. The second advantage is that variations of sequence logic for other types of orderings can be combined with underlying logics in the same way as the sequence logic of dense time presented here. These variations are not addressed here and the reader is referred to [6] for their overview. They amount only to modification or addition of some rules in the system $\mathcal{SEQ}$ presented in the following Section. Consequently, the whole approach developed in the rest of this paper can be applied without any changes for studying the respective phenomena on the class of all linear orderings, ω-orderings, etc. This generality and genericity seems an advantage in comparison with LTL and ITL which have been studied in depth only for some classes of orderings (e.g., ITL mostly for ω-orderings) and which require genuinely new investigations when one attempts to transfer results from one class of orderings to another.

[4] Strictly speaking, neither of the formulae on the right of $\models$ belongs to $\mathcal{SL}$, where only entailments between simple sequences are expressed. So we should rather say that sequence logic *would not* satisfy these formulae, *if* they had belonged to it with the expected interpretation of disjunction.

2.2.3. *Transition systems*
As a final example, we observe a close relationship of sequence logic to transition systems. As mentioned in the introduction, we can define for every sequence formula $\sigma = f_1; f_2; \ldots; f_n$ a transition system (or: rewrite system):



The *meaning* of σ can now be given as the set of all rewrite sequences of this transition system. Here some care has to be taken. First, we only consider rewrite sequences that end in the final state. Second, we consider the states modulo equivalence in the underlying logic. Third, we adopt some notation of formal language theory and denote the rewrite sequences as words $f_1^{p_1} f_2^{p_2} \ldots f_n^{p_n}$ and call them *traces* (note that we denote only the states, and not the transitions). We use Kleene $^+$ to denote one or more iterations. With these points in mind we define the semantics of σ as follows:

$$[\![\sigma]\!] = f_1^+ \ldots f_n^+$$

For example, $[\![a \wedge b; b \wedge a; c]\!] = (a \wedge b)^+ (b \wedge a)^+ c^+ = \{(a \wedge b)^p c^q \mid p > 1, q > 0\}$. Equivalence of systems can now be defined as follows.

DEFINITION 13. $\sigma_1 \cong \sigma_2$ *if* $[\![\sigma_1]\!] \cap [\![\sigma_2]\!]$ *is non-empty.*

Indeed, $\cong$ is an equivalence relation: reflexivity and symmetry are trivial, and transitivity follows after a moment's reflection on the regular expressions involved. The following lemma characterizes the semantic entailment in terms of transition systems. ($\sigma(i)$ denotes the i-th "state" of σ.)

LEMMA 14. $\sigma_1 \models\!\models \sigma_2$ *if and only if*

$$\exists\ \sigma_1' \cong \sigma_1,\ \sigma_2' \cong \sigma_2 : |\sigma_1'| = |\sigma_2'|\ \&\ \forall\ i = 1, \ldots, |\sigma_1'| : \sigma_1'(i) \models_{u.l.} \sigma_2'(i)$$

The correspondence between the traces σ_1' and σ_2' reflects the existence of a joint path $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ from Lemma 11, which verifies $\sigma_1 \models\!\models \sigma_2$.

3. The reasoning system

Given a sequence formula σ (possibly only a single formula of the underlying logic) we let $\sigma*$ denote σ or an arbitrary extension of σ to a (longer) sequence formula (analogously for $*\sigma$). $\perp$ denotes contradiction of the underlying logic (if it is present there).

DEFINITION 15. *The calculus $\mathcal{SEQ}$ contains the following rule schemata:*

$$\textit{(ex falso)}\ \frac{}{f_1;\ldots;f_n \Vdash \sigma}\ [\exists i : f_i \vdash \bot] \qquad (L)\ \frac{f* \Vdash \sigma}{f* \Vdash g;\sigma}\ [f \vdash g]$$

$$\textit{(lift)}\ \frac{}{f \Vdash g}\ [f \vdash g] \qquad (E)\ \frac{\sigma_1 \Vdash \sigma_2}{f;\sigma_1 \Vdash g;\sigma_2}\ [f \vdash g]$$

$$(R)\ \frac{\sigma \Vdash g*}{f;\sigma \Vdash g*}\ [f \vdash g]$$

The intuition behind these rules is straightforward and concerns the possible overlapping of subsequent intervals. It refers again to Figure 1, now with (E) corresponding to (i), (L) to (ii) and (R) to (iii). In either case, validity of the conclusion requires validity of $f \vdash g$ (which is placed in the side-condition). The relation between the remaining parts depends on whether the left formula, f, "lasts longer" than g (L), "shorter than" g (R), or if both have equal duration (E). Axioms are absent because side-conditions will give proof obligations determining if a given derivation is a proof. The rule (lift) terminates construction of a derivation (bottom-up). A derivation is defined in the standard way. The following gives a couple of examples.

EXAMPLE 16. *The following are two possible derivations of the sequent q from Example 8:*

$$\Delta_q:\quad \textit{(lift)}\ \frac{\dfrac{\dfrac{}{f_3 \Vdash g_3}\ [f_3 \vdash g_3]}{f_2;f_3 \Vdash g_2;g_3}\ (E)\ [f_2 \vdash g_2]}{f_1;f_2;f_3 \Vdash g_1;g_2;g_3}\ (E)\ [f_1 \vdash g_1]$$

$$\Delta'_q:\quad \textit{(lift)}\ \frac{\dfrac{\dfrac{\dfrac{}{f_3 \Vdash g_3}\ [f_3 \vdash g_3]}{f_2;f_3 \Vdash g_3}\ (R)\ [f_2 \vdash g_3]}{f_2;f_3 \Vdash g_2;g_3}\ (L)\ [f_2 \vdash g_2]}{f_1;f_2;f_3 \Vdash g_1;g_2;g_3}\ (E)\ [f_1 \vdash g_1]$$

We denote by $Der(q)$ the set of all possible derivations of q. Side-conditions in a derivation constitute the proof obligations. Given a derivation Δ_q, we denote by $po(\Delta_q)$ the sequence of all side-conditions (in the bottom-up order). For the derivations from Example 16, we have

$$po(\Delta_q) = [f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$$
$$po(\Delta'_q) = [f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_2 \vdash g_3] - [f_3 \vdash g_3]$$

This operation is extended pointwise to the set of all derivations of a sequent, i.e., $po(Der(q)) = \{po(\Delta) \mid \Delta \in Der(q)\}$.

The following lemma states the equivalence of the proof obligations obtained over all derivations of a sequent q and the possible overlappings of cuts verifying q.

LEMMA 17. *For an arbitrary sequent* $q : po(Der(q)) = Paths(q)$.

DEFINITION 18. *A derivation* Δ_q *of* $q = f_1; ...; f_n \Vdash g_1; ...; g_m$ *is a* proof *of* q *iff either*

$$\exists 1 \leq i \leq n : \ f_i \vdash \bot \quad \text{or}$$
$$\forall [f_i \vdash g_j] \in po(\Delta_q) : \ f_i \vdash g_j$$

For convenience, it is assumed that the underlying logic contains contradiction, $\bot$ (and, typically, tautology $\top$ such that $\vdash \top$ and for all $f \vdash \top$.)

EXAMPLE 19. *Assuming the underlying logic to be propositional, let the sequent* q *be as in our previous examples, specialized to actual formulae as follows:* $\top; b; \neg b \Vdash \top; \top; \neg b$. *Then the derivation* Δ_q *(from Example 16) is a proof of* q, *but the derivation* Δ'_q *is not. The latter fails to be a proof, because the side-condition* $f_2 \vdash g_3$ *is now* $b \vdash \neg b$, *which does not satisfy the second condition of Definition 18.*

Every rule, applied bottom-up, decreases the complexity of the sequent. Hence every derivation Δ_q terminates in finitely many steps. Checking if the obtained $po(\Delta_q)$ is a proof is trivially decidable, provided that provability in the underlying logic is decidable. Hence we have a simple, but useful, fact.

PROPOSITION 20. *If the relation* $\vdash$ *is decidable, then so is* $\Vdash$.

The next lemma reflects the desired property that we mentioned in the Introduction, i.e., that we do not want to differentiate between f and $f; f$.

LEMMA 21. *The following rules are admissible.*

$$(\mathrm{id}\Vdash)\ \frac{*f\star \Vdash \sigma}{*f; f\star \Vdash \sigma} \qquad (\Vdash\mathrm{id})\ \frac{\sigma \Vdash *g\star}{\sigma \Vdash *g; g\star} \qquad (\vdash)\ \frac{*g\star \Vdash \sigma}{*f\star \Vdash \sigma}\ [f \vdash g]$$

The first two rules, allowing us to ignore all adjacent duplicates in the considered sequence formulae, simplify the proof of Lemma 17.

Assuming soundness of the u.l., one verifies relatively easily soundness of $\mathcal{SEQ}$. Lemma 17 is of crucial importance in the proof of completeness, and we now sketch its main steps.

3.1. Completeness

We assume completeness of the underlying logic. We also restrict our attention to the dense ordering of non-negative rationals, $\mathcal{Q}$, but constructions and the final result generalize to arbitrary dense orderings.

Given an interval $[a,b) \sqsubseteq_I \mathcal{Q}$, and $k \geq 2$ models $c_0, ..., c_{k-1}$, one can distribute them densely, i.e., so that for any two points $a \leq o_1 < o_2 < b$, the subinterval $[o_1, o_2)$ contains all models. We register this fact without proof.

LEMMA 22. *$k \geq 2$ models $c_0, ..., c_{k-1}$ can be distributed densely over an interval $D = [a,b) \sqsubseteq_I \mathcal{Q}$ so that every non-empty subinterval $X \sqsubseteq_I D$ contains all models $c_0, ..., c_{k-1}$.*

THEOREM 23. *When the u.l. is strongly complete, then so is $\mathcal{SEQ}$.*

Proof: We show that every unprovable sequent has a counter-model. So assume a sequent q with no proof: $f_1; \ldots; f_n = \sigma_1 \Vdash \sigma_2 = g_1; \ldots; g_m$ (with no adjacent duplicates). By Definition 18 combined with Lemma 17, this means that $\forall \pi \in Paths(q)$:

- $\forall [f_i \vdash g_j] \in \pi : f_i \nvdash \bot$ – all f_i are consistent, AND
- $\exists [f_i \vdash g_j] \in \pi : f_i \nvdash g_j$.

We construct a model of $f_1; ...; f_n$ by constructing an interval $r_i \models f_i$ for every $1 \leq i \leq n$. We use rationals, so for every $i < n$, we let $r_i = [i-1, i)$, while $r_n = [n-1, \infty)$. For every r_i we assign the models as follows.

For every pair f_i, g_j, the proof obligation $f_i \vdash g_j$ occurs on some derivation path. For each f_i, collect all g_j's (from all derivation paths) such that $f_i \nvdash g_j$. (If for some f_i, there are no such g_j's, then let r_i contain any model of f_i (existing by $f_i \nvdash \bot$.) Construct r_i as follows:

Since $f_i \nvdash g_j$ (for each chosen g_j) so, by completeness of the u.l., $f_i \not\models g_j$, so we have a counter-model, m_{ij}, for each such pair. For a given f_i and all g_j with $f_i \nvdash g_j$, we collect all such m_{ij} and distribute them densely in the interval r_i. By Lemma 22, we then have that, for all j for which we have a counter-model $m_{ij} : r_i \not\models g_j$ (and $\forall s \sqsubseteq_I r_i : s \not\models g_j$).

Concatenating all the intervals $r = r_1; r_2; ...; r_n$, we obtain $r \models f_1; ...; f_n$, with the cut $r_i \models f_i$, which we now fix.

If now (*) $r \models g_1; ...; g_m$ then, by Lemma 11, there exists a path $\pi \in Paths(\sigma_1 \Vdash \sigma_2)$ such that for every node $[f_i \vdash g_j] \in \pi$, the respective subinterval $r_{ij} \models f_i \wedge g_j$. However, by Lemma 17, $Paths(q) = po(Der(q))$, i.e., π comprises the proof obligations from one of the derivation paths for q. Since no such path is a proof, it contains a node

$[f_i \vdash g_j]$ where $f_i \not\vdash g_j$. But then also $\forall s \sqsubseteq_I r_i : s \not\models g_j$ – contradicting (*). So r is a counter-model for $\sigma_1 \models \sigma_2$. □

EXAMPLE 24. *Consider the following (unprovable) sequent* $q = \sigma_1 \Vdash \sigma_2$*:*

$$a; a \vee b; b \Vdash a \vee b; a; b$$

Paths(q) *are obtained as in Example 10, and are listed with* f_i, g_j *instantiated appropriately:*

$[a \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash b]$, *(1)*
$[a \vdash a \vee b] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b]$, *(2)*
$[a \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b]$, *(3)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash b] - [b \vdash b]$, *(4)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vdash b] - [a \vee b \vdash b] - [b \vdash b]$, *(5)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [b \vdash b]$, *(6)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b]$, *(7)*
$[a \vdash a \vee b] - [a \vdash a] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b]$, *(8)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [b \vdash a] - [b \vdash b]$, *(9)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash b]$, *(10)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [a \vee b \vdash b] - [b \vdash b]$, *(11)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [a \vee b \vdash a] - [b \vdash a] - [b \vdash b]$, *(12)*
$[a \vdash a \vee b] - [a \vee b \vdash a \vee b] - [b \vdash a \vee b] - [b \vdash a] - [b \vdash b]$ *(13)*

On every path there exists a node with an unprovable obligation, either $[a \vee b \vdash a]$ *or* $[a \vee b \vdash b]$ *(as well as* $[b \vdash a]$ *or* $[a \vdash b]$*). Hence* $\sigma_1 \not\Vdash \sigma_2$*. The counter-model will be built from three intervals,* $r = [0,1)[1,2)[2,\infty)$*, where* $\forall o \in [0,1) : r(o) \models a \wedge \neg b$ *(a boolean structure assigning* `true` *to* a *and* `false` *to* b*),* $\forall o \in [2,\infty) : r(o) \models b \wedge \neg a$*, while in* $[1,2)$ *we distribute densely the counter-models for* $a \vee b \vdash b$ *(namely* $a \wedge \neg b$*) and for* $a \vee b \vdash a$ *(namely* $\neg a \wedge b$*). This will ensure that* $[1,2) \models a \vee b$*, and so* $r \models a; a \vee b; b$*, as we have the following situation*

$r =$	r_1	r_2	r_3
	$a \wedge \neg b$	$a \vee b$	$\neg a \wedge b$

A cut verifying $a \vee b; a; b$ *must first comprise some subinterval verifying* $a \vee b$ *and then some verifying* a*. But the latter can occur only within* r_1*, as* $r_3 \models \neg a$*, while every subinterval* $s \sqsubseteq_I r_2 : s \not\models a$*. But then, the rest of* r *will not satisfy* b*, since all subintervals* $s \sqsubseteq_I r_2 : s \not\models b$*. In short,* $r \not\models a \vee b; a; b$*.*

4. Bounded agents

As we have emphasized, sequence logic allows combination with arbitrary underlying logic and we have seen some examples of that. As such, it is a generic tool which merits independent study. The present paper, however, is concerned also with bounded agents and in its remaining part we will be occupied with combination of sequence logic with a particular logic of bounded agents. As described in introduction, we intend to achieve a model of bounded agents as independent from the actual agent logic, as sequence logic is independent from the underlying logic. In other words, we will introduce a logic of bounded agents which can be combined with arbitrary agent logics. Section 5 will provide means for doing this, while in this section we study such a generic logic of bounded agents.

As an introduction, however, we will use first, in 4.1, as the underlying logic the epistemic logic S4 to illustrate a verification of a simple protocol goal. We point out that some goals, concerning negative information about things agents are not able to know, require explicit treatment of the bounds of agent's knowledge. Besides signalling this point, this subsection illustrates only one possible combination of sequence logic with an arbitrary underlying logic, exemplified by S4.

In 4.2, we summarize the semantic model of finite, syntactic agents from [1] and give a new, sequent-based reasoning system for it. The presence of upper bounds in the logic enables us to utilize possibly infinite negative information (everything an agent does not know) in the reasoning about an agent's lacking knowledge of particular pieces of information.

4.1. Communicating epistemic agents

As an example, we will use a simple step in a communication protocol involving three agents: A communicating with B and the environment, or enemy, E, who can observe/listen to all communication. Initially, agent A possesses the message m_k which he will send to B and B possesses the (secret) key k with which the message m is coded. The formula $\mathbf{K}_X(m_k \wedge k \to m)$ says that agent X, knowing/possessing m_k and k, can also decode the message and obtain m. The initial state is described by the formula

$$\iota = \mathbf{K}_A(m_k) \wedge \mathbf{K}_B(k) \wedge \mathbf{K}_B(m_k \wedge k \to m) \wedge \mathbf{K}_E(m_k \wedge k \to m).$$

In the next step, this intial state is expanded by the acquisition of the message m_k, sent by A, by both B and E, $\kappa = \mathbf{K}_B(m_k) \wedge \mathbf{K}_E(m_k)$:

$$\iota; \iota \wedge \kappa \;\Vdash\; \top; \mathbf{K}_B(m) \tag{25}$$

We obviously have $\iota \vdash_{S4} \top$ and also $\iota \wedge \kappa \vdash_{S4} \mathbf{K}_B(m)$, so the above statement follows in $\mathcal{SEQ}$ by two applications of (E).

REMARK 26. *We should take some precautions in formulating the proof obligations (side-conditions in $\mathcal{SEQ}$) when using standard epistemic/modal logics like* S4 *or* S5. *For instance, we might want to prove that, given that $a \rightarrow b$, an agent A knowing first a, will eventually (be able to) know b, i.e., $\top; \mathbf{K}_A(b)$. The last step in a derivation of such a description would amount to the following:*

$$\frac{}{(a \rightarrow b) \wedge \mathbf{K}_A(a) \Vdash \mathbf{K}_A(b)} \quad [(a \rightarrow b) \wedge \mathbf{K}_A(a) \vdash \mathbf{K}_A(b)] \tag{27}$$

In many modal logics, like S4 *or* S5, *strong completeness requires that the premises are closed under the $\Box$-modality (here $\mathbf{K}$), and the premise $a \rightarrow b$ does not satisfy this condition. The general statement of this fact is as follows (e.g., exercise 1.5.3 in [7]):*

$$\Gamma \models^g \phi \iff \mathbf{K}^*(\Gamma) \models^l \phi \tag{28}$$

where $\models^g$ is the global logical consequence (which we are using), while $\models^l$ is local logical consequence (the latter with strongly complete reasoning systems).[5] *In the logics where $\mathbf{K}(\phi) \leftrightarrow \mathbf{K}(\mathbf{K}(\phi))$, we can simplify the right hand side of (28) to $\Gamma, \mathbf{K}(\Gamma) \models^l \phi$, or even drop Γ when axiom $\mathbf{T}$ is present. Consequently, when using modal epistemic logic, like* S4 *or* S5, *we would have to utilize strongly complete versions of the reasoning system, based on (28). In the example (25), the problem does not arise since all premises are under the modality $\mathbf{K}$.*

The complete goal of the sequence $\iota; \iota \wedge \kappa$ from (25) would be not only that B obtains m but also that E does *not* obtain it. The fact $\neg\mathbf{K}_E(m)$ is not, however, provable from $\iota \wedge \kappa$. The reason is not so much inadequacy of the description $\iota; \iota \wedge \kappa$, as the inadequacy of the underlying logic itself. The $\mathbf{K}$ modality allows us to state what agents know. We can also negate single propositions stating that an agent does not know a particular m, like in $\neg\mathbf{K}_E(m)$. But such negative propostions cannot be obtained from purely positive ones (as all those included in 1 and 2), since $\mathbf{K}$ captures only the "lower bound" of agent's knowledge. Certainly, E might have learnt m from other sources, and the description $\iota; \iota \wedge \kappa$ does not preclude him from knowing it. What is missing in this description is the statement that it gives a complete picture and that no other sources of possible knowledge are of interest

[5] Statement (28) can be relativised to arbitrary classes of frames which are closed under 'reachable subframes', i.e., classes M where, when $(W, R) \in \mathsf{M}$ and (W_w, R_w) is a subframe of (W, R) obtained by taking $W_w = \{w' \mid R^*(w, w')\}$ for some fixed $w \in W$ and restricting R to this subset, then also $(W_w, R_w) \in \mathsf{M}$. This closure property is enjoyed by the typical modal logics like S4 or S5.

in the analysis. We should not, however, be forced to include in our description the list of everything agents do not know in order to be able to draw such negative conclusions. At this place we have a natural need for an "upper bound" operator, $\bigtriangledown_i X$, expressing that i does not know more than X. We consider such agent descriptions in the rest of this section.

4.2. Finite epistemic agents

Following [1], an agent's epistemic states are represented as finite sets of formulae written in some logical language which is a parameter to the whole formalism. First, the agent language is defined in 4.2.1, and its semantics is given in 4.2.2. Then in 4.2.3 a new sound, complete and decidable logical system for the introduced semantics is presented.

4.2.1. *Language $\mathcal{EL}$*

The *epistemic language*, $\mathcal{EL}$, for describing agents' knowledge states, is defined as follows. It is parameterized by an arbitrary agent language, $\mathcal{AL}$, which for simplicity can be taken now to be simply a set of atomic formulae.

DEFINITION 29. *Formulae of the language $\mathcal{EL}$ are given by the following grammar (over a parameter $\mathcal{AL}$):*

$$\phi ::= \mathcal{AL} \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \bigtriangleup_i T \mid \bigtriangledown_i T \tag{30}$$

where i ranges over agent names, and T over finite sets of $\mathcal{EL}$-formulae.

The intended meaning of the epistemic operators is that agent i knows *at least* formulae $\phi_1, \ldots, \phi_n$ (he may know more): $\bigtriangleup_i\{\phi_1, \ldots, \phi_n\}$, and that he knows *at most* $\phi_1, \ldots, \phi_n$ (he may know less): $\bigtriangledown_i\{\phi_1, \ldots, \phi_n\}$. $\Box_i\{\phi_1, \ldots, \phi_n\}$ abbreviates the conjunction $\bigtriangleup_i\{\phi_1, \ldots, \phi_n\} \wedge \bigtriangledown_i\{\phi_1, \ldots, \phi_n\}$, meaning that agent i knows *exactly* the formulae $\phi_1, \ldots, \phi_n$.

An epistemic formula is one where the outermost operator is either $\bigtriangleup$ or $\bigtriangledown$. Agent identifiers will be sometimes omitted in order to simplify the generic formulations valid for all possible indices, but only when only single agent is involved.

EXAMPLE 31. *The definition allows nesting of epistemic operators in the agent's knowledge. For instance, the intended meaning of $\bigtriangleup_i\{p, \bigtriangledown_j\{q\}\}$ might be that agent i knows at least p and that agent j knows at most q.*

Such nested knowledge is not, however, addressed here, since the "meaning" of the formulae under the outermost epistemic operator ($\bigtriangleup_i$ in the

example) will be relative to the rules with which agents may process them, which we discuss in Section 5. The semantics and logic presented in the current subsection address only the outermost level. One can view it as the meta-level, at which we reason about agents' knowledge states, and which is distinct from the agent-level, containing the formulae under the outermost epistemic operator. These formulae, elements of the set under this outermost (meta)-operator appear simply as distinct "pieces of knowledge".

Thus, although possible consequences of agent i knowing that agent j knows q could be drawn, this depends on the application, i.e. specific rules with which agents can be equipped. Since we do not assume any such rules for agents' reasoning, we do not make any closure restrictions on agents' knowledge. For instance, it is possible that agent knows (has in his state) $\neg\neg p$ without knowing p. Since we do not put any consistency requirements on them, agent can have in his epistemic state e.g. $p, \neg p$. In general, agent's state is an *arbitrary* finite set of formulae.

4.2.2. *Semantics*

We are modelling explicit knowledge, i.e. not the potential (logical or other) closure but only explicitly stored pieces of knowledge. This amounts to a finite set of pieces (of information) which are available to an agent at any given time. For a given set $\mathcal{AL}$, the language is interpreted in the lattice of finite sets of $\mathcal{EL}$-formulae built over $\mathcal{AL}$.

DEFINITION 32. *A* local state s_i *of an agent* i *is a finite set of* $\mathcal{EL}$*-formulae.*

A global state *is an agent-indexed tuple* $s = \langle s_1, \ldots, s_n \rangle$ *of the local states of all agents.*

For a given set of atomic formulae $\mathcal{AL}$ (e.g., propositional variables) and an evaluation function $I: \mathcal{AL} \to \wp(S)$ associating to each atom the set of states which make it true, we define the satisfaction relation between global state s and an agent formula $\phi \in \mathcal{EL}$.

DEFINITION 33. *A (global) state* s *satisfies a formula* ϕ *under evaluation* I, $(s, I) \models \phi$, *iff:*

1. $(s,I) \models p \iff s \in I(p)$
2. $(s,I) \models \neg\phi \iff (s,I) \not\models \phi$
3. $(s,I) \models \phi \to \phi' \iff (s,I) \not\models \phi \text{ or } (s,I) \models \phi'$
4. $(s,I) \models \phi \wedge \phi' \iff (s,I) \models \phi \ \& \ (s,I) \models \phi'$
5. $(s,I) \models \phi \vee \phi' \iff (s,I) \models \phi \text{ or } (s,I) \models \phi'$
6. $(s,I) \models \triangle_i T \iff T \subseteq s_i$
7. $(s,I) \models \triangledown_i T \iff s_i \subseteq T$

$$s \models \phi \iff \forall I : (s,I) \models \phi$$
$$Mod(\phi) = \{s \mid s \models \phi\}$$

Note that satisfaction of epistemic formulae in points 6 and 7 is independent from the evaluation of atoms, i.e., for any I, J and s, we have that $(s,I) \models \triangle_i T \iff (s,J) \models \triangle_i T$. Since the epistemic aspects are of our primary concern, we will not pay much attention to evaluations.

The simplicity of the logic notwithstanding, it displays some interesting phenomena, as illustrated by the following proposition and example from [1].

PROPOSITION 34. *Let* $\mathbf{T}$ *be the axiom schema (for every agent and formula)* $\triangle_i\{\alpha\} \to \alpha$. *Then, for any set of formulae* X: $\mathbf{T} \models \neg \triangle_i \{\triangledown_i X\}$.

That is, assuming that agents are finite and that knowledge implies truth, it is impossible to know the upper bound of one's knowledge, as shown in Example 35 and 38.

EXAMPLE 35. *The fact that an agent* i *knows at least that he knows at most* p *i.e.*

$$\phi_1 = \triangle_i\{\triangledown_i\{p\}\}$$

is unsatisfiable in $Mod(\mathbf{T})$. *Assume, on the contrary,* $(s_i, I) \models \phi_1 \wedge \mathbf{T}$, *for some* (s_i, I). *Then (*)* $\triangledown_i\{p\} \in s_i$ *and, since* $(s_i, I) \models \mathbf{T}$, *also* $(s_i, I) \models \triangledown_i\{p\}$. *The last point forces* $s_i = \emptyset$ *or* $s_i = \{p\}$. *But then, by (*), we must have* $p = \triangledown_i\{p\}$, *which is impossible (at least as long as we work with standard finitary languages and well-founded set theory).*

4.2.3. *Reasoning about finite agents*

In [1] Ågotnes introduced a sound and complete Hilbert-style reasoning system for the semantics described in the preceding subsection. Below, we introduce a new reasoning system which, using sequent representation, provides direct means for concluding decidability and designing possible implementations.

Each axiom and rule concerns only one agent, i.e., all involved epistemic operators have the same subscript, which is marked by the metavariable a. Γ, Δ are finite sets of $\mathcal{EL}$-formulae. In (axE4) Δ does not contain any a-indexed operators.

DEFINITION 36. *The calculus $\mathcal{EC}$ is parameterized by the atomic formulae $\mathcal{AL}$ and agent names, and contains the following axiom and rule schemata:*

AXIOM SCHEMATA:

(axE1) $\Gamma, \triangle_a L, \bigtriangledown_a H \vdash \Delta$ *when* $L \not\subseteq H$

(axE2) $\Gamma \vdash \triangle_a \varnothing, \Delta$

(axE3) $\Gamma, \bigtriangledown_a H \vdash \bigtriangledown_a G, \Delta$ *when* $H \subseteq G$

(axE4) $\Gamma, \bigtriangledown_a H \vdash \triangle_a K_1, ..., \triangle_a K_n, \bigtriangledown_a G_1, ..., \bigtriangledown_a G_m, \Delta$
if $\forall X \in \left(\wp(H) \setminus \bigcup_j \wp(G_j)\right) \exists i : K_i \subseteq X$

RULE SCHEMATA:

$$(\cup\vdash)\ \frac{\Gamma, \triangle_a(L_1 \cup L_2) \vdash \Delta}{\Gamma, \triangle_a L_1, \triangle_a L_2 \vdash \Delta} \qquad (\vdash\cup)\ \frac{\Gamma, \triangle_a L \vdash \triangle_a K, \Delta}{\Gamma, \triangle_a L \vdash \triangle_a(K \cup L), \Delta}$$

$$(\cap\vdash)\ \frac{\Gamma, \bigtriangledown_a(H_1 \cap H_2) \vdash \Delta}{\Gamma, \bigtriangledown_a H_1, \bigtriangledown_a H_2 \vdash \Delta} \qquad (\vdash\cap)\ \frac{\Gamma, \bigtriangledown_a H \vdash \bigtriangledown_a(G \cap H), \Delta}{\Gamma, \bigtriangledown_a H \vdash \bigtriangledown_a G, \Delta}$$

The propositional connectives have usual semantics and are treated in the standard way:

PROPOSITIONAL RULES

$$(\to\vdash)\ \frac{\Gamma, \delta \vdash \Delta \ \&\ \Gamma \vdash \Delta, \phi}{\Gamma, \phi \to \delta \vdash \Delta} \qquad (\vdash\to)\ \frac{\Gamma, \phi \vdash \Delta, \delta}{\Gamma \vdash \Delta, \phi \to \delta}$$

$$(\neg\vdash)\ \frac{\Gamma \vdash \phi, \Delta}{\Gamma, \neg\phi \vdash \Delta} \qquad (\vdash\neg)\ \frac{\Gamma, \phi \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta}$$

$$(\wedge\vdash)\ \frac{\phi, \delta, \Gamma \vdash \Delta}{\phi \wedge \delta, \Gamma \vdash \Delta} \qquad (\vdash\wedge)\ \frac{\Gamma \vdash \Delta, \phi \ \&\ \Gamma \vdash \Delta, \delta}{\Gamma \vdash \Delta, \phi \wedge \delta}$$

$$(\vee\vdash)\ \frac{\Gamma, \phi \vdash \Delta \ \&\ \Gamma, \delta \vdash \Delta}{\Gamma, \phi \vee \delta \vdash \Delta} \qquad (\vdash\vee)\ \frac{\Gamma \vdash \Delta, \phi, \delta}{\Gamma \vdash \Delta, \phi \vee \delta}$$

Short comments explaining the intuitions behind the axioms and rules may be helpful.

(axE1) is a form of *ex falso* and says that the assumption of knowing at least L and at most H is contradictory when $L \not\subseteq H$.

(axE2) states that everybody knows at least nothing, $\varnothing$.

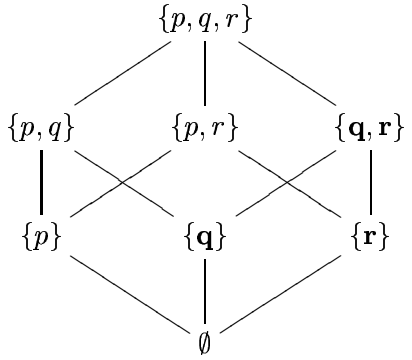
(axE3) says that if one knows no more that H then one also knows no more than G for any $G \supseteq H$.

The schema (axE4) is rather complicated and the following example illustrates its meaning.

EXAMPLE 37. *Consider the following, irreducible sequent (bottom-up application of any rule leaves it unchanged):*

$$\bigtriangledown\{p,q,r\} \vdash \bigtriangledown\{q,r\}, \bigtriangleup\{p\}$$

Any state s satisfying the assumption, i.e., $s \subseteq \{p,q,r\}$ and such that $s \not\subseteq \{q,r\}$ must contain p. This tautology is not an instance of any of the axioms (axE1), (axE2), (axE3), and (axE4) fills this gap. The possible models of $\bigtriangledown\{p,q,r\}$ are elements of the powerset lattice $\wp(\{p,q,r\})$:



$\bigtriangledown\{q,r\}$ from the sequent comprises the elements marked with bold font. If $s \models \bigtriangledown\{p,q,r\}$ is not among these, it must be one of the remaining elements which all contain p. This is captured by the side-condition of the axiom. We calculate first $\wp(\{p,q,r\}) \setminus \wp(\{q,r\}) = \{\{p,q,r\},\{p,q\},\{p,r\},\{p\}\}$. The side-condition requires each of these sets to contain some of the sets K_i occurring under $\bigtriangleup K_i$ on the right of $\vdash$, which in the present case is just $\{p\}$.

All rules are sound and inverible (sound also bottom-up). According to $(\cup\vdash)$, (not) knowing at least L_1 and L_2 is the same as (not) knowing at least $L_1 \cup L_2$ and, similarly, according to $(\cap\vdash)$, (not) knowing at most H_1 and H_2 is equivalent to (not) knowing at most $H_1 \cap H_2$. The rule $(\vdash\cup)$ read top down says that if (Γ and) knowing L implies knowing K then it also implies knowing $K \cup L$. Similarly, according to rule $(\vdash\cap)$ if (Γ and) knowing at most H implies knowing at most $G \cap H$, then it also implies knowing at most G.

EXAMPLE 38. *As an example, we give the (schema of a) proof of the statement corresponding to Proposition 34, namely,* $\triangle_i\{\triangledown_i X\} \rightarrow \triangledown_i X \vdash \neg\triangle_i\{\triangledown_i X\}$. *(All epistemic operators have the same agent index which we therefore ignore in the notation.)*

$$\dfrac{\dfrac{\dfrac{}{\triangle\{\triangledown X\}, \triangledown X \vdash}\,(axE1)}{\triangledown X \notin X} \qquad \dfrac{\dfrac{}{\triangle\{\triangledown X\} \vdash \triangle\emptyset}\,(axE2)}{\triangle\{\triangledown X\} \vdash \triangle\{\triangledown X\}}\,\vdash\cup}{\triangle\{\triangledown X\} \rightarrow \triangledown X, \triangle\{\triangledown X\} \vdash}\,\rightarrow\vdash}{\triangle\{\triangledown X\} \rightarrow \triangledown X \vdash \neg\triangle\{\triangledown X\}}\,\vdash\neg$$

The crux of the proof, as in the semantic argument in Example 35, is the side-condition $\triangledown_i X \notin X$, *i.e.,* $\{\triangledown_i X\} \not\subseteq X$, *enabling application of the axiom schema (axE1).*

The proof of completeness of $\mathcal{EC}$ is based on the invertibility of the rules and construction of a counter-model for an irreducible non-axiomatic sequent. Decidability follows easily by inspecting the rules and axioms of the calculus.

THEOREM 39. *The calculus* $\mathcal{EC}$ *is sound and complete with respect to the semantics from Definition 33. It is also decidable.*

A virtue of the above logic, besides giving a syntax-oriented semantics to epistemic language and capturing some interesting phenomena like those from Proposition 34, is that it allows deductions of what an agent does *not* know from the finite assumptions about upper bounds of his knowledge.

EXAMPLE 40. *The triviality that, if an agent* E *knows exactly* m_k, *then he does not know, for instance,* m *is proven as follows:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\triangle_E\{m_k, m\}, \triangledown_E\{m_k\} \vdash}\,(axE1)}{\triangle_E\{m_k\}, \triangledown_E\{m_k\}, \triangle_E\{m\} \vdash}\,\{m_k, m\} \not\subseteq \{m_k\}}{\triangle_E\{m_k\} \wedge \triangledown_E\{m_k\}, \triangle_E\{m\} \vdash}\,\cup\vdash}{\triangle_E\{m_k\} \wedge \triangledown_E\{m_k\} \vdash \neg\triangle_E\{m\}}\,\wedge\vdash}\,\vdash\neg$$

Another virtue of this logic is its complete independence from the possible agents' logic. It can be combined with any agent logic and in the following section we give examples of that.

5. Reasoning finite agents

The last step of our three-tiered approach amounts to enabling agents to carry out actual reasoning themselves. Since the two previous steps of the framework are entirely generic, allowing arbitrary reasoning at the agent level, while the variety of agents is virtually unlimited, we present here only one class of agents satisfying some general conditions. Applications to agents violating these conditions are not excluded but would require adjustments at appropriate places.

The main general assumption is that the agents' reasoning happens within the state(s) described by a single formula of u.l. – it does not "cross' the boundaries set by $\_;\_$, i.e., an agent described by $f_1; f_2$, carries out his entire reasoning starting with f_1 (when in the first state) or with f_2 (when in the second state) but not, for instance, with $f_1 \wedge f_2$. Of course, reference to preceding or following states is possible, though this would require particular agent logics. Also, additional requirements can be put on the sequences themselves, for instance, that they are monotonic, i.e., $f_2 \to f_1$. In general, however, no such assumptions are made and this generality also makes the presentation simpler. The second central assumption is that agents are bounded as described in the previous Section, i.e., are modelled as having finite states of "pieces of information". Their internal reasoning affects the state in which it is started. Furthermore, this internal reasoning is monotone, i.e., application of the reasoning rules results at most in an extension of agent's state. Note that this still allows handling of non-monotonicity, since non-monotonic transitions can be expressed at the level of sequence logic.

The need to equip agents with internal reasoning is, hopefully, self-explaining. We can, for instance, describe the protocol step from Section 4.1 in essentially the same way as in (25), now taking into account also the relevant upper bounds of the information gathered by the enemy agent:

$$\begin{gathered} \triangle_A\{m_k\} \wedge \triangle_B\{k\} \wedge \Box_E\emptyset \ ; \\ \triangle_A\{m_k\} \wedge \triangle_B\{m_k, k\} \wedge \Box_E\{m_k\} \\ \models \top; \triangle_B\{m\} \wedge \neg \triangle_E \{m\} \end{gathered} \tag{41}$$

The above entailment does not hold. Although the security objective, $\neg\triangle_E\{m\}$, can be proved as in Example 40, the first conjunct, $\triangle_B\{m\}$, does not follow from $\triangle_B\{m_k, k\}$.

Now, having the key k and a message m_k encoded with it, one would expect agents to be able to obtain the decoded message m. This is an ability which in ι from (25) was expressed as the additional formula describing the agents' knowledge, $\mathbf{K}_i(m_k \wedge k \to m)$. Now we model it

by providing agents with an internal logic, the rules for modifying their internal states. This ability is captured by the following internal rule of 'decoding':

$$(dec)\quad \frac{m_k, k}{m_k, k, m}$$

It says that having m_k and k, an agent can expand his state adding to it m. (One can easily formulate other rules capturing all the capacities of Dolev-Yao adversary, [9], as well as more advanced adversaries.) With its use, the agent B can pass from the state $\triangle_B\{m_k, k\}$ to $\triangle_B\{m_k, k, m\}$. But this means that the sequence on the left of $\vdash$ in (41) is *actually modified* by the agent's reasoning. Such modifications are incorporated as the lowest level of our framework. We introduce a new relation between sequence descriptions, $\sigma_1 \Vdash \sigma_2$, whose intended meaning is that starting in a sequence satisfying σ_1, agents can reason and arrive at a sequence σ_1' such that $\sigma_1' \models \sigma_2$.

We begin in 5.1 by spelling out this definition in detail. Then, we give two examples of applications to protocol description, 5.2, and to muddy children, 5.3. In the concluding Subsection 5.4, we compare the obtained framework with active logic.

5.1. Agent's reasoning and sequence descriptions

Given an agent's language $\mathcal{AL}$ (the parameter to $\mathcal{EL}$ from Definition 4.2.1), an agent's rule has the form $\frac{a}{c}$, with $a, c \subseteq \mathcal{AL}$.[6] Its application to a state $x \supseteq a$ results in the state y obtained by replacing the subset of assumptions by the set of conclusions, $y = (x \setminus a) \cup c$. The closure of a state x under application of the rules, $Cl(\{x\})$, is the collection of all finite states reachable from x by applying arbitrary rules in arbitrary order finitely many times. Closure of a set X of states is the union of closures of all states $s \in X$.[7] When, as we assume, the agent's rules are monotone, i.e., $a \subseteq c$, the result of applying a rule is simply $y = x \cup c$. Moreover, $Cl(\_)$ is then a closure operator satisfying the classical requirements, e.g., [8, 25]. Using upper-case letters for sets

[6] This generalizes trivially to an agent language being the whole $\mathcal{EL}(\mathcal{AL})$, but for simplicity we keep the stucture flat with the agent language being only the parametric atoms $\mathcal{AL}$.

[7] Different agents may have different rules, so identifying them in the notation is needed, but we ignore this aspect, addressing only a single agent.

of (finite) states, we have

$$\begin{array}{l} 1.\ X \subseteq Cl(X) \\ 2.\ Cl(Cl(X)) = Cl(X) \\ 3.\ X \subseteq Y \Rightarrow Cl(X) \subseteq Cl(Y) \end{array} \tag{42}$$

On the syntactic side, given a sequence formula, we characterize the set of sequence formulae which can be obtained from it by agents' internal reasoning. We first define such a formulae closure of a single $\mathcal{EL}(\mathcal{AL})$ formula f:

$$FCl(f) = \{f' \in \mathcal{EL}(\mathcal{AL}) \mid \forall z(z \models f \Rightarrow \exists z' \in Cl(\{z\}) : z' \models f')\} \tag{43}$$

In words, $FCl(f)$ is the set of all formulae h such that, starting from an arbitrary state z satisfying f, the agent can reach a state z' satisfying h. When f is inconsistent, $FCl(f)$ becomes the set of all formulae.

The relation $\Vdash$ is now defined as follows ($\models$ in the definiens is from (6)):

$$f_1; ...; f_n \Vdash \sigma \iff \forall 1 \leq i \leq n\ \exists f_i' \in FCl(f_i) : f_1'; ...; f_n' \models \sigma. \tag{44}$$

The next task is to reflect this definition by the appropriate rules for reasoning about the relation $\Vdash$, i.e., as a provability relation $\Vdash$ which corresponds to $\Vdash$. The observation which applies generally to all possible situations is that we must address two aspects: (i) reflection of agents' internal reasoning at the level of sequence formulae and (ii) the possible transition between $\Vdash$ and $\vdash$. The latter can be treated uniformly by the rule

$$\text{(Lift)}\ \frac{\sigma_1 \vdash \sigma_2}{\sigma_1 \Vdash \sigma_2} \tag{45}$$

which is trivially sound, since $f \in FCl(f)$ and so, choosing all $f_i' = f_i$ in (44) gives the assumption of the rule, while the right-to-left implication yields the conclusion.

The aspect (i) is more involved and we consider only a special case which is sufficient for the examples to follow. To simplify the definitions, we assume that all u.l. formulae considered are consistent. (This is easily decidable in $\mathcal{EC}$. Moreover, if some formula in the sequence on the left of $\Vdash$ is inconsistent, applying (Lift) rule gives its provability, since then the respective sequent is provable in $\vdash$ by the (ex falso) rule.)

We assume that the $\mathcal{EL}(\mathcal{AL})$ formulae are in a specific form, exemplified by (41), namely, for each agent there is at most one conjunction $\triangle x \wedge \triangledown y$ (possibly, with one conjunct missing, i.e., lonely $\triangle x$ is allowed, while lonely $\triangledown y$ is treated as $\triangle \varnothing \wedge \triangledown y$). Application of a (monotone)

rule $\frac{a}{c}$ to such a formula presupposes that $a \subseteq x \subseteq y$, the first inclusion requiring the agent's state described by $\triangle x \wedge \triangledown y$ to match the assumption of the rule and the second inclusion requiring consistency of this very formula. The result of an application is then $\triangle(x \cup c) \wedge \triangledown(y \cup c)$. The external rule (at the level of $\Vdash$) corresponding to the internal application of $\frac{a}{c}$ is thus:

$$(IR)(\frac{a}{c}) \quad \frac{...; \triangle(x \cup c) \wedge \triangledown(y \cup c)\ ; ...\ \Vdash \sigma}{...; \triangle x \wedge \triangledown y\ ; ...\ \Vdash \sigma} \text{ provided } a \subseteq x \subseteq y$$

The rule is trivially sound since the agent's rule $\frac{a}{c}$, with $a \subseteq c$, and the fact that $a \subseteq x \subseteq y$ imply $\triangle(x \cup c) \wedge \triangledown(y \cup c) \in FCl(\triangle x \wedge \triangledown y)$.

The following form of completeness is immediate from the definition of (IR):

FACT 46. *If $z' \in Cl(\{z\})$ then, for an arbitrary sequence formula σ and arbitrary (possibly empty) consistent sequence formulae $\star, *$, there is a derivation* $\dfrac{\star\ \Box z'\ *\ \Vdash \sigma}{\star\ \Box z\ *\ \Vdash \sigma}\ (IR)$.

We give two examples.

5.2. The protocol example

For the protocol example, assume the agents' language $\mathcal{AL}$ to be simply a collection of atoms, like m, m_k, etc., and (dec) be the only rule. The two conjuncts of the goal in (41) are proven using the respective agents' descriptions:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\triangle_B\{m_k,k,m\}\ \vdash\ \triangle_B\emptyset}(axE2)}{\triangle_B\{m_k,k,m\}\ \vdash\ \triangle_B\{m\}}(\vdash\cup)}{\triangle_B\{m_k,k,m\}\ \Vdash\ \triangle_B\{m\}}(\text{lift})}{\triangle_B\{k\};\triangle_B\{m_k,k,m\}\ \Vdash\ \top;\triangle_B\{m\}}(E):\triangle_B\{k\}\vdash\top}{\triangle_B\{k\};\triangle_B\{m_k,k,m\}\ \Vdash\ \top;\triangle_B\{m\}}(\text{Lift})}{\triangle_B\{k\};\triangle_B\{m_k,k\}\ \Vdash\ \top;\triangle_B\{m\}}(IR)(dec)$$

$$\dfrac{\dfrac{\dfrac{}{\Box_E\{m_k\}\ \Vdash\ \neg\triangle_E\{m\}}\text{Example 40}}{\Box_E\emptyset;\Box_E\{m_k\}\ \Vdash\ \top;\neg\triangle_E\{m\}}(E):\Box_E\emptyset\vdash\top}{\Box_E\emptyset;\Box_E\{m_k\}\ \Vdash\ \top;\neg\triangle_E\{m\}}(\text{Lift})$$

Note that the proof of the last goal is not quite sufficient. $\sigma_1 \Vdash \sigma_2$ means only that it is possible for the agents, starting from σ_1, to arrive at a

sequence entailing σ_2. The aim, however, is rather to prove that it is not possible to arrive at a situation where $\triangle_E\{m\}$, i.e., $\sigma_1 \nVdash \top; \triangle_E\{m\}$. For this particular example, or rather more generally, for every case when agent starts his reasoning in a state z for which $Cl(\{z\})$ is finite, such conclusion can be drawn provided that it follows from every $s \in Cl(\{z\})$. For each such s can be uniquely described by a single formula $\Box s$. If $\Box s \vdash \neg \triangle_E \{m\}$ for every $s \in Cl(\{z\})$, then $\triangle_E\{m\}$ will not be provable from z or any state reachable from it (due to monotonicity of agent logic). Establishing such results in general depends on the agent logic and the decidability of its combination with the rest of the system which are currently under investigation.

5.3. Muddy children

In this standard puzzle, after the initial situation in which everybody sees everybody else except himself, there follows the father's announcement that at least one child is muddy. Following that, the negative answer to the question whether anybody knows if he is muddy, brings the participants to the final state in which everybody can deduce that he is muddy. We consider only the most simple situation with two children, both of whom are muddy, and focus on one of them, A, describing only his state changes. The agent logic can be first taken to be propositional logic and identified with u.l. We use only more informative syntax indicating the intended meaning of various atoms: m_X stands for X being muddy, while $B(m_A)$ for B knowing that A is muddy (this, too, is only an atom). A passes through the following states characterizing his knowledge:

	in state	A knows that
	$s_0 = \{m_B\}$	B is muddy
$\rightsquigarrow$	$s_1 = s_0 \cup \{\neg m_A \to B(m_B)\}$	at least one is muddy (and B knows it, too)
$\rightsquigarrow$	$s_2 = s_1 \cup \{\neg B(m_B)\}$	B does not know if he is muddy

and we obtain trivially the entailment: $s_0; s_1; s_2 \Vdash \top; m_A$, i.e., the final state of A implies/involves the fact that A is muddy.

Such a simple analysis gives only a partial insight into the problem. In particular, it does not enable one to conclude that before hearing B's negative answer, A actually does *not* know that he is muddy: $\neg m_A$ does not follow from s_1 as m_A is consistent with it. To capture this aspect, explicit treatment of upper bounds is needed and we use our model of bounded agents. To simplify the presentation, we equip A with only one rule schema: $\frac{\{\neg a \to b, \neg b\}}{\{\neg a \to b, \neg b, a\}}$, which is sufficient to obtain the positive result, i.e., $\triangle_A\{s_0\}; \triangle_A\{s_1\}; \triangle_A\{s_2\} \Vdash \top; \triangle_A\{m_A\}$. The

crucial point is that the knowledge we postulate for A is not only the lower but also the upper bound. Thus, the actual sequence of states on the left of $\Vdash$ should have $\Box_A$ instead of $\triangle_A$ at all places. Then, if we stop after the first transition, we can easily prove in $\mathcal{SEQ}$ using $\mathcal{EC}$ as u.l. the assumption of the following application of (Lift):

$$\frac{\Box_A s_0; \Box_A s_1 \Vdash \neg \triangle_A \{m_A\}}{\Box_A s_0; \Box_A s_1 \Vdash \neg \triangle_A \{m_A\}} \text{ (Lift)}$$

However, as observed in the previous example, the conclusion is still not sufficient, since it means only that, after father's announcement, it is possible to reach a state where $\neg \triangle_A \{m_A\}$ holds, not that A cannot possibly reach a state which entails $\triangle_A\{m_A\}$. But now we have sufficient formal means at our disposal. Using A's rules, we have that $Cl(\{s_1\}) = \{s_1\}$, so $FCl(\Box_A s_1) = \{h \mid \Box_A s_1 \models h\}$. In particular, none of the formulae in $FCl(\Box_A s_1)$ entails more than does already $\Box_A s_1$. Thus, since $\Box_A s_1 \vdash \neg \triangle_A \{m_A\}$ and, being consistent, $\Box_A s_1 \nvdash \triangle_A\{m_A\}$, the same unprovability obtains for every formula in $FCl(\Box_A s_1)$. Therefore, we can conclude that

$$\Box_A s_0; \Box_A s_1 \nVdash \triangle_A\{m_A\}$$

which is the desired conclusion.[8]

5.4. Comparison with active logic

Active logic has been developed to reason about the changing states of beliefs, [5, 13, 17].[9] Several aspects of active logic fall naturally into our framework (e.g., stepwise reasoning, temporally evolving beliefs, possible nonmonotonicity). Other aspects addressed by active logic are in our case factored out and delegated to the choice of the underlying logic. Thus, for instance, to model bounded agents, we have to choose an appropriate logic for such agents, while to treat contradictory beliefs, some form of paraconsistent logic could be chosen. This choice is, in turn, separated from the choice of logic used by the agents (though the two may often coincide). We illustrate some of these aspects below.

[8] It is, of course, artificial to equip agents with only one rule as we did above. It would be more natural to let agents reason with full propositional logic. The corresponding conclusion can be established also for this case since $s_1 \nvdash_{Prop} m_A$ and so $\forall z' \in Cl(\{s_1\}) : \Box_A z' \nvdash \triangle_A\{m_A\}$, which implies $\Box_A s_0; \Box_A s_1 \nVdash \triangle_A\{m_A\}$. This suggests that one may hope for lifiting decidability of agent logic to decidability of $\Vdash$, with $\mathcal{EC}$ as u.l.. This issue, however, awaits a detailed treatment.

[9] Since the details vary, sometimes significantly, from one presentation to another, one should rather speak about active *logics*, which share some basic ideas. We try to address only these common basic ideas and, to the extent more specific details are commented, they are taken from [17].

5.4.1. *Stepwise unfolding of knowledge*
Active logic operates with the explicit notion of (discrete) time-points, and application of rules involves always increase of time. (This simple idea is present in the predecessor of active logic, step logic [10, 12, 14], and in its decidable fragment, [4].) For instance:

$$\frac{\mathsf{n}:\ f\ ,\ f \to g}{\mathsf{n}+1:\ g}$$

– if, at step n, f and $f \to g$ are known/available
– then, at step n+1, g can become known/available

The following rule, admissible in our system, can be used to model exactly this aspect of stepwise reasoning:

$$(step)\ \ \frac{\sigma \Vdash *f\star}{\sigma \Vdash *f; g\star}\ [f \vdash g]$$

How much an agent can do in a single step depends on a particular variant of active logic. Some variants allow to apply once all available rules in all possible ways in a single step, others only one rule, and many other variants are possible. Simply counting the rule applications is possible here as well, but we allow also much coarser granularity admitting transitions to arbitrary consequences. If we want to model the situation where only one rule is applied once in a single step, we simply restrict the proofs $[f \vdash g]$ in the side condition to those of length one. Then, for an agent possessing Modus Ponens and three formulae in his initial state as shown on the left of $\Vdash$, we can obtain a stepwise deduction:

$$\{\phi, \phi \to \psi, \psi \to \rho\} \Vdash \{\phi, \phi \to \psi, \psi \to \rho\}; \{\phi, \phi \to \psi, \psi \to \rho, \psi\}; \{\phi, \phi \to \psi, \psi \to \rho, \psi, \rho\}$$

Unlike in active logic, in our case counting of steps happens only at the meta-level (i.e., just count the number of ; in the resulting sequence). The above deduction corresponds to the active logic statement that knowing the formulae listed in the initial state, after two steps one can obtain ρ:

$$K(x, 0, \phi \wedge (\phi \to \psi) \wedge (\psi \to \rho)) \to K(x, 2, \phi \wedge (\phi \to \psi) \wedge (\psi \to \rho) \wedge \psi \wedge \rho).$$

Active logic uses a predicate OBS(erved), stating that some facts are observed at particular time-points. In sequence logic, such observations are incorporated into the descriptions simply by including the observed facts/formulae into agents' theories at the appropriate points.[10]

[10] OBS seems rather redundant from the point of view of modelling, unless one enriches the theory with more specific rules for its treatment. As it appears in [17], its only role is to give an agent knowledge of the observed facts/formulae in the next state, as captured by the only axiom involving it, **AR2**: $OBS(z, x, y) \to$

For instance, the sequence starting with the knowledge that birds fly, $\phi = \forall x(B(x) \rightarrow F(x))$, and, after some time, observation that t is a bird, $B(t)$, is expressed with the (appropriate fragment of) first-order logic as the underlying logic, as $\phi; \phi \wedge B(t)$. From this sequence, one can derive $\top; F(t)$, namely, that after some time the agent may learn that t flies. Application of (IR) at the end of the derivation refers to the appropriate derivation in agent's first-order logic.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\phi \wedge B(t) \wedge F(t) \vdash F(t)}{\phi \wedge B(t) \wedge F(t) \Vdash F(t)}\,(lift)}{\phi; \phi \wedge B(t) \wedge F(t) \Vdash \top; F(t)}\,(E)\ \phi \vdash \top}{\phi; \phi \wedge B(t) \wedge F(t) \Vdash \top; F(t)}\,(Lift)}{\phi; \phi \wedge B(t) \Vdash \top; F(t)}\,(IR)$$

However, since $\phi \nvdash F(t)$, one cannot derive $\phi; \phi \wedge B(t) \Vdash F(t)$ which would mean that agent described on the left has known that t flies from the very beginning.

Note that, in this example u.l. coincides with agent logic. This may be quite a general situation, especially when one does not focus on bounded agents as we did. In such situations, there is no need to go to the level of $\Vdash$ – the claim $\phi; \phi \wedge B(t) \vdash \top; F(t)$ can be established directly in $\mathcal{SEQ}$, without using any (IR) rules.

The above scenario, built into active logic and expressible in sequence logic, captures the idea of gradual unfolding of the implicit consequences of the explicitly given, and typically finite, knowledge, e.g., [21, 15]. Starting on the left of $\Vdash$ with the description of the initial state, the derivable right hand sides of $\Vdash$ represent then possible *implicit* consequences of such unfolding. Identifying u.l. with agent logic for simplicity, and assuming that in this logic we have a series of implications $f \rightarrow f_1$ and then, for $1 \leq i \leq n : f_i \rightarrow f_{i+1}$, the following is a valid sequence formula:

$$f \Vdash f_1; f_2; ...; f_n.$$

The advantage of sequence logic is, here as elsewhere, the possibility of combining it with an *arbitrary* underlying logic. For instance, using modal epistemic logic as the underlying logic, with the resolution rule, we can obtain the following entailment:

$$\triangle_A((a \vee b \vee c) \wedge \neg b \wedge \neg c) \ \Vdash \ \triangle_A((a \vee b \vee c) \wedge \neg b \wedge \neg c); \triangle_A((a \vee b) \wedge \neg b \wedge \neg c); \triangle_A(a \wedge \neg b \wedge \neg c)$$

$K(z, x+1, y)$ which states that if z observes y at time x, then he knows y at time $x+1$. Consequently, the non-logical axioms specifying a particular situation in terms of observations, $OBS(z, x, y)$, can be safely repalced by $K(z, x+1, y)$ and this is what happens in the following sequence logical formulation.

This particular way of using modal epistemic logic in $\mathcal{SEQ}$ becomes actually the approach taken in the logic of algorithmic knowledge, [19, 11], which incorporates the idea of counting the deduction steps into modal epistemic logic just as active logic incorporates it into first-order logic. The above statement corresponds to $K(A,0,(a\vee b\vee c)\wedge\neg b\wedge\neg c)\rightarrow K(A,2,a\wedge\neg b\wedge\neg c)$ in the language of algorithmic knowledge. (As usual, counting depends on what exactly is being counted as a single step.) Algorithmic knowledge utilizes also a more abstract version of (implicit) knowledge: $K_A^{\exists}(a\wedge\neg b\wedge\neg c)$ states that A can, at some time, know the formula given as the argument. The sequence logical expression of the same fact is $\top;\triangle_A(a\wedge\neg b\wedge\neg c)$ and its proof is obtained from the proof of the previous statement since, in $\mathcal{SEQ}$, we can abstract away any sequence of formulae/states:

$$f_1;...;f_n;f \quad \Vdash \quad \top;f.$$

We thus see that sequence logic, combined with the appropriate underlying logics, allows to capture a variety of formalisms addressing the stepwise reasoning in a generic framework. Of course, this genericity implies often that some specific aspects of other formalisms must be given more specific treatment. For instance, inclusion of the time-step, step-sequence, etc. into the language, so central in step and active logics, would require simple adjustments of the underlying logic. A difference which remains is that in sequence logic the discrete time-steps represent only an abstraction required by the description as finite series of formulae separated by ;. Unlike in active logic, its semantics uses dense (possibly continuous) time which can be considered "closer to" the modeled world. Nevertheless, as mentioned earlier, variations of sequence logic for other types of orderings are available, [6], and allow exactly the same combinations with underlying and agent logics as those described here only for dense orderings.

5.4.2. *Boundedness*

As far as modelling of agents' knowledge is concerned, active logic has actually "syntactic" semantics just like our language $\mathcal{EL}$ from 4.2. It is obtained by using the *names* of the formulae, instead of the formulae themselves, as the arguments of agents' knowledge operator. $K(A,5,\lceil f\rceil)$ means that (at time 5) agent A knows f. The machinery used to ensure that the name $\lceil f\rceil$ represents the intended formula f utilizes function names for the connectives of the agent's language. For agents working with propositional logic, one introduces the function names *con*, *imp*, etc., axiomatizes their operation, e.g., $imp(\lceil f\rceil,\lceil g\rceil)=\lceil f\rightarrow g\rceil$, and ensures by means of the additional axioms that these

meta-functions behave as the connectives they denote, e.g., $K(A,n,x) \wedge K(A,n,imp(x,y)) \rightarrow K(A,n+1,y)$.

It is exactly this syntactic lifting which enables to model in active logic non-omniscient agents, that is, agents whose theories need not be closed under any specific rules. At a given time-point, the predicate K may hold for $K(A,n,x)$ and $K(A,n,imp(x,y))$, without holding also for $K(A,n,y)$. The advantage seems to be exactly the same as that offered, in a less convoluted way, by our syntactic approach from 4.2. In the remark following Example 31, we observed that the outermost epistemic operator, handled by $\mathcal{EC}$, corresponds to the meta-level. Its argument is simply a set of mutually distinct "atoms". Even if these "atoms" have an internal structure, its meaning appears only when they either 1) are brought to the meta-level (as happens in Proposition 34, due to the axiom $\triangle_i\{\alpha\} \rightarrow \alpha$), or 2) are processed by the agents' internal rules. For instance, endowing an agent with the rule $\frac{f, f \rightarrow g}{f, f \rightarrow g, g}$, amounts to making agent's "$\rightarrow$" behave as the meta-implication, as far as modus ponens is concerned. (This corresponds to the *imp* axiom from the end of the previous paragraph.)

An important difference concerns the treatment of the bounds of agents' knowledge and is essentially the same as the difference, described in the last paragraph of 4.1, between the modal epistemic logic and our approach from 4.2. Ignoring the time-points for the moment, the K predicate of active logic corresponds to the modal **K** – it expresses only the lower bound of knowledge. Using first-order logic (which underlies active logic), one can prove, for instance, statements of the form (*) $\neg K(A, \lceil f \rceil)$. However, proving such statements will often require stronger assumptions than one would like to make (if not stronger than ones which can be expressed, cf. the last paragraph of 4.1). Moreover, [17] claims that a theory in active logic, being axiomatized using only Horn clauses, has a minimal Herbrand model which is suggested as the indended semantics. This model, however, will validate a lot of statements of the form (*) which are *not* provable. Although this opens the possibility of implementing active logic in Prolog, the treatment of the upper bounds will thus lead to the problems known from treating negation as failure.

Our model of bounded agents is both simpler and more general. It allows combination with arbitrary agent logic and, given the calculus $\mathcal{EC}$, its implementation is straightforward.

6. Conclusions and Future Work

We have presented a decidable temporal logic with a single binary modality, roughly corresponding to the chop or **U**ntil operator, and interpreted it over dense linear time. The logic is parameterized by an *arbitrary* underlying logic for describing the states. The logic inherits meta-properties of the underlying logic: it is strongly complete/sound/decidable whenever the underlying logic is.

Having then summarized the language and semantics of finite agents from [1], we presented a new decidable logic for reasoning about lower and upper bounds of epistemic agents. We have illustrated how agents can be endowed with internal reasoning mechanisms and how such mechanisms can be integrated into the unified system for reasoning about sequences of events/communications between bounded rational agents.

One of the main virtues of our approach is its thorough genericity. Sequence logic can be combined with arbitrary underlying logics while the presented logic of bounded agents can be combined with arbitrary agent logic. This genericity, however, means that combination with a particular logic is not very tight and specific applications may require additional adjustments. In particular, when agents are endowed with their internal reasoning mechanisms, additional rules for connecting their internal reasoning with the reasoning at the meta-level of sequence logic are needed. We have given a general pattern for such rules, (Lift) and (IR), and illustrated them by distinct examples in Section 5. Although specific applications may require quite specific additions, we hope that general results can be obtained determining the required (and sufficient) adjustments in terms of the properties of the agent logic. A systematic investigation of the classes of agent logics and the ways of combining them with the rest of the framework is the main topic for further work.

We have considered only dense time, but modifications of sequence logic are applicable to a wider class of orderings, [6]. For instance, a simple restriction of the (L) rule yields a sound system for *all* total orderings, but completeness requires further modifications. For ω orderings, the relation $\models$ is decidable (provided that the underlying logic is), but the problem of constructing a natural reasoning system remains open. This problem turns out to be surprisingly difficult and is currently under investigation. All such variants of sequence logic can be combined with underlying logics and agents' logics in the same way as described in this paper.

References

1. Ågotnes, T.: 2004, 'A Logic of Finite Syntactic Epistemic States'. Ph.D. thesis, Department of Informatics, University of Bergen.
2. Ågotnes, T. and M. Walicki: 2006a, 'Complete Axiomatizations of Finite Syntactic Epistemic States'. In: M. Baldoni, U. Endriss, A. Omicini, and P. Torroni (eds.): *Declarative Agent Languages and Technologies III: Third International Workshop, DALT 2005, Utrecht, The Netherlands, July 25, 2005, Selected and Revised Papers*, Vol. 3904 of *LNCS*. pp. 33–50.
3. Ågotnes, T. and M. Walicki: 2006b, 'Strongly complete axiomatizations of "knowing at most" in syntactic structures'. In: F. Toni and P. Torroni (eds.): *Computational Logic in Multi-Agent Systems, 6-the International Workshop, CLIMA 2005, London, Selected and Revised Papers*, Vol. 3900 of *LNAI*. pp. 57–76.
4. Alechina, N., B. Logan, and M. Whitsey: 2004, 'A complete and decidable logic for resource-bounded agents'. In: *3-rd Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'04)*.
5. Anderson, M. L., W. Gomaa, J. Grant, and D. Perlis: 2005, 'On the reasoning of real-world agents: towrad a semantics of active logic'. In: *7-th Annual Symposium on the Logical Formalization of Commonsense Reasoning*. Corfu, Greece.
6. Bezem, M., T. Langholm, and M. Walicki: 2007, 'Completeness and decidability in sequence logic'. In: *Proceedings of LPAR'07*, Vol. 4790 of *LNAI*.
7. Blackburn, P., M. de Rijke, and Y. Venema: 2001, *Modal Logic*. Cambridge University Press.
8. Brown, D. J. and R. Suszko: 1973, 'Abstract logics'. *Dissertationes Mathematicae* **102**, 9–42.
9. Dolev, D. and A. Yao: 1981, 'On the security of public key protocols'. In: *Proceedings of the IEEE 22nd Annual Symposium on Foundations of Computer Science*. pp. 350–357.
10. Drapkin, J. and D. Perlis: 1986, 'A preliminary excursions into step-logics'. In: *SIGART International Symposium on Methodologies for Intelligent Systems*.
11. Duc, H. N.: 2001, 'Resource-Bounded Reasoning about Knowledge'. Ph.D. thesis, University of Leipzig.
12. Elgot-Drapkin, J.: 1988, 'Step-logic: reasoning situated in time'. Ph.D. thesis, University of Maryland.
13. Elgot-Drapkin, J., S. Kraus, M. Miller, M. Nirkhe, and D. Perlis: 1999, 'Active Logics: A Unified Formal Approach to Episodic Reasoning'. Technical Report CS-TR-3680 and UMIACS-TR-99-65, University of Maryland.
14. Elgot-Drapkin, J., M. Miller, and D. Perlis: 1991, 'Memory, reason and time: the step-logic approach'. In: R. Cummins and J. Pollock (eds.): *Philosophy and AI: Essays at the Interface*. Cambridge, Mass.: MIT Press.
15. Fagin, R. and J. Y. Halpern: 1985, 'Belief, awareness, and limited reasoning'. In: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*. pp. 480–490.
16. Goranko, V., A. Montanari, and G. Sciavicco: 2004, 'A Road Map of Interval Temporal Logics and Duration Calculi'. *Journal of Applied Non-Classical Logics* **14**(1-2), 9–54.
17. Grant, J., S. Kraus, , and D. Perlis: 2000, 'A logic for characterizing multiple bounded agents'. *Autonomous Agents and Multi-agent Systems* **3**(4), 351–387.

18. Halperm, J. Y. and Y. Shoham: 1991, 'A Propositional Modal Logic of Time Intervals'. *Journal of the ACM* **38**(4), 935–962.
19. Halpern, J. Y., Y. Moses, and M. Y. Vardi: 1994, 'Algorithmic knowledge'. In: R. Fagin (ed.): *5-th Conference on Theoretical Aspects of Reasoning about Knowledge (TARK'94)*.
20. Kurucz, A., I. Németi, I. Sain, and A. Simon: 1995, 'Decidable and Undecidable Logics With a Binary Modality'. *Journal of Logic, Language, and Information* **4**(3), 191–206.
21. Levesque, H. J.: 1984, 'A logic of implicit and explicit belief'. In: *National Conference on Artificial Intelligence (AAAI-84)*. pp. 198–202.
22. Montanari, A., G. Sciavicco, and N. Vitacolonna: 2002, 'Decidability of interval temporal logics over split-frames via granularity'. In: *Logics in AI, Proceedings of 8-th European Conference, JELIA*, Vol. 2424 of *LNAI*. pp. 259–270.
23. Sistla, A. P. and E. M. Clarke: 1985, 'The complexity of propositional lineral temporal logics'. *Journal of the ACM* **32**(3), 733–749.
24. Szajnkening, W.: forthcomming, 'Sequence Logic'. Ph.D. thesis, Department of Informatics, University of Bergen.
25. Tarski, A.: 1956, *Logic, Semantics, Metamathematics*. Oxford University Press.