Sequence Logic and Bounded Agents *

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 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 (action based descriptions, active logic, bounded agents), which is a simple consequence of the parametric character of both the language and the reasoning system. We address in more detail the case of non-omniscient rational agents. Their models are syntactic structures (sets of available formulae) similar in spirit, though not in the technical formulation, to the models used in active logic. We give a sound, complete and decidable system for reasoning about such agents, and illustrate its combination with the internal reasoning of agents and the sequence logic from the first part of the paper.

* Norwegian Research Council provided partial financial support for this work under the projects MoSIS and SHIP.

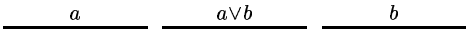
1. Introduction

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:

$$a; a \vee b; b \tag{1.1}$$

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.

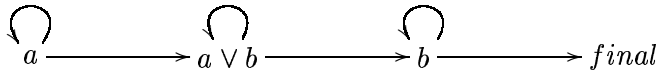
Viewing the above as a description of the result of some interactions (the announcement and the other event), it is natural to ask for possible consequences, and for an entailment relation between such descriptions in general. For example, a system

$$a \vee \neg b; a \vee b \tag{1.2}$$

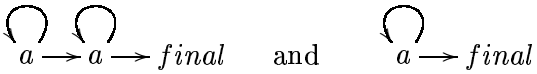
can be viewed as a consequence of the previous one, with the last two intervals concatenated. Furthermore, $a \wedge b$ during a certain interval has both (1.1) and (1.2) as consequence by appropriately cutting the interval in three and two smaller ones, respectively.

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 (1.1) 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 5.1.

The paper introduces, in Section 2, the language and semantics of such system descriptions. In Section 3 it introduces a reasoning system and provides the answer to the main question about such systems and their descriptions: given a description (specification) of the sequence of states an agent (or a group of agents) is required to pass through, like (1.1), 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 would be strongly complete with respect to intervals of total, dense orderings.

The sequence logic from Sections 2-3 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 epistemic 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 [17], but we give a new sound, complete and decidable reasoning system. We show then how bounded agents can be endowed with the internal reasoning mechanisms, and illustrate the interaction with the sequence logic on a simple example of a communication protocol. Its analysis depends in a crucial way on the ability to handle the upper bounds of agents' knowledge. In Section 5 we comment relations to the transition/rewriting systems and to the action-based descriptions and then, in more detail, to the active logic focusing on the treatment of bounded agents. Section 6 summarizes the main points and suggests further development. Proofs omitted here can be found in [16].

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 2.1 (Sequence Language SL). *The language SL – containing sequence formulae over a parameter language U – is given by the following grammar:*

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

$\top$ stands for tautology of the underlying logic – this symbol is added only if it is missing in the underlying logic.

In the sequel $\sigma, \sigma_1, \ldots$ denote (sequence) formulae of SL; $f, f_1 ... g, g_1 ...$ – atomic formulae, i.e., those without ; occurring inside. The formulae of our 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.

2.1. Semantics

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

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

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 2.2. *Satisfaction of an SL-formula σ in an 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 (2.3) and (2.4). For a semantical entailment, we write $\sigma_1 \models_{\mathcal{O},r} \sigma_2$ iff

$$\forall \mathcal{O} \forall r \forall [a, b) : [a, b) \models_{\mathcal{O},r} \sigma_1 \Rightarrow [a, b) \models_{\mathcal{O},r} \sigma_2 \tag{2.3}$$

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

$$\forall \mathcal{O} \forall r : O \models_{\mathcal{O},r} \sigma_1 \Rightarrow O \models_{\mathcal{O},r} \sigma_2 \tag{2.4}$$

It is an easy exercise to verify that $\_;\_$ is associative. One could view this operator as the until, $\mathbf{U}$, of temporal logic over linear time (depending, however, on the details of the definition which may vary). 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 and only under $\Box$, to remain true from then on.[1] Thus it is not surprising that we can express several temporal modalities, for instance:

[1] Alternatively, one can almost identify $\_;\_$ with the chop operator, common in interval logics, [13, 11]. We have, however, only a very limited fragment of such logics.

1. $r \models f$ iff f holds always in 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.

This analogy to temporal logic (of dense linear time) concerns the limited semantic interpretation. However, unlike modal logics, in general, we will offer not only sound and complete, but also strongly complete reasoning, which is also decidable. (All these properties are relative to their presence in the underlying logic.)

The semantics is based on points but, nevertheless, it is strongly interval-oriented. For the first, it 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 given in the following example.

EXAMPLE 2.5. *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* $[c, d) \not\models g$. [2] *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* 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 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* $[c, d) \models g$.

2.2. Cuts

An equivalent definition of satisfaction can be expressed by saying 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 2.6. *An* n-cut C *of* $[a, b)$ *is a partition of* $[a, b)$ *into* n *intervals*

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

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

[2] $[c, d) \sqsubseteq_I [a, b)$ denotes that $[c, d)$ is a *subinterval* of $[a, b)$, i.e., $[c, d) = \{o \in O : a \leq c \leq o < d \leq b\}$.

By $r|_\sigma$ we denote a cut of r which verifies σ, i.e., shows that $r \models \sigma$.[3]

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

EXAMPLE 2.7. *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:*

$$\begin{array}{l} r|_{\sigma_1} \\ r|_{\sigma_2} \end{array} \qquad [\ \frac{f_1}{g_1}\ |\ \frac{f_2}{\ }\ |\ \frac{f_2}{g_2}\ |\ \frac{f_2}{g_3}\ |\ \frac{f_3}{g_3}\ )$$

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

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

The following definition formalizes the superposition of two cuts. It will be used only in the 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 mark 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 2.10.)

DEFINITION 2.8. *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.

EXAMPLE 2.9. *$Paths(q)$, for q from Example 2.7, are the following:*

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

$$\frac{f_1 \mid f_2;\ldots;f_n}{g_1 \mid g_2;\ldots;g_m} \qquad \frac{f_1 \mid f_2;\ldots;f_n}{g_1 \mid g_2;\ldots;g_m} \qquad \frac{f_1 \mid f_2;\ldots;f_n}{g_1 \mid g_2;\ldots;g_m}$$

(i) (ii) (iii)

Figure 1. Possible overlappings of initial intervals.

$[f_1 \vdash g_1] - [f_2 \vdash g_2] - [f_3 \vdash g_3]$, *(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 2.7.

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 2.7.

LEMMA 2.10. *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 $r \models \pi$.*

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 $*\sigma$).

DEFINITION 3.1. *The calculus $\mathcal{SEC}$ contains the following rule schemata:*

$$\textit{(lift)}\ \frac{}{f \Vdash g}\ [f \vdash g] \qquad (L)\ \frac{f* \Vdash \sigma}{f* \Vdash g;\sigma}\ [f \vdash g]$$

$$(E)\ \frac{\sigma_1 \Vdash \sigma_2}{f;\sigma_1 \Vdash g;\sigma_2}\ [f \vdash g] \qquad (R)\ \frac{\sigma \Vdash g*}{f;\sigma \Vdash g*}\ [f \vdash g]$$

If $*$ is empty, the formula to the left of it remains unchanged. 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 3.2. *Consider the sequent q from Example 2.7. The following are two of its possible derivation:*

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

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 3.2, 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. (Its proof, in which each of the inclusions is shown by induction on the length of q, can be found [16].)

LEMMA 3.3. *For an arbitrary sequent q: $po(Der(q)) = Paths(q)$.*

DEFINITION 3.4. *A derivation Δ_q of $q = f_1; ...; f_n \Vdash g_1; ...; g_m$ is a* proof *of q iff either*

$$\exists 1 \le j \le n:\ f_j \vdash \bot \quad \textit{or}$$
$$\forall [f_i \vdash g_j] \in po(\Delta_q):\ f_i \vdash g_j$$

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 3.5. *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 3.2) 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 3.4.*

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 3.6. *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 3.7. *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 3.3.

Assuming soundness of the underlying logic, one verifies relatively easily soundness of $\mathcal{SEC}$. Lemma 3.3 is of crucial importance in the proof of completeness, and we now sketch the main steps of this proof.

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

THEOREM 3.9. *When the u.l. is strongly complete, then so is $\mathcal{SEC}$.*

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 3.4 combined with Lemma 3.3, 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, we 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 we let r_i contain any model of f_i (existing by $f_i \nvdash \bot$.) We 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 densly in the interval r_i. By Lemma 3.8, 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 2.10, 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 3.3, $Paths(q) = po(Der(q))$, 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 $f_i \not\forall g_j$. But then also $\forall s \sqsubseteq_I r_i : s \not\models g_j$ – contradicting (*). □

EXAMPLE 3.10. *Consider the following (unprovable) sequent* $q = \sigma_1 \Vdash \sigma_2$*:*

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

The $Paths(q)$ *are obtained as they were in Example 2.9, and are here listed with the formulae* 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 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 densly 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. Application to finite agents

First, in 4.1, we use as the underlying logic the epistemic logic S4 to illustrate a verification of a simple protocol goal. We point out that some goals, concerning especially the negative information about things agents will not be able to know, require explicit treatment of the bounds of agent's knowledge. In 4.2, we summarize the semantic model of finite, syntactic, epistemic agents from [17] and give a new, sequent based, reasoning system for it. In 4.3, we illustrate the use of this system as the underlying logic for the sequence logic verifying such a negative property of the protocol step from 4.1.

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, $\kappa = \mathbf{K}_B(m_k) \wedge \mathbf{K}_E(m_k)$:

$$\iota; \iota \wedge \kappa \ \Vdash \ \top; \mathbf{K}_B(m) \tag{4.1}$$

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

REMARK 4.2. *We should take some precautions in formulating the proof obligations (side-conditions in our reasoning system) when using standard epistemic/modal logics like* S4 *or* S5. *For instance, we might want to prove that, given that $a \to b$, 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:*

$$\frac{}{(a \to b) \wedge \mathbf{K}_A(a) \Vdash \mathbf{K}_A(b)} \ [(a \to b) \wedge \mathbf{K}_A(a) \vdash \mathbf{K}_A(b)] \tag{4.3}$$

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 $a \to b$ does not satisfy this condition. The general statement of this fact is as follows (e.g., exercise 1.5.3 in [3]):*

$$\Gamma \models^g \phi \iff \mathbf{K}^*(\Gamma) \models^l \phi \tag{4.4}$$

where $\models^g$ is the global logical consequence (which we are using), while $\models^l$ is local logical consequence, for which we have strongly complete reasoning systems.[4] *In the logics where $\mathbf{K}(\phi) \leftrightarrow \mathbf{K}(\mathbf{K}(\phi))$, we can simplify the right hand side of (4.4) 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 (4.4). In the example (4.1), the problem does not arise since all premises are under the modality $\mathbf{K}$.*

[4] Statement (4.4) 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.

The complete goal of the sequence $\iota; \iota \wedge \kappa$ from (4.1) 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 can not 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 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 [17], 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. Here, we use simply a set of atomic propositions, *At*.

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 sound, complete and decidable logical system for bounded agents is presented. (In 4.3 we give reasoning mechanisms to agents and describe the integration of all the parts of the framework.)

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

The *epistemic language*, $\mathcal{EL}$, in which agents represent their knowledge is defined as follows.

DEFINITION 4.5. *Formulae of the language $\mathcal{EL}$ are given by the following grammar (over a parameter of atomic formulae At):*

$$\phi ::= At \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \rightarrow \phi \mid \bigtriangleup_i T \mid \bigtriangledown_i T \tag{4.6}$$

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 agent i knows *at most* formulae $\phi_1, \ldots, \phi_n$ (he may know less): $\bigtriangledown_i\{\phi_1, \ldots, \phi_n\}$. Notation $\bowtie_i\{\phi_1, \ldots, \phi_n\}$ stands for 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 identifier will be sometimes omitted in order to simplify the generic formulations valid for all possible indices.

EXAMPLE 4.7. *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 two formulae: 1) p and 2) 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 ($\triangle_i$ in the example) will be relative to the rules with which agents may process them, which we discuss in 4.3. The semantics and logic presented in this subsection address only this 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". We will see some examples of the possible interactions betweem these two levels.

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 accessible pieces of knowledge. This amounts to a finite set of pieces (of information) which are available to an agent at any given time.

The semantics of the language, for a given set At, is just the lattice of finite sets of $\mathcal{EL}$-formulae built over At.

DEFINITION 4.8. *A* local state s_i *of an agent i is a finite set of $\mathcal{EL}$-formulae. (It represents all the formulae agent i knows/has available.)*

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 At (e.g., propositional variables) and an evaluation function $I\colon At \to \wp(S)$ 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 4.9. *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. $(s, I) \models \phi \to \phi' \iff (s, I) \not\models \phi$ *or* $(s, I) \models \phi'$
4. $(s, I) \models \phi \wedge \phi' \iff (s, I) \models \phi$ & $(s, I) \models \phi'$
5. $(s, I) \models \phi \vee \phi' \iff (s, I) \models \phi$ *or* $(s, I) \models \phi'$
6. $(s, I) \models \triangle_i T \iff T \subseteq s_i$
7. $(s, I) \models \bigtriangledown_i T \iff s_i \subseteq T$

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

PROPOSITION 4.10. *Let* $\mathbf{T}$ *denote the axiom schema (for every agent and formula)* $\triangle_i\{\alpha\} \to \alpha$. *Then, for any set of formulae X:* $\mathbf{T} \models \neg \triangle_i \{\bigtriangledown_i X\}$.

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

EXAMPLE 4.11. *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, $s_i \models \phi_1 \wedge \mathbf{T}$. Then (*) $\triangledown_i\{p\} \in s_i$ and, since $s_i \models \mathbf{T}$, also $s_i \models \triangledown_i\{p\}$. 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).

Note that satisfaction of epistemic formulae in points 6 and 7 of Definition 4.9 is independent from the evaluation of atoms, e.g., for any I, J and s, we have that $(s, I) \models \triangle_i T \iff (s, J) \models \triangle_i T$. It is the epistemic aspects which is of primary concern to us, so we will not pay much attention to evaluations.

4.2.3. *Reasoning about finite agents*
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. $\triangle_a L$ and $\triangledown_a H$ may be absent in axioms which do not mention them other places than on the left (e.g., L may be absent in (axE3) and (axE4)). In the axioms (axE1), (axE3), (axE4) Γ on the left of $\vdash$ contains no operators with subscript a, and neither does Δ in axiom (axE4), i.e., all present operators with a-subscript are mentioned explicitly in the formulation.

DEFINITION 4.12. *The calculus $\mathcal{EC}$ is parameterized by the atomic formulae At and agent names, and contains the following axiom and rule schemata:*

AXIOM SCHEMATA:

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

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

(axE3) $\Gamma, \triangle_a L, \triangledown_a H \vdash \triangledown_a G, \Delta$ *when* $H \subseteq G$

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

RULE SCHEMATA:

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

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

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

Prop

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

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

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

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

EXAMPLE 4.13. *As an example, we give the (schema of a) proof of the statement corresponding to Proposition 4.10, namely,* $\triangle_i\{\bigtriangledown_i X\} \to \bigtriangledown_i X \vdash \neg\, \triangle_i\, \{\bigtriangledown_i X\}$. *(All epistemic operators have the same agent index which we therefore ignore in the notation.)*

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

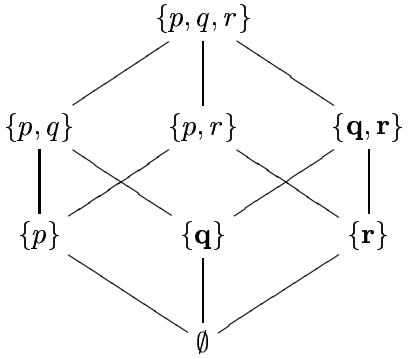
The crux of the proof, as in the semantic argument in Example 4.11, is the side-condition $\bigtriangledown_i X \notin X$, *i.e.,* $\{\bigtriangledown_i X\} \not\subseteq X$, *enabling application of the axiom schema (axE1).*

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

EXAMPLE 4.14. *Consider the following, irreducible sequent (bottom-up application of any rule leaves it unchanged):*

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

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* $\{p, q, r\}$ *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* $\{p\}$*.*

THEOREM 4.15. *The calculus* $\mathcal{EC}$ *is sound and complete with respect to the semantics from Definition 4.9. 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 4.10, is that it allows deductions of what an agent does *not* know from the assumptions about upper bounds of his knowledge.

EXAMPLE 4.16. *The triviality that, if an agent* E *knows exactly* m_k*, then he does not know, for instance,* m *is proven as follows:*

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

4.3. Reasoning finite agents

The rule $(\vdash)\ \dfrac{*\ g\ \star \Vdash \sigma}{*\ f\ \star \Vdash \sigma}\ [f\vdash g]$, admissible in $\mathcal{SEC}$ by Lemma 3.7, connects the consequence relation $\vdash$ in the underlying logic with the consequence relation $\Vdash$ at the level of sequences. However, the finite agents as described above, do not have any proper reasoning tools. The logic $\mathcal{EC}$ relates only different set-descriptions. We can, for instance, describe the protocol step from 4.1 in essentially the same way as in (4.1), now taking into account also the relevant upper bounds of the information gathered by the enemy agent:

$$\begin{array}{c}\bigtriangleup_A\{m_k\}\wedge\bigtriangleup_B\{k\}\wedge\bowtie_E\emptyset\ ;\\ \bigtriangleup_A\{m_k\}\wedge\bigtriangleup_B\{m_k,k\}\wedge\bowtie_E\{m_k\}\\ \Vdash\top;\bigtriangleup_B\{m\}\wedge\neg\bigtriangleup_E\{m\}\end{array}\qquad(4.17)$$

The above goal is not provable. In particular, the first conjunct, $\bigtriangleup_B\{m\}$, does not follow in $\mathcal{EC}$ from the fact that $\bigtriangleup_B\{m_k,k\}$.

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 (4.1) 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 their internal logic, the rules for modifying their internal states. We consider only monotone agents here, and the above ability is captured by the following internal rule of 'decoding':

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

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, [4], as well as more advanced adversaries.)

We now fix the underlying logic to be $\mathcal{EC}$. Given a description (of a protocol) $f_1;...;f_n$ for which we want to establish some goal $g_1;...;g_m$, we are now asking whether agents can transform their states f_i, using their internal rules, so that the goal becomes provable. This is captured by the following – $\mathcal{SEC}(\mathcal{EC})$-special – version (IR) of the $\mathcal{SEC}$-rule ($\vdash$):

$$(IR)(\tfrac{F}{G})\ \ \frac{*\triangle(G\cup L)\wedge\bigtriangledown(H\cup G)\star' \Vdash \sigma}{*\triangle(F\uplus L)\wedge\bigtriangledown H\star \Vdash \sigma}$$

Since application of internal rules amounts to extending the state, we have to adjust possible upper bounds. If before the rule application (bottom-up) agent knows at most H, $\bigtriangledown H$, after the application $\frac{F}{G}$ it becomes $\bigtriangledown(H\cup G)$. By the assumption of monotonicity, we have that $F\subseteq G$. In the lower bound, we therefore replace F by G.[5]

Note that this gives a new relation, marked at the syntactic level with $\Vdash$, and which corresponds to the semantic

$$f_1;...;f_n \Vdash \sigma \iff Cl(f_1);...;Cl(f_n)\models\sigma \tag{4.18}$$

where $Cl(f_i)$ represents the closure of f_i under the agents' internal rules, and the satisfaction on the right hand side is that from Section 2, as given in (2.3-2.4), with $\mathcal{EC}$ as the underlying logic.

The following rule (Lift) captures this connection allowing to continue the proof in $\mathcal{SEC}$, once the closure under agents' internal rules has been reached using (IR).

$$\text{(Lift)}\ \frac{\sigma_1\vdash\sigma_2}{\sigma_1\Vdash\sigma_2}\ \ \sigma_1=Cl(\sigma_1) \tag{4.19}$$

Of course, this rule relies on the crucial assumption that such a closure can be reached in finitely many steps. Although this may be the case in many practical applications, a general treatment of agents with arbitrary internal logics would require different versions of this rule and is under ongoing investigation.

Endowing agents with the rule (dec), and extending $\mathcal{SEC}$ with (IR)(dec) and (Lift), the two conjuncts of the goal in (4.17) are proven using the respective agents' descriptions:

$$\begin{array}{rl}
 & (axE2)\\
\triangle_B\{m_k,k,m\} \vdash \triangle_B\emptyset & \\
\hline
\triangle_B\{m_k,k,m\} \vdash \triangle_B\{m\} & (\vdash\cup)\\
\hline
\triangle_B\{m_k,k,m\} \Vdash \triangle_B\{m\} & \\
\hline
\triangle_B\{k\};\triangle_B\{m_k,k,m\} \Vdash \top;\triangle_B\{m\} & \text{(E): } \triangle_B\{k\}\vdash\top\\
\hline
\triangle_B\{k\};\triangle_B\{m_k,k,m\} \Vdash \top;\triangle_B\{m\} & \text{(Lift): } \triangle_B\{m_k,k,m\}=Cl(\triangle_B\{m_k,k,m\})\\
\hline
\triangle_B\{k\};\triangle_B\{m_k,k\} \Vdash \top;\triangle_B\{m\} & \text{(IR)(dec)}
\end{array}$$

[5] Monotonicity of agents may require also propagation of such updates through the rest of the sequence which is marked in the premise of the rule by $\star'$.

Example 4.16

$$\frac{\triangle_E\{m_k\} \wedge \bigtriangledown_E\{m_k\} \Vdash \neg \triangle_E \{m\}}{\bowtie_E \emptyset; \triangle_E\{m_k\} \wedge \bigtriangledown_E\{m_k\} \Vdash \top; \neg \triangle_E \{m\}} \text{(E): } \bowtie_E \emptyset \vdash \top$$

$$\frac{\bowtie_E \emptyset; \triangle_E\{m_k\} \wedge \bigtriangledown_E\{m_k\} \Vdash \top; \neg \triangle_E \{m\}}{\bowtie_E \emptyset; \triangle_E\{m_k\} \wedge \bigtriangledown_E\{m_k\} \Vdash \top; \neg \triangle_E \{m\}} \text{(Lift): } \bowtie_E \{m_k\} = Cl(\bowtie_E \{m_k\})$$

$$\bowtie_E \emptyset; \triangle_E\{m_k\} \wedge \bigtriangledown_E\{m_k\} \Vdash \top; \neg \triangle_E \{m\} \quad \text{(IR)(dec)}$$

The example in this section has been intentionally chosen so simple to give a clear impression of the main components of the system and steps involved in its applications. Less trivial applications are to be found in [16].

5. Related systems and applications

The presented system, being interval-based, seems to require a comparison with other interval logics. However, since it possesses a number of desirable properties usually missing in such logics, it promises also potential for applications, and we comment briefly such possibilities. (An overview of other interval logics can be found in [11].) Relations to other frameworks – transition systems, 5.1, action-based descriptions, 5.2, and active logic, 5.3 – indicate also possible applications in the areas where these other techniques are applied.

5.1. 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 5.1. $\sigma_1 \cong \sigma_2$ *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 5.2. *$\sigma_1 \models \sigma_2$ 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 2.10, which verifies $\sigma_1 \models \sigma_2$.

5.2. Representing actions

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 an 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, as we did in Section 4, 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 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. Addressing such a worst case scenario, we rewrite the sending event as we did in (4.1) and (4.17):

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

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

Once a series of actions is rewritten as a series of their effects, σ_1, we can apply $\mathcal{SEC}$ for deriving its consequences, by asking for σ_2 such that $\sigma_1 \Vdash \sigma_2$, as we did in 4.1 and 4.3. 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.

5.3. Active logic

Active logic has been developed to reason about the changing states of beliefs, [2, 8, 12].[6] Although our framework does not aim at equal completeness of modelling, several aspects of active logic fall naturally into it (e.g., stepwise reasoning, temporally evolving beliefs, possible nonmonotonicity). Other aspects addressed by active logic have been 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.

5.3.1. *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 [5, 7, 9], and in its decidable fragment, [1].) 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:

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

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.[7]

[6] 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 [12].

[7] 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 [12], its only role is to give an agent knowledge of the observed

For instance, the sequence starting with the knowledge that birds fly, $\phi = \forall x(B(x) \to F(x))$, and, after some time, observation that t is a bird, is expressed with the (appropriate fragment of) first-order logic as the underlying logic, as given on the left of $\Vdash$. From this sequence, one can derive $\top; F(t)$, namely, that after some time (in the next step) the agent may learn that t flies.

$$\phi; \phi \wedge B(t) \quad \Vdash \quad \phi; \phi \wedge B(t); \phi \wedge B(t) \wedge F(t) \quad \Vdash \quad \top; F(t)$$

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.

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., [15, 10]. 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. 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:

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

This particular way of using modal epistemic logic in $\mathcal{SEC}$ becomes actually the approach taken in the logic of algorithmic knowledge, [14, 6], 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 $K(A, 0, (a \vee b \vee c) \wedge \neg b \wedge \neg c) \to K(A, 2, a \wedge \neg b \wedge \neg c)$ 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 $\mathcal{SEC}$, 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, our semantics uses dense (possibly continuous) time which can be considered "closer to" the modeled world.

5.3.2. *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

facts/formulae in the next state, as captured by the only axiom involving it, **AR2**: $OBS(z, x, y) \to K(z, x+1, y)$ 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.

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., $imp(\lceil f \rceil, \lceil g \rceil) = \lceil f \to g \rceil$, and ensures by means of the additional axioms that these meta-functions behave as the connectives they denote, e.g., $K(A, n, x) \wedge K(A, n, imp(x, y)) \to K(A, n+1, y)$.

It is exactly this syntactic lifting which enables to model 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 4.7, we pointed out that the outermost epistemic operator corresponds to the meta-level (which is addressed by the reasoning in $\mathcal{EC}$). 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 4.10, due to the axiom $\triangle_i\{\alpha\} \to \alpha$), or 2) are processed by the agents' internal rules. For instance, endowing an agent with the rule $\frac{f, f \to g}{f, f \to g, g}$, amounts to making agent's "$\to$" 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, [12] 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.

6. Conclusions and Future Work

We have presented a temporal logic with a single binary modality, roughly corresponding to **U**ntil, 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 [17], 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, under special assumptions, into the unified system for reasoning about sequences of events/communications between bounded rational agents.

One of the main virtues of sequence logic is its thoroughly generic character allowing combination with arbitrary underlying logics. 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 simple example of such additional rules (IR) and (Lift) in Section 4.3. 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 agents' logic.

We have addressed only dense time, but modifications of sequence logic are applicable to a wider class of orderings. For instance, a simple restriction of the (L) rule yields a sound system for *all* total orderings, but completeness requires further modifications. For discrete 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.

References

1. 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)*.
2. 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.
3. Blackburn, P., M. de Rijke, and Y. Venema: 2001, *Modal Logic*. Cambridge University Press.
4. 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.
5. Drapkin, J. and D. Perlis: 1986, 'A preliminary excursions into step-logics'. In: *SIGART International Symposium on Methodologies for Intelligent Systems*.
6. Duc, H. N.: 2001, 'Resource-Bounded Reasoning about Knowledge'. Ph.D. thesis, University of Leipzig.
7. Elgot-Drapkin, J.: 1988, 'Step-logic: reasoning situated in time'. Ph.D. thesis, University of Maryland.
8. 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.
9. 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.
10. 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.
11. 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.
12. 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.
13. Halperm, J. Y. and Y. Shoham: 1991, 'A Propositional Modal Logic of Time Intervals'. *Journal of the ACM* **38**(4), 935–962.
14. 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)*.
15. Levesque, H. J.: 1984, 'A logic of implicit and explicit belief'. In: *National Conference on Artificial Intelligence (AAAI-84)*. pp. 198–202.
16. Szajnkening, W.: forthcomming, 'Sequence Logic'. Ph.D. thesis, Department of Informatics, University of Bergen.
17. Ågotnes, T.: 2004, 'A Logic of Finite Syntactic Epistemic States'. Ph.D. thesis, Department of Informatics, University of Bergen.