

Inference as doxastic agency.  
Part II: Ramifications and refinements

Grigory K. Olkhovikov  
Ruhr University Bochum  
Department of Philosophy II, Universitätsstraße 150  
D-44780 Bochum, Germany  
Ural Federal University  
Department of Philosophy, 19 Mira street  
620002 Ekaterinburg, Russia  
grigory.olkhovikov@rub.de, grigory.olkhovikov@gmail.com

Heinrich Wansing  
Ruhr University Bochum  
Department of Philosophy II, Universitätsstraße 150  
D-44780 Bochum, Germany  
Heinrich.Wansing@rub.de

Abstract

Justification stit logic is a logic for reasoning about proving as a certain kind of activity, namely seeing to it that a proof is publicly available. It merges the semantical analysis of deliberately seeing-to-it-that from stit theory (Belnap, Perloff, Xu 2001) and the semantics of the epistemic logic with justification from (Artemov and Nogina 2005). In this paper, after recalling its language and basic semantical definitions, various ramifications and refinements of justification stit logic are presented and discussed: imposing natural restrictions upon the class of models under consideration, making use of modalities that assert the existence of a proof, introducing a variant of justification stit logic based on a semantics introduced by M. Fitting, and adding variable-binding operators and extending the set of proof polynomials.

Keywords: proofs as acts, doxastic agency, epistemic logic, justification logic, dstit logic

## 1 Introduction

In [10], the first part of this paper, we introduced a relatively simple modal propositional logic for reasoning about the activity of proving. Within this logic, (which throughout this paper will be called basic justification stit logic) one can systematically distinguish between dynamic and static aspects of proving, that is to say, between proofs as acts and proofs as objects, compare different sets of intuitions concerning the semantics of the verb ‘to prove’, and answer some philosophically meaningful questions like whether actions themselves (rather than static objects) can be a proper object of proof theory.

The phenomena modelled in justification stit logic naturally call for a relatively rich set of modalities connected by rather involved logical relations. Indeed, one needs a separate set of action modalities (which we borrowed from stit logic, see [2]), a separate set of proof modalities and epistemic modalities (which we borrowed from the justification logic S4LP by Artemov and Nogina, see [1]), and also a separate set of modalities for proving as an activity, which, according to the detailed motivation given in [10], come at least in four different kinds.

Therefore, in the first part of this paper we were mainly focusing on exploring these relations under a most straightforward set of choices as to the semantics and syntax of justification stit logic. Indeed, the Artemovian semantics presented in [10] is rather uncritically built upon the semantics given in [2] and [1], and the set of modalities for proving as an activity was an obvious attempt to carry over some modes of speaking about acts of proving in natural languages and implant them into the environment of modal propositional logic. Both of these choices admit of at least *prima facie* interesting alternatives which deserve some discussion and comparison, and this is the main goal of this second part of the present paper. In this part, therefore, our intention is to look at the logic defined in the first part as an entry point into a rich landscape of related logical systems which naturally arise from this basic system not unlike the way in which different modal logics arise from the system  $K$  by either restricting its semantics, or tampering with its set of expressive means (say by introducing binary modalities rather than just unary ones), or else switching to a richer semantical setting (say, neighborhood semantics). This landscape of further justification stit systems will be mostly explored in this paper with a conceptual focus, although we will cite some more technical results in one or two places when they seem to be especially relevant.

The paper has the following layout. In Section 2 we recall the basic justification stit logic with Artemovian semantics defined in [10]. Then, in

Section 3, we consider some alternatives to this logic which can be obtained by different natural restrictions placed upon the class of its models. In Section 4 we consider justification stit logic based on so-called *E*-notions (basically, modalities asserting the existence of a proof) as an alternative to a logic based on modalities for acts of proving. We are mainly dwelling on the properties of the explicit fragment of the logic of *E*-notions which seems to constitute a natural extension for the explicit fragment of the basic justification stit logic. However, once one switches to the level of full basic justification stit logic, it appears that one cannot re-define all of its modalities in terms of *E*-notions unless one extends the language with quantifiers over proof polynomials. The latter move, however, leads to quite a bit of inconvenience in the context of Artemovian semantics. Therefore, the logic of *E*-notions cannot be seen as a proper substitute of the basic justification stit logic. We conclude that at least in the context of Artemovian semantics, *E*-notions serve better as an addition to the set of modalities for acts of proving than as an alternative to this set.

Section 5 is then devoted to an alternative semantics for basic justification stit logic which is based on a semantics introduced by M. Fitting in [4]. This semantics is only explored here on the propositional level, but it is clear that it allows for a much more natural upgrade to quantification over proof polynomials as compared to Artemovian semantics. We then briefly discuss strengths and weaknesses of the two semantical settings for justification stit logic. In Section 6, we consider a further natural extension of justification stit logic obtained by admitting lambda-abstraction as a variable-binding, additional operation on proof polynomials. We also briefly compare this refinement of justification stit logic with Fitting's introduction of quantifiers into the logic of proofs.

Finally, in Section 7 we briefly sum up our conclusions and mention some directions for future research.

## 2 Basic justification stit logic

### 2.1 Language

We start with a brief recapitulation of basic justification stit logic introduced in [10]. Its language is built on top of three sets: the set *Var* of propositional variables, the set *Pol* of proof polynomials and the set *Ag* of agents, where the first two sets are countably infinite and *Ag* is finite. We will denote propositional variables by  $p, q, r$  and agents by  $i, j, k$ , using subscripts if necessary. The set of proof polynomials is obtained by taking a

countably infinite set of proof variables  $PVar$  and a countably infinite set of proof constants  $PConst$  and then closing it w.r.t. the binary operations  $+$ ,  $\cdot$ , and the unary operation  $!$ . We will denote proof variables by  $x, y, z$ , proof constants by  $a, b, c$  and proof polynomials by  $s, t, u$ , sometimes using subscripts. Informally, the meanings of operations on proof polynomials are given below:

$s + t$	the sum of proofs $s$ and $t$
$s \cdot t$	the application of proof $s$ to proof $t$
$!s$	'proofchecker', i.e. a proof deciding whether $s$ is a correct proof

Members of the set  $Form$  of justification stit formulas will be denoted by  $A, B, C, D$ , possibly with subscripts. The set  $Form$  is built upon  $Var$  using Boolean connectives  $\wedge$  and  $\neg$  and the following set of modalities:

Notation	Informal interpretation
$[c]_j A$	Agent $j$ sees to it that $A$
$\square A$	$A$ is historically necessary
$t:A$	$t$ is a proof of $A$
$KA$	$A$ is provable/knowable
$Prove(j, A)$	Agent $j$ proves $A$
$Prove(j, t, A)$	Agent $j$ proves $A$ by $t$
$Proven(A)$	$A$ has been proven
$Proven(t, A)$	$A$ has been proven by $t$

Let  $\pi = \{Prove( , ), Prove( , , ), Proven( , ), Proven( , , )\}$ . We will refer to the elements of  $\pi$  as  $\pi$ -modalities. In the above language, the Boolean connectives  $\vee, \rightarrow, \leftrightarrow, \perp, \top$  can be defined in a standard way. We also define  $\diamond$  as the dual of  $\square$ . In the above notation for our action modality,  $[c]_j$ ,  $c$  stands for *cstit* (Chellas stit), so that the criterion for  $j$  doing  $A$  is, roughly speaking, that  $A$  is sure to obtain after  $j$ 's actions. This also applies to situations when  $A$  is guaranteed to obtain irrespectively of any actions by  $j$ , simply by itself. To rule out such situations, a *dstit* (deliberative stit) action modality  $[d]_j$  is normally introduced, with the following definition:

$$[d]_j A =_{df} [c]_j A \wedge \neg \square A.$$

## 2.2 Artemovian semantics

For the language at hand, we introduced in [10] the following semantics. An Artemovian model of basic justification stit logic is a tuple

$$\mathcal{M} = \langle Tree, \leq, Choice, R, R_e, \mathcal{E}, Act, V \rangle.$$

In this structure,  $Tree$  is a non-empty set of moments which are conceived, roughly, as momentary ‘snapshots’ of the universe. These moments are causally connected by  $\leq$ , which is assumed to be a forward-branching partial order. Moreover, it is assumed that  $\leq$  weaves  $Tree$  into a single causally connected structure so that all the moments are possible states of one and the same World. More precisely, it is assumed that

$$(\forall m, m_1 \in Tree)(\exists m_2 \in Tree)(m_2 \leq m \wedge m_2 \leq m_1).$$

The possible histories of this World are then defined as all  $\leq$ -maximal chains in  $Tree$ . We denote the set of all histories by  $Hist$ . The set

$$H_m = \{h \in Hist \mid m \in h\}$$

is called the set of all histories passing through  $m$ . Justification stit logic shares with stit logic the intuition that truth values of formulas at a given moment may depend on which of the histories passing through a given moment is considered. This holds not so much for hard facts like ‘this table is round’ but rather for events dependent on the future, like tomorrow’s sea-battles and outcomes of unfinished actions. Thus the set of evaluation points for the formulas of justification stit logic in a given model  $\mathcal{M}$ , much like in stit logic, is the following set of moment-history pairs:

$$MH(\mathcal{M}) = \{(m, h) \mid m \in Tree, h \in H_m\}.$$

At any given moment  $m$  the agents perform their actions, and this affects the determination of which histories in  $H_m$  are realized. The complete information about such influences is encoded in  $Choice$  which is assumed to be a function computing, for a given moment  $m$  and agent  $j$ , a set of choices of  $j$  at  $m$  which is given as a partition of  $H_m$ . This partition is denoted by  $Choice_j^m$  and its elements are different choices of the agent. If  $h \in H_m$  then  $Choice_j^m(h)$  stands for the choice of  $j$  at  $m$  to which  $h$  belongs.

In the context of justification stit logic, unlike in the case of general stit logic, it is assumed that agents living and acting in  $Tree$  are engaged in a specific activity of proving things. At any given moment, a totality of proofs is available, and this totality is represented by  $Pol$ . This totality does not change from moment to moment, but the status of proofs themselves may change: some proofs are not admissible, the others are, and non-admissible proofs may become admissible as history evolves (but not vice versa — once a proof has become admissible this cannot be undone). And in order to be valid, a proof must not only be admissible but also correct in the sense that the proved sentence is true in all the relevant moment-history pairs.

This latter aspect of proofs is handled in a (by now) traditional Hintikkaean fashion by the binary relation  $R_e$  in our models. Relation  $R_e$ , called an evidence accessibility relation, is assumed to be a pre-order on  $Tree$ , and we further demand that  $\leq$  is a subset of  $R_e$ , so that the possible future of a given moment is always relevant to weighing evidence in it. The admissibility aspect of proofs is then encoded by a function  $\mathcal{E}$ , called admissible evidence function. For every moment  $m$  and proof polynomial  $t$ , this function yields a set of justification stit formulas for which  $t$  is an admissible proof in  $m$ . This function is assumed to satisfy the following closure properties w.r.t. operations on proof polynomials:

- $A \rightarrow B \in \mathcal{E}(m, s) \wedge A \in \mathcal{E}(m, t) \Rightarrow B \in \mathcal{E}(m, s \cdot t)$ ;
- $A \in \mathcal{E}(m, t) \Rightarrow t:A \in \mathcal{E}(m, !t)$ ;
- $\mathcal{E}(m, s) \cup \mathcal{E}(m, t) \subseteq \mathcal{E}(m, s + t)$ .

It is easy to see how these closure properties correspond to the interpretation of the respective operations; e.g., the last clause says that  $s + t$  is an admissible proof for any sentence, for which either  $s$  or  $t$  is admissible. Further, since admissibility of evidence is assumed to be cumulative, the following restriction is also imposed on  $\mathcal{E}$ :

$$(\forall t \in Pol)(\forall m, m' \in Tree)(R_e(m, m') \Rightarrow \mathcal{E}(m, t) \subseteq \mathcal{E}(m', t)).$$

Relation  $R_e$  and function  $\mathcal{E}$  are both necessary to interpret modalities of the form  $t:A$  which explicitly mention the proof involved in showing  $A$ . However, the modality  $KA$  could already be interpreted with the help of  $R_e$  only. Still, in justification logic, which (in addition to stit logic) is the second source of justification stit logic, a separate pre-order  $R$  is assigned to that task with the condition that  $R \subseteq R_e$ .<sup>1</sup> In the context of justification stit logic,  $R$ , just like  $R_e$  is defined on  $Tree$ .

These constructions embed the totality of proofs into our model as some objective feature about the way the World is; but they do not yet say anything about how agents can relate themselves to this totality. In basic justification stit logic this relation is established by *Act* which is assumed to be a function mapping  $MH(\mathcal{M})$  into  $2^{Pol}$ . The underlying intuition here is something like this: agents are supposed to search *Pol* together and present to the community different elements of *Pol* which they happen to find. One can think, for instance, of a group of researchers summoned in a conference room and

---

<sup>1</sup>In pure justification logic this somewhat more general setting does not affect the set of theorems and we expect the same for justification stit logic.

putting proofs they come up with on a common whiteboard. These proofs, once they are presented to the community in this way, are supposed to remain on the whiteboard forever. Function  $Act$  encodes the state of this whiteboard in that it assigns every moment-history pair a set of proofs presented (or being presented) to the community in it; and the fact that presented proofs cannot vanish later is captured by the following restriction:

$$(\forall m, m' \in Tree)(m' < m \Rightarrow \forall h \in H_m(Act(m', h) \subseteq Act(m, h))).$$

The only remaining element of the above model,  $V$ , is a standard evaluation function for propositional variables mapping  $Var$  into  $2^{MH(M)}$ .

In the above discussion, we omitted some of the less trivial and more technical restrictions on our models; we now list all of them for the sake of completeness:

1. No choice between undivided histories:

$$(\forall m, m' \in Tree)(\forall h, h' \in H_m)(\forall j \in Ag)(m < m' \wedge m' \in h \cap h' \Rightarrow \\ Choice_j^m(h) = Choice_j^m(h')).$$

2. Independence of agents:

$$(\forall m \in Tree)(\forall f : Ag \rightarrow 2^{H_m})(\forall j \in Ag)(f(j) \in Choice_j^m \Rightarrow \\ \bigcap_{j \in Agent} f(j) \neq \emptyset).$$

3. No new proofs guaranteed:

$$(\forall m \in Tree)(\bigcap_{h \in H_m} (Act(m, h)) \subseteq \bigcup_{m' < m, h \in H_{m'}} (Act(m', h))).$$

4. Presenting a new proof makes histories to divide:

$$(\forall m, m' \in Tree)(\forall h, h' \in H_m)(m < m' \wedge m' \in h \cap h' \Rightarrow \\ (Act(m, h) = Act(m, h'))).$$

5. Presented proofs are epistemically transparent:

$$(\forall m, m' \in Tree)(R_e(m, m') \Rightarrow (\bigcap_{h \in H_m} (Act(m, h)) \subseteq \bigcap_{h' \in H_{m'}} (Act(m', h')))).$$

We now briefly comment on the conditions given in this latter list. The first two restrictions are inherited by justification stit logic from stit logic; the first one says that agents cannot by their actions distinguish between histories which are, by the causal order of the world, bound to run together for some more time. The second one says that the joint choice of any group of agents cannot prevent any parallel choices by other agents. The remaining three restrictions deal with specific phenomena studied in justification stit logic. The third one says that a proof cannot occur as a hard fact out of nothing. If a proof's presence on the community's 'whiteboard' is an accomplished fact, then it must have been introduced there by some agent at an earlier moment and at this moment its introduction to the community was not an accomplished fact, but was dependent on the actions of the introducing agent. The fourth restriction can be viewed as a corollary to the first one. The difference between the sets of proofs presented to the community at  $(m, h)$  and at  $(m, h')$  may only obtain as a result of different choices of agents made at  $m$  (or earlier), since the presentation of proofs can be effected by agents only. But in this case  $(m, h)$  and  $(m, h')$  should have been in different choice cells for at least one agent, which contradicts the fact that  $h$  and  $h'$  are undivided at  $m$ . The fifth restriction says that whenever the presence of a given proof on the community whiteboard has become an accomplished fact, the community knows it, so that this presence also obtains in all the epistemic alternatives of a given state.

With all the restrictions in place, we can finally define the satisfaction relation for basic justification stit logic. For every model  $\mathcal{M} = \langle Tree, \leq, Choice, Act, R, R_e, \mathcal{E}, V \rangle$  of the above-described kind, and for every  $(m, h) \in MH(\mathcal{M})$  we stipulate that:

$$\begin{aligned}
\mathcal{M}, (m, h) \models p &\Leftrightarrow (m, h) \in V(p); \\
\mathcal{M}, (m, h) \models [c]_j A &\Leftrightarrow (\forall h' \in Choice_j^m(h))(\mathcal{M}, (m, h') \models A); \\
\mathcal{M}, (m, h) \models \Box A &\Leftrightarrow (\forall h' \in H_m)(\mathcal{M}, (m, h') \models A); \\
\mathcal{M}, (m, h) \models t:A &\Leftrightarrow A \in \mathcal{E}(m, t) \wedge \forall m' \forall h'(R_e(m, m') \wedge h' \in H_{m'} \Rightarrow \mathcal{M}, (m', h') \models A); \\
\mathcal{M}, (m, h) \models KA &\Leftrightarrow \forall m' \forall h'(R(m, m') \wedge h' \in H_{m'} \Rightarrow \mathcal{M}, (m', h') \models A); \\
\mathcal{M}, (m, h) \models Prove(j, A) &\Leftrightarrow (\forall h' \in Choice_j^m(h))(\exists t \in Act(m, h'))(\mathcal{M}, (m, h) \models t:A) \wedge \\
&\quad \wedge (\forall s \in Pol)(\exists h'' \in H_m)(\mathcal{M}, (m, h) \models s:A \Rightarrow s \notin Act(m, h'')); \\
\mathcal{M}, (m, h) \models Proven(A) &\Leftrightarrow (\exists t \in Pol)(\forall h' \in H_m)(t \in Act(m, h') \wedge \mathcal{M}, (m, h) \models t:A); \\
\mathcal{M}, (m, h) \models Prove(j, t, A) &\Leftrightarrow (\forall h' \in Choice_j^m(h))(t \in Act(m, h') \wedge \mathcal{M}, (m, h) \models t:A) \wedge \\
&\quad \wedge (\exists h'' \in H_m)(t \notin Act(m, h'')); \\
\mathcal{M}, (m, h) \models Proven(t, A) &\Leftrightarrow (\forall h' \in H_m)(t \in Act(m, h') \wedge \mathcal{M}, (m, h) \models t:A).
\end{aligned}$$

In the above clauses we assume that  $p \in Var$  and that  $j \in Ag$ ; we also assume standard clauses for Boolean connectives. We observe that the above clauses for  $\Box A$  and  $[c]_j A$  are borrowed from stit logic and that the clauses for  $t:A$  and  $KA$  are an obvious adaptation of the corresponding clauses for S4LP given in [1]. As for the remaining parts of the definition of the satisfaction relation, the ones for  $Proven(A)$  and  $Proven(t, A)$  are rather transparent. A sentence has been proven (or has been proven by  $t$ ) iff some proof of it (or this specific proof  $t$ ) is correct and exists independently of what agents may do about switching from one history through a given moment to another one, or, in other words, it exists at a given moment under all such histories. As for  $Prove(j, t, A)$ ,  $j$  proves  $A$  by  $t$  iff  $t$  is a correct proof of  $A$  and  $j$  ensures by her actions that this proof is presented to the community. With  $Prove(j, A)$  it is more or less along the same lines only this time  $j$  must ensure by her actions that some correct proof of  $A$  is presented, and every such proof is novel in the sense that no proof of  $A$  was presented earlier.<sup>2</sup>

Admittedly, these are not the only plausible choices when it comes to defining a semantics of ascriptions of proving; in [10] we offered some informal motivation as to why the choices presented above seem to be the most adequate. We have also shown how some other appealing choices as to notions of proving can be defined within basic justification stit logic in terms of the above  $\pi$ -modalities. We list these definitions below:

$$Prove'(j, A) =_{df} Prove(j, A) \wedge \Diamond \neg Prove(j, A);$$

$$Prove'(j, t, A) =_{df} Prove(j, t, A) \wedge \neg Proven(A);$$

$$Prove''(j, t, A) =_{df} Prove'(j, t, A) \wedge \Diamond \neg Prove(j, A).$$

### 3 Refinements by restriction

Perhaps the simplest way to refine the basic justification stit logic explained above, would be to restrict its class of models. We briefly survey some strategies and ideas for such restrictions which seem to be *prima facie* promising from a conceptual viewpoint.

First, one could impose new restrictions immediately on the level of Artemovian justification stit models. One interesting direction of search for new principles is about different intuitively appealing patterns of non-trivial connection between the structure of *Act* and operations on proof polynomials.

---

<sup>2</sup>The semantics of justification stit logic merges the semantics of the *dstit*-operator from stit theory [2, 7, 8] and the semantics of the epistemic logic with justification presented in [1]. A logic of imagination the semantics of which merges the *dstit*-modality and a neighbourhood modality can be found in [9, 11].

Consider  $+$ , for instance. If  $s+t$ , i.e. the sum of proofs  $s$  and  $t$  is presented to the community, it is hard to imagine how this could be done without somehow presenting both  $s$  and  $t$ . But if we assume this connection in general, this assumption already amounts to the following non-trivial restriction on *Act*:

$$(\forall s, t \in Pol)(s + t \in Act(m, h) \Rightarrow s, t \in Act(m, h)) \quad (1)$$

for every  $(m, h) \in MH(\mathcal{M})$ . Imposing (1) on Artemovian models affects the set of valid formulas of justification stit logic. For example, the following formula is not valid over the class of all Artemovian models but becomes a validity over the subclass satisfying (1):

$$(Prove(j, s + t, A) \wedge s : B) \rightarrow (Prove(j, s, B) \vee Proven(s, B)).$$

One may also come up with similar restrictions on *Act* with respect to other operations on proofs, but these tend to be less appealing intuitively. For example, what exactly is entailed by a presence of  $s \cdot t$  in  $Act(m, h)$ ? To assume that  $s$  will be also presented to the community would seem to be a little bit too strong a claim. Indeed, it often happens that a certain proof method is only known by its application in one or two particular results and it may take decades to extract this proof method from this very particular environment and find an appropriate generalized formulation for it. With respect to the proof  $t$  to which  $s$  is applied in  $s \cdot t$  our intuitions seem to be more favorable. Indeed, even if  $t$  was not known earlier, one would have to at least create a copy of  $t$  for this particular application of  $s$  to it. If we agree with this, this would amount to the following non-trivial restriction on *Act*:

$$(\forall s, t \in Pol)(s \cdot t \in Act(m, h) \Rightarrow t \in Act(m, h)) \quad (2)$$

for every  $(m, h) \in MH(\mathcal{M})$ . And this new restriction, again, would yield even more new validities like the following one:

$$(Prove(j, s \cdot t, A) \wedge t : B) \rightarrow (Prove(j, t, B) \vee Proven(t, B)).$$

Another strategy to look for promising restrictions on the class of Artemovian justification stit models would be to restrict admissible models rather than admissible frames. In this case one could attempt to start with some new validities and then impose them as new principles of justification stit logic and see which restrictions on models correspond to these new principles. This alone can constitute a substantial line of research. For now we only mention some of the new validities which would capture some natural properties of proofs. First, one could try to capture proof-theoretic properties of connectives and modal operators occurring in justification stit formulas.

With respect to conjunction, for example, one would expect that in order to prove  $A \wedge B$  it is necessary to prove both  $A$  and  $B$ . This principle could be expressed as follows:

$$\text{Prove}(j, A \wedge B) \rightarrow (\text{Prove}(j, A) \vee \text{Proven}(A)),$$

and similarly for  $B$ . If, in addition, one has intuitionistic leanings towards the disjunction property, then a similar principle for disjunction would be of interest, that is to say, that a proof for  $A \vee B$  always means that there was a proof for  $A$  or a proof for  $B$  somewhere along the way. This will be expressed in basic justification stit logic as follows:

$$\text{Prove}(j, A \vee B) \rightarrow (\text{Prove}(j, A) \vee \text{Proven}(A) \vee \text{Prove}(j, B) \vee \text{Proven}(B)).$$

Secondly, one could look for principles capturing a ‘dynamic meaning’ of operations on proofs themselves. Indeed, it is often assumed that these operations are nothing but generic actions performed by agents with respect to their argument proofs considered as objects. If so, then once any given proofs  $s$  and  $t$  are in place, any agent can perform the next step and choose to construct and present to the community  $s \cdot t$ , or  $s + t$ , or  $!s$ . This, of course, is a further idealization since in the real world agents can be distracted or otherwise put in some unfavorable conditions preventing them from constructing a further proof. But, again, if we picture the community of agents as assembled in a conference room and specifically engaged in finding and writing down new proofs, this additional no-distractions idealization seems to be a very natural and realistic thing to suppose. Once we agree to this further idealization, the agentive meaning of the three operations on proof polynomials assumed in justification logic can be represented as follows. For the sum operation we have:

$$(\text{Proven}(s, A) \wedge \text{Proven}(t, B)) \rightarrow (\text{Proven}(s + t, A) \vee \Diamond \text{Prove}(j, s + t, A)). \quad (3)$$

For the application operation we have the following principle:

$$(\text{Proven}(s, A \rightarrow B) \wedge \text{Proven}(t, A)) \rightarrow (\text{Proven}(s \cdot t, B) \vee \Diamond \text{Prove}(j, s \cdot t, B)). \quad (4)$$

And finally, the agentive meaning of the proofchecker can be formalized as follows:

$$\text{Proven}(s, A) \rightarrow (\text{Proven}(!s, s:A) \vee \Diamond \text{Prove}(j, !s, s:A)). \quad (5)$$

Note, incidentally, that these principles provide sort of an inversion for the validities resulting from restrictions (1) and (2).

Yet another family of possible restrictions would concern the interaction of the admissible evidence function  $\mathcal{E}$  with stit modalities and  $\pi$ -modalities. Consider, for example the interaction of  $\Box$  with justification modalities. Some of the natural validities associated with this type of interaction are  $t:A \rightarrow \Box t:A$  and  $t:A \rightarrow \Box A$ . These validities say, roughly speaking, that if  $t$  proves  $A$ , then it is necessarily so, and also that whatever, which is proved, obtains necessarily. Now, the natural question is, when  $A$  is proved, does it mean that eo ipso the fact that  $A$  obtains necessarily is proved, or this would require a separate proof? This is basically a question of how explicit one wants to be about the apodeictic nature of proof. Classical mathematics, for example, has often been construed along the lines that whatever is proven, is proven to hold necessarily, so that the proof that  $A$  and the proof that, necessarily,  $A$  would be one and the same proof. If this idea is to be introduced in jstit logic, it will amount to the following additional restriction on  $\mathcal{E}$ :

$$A \in \mathcal{E}(m, t) \Rightarrow \Box A \in \mathcal{E}(m, t),$$

for every formula  $A$  and proof polynomial  $t$ . Such a modification of closure conditions for  $\mathcal{E}$  would yield some additional validities, for example  $t:A \rightarrow \Box t\Box A$ . Note that in this case we get more uniformity in the interactions of justification modalities with  $\Box$ , since the implicit version of the above validity, namely,  $KA \rightarrow \Box K\Box A$ , holds already within the basic jstit logic. The other option, however, would be to say that a proof of  $\Box A$  not only can, but even must be different from the proof of  $A$ , since the former proof can only be obtained after applying something like the Gödel rule to the last line of the latter proof. This approach would be, perhaps, more in line with the tendency of justification logic to mirror the functions of all the logical means available within the language at the level of proof polynomials. When pursued further, this approach would lead to the extension of the set of operations on proof polynomials with an operation  $\Box$  together with the following additional constraint on  $\mathcal{E}$ :

$$A \in \mathcal{E}(m, t) \Rightarrow \Box A \in \mathcal{E}(m, \Box(t)).$$

Of course, the choice between these two strategies of accomodating the interactions between justifications and historical necessity would depend on one's modelling aims and philosophical inclinations.

One thing to note about the language of basic justification stit logic and its Artemovian semantics is that this whole system is rather conservative w.r.t. its sources of inspiration. It is clear, for instance that if  $\mathcal{M} = \langle Tree, \leq, Choice, R, R_e, \mathcal{E}, Act, V \rangle$  is an Artemovian model for basic justification stit logic, then  $\langle Tree, \leq, Choice \rangle$  is just a  $BT + AC$  structure in the

sense of [2] and  $\langle Tree, R, R_e, \mathcal{E}, V \rangle$  is just a model of S4LP as defined in [1]. Further, all the stit and justification modalities definable on these respective substructures are inherited by basic justification stit logic in their original form. The choice of the new modalities is also rather straightforward and conservative: the four new modalities are just different modes of the English verb ‘to prove’ defined so as to accommodate for the two crucial dichotomies introduced in stit logic and justification logic respectively, namely the dichotomy of agentive vs. non-agentive w.r.t. sentences and the dichotomy of explicit vs. implicit w.r.t. knowledge.

This conservative character of basic justification stit logic with Artemovian semantics makes it relatively accessible and easy to introduce and explain for readers already familiar with stit logic and justification logic, which is one of the reasons why we have chosen it to be the main subject of the first part of this paper. However, this neither means that this choice is the only possible one nor that legitimate arguments in favor of other choices cannot be adduced. These alternatives may depart from the one described above either in their choice of the language for basic justification stit logic or in their semantical definitions. In the following sections, we consider and discuss some of these alternatives to basic justification stit logic with Artemovian semantics and consider the tradeoffs involved in switching to these alternatives.

## 4 $E$ -notions as an alternative to $\pi$ -modalities

We start by considering possible criticism of the choice of  $\pi$ -modalities for basic justification stit logic. One possible argument against this choice might be that it takes as its basic elements the concepts which are defined by an interplay of several essentially different elements of our semantics. While the modalities inherited by justification stit logic from stit or justification logic are normally connected with just one element in the model tuple and are defined very much along the lines of the classical Kripkean definition of necessity-type modalities,<sup>3</sup> with  $\pi$ -modalities the situation is very different. Their definitions invoke *Act*, the only element of justification stit models which is not inherited from either stit logic or justification logic, but they only invoke it in a non-trivial connection with all the other elements in the model, like *Choice* and  $R_e$ . And the definitions of the  $\pi$ -modalities at least in some cases look much more involved than those typically used in modal logic.

---

<sup>3</sup>In this respect  $t:A$  is somewhat different, being based on two elements ( $\mathcal{E}$  and  $R_e$ ) and having a slightly more involved definition. But this case is also well-known in existing literature under the name of modalities with an awareness condition (see, e.g. [3]).

For example, in the semantic definition for  $Prove(j, A)$  an  $\forall\exists$ -prefix is used which is not very common in modal propositional logic. If one could clear this logical mess up by introducing one or two relatively simple modalities based solely on  $Act$  but not invoking the other elements of the model and then retrieve the whole variety of the  $\pi$ -modalities in the resulting language by using expressive powers of modal propositional logic, this would make the basic justification stit logic much easier to handle and comprehend.

Unfortunately, it is not quite clear at the moment whether it is possible to substitute proving modalities with the simpler ones based on  $Act$  only without losing any expressive power. We illustrate the problems involved here by considering a typical and natural alternative to  $\pi$ -modalities which we will call  $E$ -notions.

Indeed, a modality directly representing the contents of  $Act$  at a given moment-history pair can be defined at least in the following two versions, corresponding to the distinction between explicit modalities (displaying a proof polynomial) and implicit modalities (not displaying a proof polynomial):

$$\mathcal{M}, (m, h) \models Et \Leftrightarrow t \in Act(m, h);$$

$$\mathcal{M}, (m, h) \models EA \Leftrightarrow (\exists t \in Pol)(t \in Act(m, h) \wedge \mathcal{M}, (m, h) \models t:A).$$

Thus  $Et$  says, in effect, that the proof  $t$  exists in the sense that  $t$  is presented to the community. On the other hand  $EA$  says that some proof of  $A$  is presented to the community (or exists). Since these modalities are connected with a certain kind of existence claims we have chosen for them the above notation and will hereafter refer to them as  $E$ -notions.

Is it possible to analyze the four  $\pi$ -modalities away as logical combinations of  $E$ -notions with stit and justification modalities already borrowed from justification stit logic? The following theorem shows that at least in some cases the answer is in the affirmative:

Theorem 1. Assume justification stit logic with  $Et$  and  $EA$  added as the new modalities. In this logic, the following formulas will be universally valid for arbitrary justification stit formula  $A$ ,  $j \in Ag$ ,  $t \in Pol$ :

$$Prove(j, t, A) \leftrightarrow [d]_j Et \wedge t:A$$

$$Proven(t, A) \leftrightarrow \Box Et \wedge t:A$$

$$Prove'(j, A) \leftrightarrow [d]_j(EA)$$

$$Proven(A) \rightarrow \Box EA,$$

where the last implication is irreversible.

It is evident from the above theorem that even though  $E$ -notions are sufficient to retrieve some of the  $\pi$ -modalities, they do not seem to be able to define all of them — we basically get three modalities out of seven in this way. It is not clear, for example, how one would get a definition of  $Prove(j, A)$  in the propositional setting, even though the stronger  $Prove'(j, A)$  is definable. Also, the definition of  $Proven(A)$  seems to be impossible. One would think of  $\Box EA$  as perhaps the most natural defining construction, but the difference between the two formulas is in the position of the existential quantifier w.r.t. the universal one, which results in  $Proven(A)$  being a strictly stronger proposition than  $\Box EA$ .

However, the introduction of quantifiers over proof polynomials would seem to set the potential of  $E$ -notions in a much more favourable light. Indeed, the following theorem holds:

Theorem 2. Consider justification stit logic augmented with  $E$ -notions. The following conditionals hold for arbitrary justification stit model  $\mathcal{M}$ ,  $(m, h) \in MH(\mathcal{M})$ , for arbitrary justification stit formula  $A$  and  $j \in Ag$ :

$$\mathcal{M}, (m, h) \models Prove(j, A) \Leftrightarrow$$

$$(\mathcal{M}, (m, h) \models [c]_j(EA) \wedge (\forall t \in Pol)(\mathcal{M}, (m, h) \models t:A \rightarrow \Diamond \neg Et));$$

$$\mathcal{M}, (m, h) \models Proven(A) \Leftrightarrow (\exists t \in Pol)(\mathcal{M}, (m, h) \models \Box Et \wedge t:A);$$

$$\mathcal{M}, (m, h) \models EA \Leftrightarrow (\exists t \in Pol)(\mathcal{M}, (m, h) \models Et \wedge t:A).$$

Theorems 1 and 2 combined show that in a first-order setting  $Et$  alone would be sufficient to get both  $EA$  and all of the proving modalities. We could use, for instance, the following definitions:

$$EA =_{df} \exists t(Et \wedge t:A);$$

$$Prove(j, A) =_{df} [c]_j(EA) \wedge \forall t(t:A \rightarrow \Diamond \neg Et);$$

$$Proven(A) =_{df} \exists t(\Box Et \wedge t:A).$$

Then we could define the remaining  $\pi$ -modalities as shown above. However, switching to a first-order setting within modal logic typically comes with a price — both computationally and philosophically. In case of justification logic this holds even more so, since, as the next section shows, Artemovian semantics is not likely to allow for a nice upgrade to a first-order setting. Therefore, at least in the context of propositional modal logic, the tradeoff in replacing the  $\pi$ -modalities of basic justification stit logic with  $E$ -notions

seems to be along the lines of exchanging more simplicity for less expressive power rather than exchanging a less involved set of basic modalities for the same logic for a more involved one.

On the other hand, the reverse definability also seems to fail, that is to say, one cannot define  $E$ -notions in terms of the  $\pi$ -modalities. Therefore, a simple addition of  $E$ -notions to the full language of basic justification logic appears to yield a strictly more expressive formalism. Note, further, that if one is only interested in the explicit fragment of basic justification stit logic, given by the pair of  $\pi$ -modalities  $\{Prove(j, t, A), Proven(t, A)\}$ , then it follows from Theorem 1, that the explicit fragment of the logic of  $E$ -notions (i.e., the one which includes  $Et$  but not  $EA$ ) both provides a nice reduction of explicit  $\pi$ -modalities to a much simpler set of modalities and a natural extension of its expressive means.

The explicit fragment of the logic of  $E$ -notions has a number of other merits, which we briefly mention here without proof. First, this fragment is simple enough to admit of a strongly complete finitary axiomatization which is given by the Hilbert-type system with the following set of axiom schemata:<sup>4</sup>

A full set of axioms for classical propositional logic (A0)

$S5$  axioms for  $\Box$  and  $[j]$  for every  $j \in Agent$  (A1)

$\Box A \rightarrow [j]A$  for every  $j \in Agent$  (A2)

$(\Diamond[j_1]A_1 \wedge \dots \wedge \Diamond[j_n]A_n) \rightarrow \Diamond([j_1]A_1 \wedge \dots \wedge [j_n]A_n)$  (A3)

$(s:(A \rightarrow B) \rightarrow (t:A \rightarrow (s \times t):B)$  (A4)

$t:A \rightarrow (!t:(t:A) \wedge KA)$  (A5)

$(s:A \vee t:A) \rightarrow (s + t):A$  (A6)

$S4$  axioms for  $K$  (A7)

$KA \rightarrow \Box K\Box A$  (A8)

$\Box Et \rightarrow K\Box Et$  (A9)

The assumption is that in (A3)  $j_1, \dots, j_n$  are pairwise different.

---

<sup>4</sup>We present here the axiomatization for a particular format of proof constant specification given by rule (R3), but this system accommodates to the other constant specifications in a straightforward way. Another thing to note here is that this axiomatization was proved to be complete over a somewhat simplified class of models with the additional constraint  $R = R_e$ . We have mentioned above that this particular constraint does not change anything in the context of pure justification logic. However, it is an open question, whether this is still the case within the richer environment of the logic at hand.

To this set of axiom schemes we add the following rules of inference:

From  $A, A \rightarrow B$  infer  $B$ ; (R1)

From  $A$  infer  $KA$ ; (R2)

If  $A$  is an instance of (A0)–(A9) and  $c \in PConst$ , then infer  $c:A$ ; (R3)

From  $KA \rightarrow (\neg \Box Et_1 \vee \dots \vee \neg \Box Et_n)$  infer  $KA \rightarrow (\neg Et_1 \vee \dots \vee \neg Et_n)$ . (R4)

Secondly, the explicit fragment of the logic of  $E$ -notions is expressive enough to distinguish between different time structures in the underlying model. For example the logic of the explicit  $E$ -notion over justification stit models with discrete time<sup>5</sup> is already different from the same logic over arbitrary justification stit models, and it is an open question, whether the explicit fragment of basic justification stit logic can distinguish between discrete and non-discrete times in the underlying justification stit models in the same way. One example of a formula which is valid over models with discrete times but not over arbitrary justification stit models is

$$K(\neg \Box Ex \vee \Box Ey) \rightarrow (\neg Ex \vee Ey);$$

in order to get a system which would be strongly complete w.r.t. the class of justification stit models based on discrete time, one has to replace rule (R4) in the above axiomatic system with the following stronger rule of inference:

From  $KA \rightarrow (\neg \Box Et_1 \vee \dots \vee \neg \Box Et_n \vee \Box Es_1 \vee \dots \vee \Box Es_k)$  infer  
 $KA \rightarrow (\neg Et_1 \vee \dots \vee \neg Et_n \vee Es_1 \vee \dots \vee Es_k)$ . (R4')

We end this section with some positive illustrations of the expressive power which the explicit  $E$ -notion  $Et$  seems to add to justification stit logic. For example, we saw that restrictions (1) and (2) affect the set of validities of basic justification stit logic. However, it is not immediately clear how to axiomatize these restrictions over the basic justification stit logic, or, in other words, it is not quite clear what formulas one needs to add to the set of basic justification stit validities in order to ensure satisfaction of (1), or (2), or both. With  $E$ -notions the situation is much clearer: the two restrictions are axiomatized by the schemes

$$E(s + t) \rightarrow (Es \wedge Et) \text{ and } E(s \cdot t) \rightarrow Et,$$

---

<sup>5</sup>Discreteness may be defined in different ways. One way of making the above statement fully precise is to say that a justification stit model is based on discrete time iff every history in it is embeddable into the set of natural numbers with its intended order. There exist, however, a number of less restrictive alternative readings of ‘discrete time’ which still verify the above statements.

respectively.

The situation is very much the same with formalizing the agentive meaning of the proof operations — again,  $E$ -notions offer us a much more elegant and concise way to capture this meaning and they even allow for a more nuanced picture of this type of meaning. For example, if we are only looking for the counterparts of (3), (4), and (5), these can be expressed in terms of  $E$ -notions as follows:

$$\begin{aligned} (\Box Es \wedge \Box Et) &\rightarrow \Diamond[c]_j E(s \circ t) && \text{for } \circ \in \{+, \cdot\}; \\ \Box Es &\rightarrow \Diamond[c]_j E(!s). \end{aligned}$$

These principles, just like (3), (4), and (5) above, tell us that if the constituent terms of a proof polynomial are presented to the community, then any agent may take the next step and present the proof polynomial as well. Note that this implies a rather strong presence of the proof polynomials in question within the fan of histories passing through a given moment: for every agent in the community, these histories must make up at least one entire choice at a given moment in its entirety. However, with  $E$ -notions a weaker version of the agentive meaning for the proof operations is also available, namely,  $E$ -notions allow us to say that if the constituent terms of a proof polynomial are presented to the community, then the proof polynomial itself may end up being presented to the community, even though, perhaps, no agent can ensure this. In order to express this one only needs to omit the cstit modalities in the consequents of the above formulas.

On the other hand, we might prefer to restrict presented proofs in the opposite direction. We might want to say that presenting proof polynomials is always something brought about by an agent. Even though one may fail to recognize that a given proof  $t$  is a proof of a given sentence  $A$ , and thus may not intend to present  $t$  as a proof of  $A$ , yet it is hardly possible to present  $t$  without someone seeing to it that  $t$  is present. It is not clear how to express this sort of agentiveness w.r.t. presented proofs in basic justification stit logic, i.e., in terms of  $\pi$ -modalities alone, but with  $E$ -notions it is sufficient to assume the following schema

$$Et \rightarrow [c]_{j_1} Et \vee \dots \vee [c]_{j_n} Et,$$

where  $\{j_1, \dots, j_n\} = Ag$ , for every  $t \in Pol$ .

All these examples show, in our opinion, that justification stit logic augmented with  $E$ -notions becomes a more flexible and expressive formalism while staying within the confines of both the propositional modal setting and Artemovian semantics.

Another obvious extension, incidentally, would be to add temporal operators, but we do not explore this option in any detail here, since the method of extending a stit formalism with temporal operators is well-known from existing literature. We only mention here that in the case of  $E$ -notions already the past-tense half of Priorian temporal logic would be enough to retrieve the full set of  $\pi$ -modalities. Indeed, let  $P$  be the standard weak past tense operator which in the context of stit formalisms is normally defined by the following clause:

$$\mathcal{M}, (m, h) \models PA \Leftrightarrow (\exists m' \in Tree)(m' < m \wedge \mathcal{M}, (m', h) \models A).$$

Then one can define  $Proven(A)$  as  $PEA$  and  $Prove(j, A)$  as  $[c]_jEA \wedge \neg PEA$ , which, together with the definitions given in Theorem 1, allows to express in terms of  $E$ -notions,  $P$ , stit, and justification modalities all the four  $\pi$ -modalities of the basic jstit logic.

## 5 Fitting's semantics as an alternative to Artemovian semantics

We now turn to a possible criticism of Artemovian semantics. In the above sections, we have already mentioned a possibility of upgrading the underlying system of justification stit logic from classical propositional logic to a first-order one with quantifiers over proofs. However, one has to admit that Artemovian semantics is not particularly well-suited for such a transition. Note that in the above setting the realm of objectively existing proofs is embodied by  $Pol$  once and for all, and never changes however one may switch moments and histories through a given model. In particular, proof variables under this reading are just another type of members in this realm of objective proofs rather than genuine variables actually running over some domain. Therefore, were we to switch to a first-order setting, we would be compelled to introduce a further set of variables running over  $Pol$  to be bound by quantifiers. For this further sort of variables, proof variables would be just as legitimate a substitution value as proof constants! On the other hand the realm of proofs presented to the community at any given moment-history pair  $(m, h)$  is realised by the domain  $Act(m, h)$  which may change if one changes the value of  $m$  or  $h$ . This is more in line with standard ways of setting up possible world semantics for modal quantified logic, and one would expect a more smooth transition to a first-order setting if the realm of objectively existing proofs were not fixed for every model once and for all, but would rather be allowed to vary from one moment-history pair to another. In the

existing literature, this type of approach to the semantics of justification statements was implemented by M. Fitting in his paper [4]. Not surprisingly, the main objective of the paper was a first-order generalisation of justification logic.

Therefore, one might expect that justification stit logic would be better suited to an upgrade to a first-order version if the justificational part of our semantics were given along the lines of [4] rather than in accordance with the standard semantics of S4LP. However, a simple-minded substitution of Fitting-style justificational semantics for the Artemovian one runs into certain difficulties. First, an upgrade to first-order justificational logic from a propositional one entails also an enrichment of the algebra of proof polynomials. Fitting himself had to introduce one new operator on proofs in order to handle the properties of uniform proofs. It is not clear how many further proof operations one would have to introduce into justification stit logic in order to do justice to its new first-order basis. In any case, this question is outside our present scope. Therefore, we would like to give here not so much a Fitting-style semantics for a first-order version of justification stit logic, but rather re-style the semantics of propositional justification stit logic according to the ideas of [4]. In our opinion, such a propositional system would allow for a much easier upgrade to a first-order version than the one described in Section 4. A second limitation is imposed by the very character of results presented and explained in [4]: Fitting explicitly mentions that the propositional version of his semantics leads to just LP, that is to say, to S4LP without  $K$ , but it is not known whether retaining  $K$  in this semantics will give us back exactly S4LP. Therefore, in our re-definition of the justification stit semantics according to Fitting, we will omit the modality  $K$  from our language and its semantic counterpart  $R$  from the structure of justification stit model so that now justification stit models will extend the models of LP rather than S4LP.

With these provisions, the alternative justification stit semantics will look as follows. A Fitting justification stit model is a tuple

$$\mathcal{M} = \langle Tree, \leq, Ag, Choice, R_e, \mathcal{E}, \mathcal{D}, Act, \mathcal{I}, V \rangle$$

such that:

- $Tree, \leq, Ag, Choice, R_e, V$  are of the same type and are subject to the same limitations as in Artemovian models;
- $\mathcal{D}$  maps  $Tree$  into a family of non-empty sets disjoint from the other elements of the model. It is assumed to be monotonic w.r.t.  $R_e$ , that is to say, the conditional

$$mR_em' \Rightarrow \mathcal{D}(m) \subseteq \mathcal{D}(m')$$

must be satisfied for all  $m, m' \in Tree$ .

We will denote the union of these sets as follows:

$$\overline{\mathcal{D}} = \bigcup_{m \in Tree} (\mathcal{D}(m)).$$

- $Act$  is a function mapping  $MH(\mathcal{M})$  into  $\overline{\mathcal{D}}$  such that for every  $(m, h) \in MH(\mathcal{M})$  it is true that  $Act(m, h) \subseteq \mathcal{D}(m)$ .
- $\mathcal{I}$  is a function interpreting proof polynomials. It assigns an element of  $\overline{\mathcal{D}}$  to every proof constant and assigns to  $+$ ,  $\cdot$  and  $!$  operations on  $\overline{\mathcal{D}}$  of respective arity. This function is connected with  $\mathcal{D}$  so that in order for  $\mathcal{M}$  to qualify as a Fitting justification stit model,  $\mathcal{D}(m)$  has to contain  $\{c^{\mathcal{I}} \mid c \in Const\}$  and be closed under the three operations  $+^{\mathcal{I}}$ ,  $\cdot^{\mathcal{I}}$  and  $!^{\mathcal{I}}$  for every  $m \in Tree$ .

In the above description we omitted restrictions placed on the function  $\mathcal{E}$ . The reason is that explaining these restrictions requires developing some further auxiliary notions related to evaluation functions. Since proof polynomials are now assumed to be evaluated by elements of  $\overline{\mathcal{D}}$ , these evaluations begin to play an important role in assigning truth-values to justification stit formulas. So one can define the set of evaluations in  $\mathcal{M}$  (write  $Eval(\mathcal{M})$ ) to be the set of all functions mapping  $PVar$  into  $\overline{\mathcal{D}}$ . Elements of  $Eval(\mathcal{M})$  will be denoted by  $u, v, w$ , possibly with natural indices. As usual, for a  $v \in Eval(\mathcal{M})$ ,  $x \in PVar$ , and  $r \in \overline{\mathcal{D}}$ , we will denote by  $v_r^x$  the evaluation of proof variables in  $\mathcal{M}$  such that  $v_r^x$  always coincides with  $v$  except possibly on  $x$ , for which we have  $v_r^x(x) = r$ . We can now proceed to define, for arbitrary  $t \in Pol$ , the value of  $t$  under evaluation function  $v$  denoted by  $t^v$ . The definition is by induction on the construction of  $t$  and proceeds as follows:

1.  $x^v = v(x)$ , if  $x \in PVar$ ;
2.  $c^v = c^{\mathcal{I}}$ , if  $c \in Const$ ;
3.  $(t \circ s)^v = \circ^{\mathcal{I}}(t^v, s^v)$ , if  $\circ \in \{+, \cdot\}$ ;
4.  $(!t)^v = !^{\mathcal{I}}(t^v)$ .

This new dependence of formulas on the evaluation of proof terms, however, affects not only the definition of the satisfaction relation for this type of semantics but also the definition of the models itself, since the function  $\mathcal{E}$  also has to take such evaluations into an account. Therefore, the domain of  $\mathcal{E}$  in Fitting semantics is re-defined as  $Tree \times \overline{\mathcal{D}} \times Eval(\mathcal{M})$  with the following

restrictions assumed for all Fitting justification stit models for all  $r, r' \in \overline{\mathcal{D}}$ ,  $v, v' \in Eval(\mathcal{M})$ ,  $m, m' \in Tree$ ,  $t \in Pol$ , and for all justification stit formulas  $A, B$ :

- $r \notin \mathcal{D}(m) \Rightarrow \mathcal{E}(m, r, v) = \emptyset$ ;
- $mR_e m' \Rightarrow \mathcal{E}(m, r, v) \subseteq \mathcal{E}(m', r, v)$ ;
- $(A \rightarrow B \in \mathcal{E}(m, r, v) \wedge A \in \mathcal{E}(m, r', v)) \Rightarrow B \in \mathcal{E}(m, (r \cdot^I r'), v)$ ;
- $(A \in \mathcal{E}(m, r, v) \wedge t^v = r) \Rightarrow t : A \in \mathcal{E}(m, !^I(r), v)$ ;
- $\mathcal{E}(m, r, v) \cup \mathcal{E}(m, r', v) \subseteq \mathcal{E}(m, (r +^I r'), v)$ ;
- If  $v$  and  $v'$  agree on all proof variables occurring in  $A$ , then  $A \in \mathcal{E}(m, r, v) \Leftrightarrow A \in \mathcal{E}(m, r, v')$ .

In this list the first condition says that proofs outside  $\mathcal{D}(m)$  cannot be admitted (let alone presented to the community) at  $m$ , whereas the last condition requires that evaluations can affect the sets of admissible proofs for a formula only by affecting the values of proof variables occurring in the formula. The rest are just reformulations of monotonicity, application, sum and inspection conditions imposed onto  $\mathcal{E}$  in LP.

Having thus completed the definition of Fitting justification stit models, we now turn to the inductive definition of the satisfaction relation. We assume a propositional language which extends both the language of justification logic (without  $K$ ) and of stit logic with the following set of modalities:

$$Proven(A), Proven(t, A), Prove(j, A), Prove(j, t, A), Et, EA.$$

We will say that a formula  $A$  in this language is true in  $\mathcal{M}$  at  $(m, h) \in MH(\mathcal{M})$  under the evaluation  $v \in Eval(\mathcal{M})$  and will write this fact down as  $\mathcal{M}, (m, h) \models_v A$ . The inductive definition of  $\mathcal{M}, (m, h) \models_v A$  contains then the standard clauses for the atomic and Boolean cases plus the following clauses for modalities:

$$\mathcal{M}, (m, h) \models_v t:A \Leftrightarrow A \in \mathcal{E}(m, t^v, v) \wedge (\forall m' \in Tree)(mR_e m' \wedge h' \in H_{m'} \Rightarrow \mathcal{M}, (m', h') \models_v A)$$

$$\mathcal{M}, (m, h) \models_v Proven(A) \Leftrightarrow (\exists r \in \mathcal{D}(m))(\forall h' \in H_m)(r \in Act(m, h') \wedge \mathcal{M}, (m, h') \models_{v \frac{x}{r}} x:A)$$

$$\mathcal{M}, (m, h) \models_v Proven(t, A) \Leftrightarrow (\forall h' \in H_m)(t^v \in Act(m, h') \wedge \mathcal{M}, (m, h') \models_v t:A).$$

$$\begin{aligned} \mathcal{M}, (m, h) \models_v Prove(j, A) \Leftrightarrow & (\forall h' \in Choice_j^m(h))(\exists r \in Act(m, h'))(\mathcal{M}, (m, h') \models_{v \frac{x}{r}} x:A) \wedge \\ & \wedge (\forall r' \in \mathcal{D}(m))(\exists h'' \in H_m)(\mathcal{M}, (m, h'') \models_{v \frac{x}{r'}} x:A \Rightarrow r' \notin Act(m, h'')). \end{aligned}$$

$$\mathcal{M}, (m, h) \models_v \text{Prove}(j, t, A) \Leftrightarrow (\forall h' \in \text{Choice}_j^m(h))(t^v \in \text{Act}(m, h') \wedge \mathcal{M}, (m, h') \models_v t:A) \wedge \\ \wedge (\exists h'' \in H_m)(t^v \notin \text{Act}(m, h'')).$$

$$\mathcal{M}, (m, h) \models_v Et \Leftrightarrow t^v \in \text{Act}(m, h);$$

$$\mathcal{M}, (m, h) \models_v EA \Leftrightarrow (\exists r \in \text{Act}(m, h))(\mathcal{M}, (m, h) \models_{v \frac{x}{r}} x:A).$$

In the above clauses,  $x$  is assumed to be the first (in some alphabetical order) proof variable not occurring in either  $t$  or  $A$ . These semantical definitions are mostly faithful versions of the respective clauses given in the previous sections; however, note the complications arising for implicit modalities. These complications are due to the fact that not all of the admissible or even presented proofs are covered by arbitrary evaluation of proof terms.

With this, the difficulties are still not quite over. A further complication is necessitated by the fact that with Fitting models one cannot simply define a valid formula as a formula valid in all models w.r.t. all moment-history pairs under all possible evaluations, for such a definition would make some intuitively appealing validities of justification logic (and also of justification stit logic with Artemovian semantics) to fail. Consider, e.g., the following formula:

$$x:p \rightarrow (x+y):p.$$

Assume that an evaluation  $v$  assigns to  $x$  a proof of  $p$  which is both valid and admissible at a given moment  $m$ , whereas to  $y$  it assigns a proof outside  $\mathcal{D}(m)$  in such a way that the proof denoted by  $x+y$  also ends up being outside  $\mathcal{D}(m)$ . By the first condition imposed on function  $\mathcal{E}$  in Fitting models, this means that  $x+y$  is not an admissible proof for anything, including  $p$ , of course. Therefore, the above formula fails, even though it is actually an instance of an axiom of justification logic as given in [1]. In order to avoid such counter-examples one would have to take a little detour and first define an evaluation  $v$  to be meaningful for  $A$  in  $m \in \text{Tree}$  iff  $v$  assigns to every proof variable occurring in  $A$  an element from  $\mathcal{D}(m)$ . After that, one can define a valid formula of Fitting justification stit logic as a formula which is true in every model at every moment-history pair  $(m, h)$  for every evaluation of proof variables which is meaningful for  $A$  in  $m$ .

It was shown in [4] that the justificational fragment of Fitting's justification logic is simply justification logic without  $K$ . Moreover, all of the claims about validities and invalidities given in the first part of this paper for basic justification stit logic, and also Theorem 1 given above, can be shown to hold for Fitting justification stit logic as well. On the basis of this coincidence, our hypothesis would be that the class of justification stit formulas valid over Fitting models is the same as the class of formulas valid over Artemovian models, provided that these formulas do not contain  $K$ . However, if one

adds valid inferences to the picture, then this coincidence looks much more problematic. It is easy to see that under Artemovian semantics justification stit logic lacks compactness, since the set

$$\{Proven(p)\} \cup \{\neg t:p \mid t \in Pol\}$$

is unsatisfiable, even though every finite subset of it can be satisfied. Therefore, if Artemovian semantics is assumed, we cannot have a finitary axiomatic system for basic justification logic with the property of strong completeness. However, no such simple proof of compactness failure is available for Fitting semantics, since one can easily define  $v \in Eval(\mathcal{M})$  in such a way that all the terms in  $Pol$  at a given moment such that the set  $\{t^v \mid t \in Pol\}$  is a proper subset of  $\mathcal{D}(m)$ , and so some valid and admissible proof of  $p$  may end up being outside  $\{t^v \mid t \in Pol\}$ . Therefore, justification stit logic with Fitting semantics may actually turn out to be compact and to admit of a finitary axiomatic system with the property of strong completeness in the end.

One of the conceptually interesting aspects of Fitting justification stit models is, in our opinion, the interplay between the bigger domain of proofs  $\mathcal{D}(m)$  associated with a given moment  $m$  and the ‘fan’ of its subsets  $Act(m, h)$  associated with moment-history pairs connected with  $m$ . How should one interpret this new type of moment-localized domains? Apparently, different interpretations are possible here. Our favored answer is that  $\mathcal{D}(m)$  is a domain of all proofs, which are objectively possible at  $m$  in the sense that, were any of these proofs presented to the community at any  $(m, h)$  for  $h \in H_m$ , it would be successfully recognized as some proof. The community’s abilities to recognize proofs vary from situation to situation depending on the actual conventions as to what counts as a proof. These conventions may be affected when the community revises its definition of what a proof is, or — indirectly — when it introduces new derived forms of inference, or defines new notions which may thereafter occur in proofs and thus create new particular instances of old proof forms. Within the class of Artemovian justification stit models, the set  $Pol$  of proof polynomials constituted a sort of Platonic universe of ideas for the community, where all the possible proofs were written in advance, once and for all. The role of agents in these conditions was reduced to bringing to light one or another corner of this universe thus drawing the community’s attention to this or that item in this universe. In Fitting models, the community is rather presented as forging its own system of notions while proving new facts about these notions. Doxastic actions of the agents in Fitting semantic may be necessary or sufficient for acquiring some new members in the domain of objectively admissible proofs, and vice versa. This captures — even though, admittedly, very roughly — real phenomena arising in development of mathematical sciences, which one tends to describe

as ‘this or that defined notion played a crucial role in the proof of this or that theorem’, or ‘this group of results led to the emergence of that theory’.

## 6 Refinement by extending the set of proof polynomials

### 6.1 $\lambda$ -abstraction and proofs from assumptions

Another way of refining justification stit logic consists of modifying or extending the set of proof polynomials and, as a result, modifying or extending the underlying logic of proofs. Although in Artemov’s logic of proofs a distinction is drawn between proof constants and proof variables, perhaps surprisingly the language of proof polynomials does not comprise any variable binding operators. Once proof variables are indeed treated as variables proper, as in Fitting’s semantics that uses assignments of proofs to proof variables, introducing variable binders into the language of proof polynomials seems to be even more natural. An obvious variable-binding operator to be introduced is the  $\lambda$ -abstractor. In Backus-Naur form, the syntax of proof polynomials is then defined as follows:

$$t := x \mid c \mid (t + t) \mid (t \cdot t) \mid !t \mid \lambda x.t$$

where  $x \in PVar$  and  $c \in PConst$ .

The set  $fv(t)$  of free variables of a proof polynomial  $t$  can be defined inductively in the following way:  $fv(x) = \{x\}$ ,  $fv(!t) = \emptyset$ ,  $fv((t\#s)) = fv(t) \cup fv(s)$  for  $\# \in \{+, \cdot\}$ , and  $fv((\lambda x.t)) = fv(t) \setminus \{x\}$ . If  $x \in PVar$  and  $t \in Pol$ , then we denote by  $s[x := t]$  the result of substituting  $t$  for the free occurrences of  $x$  in  $s$ .

The use of proof variables allows one to encode proofs from assumptions, as is well-known from the Curry-Howard encoding of proofs in intuitionistic logic by means of typed  $\lambda$ -terms, see, for instance, [6]. The idea is to read  $x:A$  not as “ $x$  is a proof of  $A$ ” but as “ $x$  is a proof of  $A$  from the assumption  $A$ ”. If  $fv(t) = \{x_1, \dots, x_n\}$ , then for all  $A_1, \dots, A_n$  for which  $x_1:A_1, \dots, x_n:A_n$  hold,  $t:A$  can be understood as saying that  $t$  is a proof of  $A$  from the assumptions  $A_1, \dots, A_n$ , and the polynomial (or proof term)  $\lambda x_1. \dots \lambda x_n. t$  emerges as a proof of  $A_1 \rightarrow (A_2 \rightarrow (\dots (A_n \rightarrow A) \dots))$  from  $\emptyset$ . In general, a formula  $t:A$  may be read as “ $t$  is a proof of  $A$  depending on the assumptions of which  $t$ ’s free variables are proofs”.<sup>6</sup> It is well-known that typed  $\lambda$ -terms can be interpreted

<sup>6</sup>Note that whilst in Artemovian semantics it is not assumed that each proof polynomial is a proof of exactly one formula, in simply typed  $\lambda$ -calculus with explicit types, every term has a unique type of which it is a proof.

as functionals of finite type in so-called type structures over infinite sets, see [5]. Semantically, if  $\lambda x.t$  is a proof of  $A \rightarrow B$  it may be seen as denoting the set of all functions from a domain  $D^A$  of proofs of  $A$  to a domain  $D^B$  of proofs of  $B$ . The above reinterpretation, of course, suggests a justification logic very different from justification logic LP initially proposed by Artemov (see [1, Section 3] for a Hilbert-style axiomatization). In fact, it is not even an extension of the Artemovian justification logic. Most importantly, the factivity of justification is now lost, since  $x:A \rightarrow A$  is no longer valid: the availability of a proof of  $A$  from premise  $A$  does not, by any means, show that  $A$  must be true. Other axioms of LP are still valid under this interpretation, although sometimes their meaning becomes more nuanced. For example, summing proofs under this reading not only involves accumulation of their conclusions but also accumulation of their premises. As a result,  $s + t$  not only proves everything that is proved by either  $s$  or  $t$  but, in general, depends on more premises than either  $s$  or  $t$ . Therefore, even though the following axiom scheme of LP is still valid, under the proposed reading it starts to look, among other things, like a form of monotonicity principle:

$$s:A \rightarrow (s + t):A.$$

The satisfaction clause for formulas  $t:A$  in Artemovian semantics can be adjusted to the new reading as follows:

$$\begin{aligned} \mathcal{M}, (m, h) \models t:A &\Leftrightarrow A \in \mathcal{E}(m, t) \wedge \forall m' \forall h' (((R_e(m, m') \wedge h' \in H_{m'}) \wedge \\ &\wedge (\forall x \in fv(t)) (\forall B \in \mathcal{E}(m', x)) (\mathcal{M}, (m', h') \models B) \Rightarrow \mathcal{M}, (m', h') \models A). \end{aligned}$$

Note that in the special case where  $t = x \in PVar$  this clause boils down to:

$$\mathcal{M}, (m, h) \models x:A \Leftrightarrow A \in \mathcal{E}(m, x),$$

since the proof of  $A$  from  $A$  represented by  $x$  is bound to be valid in all states. Also, note that it follows that the general clause for  $t:A$  given above, can be equivalently re-written as follows:

$$\begin{aligned} \mathcal{M}, (m, h) \models t:A &\Leftrightarrow A \in \mathcal{E}(m, t) \wedge \forall m' \forall h' (((R_e(m, m') \wedge h' \in H_{m'}) \wedge \\ &\wedge (\forall x \in fv(t)) \forall B (\mathcal{M}, (m', h') \models B \wedge x:B) \Rightarrow \mathcal{M}, (m', h') \models A). \end{aligned}$$

Note, however, that if one only allows closed proof polynomials, then all the axioms of Artemovian justification logic are still preserved, including factivity.

The above satisfaction clause for  $t:A$  also shows why we need  $fv(!t)$  to be empty. Indeed, were we to set  $fv(!t) = fv(t)$ , this would mean that the result

of checking the correctness of  $t$  would depend on the truth of all the premises represented by the free variables from  $t$ . For instance, suppose  $x$  is a proof of  $A$  from the premise  $A$ . We have then  $x:A$ , from which we get that  $!x:x:A$  by the reflection axiom. Under our current reading,  $!x$  must be then a proof of  $x:A$  from zero assumptions, since we have  $fv(!x) = \emptyset$ . However, were we to have  $fv(!x) = \{x\}$ , this would mean that  $!x$  shows that  $x$  correctly infers  $A$  from  $A$  only when we assume  $A$ , which does not seem natural.

It is clear that in view of adding the  $\lambda$ -abstractor, the underlying logic of proofs calls for axiomatic extension. In addition to the functional application axiom that captures implication elimination, we need an axiom scheme that takes care of implication introduction and the cancellation of assumptions:

$$x:A \rightarrow (t:B \rightarrow \lambda x.t:(A \rightarrow B)).$$

The concept of proof normalization invites adding an axiom scheme that captures  $\beta$ -reduction:

$$s:A \rightarrow (x:A \rightarrow ((\lambda x.t \cdot s):B \rightarrow t[x := s]:B)).$$

and another one that takes care of  $\eta$ -reduction for terms  $t$  with  $x \notin fv(t)$ :

$$x:A \rightarrow (\lambda x.(t \cdot x):(A \rightarrow B) \rightarrow t:(A \rightarrow B)).$$

Moreover,  $\alpha$ -reduction can be taken into account by considering all implications  $t:A \rightarrow s:A$  as axiomatic if  $t$  and  $s$  are obtainable from each other by the clash-free renaming of bound variables. In order to integrate the expanded underlying logic of proofs into justification stit logic with Artemovian semantics, we also need to consider an interaction of  $\lambda$ -abstracted proof polynomials with the  $\mathcal{E}$ -function in justification stit models and postulate:

$$(A \in \mathcal{E}(m, x) \wedge B \in \mathcal{E}(m, t)) \Rightarrow (A \rightarrow B) \in \mathcal{E}(m, \lambda x.t),$$

$$(A \in \mathcal{E}(m, s) \wedge (A \in \mathcal{E}(m, x) \wedge B \in \mathcal{E}(m, \lambda x.t \cdot s))) \Rightarrow B \in \mathcal{E}(m, t[x := s]),$$

$$(A \in \mathcal{E}(m, x) \wedge (A \rightarrow B) \in \mathcal{E}(m, \lambda x.(t \cdot x))) \Rightarrow (A \rightarrow B) \in \mathcal{E}(m, t), \text{ for } x \notin fv(t).$$

We now may introduce generalized  $\pi$ -modalities for making statements about proving a sentence  $A$  from a set of assumptions  $\Gamma$ :

Notation	Informal interpretation
$Prove(j, A, \Gamma)$	Agent $j$ proves $A$ from $\Gamma$
$Prove(j, t, A, \Gamma)$	Agent $j$ proves $A$ by $t$ from $\Gamma$
$Proven(A, \Gamma)$	$A$ has been proven from $\Gamma$
$Proven(t, A, \Gamma)$	$A$ has been proven by $t$ from $\Gamma$

Such statements about proving from assumptions can be interpreted by adding a suitable condition to the above satisfaction clauses for  $\pi$ -modalities in Artemovian semantics:

$$\begin{aligned}
\mathcal{M}, (m, h) \models \text{Prove}(j, A, \Gamma) &\Leftrightarrow (\forall h' \in \text{Choice}_j^m(h))(\exists t \in \text{Act}(m, h'))(\mathcal{M}, (m, h) \models t:A) \wedge \\
&\quad \wedge (\forall s \in \text{Pol})(\exists h'' \in H_m)(\mathcal{M}, (m, h) \models s:A \Rightarrow s \notin \text{Act}(m, h'')) \wedge \\
&\quad \wedge (\forall x \in \text{fv}(t))(\exists B \in \Gamma) \mathcal{M}, (m, h) \models x:B; \\
\mathcal{M}, (m, h) \models \text{Prove}(j, t, A, \Gamma) &\Leftrightarrow \mathcal{M}, (m, h) \models \text{Prove}(j, t, A) \wedge \\
&\quad \wedge (\forall x \in \text{fv}(t))(\exists B \in \Gamma) \mathcal{M}, (m, h) \models x:B; \\
\mathcal{M}, (m, h) \models \text{Proven}(A, \Gamma) &\Leftrightarrow (\exists t \in \text{Pol})(\forall h' \in H_m)(t \in \text{Act}(m, h') \wedge \mathcal{M}, (m, h) \models t:A) \wedge \\
&\quad \wedge (\forall x \in \text{fv}(t))(\exists B \in \Gamma) \mathcal{M}, (m, h) \models x:B; \\
\mathcal{M}, (m, h) \models \text{Proven}(t, A, \Gamma) &\Leftrightarrow \mathcal{M}, (m, h) \models \text{Proven}(t, A) \wedge \\
&\quad \wedge (\forall x \in \text{fv}(t))(\exists B \in \Gamma) \mathcal{M}, (m, h) \models x:B.
\end{aligned}$$

The additional clause requires that for some proof  $t$  of  $A$  from  $\Gamma$ , respectively the proof  $t$  of  $A$  from  $\Gamma$  under consideration, every free variable in  $t$  proves a premise from  $\Gamma$ . For finite  $\Gamma$  one could require that, conversely, for any  $A$  in  $\Gamma$ , there exists an  $x \in \text{fv}(t)$  such that  $x$  is a proof of  $A$ . Note that the extended set of proof polynomials does not come with a commitment to intuitionistically acceptable proofs only. One could, for instance, assume proof constants  $c_{AV-A}$ , for any formula  $A$ .

Clearly, in light of the typed  $\lambda$ -calculus, the set of Artemovian proof polynomials can also be modified and extended in other respects. The sum operator  $+$  can be replaced by operators denoting pairing and projection functions. Moreover, the term-forming operations corresponding to disjunction introduction and elimination in natural deduction can be added. In this way, justification stit logic would allow one to talk about acts of proving in a way that more specifically records certain natural deduction inferential steps taken by the agents in order to present proofs.

## 6.2 Another modification of *Pol*

Another extension of the set of proof polynomials is implemented in Fitting's [4] quantified logic of evidence, *QLP*. First of all, Fitting adds universal quantification over proof variables to Artemov's logic of proofs. Syntactically, the quantifier  $\forall$  is thus not a term-forming device but combines a proof variable  $x$  and a formula  $A$  into a more complex formula  $\forall xA$ . The existential quantifier is defined as usual. Moreover, the set *PConst* of proof constants is replaced by a denumerably infinite set of primitive function symbols of finite arity. Proof constants are still present as function symbols of arity 0.

If  $f$  is an  $n$ -place primitive function symbol and  $x_1, \dots, x_n$  are proof variables, then  $f(x_1, \dots, x_n)$  is said to be a primitive proof term. Then a new variable-binding operation on proof terms is introduced: if  $t$  is a proof term and  $x$  is a proof variable, then  $(t \forall x)$  is a proof term as well. As Fitting [4, p. 69] explains (notation adjusted), the “intention is that  $(t \forall x)$  should serve as a justification of  $\forall xA$  if  $t$  serves as a uniform justification of each instance of  $A$ ”, i.e., each instance of substituting all free occurrences of  $x$  in  $A$  with some constant. The extension of *Pol* by terms  $(t \forall x)$  thus goes hand in hand with the addition of the universal quantifier  $\forall$  and is not an extension motivated independently of adding  $\forall$ .<sup>7</sup>

If we want to reason about proving as an activity that may result in

---

<sup>7</sup>Proof-theoretically, the meaning of  $\forall$  and terms  $(t \forall x)$  is given by two standard axiom schemes for universal quantification, one axiom scheme involving both quantifiers and  $(t \forall x)$ , and the so-called “Justified Universal Quantification Rule”, in Fitting’s numbering:

$$(6) \quad \forall xA(x) \rightarrow A(t), \text{ for any proof term } t \text{ that is free for } x \text{ in } A(x),$$

$$(7) \quad \forall x(A \rightarrow B(x)) \rightarrow (A \rightarrow \forall xB(x)), \text{ where } x \text{ does not occur free in } A,$$

$$(8) \quad \exists y: \forall x t: A \rightarrow (t \forall x): \forall xA \text{ if } y \text{ does not occur free in } A \text{ or } t,$$

$$\frac{t:A(x)}{(t \forall x): \forall xA(x)}.$$

Axiom (8) says that  $t$  is a uniform justification of  $\forall xA$  if there is a proof  $y$  showing that  $t$  is such a uniform recipe. (Fitting shows that the usual universal generalization rule and a quantifier version of the necessitation rule, which are both  $(t \forall x)$ -free, can replace the justified universal generalization rule.)

$\lambda$ -abstraction is then used in the meta-language in order to evaluate terms  $(t \forall x)$ . The term-forming  $\forall$  is interpreted as a mapping from the function space of  $\overline{\mathcal{D}}$  to  $\overline{\mathcal{D}}$ , i.e.,  $\forall^I: \overline{\mathcal{D}}^{\overline{\mathcal{D}}} \rightarrow \overline{\mathcal{D}}$ . For every variable  $x$ , a function  $\langle \lambda x.t \rangle^v$  on  $\overline{\mathcal{D}}$  is defined as follows:  $\langle \lambda x.t \rangle^v(r) = t^w$ , where  $w = v^{\frac{x}{r}}$ . The value  $(t \forall x)^v$  of  $(t \forall x)$  under evaluation function  $v$  is then defined using  $\langle \lambda x.t \rangle^v$ :

$$(t \forall x)^v = \forall^I(\langle \lambda x.t \rangle^v).$$

In the presence of primitive function symbols, the sets  $\mathcal{D}(m)$  must now be closed under  $f^I$  for every primitive function symbol  $f$ . Moreover, if  $t$  is a proof term,  $v$  an evaluation and  $v(y) \in \mathcal{D}(m)$  for every variable  $y$  occurring in  $t$  with  $x \neq y$ , then  $(t \forall x)^v$  is required to be in  $\mathcal{D}(m)$  as well.

As to an interplay between the evidence function  $\mathcal{E}$  and the value of terms  $(t \forall x)$ , Fitting stipulates:

$$(mR_e m' \wedge r \in \mathcal{D}(m') \wedge A \in \mathcal{E}(m', \langle \lambda x.t \rangle^v, v^{\frac{x}{r}})) \Rightarrow \forall xA \in \mathcal{E}(m, (t \forall x)^v, v).$$

The evaluation clause for formulas  $\forall xA$  in Fitting justification stit models would then be as one may expect:

$$\mathcal{M}, (m, h) \models_v \forall xA \Leftrightarrow \mathcal{M}, (m, h) \models_w A \text{ for all } r \in \mathcal{D}(m) \text{ and } w = v^{\frac{x}{r}}.$$

uniform proofs of each instance of a formula  $A$  from possibly non-empty sets of assumptions, we may think of combining  $\lambda$ -abstraction over proof variables with universal quantification over such variables.

## 7 Conclusion

We considered various ways of refining the justification stit logic from [10]. In summary, the tradeoff involved in switching from the Artemovian semantics to Fitting's semantics can be described as follows. On the positive side, we get a more nuanced picture of the activity of proving within the community of agents; we also end up in a better position for a possible introduction of quantifiers over proofs and thus for a better employment of the full potential of  $E$ -notions. Moreover, there is a reasonable hope that in this way we will get a compact logic with strongly complete axiomatization. On the negative side we probably lose some expressive power by giving up the  $K$  modality, and our basic semantic definitions begin to look rather formidable and less transparent. Also, Fitting-style semantics is much less visible in the existing literature on justification logic, which might well be a sign that with it the balance of complexity versus expressivity is already more on the complexity side. Fitting's semantics is a framework in which the use of proof variables is taken seriously insofar as assignments to proof variables are introduced and, moreover, variable binding quantifiers are introduced in  $QLP$ . It seems that up to now introducing  $\lambda$ -abstraction into the language of proof polynomials has not been considered in the literature on the logic of proofs. This extension is one way of representing reasoning about proving from assumptions.

We hope that this second part of the paper succeeds in putting the basic justification logic described in its first part, [10], into a suitable context as to exploring the manifold and rich variety of its related systems, and plausibly conveys the point that it is much more to the combinations of justification modalities and stit modalities than just one isolated logical system. Thus far we were mainly focusing on elementary observations concerning interdefinability between  $\pi$ -modalities and the validity and non-validity of particularly interesting formulas of justification stit logic. But this is just a beginning, and formal properties of justification stit logic and its possible refinements remain to be investigated. The most obvious task is finding axiomatizations, both for the basic logic and for the related systems, most notably the variants of the logic of  $E$ -notions. A natural first step in this direction will be publishing the completeness proof for the axiomatization of the explicit fragment of the logic of  $E$ -notions presented in Section 4. Another task is working out a

combination of  $\lambda$ -abstraction and quantification over proof polynomials.

Acknowledgements We would like to thank an anonymous reviewer for her/his comments. Also, we would like to acknowledge financial support from the Deutsche Forschungsgemeinschaft, DFG, project WA 936/11-1.

## References

- [1] Artemov, S. and Nogina E., ‘Introducing Justification into Epistemic Logic’, *Journal of Logic and Computation* 15 (2005), 1059–1073.
- [2] Belnap, N.D., M. Perloff and M. Xu, *Facing the Future: Agents and Choices in our Indeterminist World*. Oxford University Press, New York, 2001.
- [3] Fagin, R. and Halpern, Y., ‘Belief, Awareness, and Limited Reasoning’, *Artificial Intelligence* 34 (1988), 39–76.
- [4] Fitting, M., ‘A quantified logic of evidence’, *Annals of Pure and Applied Logic* 152 (2008), 67–83.
- [5] Friedman, H., ‘Equality between functionals’, in: R. Parikh (ed.), *Proceedings Logic Colloquium ’73*, Springer Lecture Notes in Mathematics 453, Springer, Berlin, 1975 22–37.
- [6] Girard, J.-Y., Taylor, P. and Lafont, Y., *Proofs and Types*, Cambridge University Press, 1989.
- [7] Horty, J., ‘An alternative stit operator’, unpublished manuscript, Philosophy Department, University of Maryland, 1989.
- [8] von Kutschera, F., ‘Bewirken’, *Erkenntnis* 24 (1986), 253–281.
- [9] Olkhovikov G.K. and Wansing, H., ‘An axiom system and a tableau calculus for STIT imagination logic’, *Journal of Philosophical Logic*, published online January 28, 2017.
- [10] Olkhovikov G.K. and Wansing, H., ‘Inference as doxastic agency. Part I: The basics of justification stit logic’, 2016, to appear in: *Studia Logica*.
- [11] Wansing, H., ‘Remarks on the logic of imagination. A step towards understanding doxastic control through imagination’, *Synthese* 194 (2017), 2843–2861, published online October 2015.