Model refinement and model checking for S5^n

Alessio Lomuscio and Mark Ryan
School of Computer Science
University of Birmingham
Birmingham B15 2TT, UK
`www.cs.bham.ac.uk/{~arl,~mdr}`

8 August 1998

Abstract

Halpern and Vardi have proposed the notion of refinement of S5^n Kripke models in order to solve multi-agent problems in which knowledge evolves. We argue that there are some problems with their proposal and attempt to solve them by moving from Kripke models to their corresponding *trees*. We define refinement of a tree with a formula, show some properties of the notion, and illustrate with the muddy children puzzle.

1 Introduction

The modal logic S5^n (see for example [HC96]), also known as KT45^n, has been used to model knowledge in multi-agent systems (MAS) for some years now [Hin62, FHMV95]. S5^n is a classical modal logic containing n modalities expressing private knowledge, and operators for expressing common knowledge and distributed knowledge within a group.

The standard (*consequence relation*) approach to using S5^n is to describe a situation as a set of formulas Γ, and to attempt to show that the situation satisfies a property ϕ by establishing $\Gamma \vdash \phi$ or $\Gamma \vDash \phi$. Establishing $\Gamma \vdash \phi$ involves finding a proof of ϕ from Γ, while establishing $\Gamma \vDash \phi$ involves reasoning about all (usually infinitely many) Kripke models satisfying Γ to show that they also satisfy ϕ. The completeness of S5^n shows that these two notions are equivalent. However, experience has shown that this approach is computationally very expensive.

In order to overcome the intractability of this approach, Halpern and Vardi have proposed to use *model checking* as an alternative to theorem proving [HV91]. In the model checking approach, the situation to be modelled is codified as a single Kripke model M rather than as a set of formulas Γ. The task of verifying that a property ϕ holds boils down to checking that M satisfies ϕ, written $M \vDash \phi$. This task is computationally much easier than the theorem proving task, being linear in the size of M and the size of ϕ [HV91].

Halpern and Vardi informally illustrate their approach by modelling the muddy children puzzle. In that puzzle, there are n children and n atomic propositions $p_1, p_2, \ldots, p_n$ representing whether each of the children have mud on their faces or not. Various announcements are made, first by the father of the children and then by the children themselves. The children thus acquire information about what other children know, and after some time the muddy ones among them are able to conclude that they are indeed muddy. We describe the problem in greater detail below.

Halpern and Vardi propose the following way of arriving at the model M to be checked. They start with the most general model for the set of atomic propositions at hand. In order to deal with the announcements made, they successively *refine* the model with formulas expressing the announcements made. This refinement process consists of removing some links from the Kripke model. At any time during this process, they can check whether child i knows p_i (for example), by checking whether the current model satisfies $\Box_i p_i$.

This method is illustrated in the paper [HV91] and the book [FHMV95], but a precise definition of the refinement operation is not given. Our original aim for this paper was to provide such a definition and explore its properties. However, we soon came to the opinion that there is no definition of model refinement on arbitrary S5^n Kripke structures that will have intuitively acceptable properties. We explain our reasons for this view in section 2. We believe the refinement and model checking ideas can still be made to work, however. In section 3 we introduce a structure derived from a Kripke model, which we call a *Kripke tree*, and define the refinement operation on Kripke trees. We illustrate this notion using the muddy children example in section 4. In section 5 we state and prove some properties of the refinement operation on Kripke trees, and conclude in section 6.

1.1 Syntax and semantics

We assume finite sets P of *propositional atoms*, and A of *agents*. Formulas are given by the usual grammar:

$$\phi \ ::= \ p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \Box_i\phi \mid C\phi$$

where $p \in P$ and $i \in A$. Intuitively the formula $\Box_i\phi$ represents the situation in which the agent i knows the fact represented by the formula ϕ. The other propositional connectives can be defined in the usual way. The modal connectives $\Diamond_i$, E and B are defined as:

$$\begin{array}{rll} \Diamond_i\phi & \text{means} & \neg\Box_i\neg\phi \\ E\phi & \text{means} & \bigwedge_{i\in A}\Box_i\phi \\ B\phi & \text{means} & \neg C\neg\phi \end{array}$$

$\Diamond_i\phi$ means "it is consistent with i's knowledge that ϕ", $E\phi$ means that everyone knows ϕ, while $C\phi$ is the much stronger statement that ϕ is common knowledge. In a multi-agent setting, a formula ϕ is said to be common knowledge if it is known by all the agents, and moreover that each agent knows that it is known by all the agents; and moreover, each agent knows that fact, and that one, etc. An announcement of ϕ results in common knowledge of ϕ among the hearers, because as well as hearing ϕ they also see that the others have heard it too (we assume throughout that all the agents are perceptive, intelligent, truthful). If one agent secretly informs all the others of ϕ, the result will be that everyone knows ϕ, but ϕ will not be common knowledge. B is the dual of C. Although not particularly useful intuitively, we will need it for technical reasons.

Definition 1.1 A formula is *universal* if it has only the modalities $C, E, \Box_i$ and no negations outside them.

Definition 1.2 An S5^n *Kripke model* $M = (W, \sim, \pi, w)$ of the modal language over atomic propositions P and agents A is given by:

1. A set W, whose elements are called *worlds*;

2. An A-indexed family of relations $\sim = \{\sim_i\}_{i \in A}$. For each $1 \leqslant i \leqslant n$, $\sim_i$ is an equivalence relation on W ($\sim_i \subseteq W \times W$), called the *accessibility relation*;
3. A function $\pi : W \to \mathcal{P}(P)$, called the *assignment function*;
4. A world $w \in W$, the *actual world*.

See figure 1 for an illustration.

Let $x \in W$. We define the relation of satisfaction of ϕ by M at x, written $M \models_x \phi$, in the usual way:

$$\begin{array}{rl}
M \models_x p & \text{iff } p \in \pi(x) \\
M \models_x \neg\phi & \text{iff } M \not\models_x \phi \\
M \models_x \phi \wedge \psi & \text{iff } M \models_x \phi \text{ and } M \models_x \psi \\
M \models_x \Box_i \psi & \text{iff for each } y \in W,\ x \sim_i y \text{ implies } M \models_y \psi \\
M \models_x C\psi & \text{iff for each } k \geqslant 0 \text{ and } i_1, i_2, \ldots, i_k \in A, \text{ we have } M \models_x \Box_{i_1} \ldots \Box i_k \psi
\end{array}$$

We say that y is *reachable in k steps* from x if there are $w_1, w_2, \ldots w_{k-1} \in W$ and $i_1, i_2, \ldots i_k$ in A such that $x \sim_{i_1} w_1 \sim_{i_2} w_2 \ldots \sim_{i_{k-1}} w_{k-1} \sim_{i_k} y$. We also say that y is *reachable* from x if there is some k such that it is reachable in k steps. The following fact is useful for understanding the technical difference between E and C.

Theorem 1.3 ([FHMV95])

1. $M \models_x E^k\phi$ iff for all y that are reachable from x in k steps, we have $M \models_y \phi$.
2. $M \models_x C\phi$ iff for all y that are reachable from x, we have $M \models_y \phi$.

2 Refining Kripke models

Halpern and Vardi propose to refine Kripke models in order to model the evolution of knowledge. They illustrate their method with the muddy children puzzle.

2.1 The muddy children puzzle

There is a large group of children playing in the garden. A certain number (say k) get mud on their foreheads. Each child can see the mud (if present) on others but not on his own forehead. If $k > 1$ then each child can see another with mud on its forehead, so each one knows that at least one in the group is muddy. The father first announces that at least one of them is muddy [which, if $k > 1$, is something they know already]; and then he repeatedly asks them 'Does any of you know whether you have mud on your own foreheadΓ' The first time they all answer 'no'. Indeed, they go on answering 'no' to the first $k - 1$ questions; but at the kth those with muddy foreheads are able to answer 'yes'.

At first sight, it seems rather puzzling that the children are eventually able to answer the father's question positively. The clue to understanding what goes on lies in the notion of common knowledge. Although everyone knows the content of the father's initial announcement, the father's saying it makes it common knowledge among them, so now they all know that everyone else knows it, etc. Consider a few cases of k.

$k = 1$, i.e. just one child has mud. That child is immediately able to answer 'yes', since she has heard the father and doesn't see any other child with mud.

$k = 2$, say a and b have mud. Everyone answers 'no' the first time. Now a thinks: since b answered 'no' the first time, he must see someone with mud. Well, the only person I can see with mud is b, so if b can see someone else it must be me. So a answers 'yes' the second time. b reasons symmetrically about a, and also answers 'yes'.

$k = 3$, say a, b, c. Everyone answers 'no' the first two times. But now a thinks: if it was just b and c with mud, they would have answered 'yes' the second time. So there must be a third person with mud; since I can only see b, c having mud, the third person must be me. So a answers 'yes' the third time. For symmetrical reasons, so do b, c.

And similarly for other cases of k.

To see that it was not common knowledge before the father's announcement that one of the children was muddy, consider again $k = 2$, say a, b. Of course a and b both know someone is muddy (they see each other), but, for example, a doesn't know that b knows that someone is dirty. For all a knows, b might be the only dirty one, and therefore not be able to see a dirty child.

2.1.1 The formalisation in [HV91]

Suppose $A = \{1, \ldots n\}$ and $P = \{p_1, \ldots, p_n\}$; p_i means that the ith child has mud on its forehead. Suppose $n = 3$. The assumption of this puzzle is that each child can see the other children but cannot see itself, so each child knows whether the others have mud or not, but does not know about itself. Under these assumptions, Halpern and Vardi propose the Kripke structure of figure 1 to model the initial situation.

Let w be any world in which there are at least two muddy children (i.e. w is one of the four upper worlds). In w, every child knows that at least one of the children has mud. However, it is not the case that it is common knowledge that each child has mud, since the world at the bottom of the lattice is reachable (cf. theorem 1.3).

To model the father's announcement, Halpern and Vardi refine the model M_1 in figure 1, arriving at M_2 in figure 2 (these figures also appear in [HV91, FHMV95]). The refinement process is not precisely defined in [HV91, FHMV95], though arguments in favour of the transformation from M_1 to M_2 are given.

Suppose now that the father asks the children whether they know whether they are muddy or not, and the children answer simultaneously that that they do not. Halpern and Vardi argue that this renders all models in which there is only one muddy child inaccessible, resulting in M_3 (figure 3).

If there are precisely two children with mud (i.e. the actual world is one of the three in the second layer), then each of the muddy children now knows it is muddy. For suppose the actual world is the left one of those three, i.e. w with $\pi(w) = \{p_1, p_2\}$. We easily verify that $M_3 \models_w \Box_1 p_1$ and $M_3 \models_w \Box_2 p_2$.

If all three children are muddy, i.e. the actual world w is the top one, then we are not yet done, for we do not have $M_3 \models_w \Box_i p_i$ for any i. The father again asks each of the children if they know if they are muddy, and the model is refined again according to their answer "no", resulting in M_4 which is M_3 with the last remaining links removed. (M_4 is not illustrated.) We can easily check that $M_4 \models_w \Box_i p_i$ for each i.

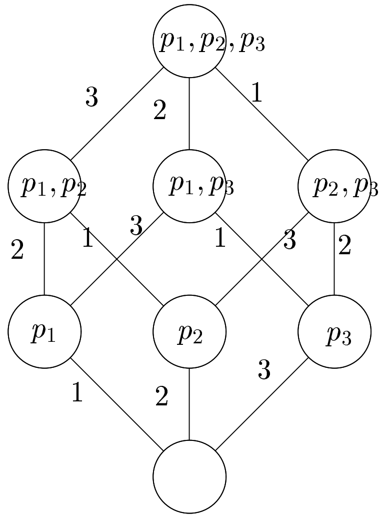

Figure 1: M_1: The Kripke model for the muddy children puzzle with $n = 3$.

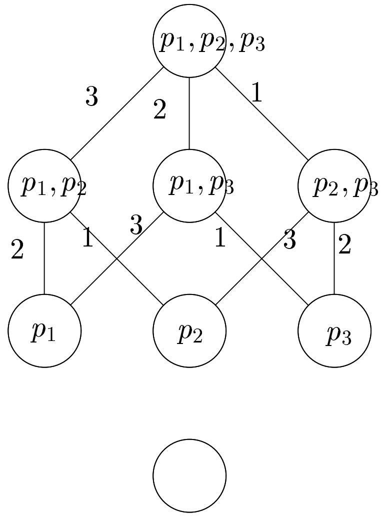

Figure 2: M_2: The Kripke structure after the father speaks.

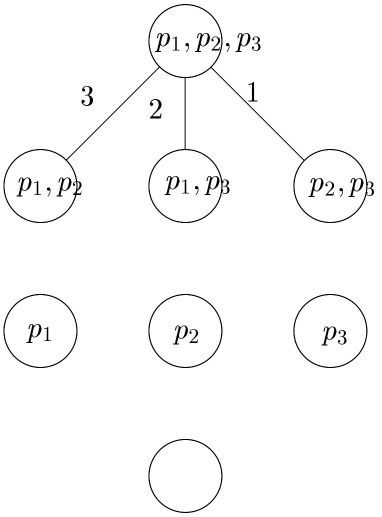

Figure 3: M_3: The Kripke structure after the children announce that they don't know whether they are muddy.

In summary, the method proposed by Halpern and Vardi for solving muddy-children-type puzzles is the following. Start with a suitably general model M_1 reflecting the initial set-up of the puzzle. Refine it successively by the announcements made. At the end of the announcements, check formulas against the refined model. In the example above, we refined M_1 first by $\phi_1 = C(p_1 \vee p_2 \vee p_3)$ (the father's announcement), and then twice by

$$\phi_2 = C(\neg\Box_1 p_1 \wedge \neg\Box_1 \neg p_1) \wedge C(\neg\Box_2 p_2 \wedge \neg\Box_2 \neg p_2) \wedge C(\neg\Box_3 p_3 \wedge \neg\Box_3 \neg p_3)$$

which corresponds to each of the three children announcing that they don't know whether they are muddy or not.

Halpern and Vardi do not precisely define what refinement by a formula means. The intuition they give is that refinement removes a minimal set of links of the model, so that the model satisfies the formula at the actual world. Removing links means that epistemic possibilities are removed, that is, knowledge is gained, so this seems intuitively the right thing to do.

2.2 Problems with refinement of Kripke models

Let us write $M * \phi$ to denote the result of refining the model M by the formula ϕ. Thus, in the example above, $M_2 = M_1 * \phi_1$, etc.

Our original aim was to make precise this notion of refinement of a Kripke model by a formula, and to investigate its properties. We investigated several possible definitions: essentially a refinement procedure will remove the links to the states that are responsible for the non-satisfaction of the formula we are refining with. However, we quickly came upon examples which showed that it will not be easy to achieve all the reasonable properties one could wish for.

Example 2.1 Let M_5 be the Kripke model illustrated in figure 4, with the left-hand world w the actual world, and consider refining by $\Box_1 p$. The definitions we examined differed in

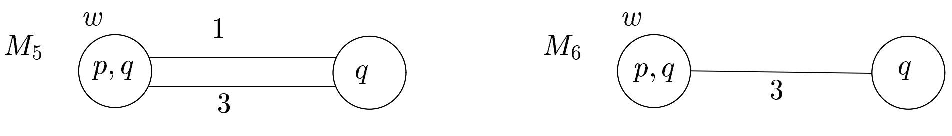

Figure 4: M_5 and M_6 (example 2.1)

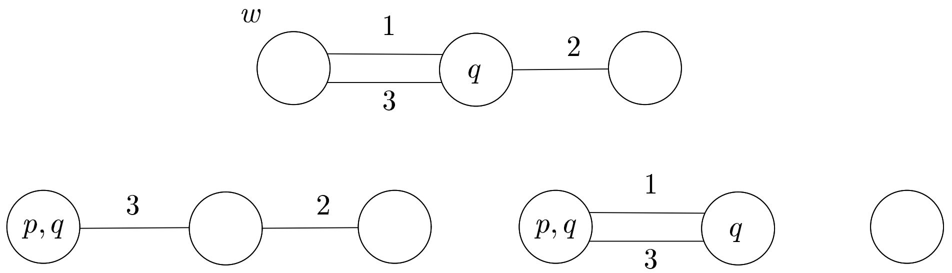

Figure 5: Two outcomes for refinement of the top model by $\Box_1\Box_2(p \vee q)$ (example 2.2)

subtle cases involving quite complex formulas and models, but they all agreed in this one: the resulting model must be M_6 (see figure). What happens is that agent 1 gains the knowledge of p, and so must eliminate the epistemic possibility of $\neg p$ by removing the link.

The counterintuitive property of this example is that $M_5 \models_w \Box_3\Diamond_1 p$, while $M_6 \not\models_w \Box_3\Diamond_1 p$. Thus, in M_5, agent 3 knows that p is consistent with 1's knowledge. But after 1 learns p for sure in M_6, 3 no longer knows this!

Example 2.2 Figure 5 shows a model and (the only) two outcomes one could consider for its refinement by $\Box_1\Box_2(p \vee q)$. One must remove either the 1 link or the 2 link in order to prevent the 1–2 path to the world exhibiting $\neg(p \vee q)$. The choice is which link to remove. Both outcomes reveal undesirable properties of the refinement operator. In the first case, removing the 1 link adds too much to 1's knowledge (he learns p), while the second case gives us a situation in which a model satisfies $\Box_3\Diamond_2\neg q$ but its refinement by $\Box_1\Box_2(p \vee q)$ does not. It is counterintuitive that 3's knowledge should change in this way when we refine by $\Box_1\Box_2(p \vee q)$.

The second case at least has the desirable property that a minimal change of the knowledge of agents at the actual world w is made, since the set of reachable states from w is maximised (cf. theorem 1.3).

Example 2.3 Refinement by universal formulas (definition 1.1) ought to be cumulative, and such formulas ought to commute with each other (i.e. $M * \phi * \psi = M * \psi * \phi$). However, another example shows that this will be hard to achieve. Consider the model M_7 shown at the top of figure 6, and let $\phi = \Box_1 p$ and $\psi = \Box_1\Box_2(p \vee q)$. Whatever way one thinks about defining $*$, the result in the left-hand branch seems clear. Note that $M_7 * \Box_1 p$ already satisfies $\Box_1\Box_2(p \vee q)$ and therefore $M_7 * \Box_1 p * \Box_1\Box_2(p \vee q) = M_7 * \Box_1 p$.

An argument for the stated result of $M_7 * \Box_1\Box_2(p \vee q)$ was given in example 2.2, and

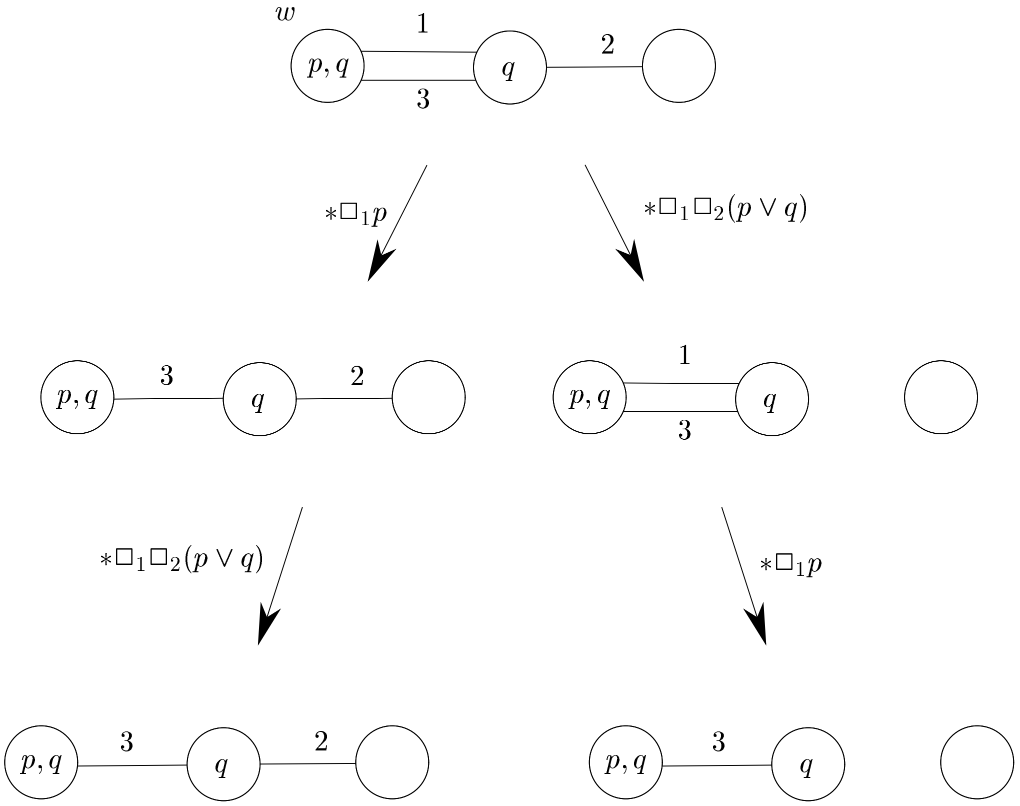

Figure 6: Two evolutions of M_7 (example 2.3), showing that $M * \phi * \psi \neq M * \psi * \phi$

further refining by $\Box_1 p$ leaves little room for maneuver. However, the resulting models differ on whether they satisfy (for example) $\Box_3\Box_2 q$.

Example 2.3 shows that even universal formulas, do not enjoy commutativity in any reasonable refinement setting. However, commutativity for universal formulas seems intuitively correct: the order in which ideal agents acquire information should not matter. Non-universal formulas are a different matter, since they can express absence of knowledge, and this will not commute with the acquisition of new knowledge.

3 Refining Kripke trees

Some of the problems exhibited by the three examples at the end of the preceding section seem to be due to the following fact: when we remove a link in a Kripke model in order to block a certain path, we also block other paths that used that link. To overcome this problem, we would like to unravel Kripke models into trees, in which each link participates in just one path. At first sight this looks like it will destroy the finiteness of our models, a feature on which effective refinement operators and model checking operators rely. To retain finiteness, we will need to limit in advance the maximum nesting of boxes that is allowed, and construct a tree to depth greater than this number. Semantic structures similar to Kripke trees have been defined in [HC84]. Our definition differs in detail from the one in [HC84], but it largely agrees with it in spirit.

In this section we define the notion of Kripke tree, show a translation of equivalence Kripke models into Kripke trees, define and algorithm for refining knowledge structures and prove a few properties about it.

3.1 Kripke trees: basic definitions

Definition 3.1 (Kripke tree) A *Kripke tree* $T = (V, E, \sigma)$ is

- a set V, of *vertices*;
- an A-indexed family E of *edges*: $E_i \subseteq V \times V$, such that the structure (V, E) forms a tree, that is,
  - there is a unique vertex $v_0 \in V$ such that for all $v \in V$ and $i \in A$, $(v, v_0) \notin E_i$. v_0 is called the *root* of T.
  - for every vertex v there is a unique and finite *path* from the root to v, i.e. unique sequences $(v_1, \ldots, v_k)$ and $(i_1, \ldots, i_k)$ such that $(v_i, v_{i+1}) \in E_{i+1}$ $(i \geq 0)$ and $v_k = v$. The *distance* of v from the root is k.
- a function $\sigma : V \to \mathcal{P}(P)$.

We write E^* to mean the transitive closure of the union of relations in E, i.e. $(v, v') \in E^*$ if there is a path from v to v', i.e. sequences $(v_1, \ldots, v_k)$ and $(i_1, \ldots, i_{k-1})$ such that $(v_i, v_{i+1}) \in E_i$, $v_1 = v$ and $v_k = v'$.

We also allow the empty tree $(\varnothing, \varnothing, \varnothing)$ which we write as $-$. It has no root.

Definition 3.2 (Generated Kripke Tree) Let $M = (W, \sim, \pi, w_0)$ be an $\mathrm{S5}^n$ Kripke model. The Kripke tree $T_M = (V, E, \sigma)$ *generated by* M is given as follows:

- The set of *vertices* is the set of paths in M:

$$V = \left\{ (w_0, i_1, w_1, \ldots, w_{k-1}, i_k, w_k) \;\middle|\; \forall j.\, w_j \in W,\, i_j \in A,\, w_j \sim_{j+1} w_{j+1} \right\}$$

- E is an A-indexed family of sets of edges. For $s, s' \in V$, there is an i-edge between s, s', written $(s, s') \in E_i$, iff s' equals s extended by an i link, i.e. $s = (w_0, i_1, w_1, \ldots, w_k)$, $s' = (w_0, i_1, \ldots, w_k, i, w)$ for some w.
- The *valuation* σ is defined by $\sigma((w_0, i_1, w_1, \ldots, w_k)) = \pi(w_k)$.

The vertex $w_0 \in V$ is the root of the tree. When the model M is clear from the context or not relevant we will simply indicate the tree as T.

Kripke trees are irreflexive, intransitive, anti-symmetric, anti-convergent and serial.

If M has at least two distinct worlds related by some $\sim_i$, then T_M is infinite. For our purposes of model refinement, we usually want to deal with finite trees. T^k_M is T_M with paths truncated at length k. Obviously by truncating the tree we will lose seriality.

Definition 3.3 (Truncated tree of depth k) Given a tree $T = (V, E, \sigma)$, the truncated tree of depth k is defined as $T^k = (V', E', \sigma')$, where

- $V' = \{v \in V \mid \text{the distance of } v \text{ from the root } \leq k\}$.
- $E' = E|_{V'}$ is the restriction of E to V',
- $\sigma' = \sigma|_{V'}$ is the restriction of σ to V'.

Infinite and finite trees satisfy modal formulas in the expected way:

Definition 3.4 (Interpretation) Let ϕ be an S5^n formula, and T a tree. The satisfaction of ϕ by T at vertex v, written $T \models_v \phi$, is inductively defined as follows:

- $T \models_v p$ if $p \in \sigma(v)$;
- $T \models_v \neg\phi$ if not $T \models_v \phi$;
- $T \models_v \phi \wedge \psi$ if $T \models_v \phi$ and $T \models_v \psi$;
- $T \models_v \Box_i \phi$ if $\forall v' \in V$, $(v, v') \in E_i$ implies $T \models_{v'} \phi$;
- $T \models_v C\phi$ if $\forall v' \in V$, $(v, v') \in E^*$ implies $T \models_{v'} \phi$.

The tree T satisfies ϕ, written $T \models \phi$, if it satisfies ϕ at its root. The empty tree $-$ satisfies no formula.

An infinite tree T_M is semantically equivalent to its generating model M as the following shows:

Lemma 3.5 Let $M = (W, \sim, \pi, w_0)$ be an equivalence Kripke model and $T_M = (V, E, \sigma)$ its associated Kripke tree. Let $v = (w_0, i_1, w_1, \ldots, w)$ be any vertex ending in w, and ϕ any formula. Then:

$$M \models_w \phi \quad \text{iff} \quad T_M \models_v \phi.$$

Proof There is a one-to-one correspondence between paths in M from w and extensions of the path v. □

Corollary 3.6 $M \models \phi$ if and only if $T_M \models \phi$.

For the case of truncated tree, Lemma 3.5 is not valid. However, we can prove a related result for formulas up to a certain level of modal nesting.

We inductively define the *rank* of a formula as follows:

Definition 3.7 (Rank of a formula) The rank rank(ϕ) of a formula ϕ is defined as follows:

- $\text{rank}(p) = 0$, where p is a propositional atom.
- $\text{rank}(\neg\phi) = \text{rank}(\phi)$.
- $\text{rank}(\phi_1 \wedge \phi_2) = max\{\text{rank}(\phi_1), \text{rank}(\phi_2)\}$.
- $\text{rank}(\phi_1 \vee \phi_2) = max\{\text{rank}(\phi_1), \text{rank}(\phi_2)\}$.
- $\text{rank}(\Box_i \phi) = \text{rank}(\phi) + 1$.
- $\text{rank}(C\phi) = \infty$.

The rank of a formula ϕ intuitively represents the maximum number of nested modalities that occur in ϕ. If an operator C occurs in ϕ we take the value of $\mathrm{rank}(\phi)$ to be infinite. The rank of a formula reflects the maximal length of any path that needs to be explored to evaluate ϕ on an infinite tree. In other words, to evaluate a formula ϕ of rank k at w_0 we need not examine worlds whose distance from w_0 is greater than k, where distance here is the number of points which appear in the minimal path connecting the two points. The following lemma formalises this.

Lemma 3.8 If $\mathrm{rank}(\phi) \leqslant k$, $M \models \phi$ if and only if $T^k_M \models \phi$.

Proof By corollary 3.6, $M \models \phi$ if and only if $T_M \models \phi$, but, by induction, the evaluation of a formula of $\mathrm{rank}(\phi) \leqslant k$ does not involve the evaluation of nodes of depth greater than k. So $T_M \models \phi$ if and only if $T^k_M \models \phi$, which gives the result. □

In the following we shift our attention from an equivalence Kripke model to its truncated generated tree. Truncated generated trees satisfy $\mathrm{S5}^n$-axioms as the following shows:

Lemma 3.9 Let M be an equivalence model and T^k_M its generated model truncated at k.

1. $T^k_M \models \phi$, where ϕ is a tautology, and $\mathrm{rank}(\phi) \leqslant k$.
2. $T^k_M \models \Box_i(\phi \Rightarrow \psi) \Rightarrow \Box_i\phi \Rightarrow \Box_i\psi$, where $\max\{\mathrm{rank}(\phi), \mathrm{rank}(\psi)\} \leqslant k-1$.
3. $T^k_M \models \Box_i\phi \Rightarrow \phi$, where $\mathrm{rank}(\phi) \leqslant k-1$.
4. $T^k_M \models \Box_i\phi \Rightarrow \Box_i\Box_i\phi$, where $\mathrm{rank}(\phi) \leqslant k-2$.
5. $T^k_M \models \Diamond_i\phi \Rightarrow \Box_i\Diamond_i\phi$, where $\mathrm{rank}(\phi) \leqslant k-2$.
6. If for every vertex $v \in V$ of T^k_M, $T^k_M \models_v \phi$, then $T^k_M \models_v \Box_i\phi$, for any $i \in A$.
7. If for every vertex $v \in V$ of T^k_M, $T^k_M \models_v \phi$, and $T^k_M \models \phi_v \Rightarrow \psi$ then $T^k_M \models_v \psi$.

Proof We prove item number 5. The others can be done similarly. Suppose $T^k_v \not\models \Box_i\phi \Rightarrow \Box_i\Box_i\phi$, where $\mathrm{rank}(\phi) \leqslant k-2$. Since T^k is generated by M, and $\mathrm{rank}(\Box_i\phi \Rightarrow \Box_i\Box_i\phi) \leqslant k$, then by lemma 3.8 $M \not\models \Box_i\phi \Rightarrow \Box_i\Box_i\phi$. But by hypothesis M is an equivalence model. This is absurd. □

So Kripke trees are models for $S5_n$.

Before we proceed further, we introduce a few basic definitions and operations on subtrees.

Definition 3.10 (Rooted-subtrees) Let $T' = (V', E', \sigma')$, $T = (V, E, \sigma)$ be trees with roots v'_0, v_0. T' is a *rooted subtree* of T, written $T' \leqslant T$, if $v_0 \in V' \subseteq V$, and $E|_{V'} = E'$, and $\sigma|_{V'} = \sigma'$.

Definition 3.11 (Intersection of trees) Let $T' = (V', E', \sigma')$ and $T = (V, E, \sigma)$ be trees such that $\sigma|_{V' \cap V} = \sigma'|_{V' \cap V}$. The intersection of T and T' is $T \sqcap T' = (V' \cap V, E' \cap E, \sigma'|_{V' \cap V})$.

It is easy to see that definition 3.11 (when applicable) defines a tree.

Definition 3.12 (Restriction of trees) Let $T = (V, E, \sigma)$ be a tree with root v, and V' a subset of V. The restriction of T to V', written $T|_{V'}$, is the largest rooted subtree of T generated by v whose vertices are in V'. If the root of T is not in V', then $T|_{V'} = -$.

3.2 Kripke trees: refinement

In section 2.2, we discussed the difficulties that arise when using $\mathrm{S5}_n$ Kripke models as knowledge structures for refinement. Example 2.3 showed that any straightforward procedure to refine an equivalence Kripke model will be non-commutative even for universal formulas, i.e. there will be universal α, β, such that $M * \alpha * \beta \not\equiv M * \beta * \alpha$.

Commutativity for universal formulas can be achieved by shifting to Kripke trees. Before we can show this, we must define refinement on Kripke trees.

The typical working scenario in which we operate is the same one as that advocated by [HV91], except that we refine T_M^k instead of M. It can be described as follows: we are given an initial configuration of a MAS, and a set of formulas $\{\phi_1, \ldots, \phi_m\}$ that represent the update of the scenario. The question is whether the updated configuration will validate a set of formulas $\{\psi_1, \ldots, \psi_l\}$. We assume every ψ to have finite rank, i.e. we cannot check a formula containing the operator of common knowledge. There is no restriction on the ϕs.

Our method operates as follows:

1. Start from the most general equivalence Kripke model M that represents the MAS.

2. Generate the infinite tree T_M, as given in Definition 3.2.

3. Generate from T_M^k, the truncated tree of depth k, for some sufficiently large k.

4. Sequentially refine T_M^k with $\{\phi_1, \ldots, \phi_m\}$,

5. Check whether the resulting tree structure satisfies $\{\psi_1, \ldots, \psi_l\}$.

The method describes above needs some further explanation. First, what is the most general Kripke model representing a MAS configurationΓ How are we to build itΓ Our answer is the same as that given by Halpern and Vardi. Assume the set of atoms P is finite. We take the model whose universe W is equal to 2^P, i.e. the universe will cover all the possible assignments to the atoms. We take $\sim_i, i \in A$ to be the universal relations on $W \times W$, and w_0 to be the actual world of the given MAS.

In general we will require that M is more specific than the most general model, e.g. some agent will have a certain knowledge about the world. We can add all the formulas that need be satisfied to the set of updates $\{\phi_1, \ldots, \phi_m\}$. For example in the muddy children example we can start from the model with universal relations and add

$$\bigwedge_{i=1}^{3} C(p_i \Rightarrow K_i p_i)$$

to the set of updates.

We have already explained how to execute steps 1, 2, 3, and 5. We now present a notion of refinement to execute step 4.

Definition 3.13 (Refinement of Kripke tree structures) Given a truncated Kripke tree $T^m = (V, E, \sigma)$, a point $v \in V$, and a formula ϕ, the result $T' = (T, v) * \phi$ of refining T by ϕ at v is procedurally defined as follows. We assume that the negation symbols in ϕ apply only to atomic propositions (to achieve this, negations may be pushed inwards using de Morgan laws and dualities $\Box/\Diamond$ and C/B).

- If $T = -$, then $T' = -$.

- If $T \models_v \phi$, then $T' = T$.
- Otherwise the result is defined inductively on ϕ:
  - $\phi = p$. If $p \in \sigma(v)$, then $T' = T$, else $T' = -$.
  - $\phi = \neg p$. If $p \notin \sigma(v)$ then $T' = T$, else $T' = -$.
  - $\phi = \psi \wedge \chi$. $T' = ((T, v) * \psi) \sqcap ((T, v) * \chi)$.
  - $\phi = \psi \vee \chi$. If $(T, v) * \psi \leqslant (T, v) * \chi$ then $T' = (T, v) * \chi$, and if $(T, v) * \chi \leqslant (T, v) * \psi$ then $T' = (T, v) * \psi$. Otherwise T' is non-deterministically given as $(T, v) * \psi$ or $(T, v) * \chi$.
  - $\phi = \Box_i \psi$. T' is given by computing as follows:

```
$T' := T$;
for each $v'$ such that $(v, v') \in E_i$ do
    if $(T', v') * \psi = -$, then
        $T' := T'|_{V - \{v'\}}$
    else
        $T' := (T', v') * \psi$
```

  - $\phi = \Diamond_i \psi$. Let X be the set $X = \{(T, v') * \phi \mid (v, v') \in E_i)$.

    If $X = \varnothing$, then $T' = -$,
    else T' is nondeterministically chosen to be a $\leqslant$-maximal element of X.

  - $\phi = C\psi$. T' is given by computing as follows:

```
$T' := T$;
for each $v'$ such that $(v, v') \in E^*$ do
    if $(T', v') * \psi = -$, then
        $T' := T'|_{V - \{v'\}}$
    else
        $T' := (T', v') * \psi$
```

  - $\phi = B\psi$. Let X be the set $X = \{(T, v') * \phi \mid v' \in V\}$.

    If $X = \varnothing$, then $T' = -$,
    else T' is nondeterministically chosen to be a $\leqslant$-maximal element of X.

$T * \phi$ means $(T, v) * \phi$, where v is the root of T.

Lemma 3.14 Given a tree T, a formula α and a point v, $(T, v) * \alpha$ is a tree.

Proof Follows from the fact that if T is a tree then $T|_{V'}$ is also a tree. □

The intuition behind $(T, v) * \phi$ is that it is obtained by removing as small a set of links from T as possible, in order to satisfy ϕ. Note that, due to the clauses for $\vee, \Diamond_i, B$, $(T, v) * \phi$ is not uniquely defined. However, we will see that running the procedure on the muddy children example does not introduce nondeterminism.

4 The muddy children puzzle using Kripke trees

In section 2.1, we described the muddy children puzzle and we reported the formalisation that was given in [FHMV95, HV91]. The aim of the present section is to solve an instance of it (where the actual situation is coded by the tuple p_1, p_2, p_3 that we equivalently write as $(1, 1, 1)$ - all the children are muddy) by using Kripke trees and the methods we introduced in section 3.

We start with the most general model to represent the puzzle: this is the model M_1 of figure 1[1]. Given M_1, we generate the infinite tree T_{M_1} for M_1 and then the truncation T_1 of T_{M_1}. In this example, we only need three levels to be unravelled. The starting tree and the three successive refinements are in figure 7, and 8. Let $\phi_1 = C(p_1 \vee p_2 \vee p_3)$ (this is the father's announcement), and $\phi_2 = C(\neg\Box_1 p_1 \wedge \neg\Box_1\neg p_1) \wedge C(\neg\Box_2 p_2 \wedge \neg\Box_2\neg p_2) \wedge C(\neg\Box_3 p_3 \wedge \neg\Box_3\neg p_3)$ (the children's simultaneous reply that they don't know whether or not they are muddy). We now sequentially update T_1 by ϕ_1 and then by ϕ_2 three times. Note that since all children are muddy, they will have to speak three times before everyone knows he is muddy.

Consider the algorithm of definition 3.13 and T_1. Following the algorithm, the refined tree $T_1 * \phi_1 = T_2$ in figure 7 is T_1 in which the links to states where no children are muddy have been removed. $T_2 * \phi_2$ is then achieved by isolating worlds that do not see two worlds for every relation. In fact, only in this case one of the formulas $\Diamond_i p_i \wedge \Diamond_i \neg p_i$ can fail on a point of T_2. We can now obtain T_3, and T_4 (shown in figure 8) similarly.

Having made all the refinements, we can now check whether or not the muddy children know that they are muddy. This involves checking

$$T_4 \models p_i \Rightarrow K_i p_i,$$

which is indeed the case.

Analogously we can prove that the procedure given in section 3 produces solutions for the other cases of the muddy children.

Note that had we decided to consider the Kripke tree truncated at $n \geq 4$, the formula $\bigwedge_{i=1}^{3} p_i$ would still be satisfied at the root after three refinements.

5 Properties of refinement on Kripke trees

In the rest of the paper we analyse some more properties of the refinement procedure that we defined in definition 3.13.

The first remark that we should make is that refining a scenario by some agent's knowledge cannot affect other agents' knowledge, as was the case in example 2.1 for Kripke models. This is because by unravelling a Kripke model we produce a tree whose leaves are in a bijection with paths of the original model. We formalise this as follows:

[1] According the the notion of most general model as described in section 3.2 the model M should actually be $M = (2^{\{p_1, p_2, p_3\}}, U, \pi, w)$, where U is the universal relation on $W \times W$, and $\pi(w) = \{p_1, p_2, p_3\}$. The model M_1 we analyse is the result of the update of M by

$$C(p_i \Rightarrow K_j p_i) : i \neq j; i, j \in \{1, 2, 3\},$$

where the formula above represents the fact that children can see each other. For brevity (as in [FHMV95, HV91]) we start our analysis from M_1; i.e. rather than building the tree for M and update it first by $C(p_i \Rightarrow K_j p_i)$, we directly build the tree for M_1. The reader can check that this leads to the same result.

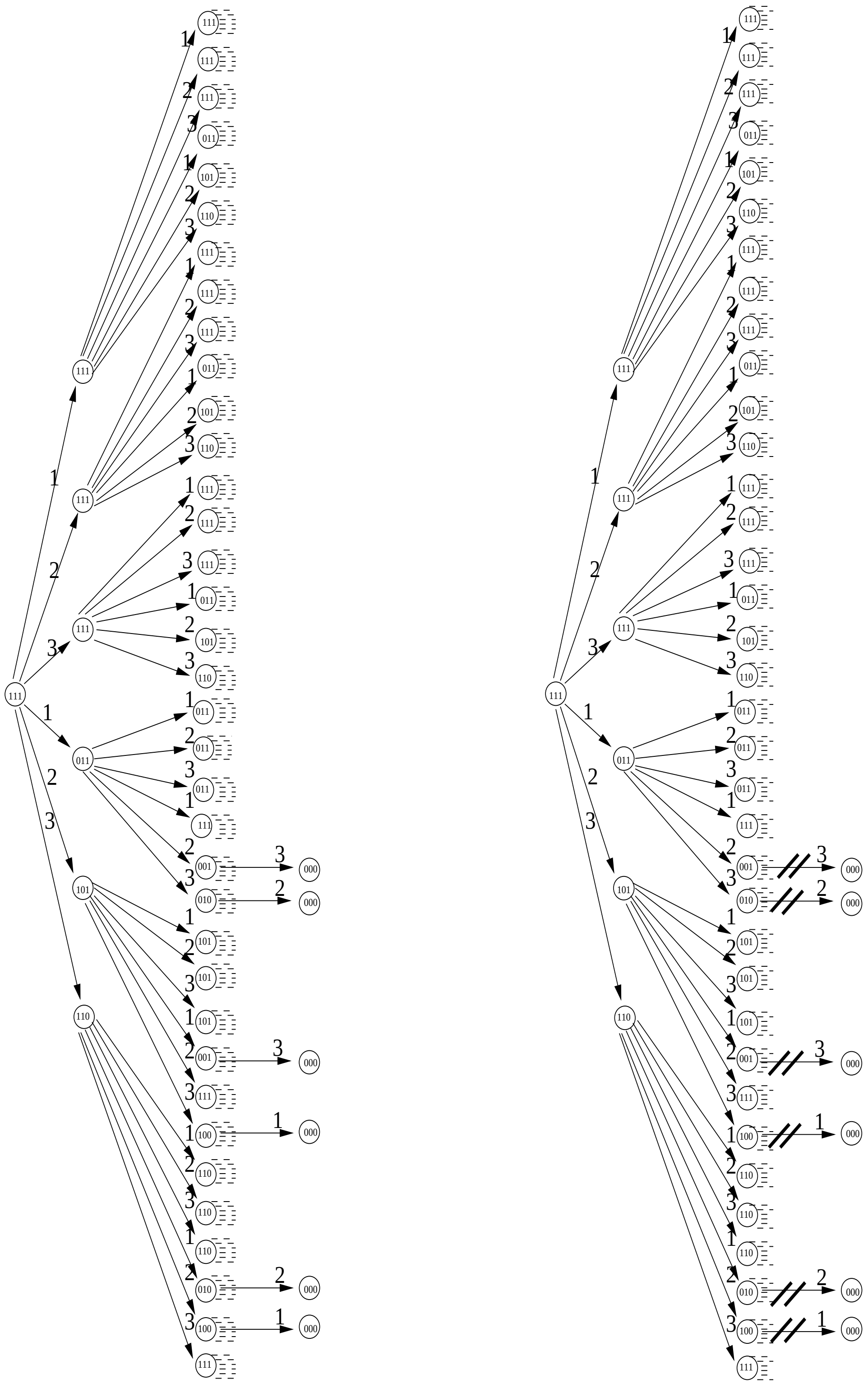

Figure 7: T_1, T_2: The Kripke trees before and after the father speaks; and after the children speak the first time.

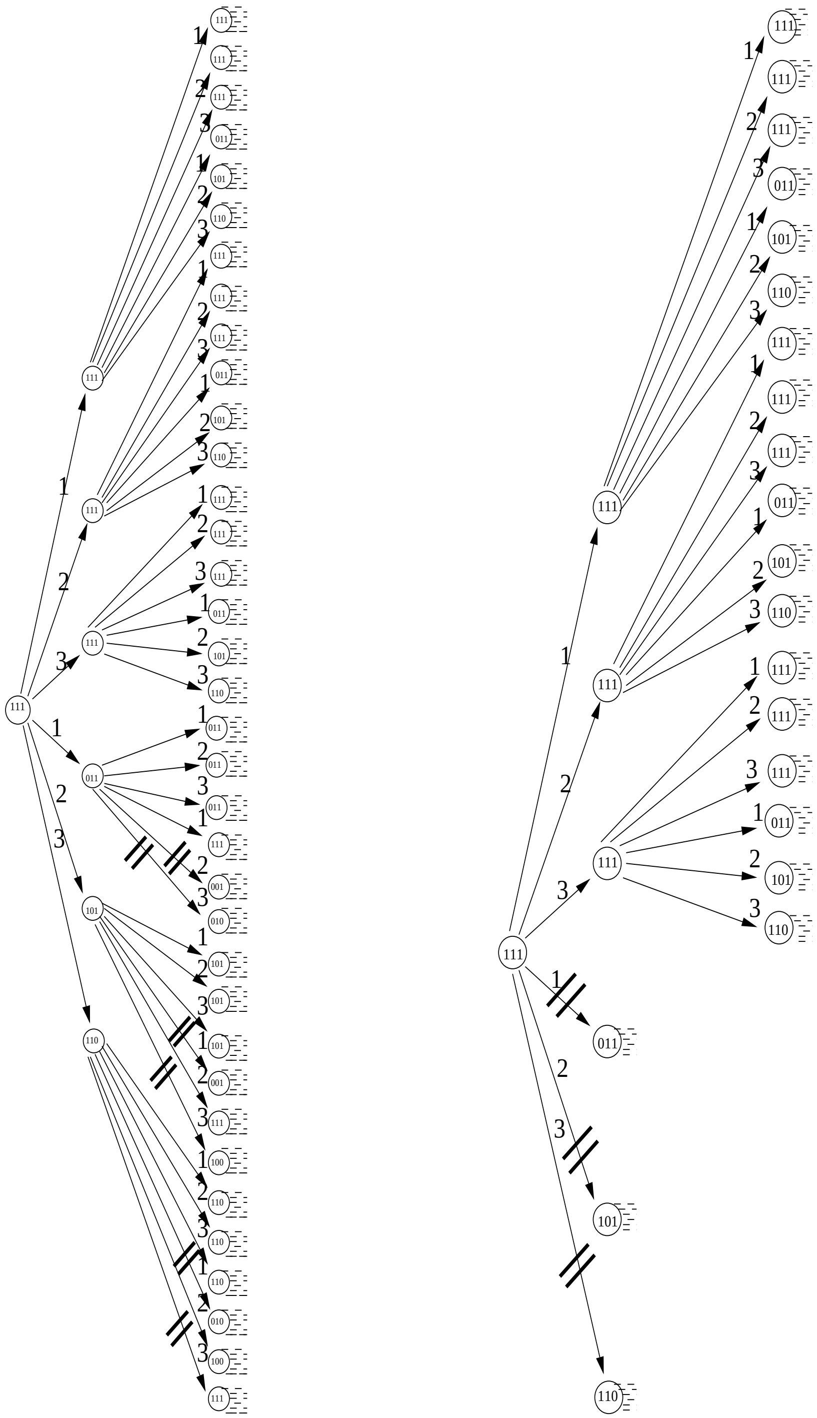

Figure 8: T_3, T_4: The Kripke trees after the children speak the second and third time.

Theorem 5.1 Let T be a tree, and ϕ, ψ two formulas. Then:

$$T \models \Box_i \phi \text{ implies } (T * \Box_j \psi \models \Box_i \phi), \text{ with } i \neq j.$$

Proof Nodes of a Kripke tree are in a bijection with paths of the generating model. Therefore by removing some j-links we cannot affect the interpretation of any modality whose index is not j. □

Although the theorem above refers to infinite trees, an analogue version can be proved for truncated trees. In that case we need the rank of the formulas to be less or equal to the depth of the truncated tree minus 1.

The second point worth stressing is that Kripke trees solve the problem of example 2.3, i.e. we can prove commutativity although the result is limited to *safe* formulas:

Definition 5.2 A formula is *safe* if it is universal and, after negations are pushed inwards, no $\Box_i$ and no C appears in the scope of $\vee$. A formula is *disjunction-free* if it is universal and, after negations are pushed inwards, has no $\vee$.

We need a few results before proving commutativity of safe formulas.

Lemma 5.3 Let ϕ, α be any formula and v any point of T.

1. $(T, v) * \phi \leqslant T$.
2. If α is disjunction-free, $T_1 \leqslant T_2$ implies $(T_1, v) * \alpha \leqslant (T_2, v) * \alpha$, where $v \in V_1 \cap V_2$.
3. If α is universal then $T \models \alpha$ and $- \neq T' \leqslant T$ imply $T' \models \alpha$.

Proof
1. The procedure for obtaining $(T, v) * \phi$ only removes links.
2. Induction on α. We assume all negations pushed inwards. Let $T_1' = (T_1, v) * \alpha$ and $T_2' = (T_2, v) * \alpha$. Suppose α is of the form:
   - p. If $p \in \sigma(v)$ then $T_1' = T_1$, $T_2' = T_2$; else $T_1' = T_2' = -$.
   - $\neg p$. Similar.
   - $\beta \wedge \gamma$.
     $$\begin{array}{rcll} (T_1, v) * \alpha & = & (T_1, v) * \beta \sqcap (T_1, v) * \gamma & \\ & \leqslant & (T_2, v) * \beta \sqcap (T_2, v) * \gamma & \text{IH} \\ & = & (T_2, v) * \alpha & \end{array}$$
   - $\Box_i \beta$. Set $T_1' = T_1$ and $T_2' = T_2$ and we execute the loops of definition 3.13 ($\Box_i$-case) synchronously. We will show that $T_1' \leqslant T_2'$ is an invariant of the execution.
     Suppose $(v, v') \in E_{2i}$.
     - If $(v, v') \in E_{1i}$, then consider the following cases:
       - $(T_1, v') * \beta = -$ and $(T_2, v') * \beta = -$.
         $T_1' := T_1'|_{V - \{v'\}}$ and $T_2' := T_2'|_{V - \{v'\}}$, so $T_1' \leqslant T_2'$ is not violated.
       - $(T_1, v') * \beta = -$ and $(T_2, v') * \beta \neq -$.
         $T_1' := T_1'|_{V - \{v'\}}$ and $T_2' := (T_2', v') * \beta$, and $T_1' \leqslant T_2'$.

  - $*$ $(T_1, v') * \beta \neq -$ and $(T_2, v') * \beta = -$.
    Contradicts hypothesis that $T_1' \leqslant T_2'$.
  - $*$ $(T_1, v') * \beta \neq -$ and $(T_2, v') * \beta \neq -$.
    $T_1' := (T_1', v') * \beta$, $T_2' := (T_2', v') * \beta$, and $T_1' \leqslant T_2'$ by induction hypothesis.
- If If $(v, v') \notin E_{1i}$ then T_1' is unchanged by the body of the loop, while T_2' becomes one of $T_2' := T_2'|_{V-\{v'\}}$ and $(T_2', v') * \beta$. In either case, we are removing links in T_2 which are not present in T_1, so $T_1' \leqslant T_2'$ is preserved.

- $C\beta$. Similar to $\Box_i \beta$.

3. Induction on α.

□

Theorem 5.4 (Success) If α is universal, $(T, v) * \alpha = -$ or $(T, v) * \alpha \models_v \alpha$.

Proof Induction on α. The cases $\alpha = p, \neg p, \psi \vee \chi$ are straightforward; the case $\alpha = \psi \wedge \chi$ requires part 3 of lemma 5.3. □

Lemma 5.5 If α is safe, then the outcome of $(T, v) * \alpha$ is deterministically defined.

Proof Suppose ϕ contains no $\Box_i, C$. Then it's an easy induction to see that $(T, v) * \phi$ is either T or $-$. Now consider $(T, v) * (\phi \vee \psi)$, where ϕ, ψ are $\Box_i, C$-free. We see that either $(T, v) * \phi \leqslant (T, v) * \psi$ or $(T, v) * \psi \leqslant (T, v) * \phi$, so the result is again T or $-$. □

We show that, for universal formulas, the change made be a refinement is the minimal one possible in order to satisfy the formula:

Theorem 5.6 If α is safe, $(T, v) * \alpha$ is $\leqslant$-maximum in $\{T' \leqslant T \mid T' \models_v \alpha \text{ or } T' = -\}$.

Proof Let $T' = (T, v) * \alpha$. By part 1 of lemma 5.3 and theorem 5.4, we know T' is in the set. To prove that it is maximum, take any T'' in the set; we will show $T'' \leqslant T'$. If $T'' = -$ the result is immediate; otherwise, we have $T'' \models_v \alpha$ and $T'' \leqslant T$. Since $T'' \leqslant T$, we get $(T'', v) * \alpha \leqslant (T, v) * \alpha$ by part 2 of lemma 5.3. But $(T'', v) * \alpha = T''$ (since $T'' \models_v \alpha$) and $(T, v) * \alpha = T'$, so $T'' \leqslant T'$. □

Theorem 5.7 If α, β are safe $(T, v) * \alpha * \beta$ is maximum in $\{T' \leqslant T \mid T' \models_v \alpha \wedge \beta \text{ or } T' = -\}$.

Proof Let $T' = (T, v) * \alpha * \beta$. By parts 1 and 3 of lemma 5.3 and theorem 5.4, we know T' is in the set. The argument that it is maximum is similar to the proof of theorem 5.6. Take any T'' in the set; we will show $T'' \leqslant T'$. If $T'' = -$ the result is immediate; otherwise, we have $T'' \models_v \alpha \wedge \beta$ and $T'' \leqslant T$. Since $T'' \leqslant T$, we get $(T'', v) * \alpha * \beta \leqslant (T, v) * \alpha * \beta$ by part 2 of lemma 5.3. But $(T'', v) * \alpha * \beta = T''$ (since $T'' \models_v \alpha$, $(T'', v) * \alpha = T''$, and since $T'' \models_v \beta$, $(T'', v) * \beta = T''$), and $(T, v) * \alpha * \beta = T'$, so $T'' \leqslant T'$. □

Theorem 5.8 (Commutativity) If α, β are safe, then $T * \alpha * \beta = T * \beta * \alpha$.

Proof By theorem 5.7, $T * \alpha * \beta$ and $T * \beta * \alpha$ are maximum in the same set. Therefore they are equal. □

We conjecture that the premise can be relaxed to universal formulas. It is worth mentioning an example of which non-universal formulas can make commutativity to fail.

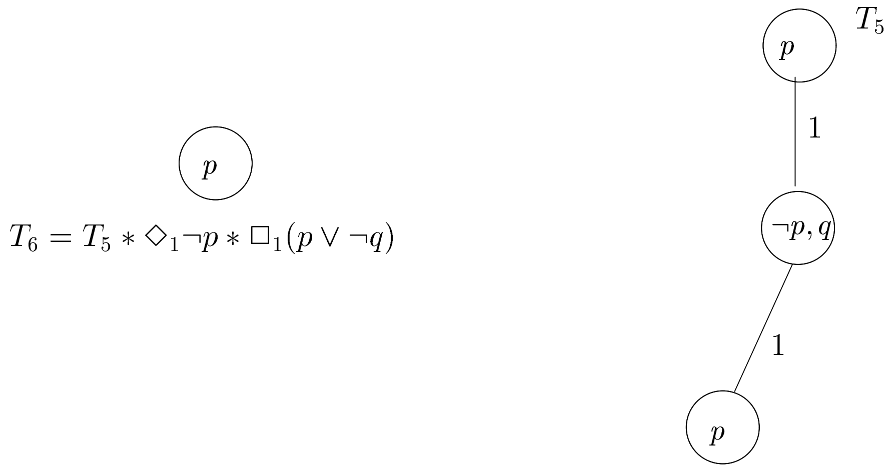

Figure 9: T_5 and T_6 discussed in example 5.9. While $T_6 = T_5 * \Diamond_1 \neg p * \Box_1 (p \vee \neg q)$ is defined and shown above, $T_7 = T_5 * \Box_1 (p \vee \neg q) * \Diamond_1 \neg p$ is undefined.

Example 5.9 Commutativity can fail for arbitrary formulas. The problem is that if the formulas are non-universal, the order of updating can play a role in the outcome of the update and we might have that one of the two cases fail. We are so far unable to find examples in which the two updates succeed but produce different result (we conjecture this is impossible; see also conclusions about this). The example we report here is the tree T_5, illustrated in figure 9, where the root is the top vertex. Consider now $T_6 = T_5 * \Diamond_1 \neg p * \Box_1 (p \vee \neg q)$, illustrated, and $T_7 = T_5 * \Box_1 (p \vee \neg q) * \Diamond_1 \neg p = -$.

6 Conclusions and further work

In [Woo97] Mike Wooldridge, discussing the problems of using possible worlds semantics as formal specification for MAS, writes:

> [... possible worlds semantics are generally *ungrounded*. That is, there is usually no precise relationship between the abstract accessibility relations that are used to characterise an agent's state and any concrete computational model... this makes it difficult to go from a formal specification of a system in terms of beliefs, desires, and so on, to a concrete computational system...] (page 9)

This is indeed very often the case and this line of research aims at bringing us a step towards the use of possible worlds semantics as specification and reasoning tool for MAS.

In this paper we have developed the proposal in [HV91] for model refinement and model checking. We argued that model refinement could not be defined satisfactorily on Kripke models, and proposed a definition on Kripke trees obtained from Kripke models instead.

The shift from Kripke models to Kripke trees let us achieve two main results. First, we showed that it is possible to refine trees by a formula expressing knowledge of a formula without affecting the knowledge of the other agents (theorem 5.8) - this was not apparently possible on standard Kripke models (see example 2.1). Secondly, while it seems impossible to obtain commutativity for even safe formulas on Kripke models, we showed this is possible for Kripke trees.

Many of the issues we discussed in this note still need investigating. The following is a list of conjectures that we have not proved (or refuted) yet.

1. If $T * \phi * \psi \neq -$, and $T * \psi * \phi \neq -$, then $T * \phi * \psi \equiv T * \psi * \phi$.
2. If ϕ, ψ are universal (not necessarily safe) then $T * \phi * \psi \equiv T * \psi * \phi$. This is implied by 1.
3. Let $T = M_0$, where M_0 is the most general model and $\phi_1, \ldots, \phi_n$ be S5^n-consistent formulas. Then $T * \phi_1 * \ldots * \phi_n \neq -$. (Item 2 implies they would commute).
4. $\phi \equiv \psi$ implies $T * \phi \equiv T * \psi$.

Further work will be focused on proving the above and trying to address the following more general issues:

- What is the appropriate level of truncation that we need applyΓ Is there a mathematical formula that can compute itΓ
- Although every generated tree will satisfy S5^n, an update of it in general will not. Is this a strong point against model refinement as it is defined hereΓ
- We have proved that model refinement satisfies the properties like success (theorem 5.4), commutativity (theorem 5.8). Are there other important properties that we should check or advocate, for example the ones discussed in [Gär88] and [KM91]Γ

References

[FHMV95] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge, 1995.

[Gär88] P. Gärdenfors. *Knowledge in Flux: Modelling the Dynamics of Epistemic States.* MIT Press, 1988.

[HC84] G. E. Hughes and M. J. Cresswell. *A Companion to Modal Logic.* Methuen, London, 1984.

[HC96] G. E. Hughes and M. J. Cresswell. *A new introduction to modal logic.* Routledge, New York, 1996.

[Hin62] J. Hintikka. *Knowledge and Belief, an introduction to the logic of the two notions.* Cornell University Press, Ithaca (NY) and London, 1962.

[HV91] Joseph Halpern and Moshe Vardi. *Model checking vs. theorem proving: a manifesto*, pages 151–176. Artificial Intelligence and Mathematical Theory of Computation. Academic Press, Inc, 1991.

[KM91] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 3(52):263–294, December 1991.

[Woo97] M. Wooldridge. Agent-based software engineering. *IEE Proceedings Software Engineering*, 144(1):26–37, 1997.