Department of Computer Science,
University of Otago

Technical Report OUCS-2006-07

What is my number? - A new epistemic riddle

Authors:

Hans van Ditmarsch
Department of Computer Science, University of Otago

Ji Ruan
Computer Science, University of Liverpool

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.html

What is my number? – A new epistemic riddle

H.P. van Ditmarsch[1⋆] and J. Ruan[2]

[1] University of Otago, New Zealand, hans@cs.otago.ac.nz
[2] University of Liverpool, United Kingdom, jruan@csc.liv.ac.uk

Abstract. A common theme in epistemic riddles is that announcements of ignorance may eventually result in knowledge. We present a fairly new epistemic riddle, including some variants that were partly accidentally designed due to a miscommunication between logic puzzle enthusiasts. The design was facilitated because such riddles can be specified, and fairly easily checked, in 'public announcement logic', a modal logic with both dynamic and epistemic operators; and because of the availability of epistemic model checking tools for the finetuning and verification of results. Logic puzzle design could benefit from similar future efforts.

Keywords: modal logic, logic puzzle, model checking, action logic

1 Introduction

Consider the following riddle—transcribed in our preferred terminology—that appeared in Math Horizons in 2004, as 'Problem 182' in a regular problem section of the journal, edited by A. Liu [1].

> *Each of agents Anne, Bill, and Cath has a positive integer on its forehead. They can only see the foreheads of others. One of the numbers is the sum of the other two. All the previous is common knowledge. The agents now successively make the truthful announcements:*
> *i. Anne: "I do not know my number."*
> *ii. Bill: "I do not know my number."*
> *iii. Cath: "I do not know my number."*
> *iv. Anne: "I know my number. It is 50."*
> *What are the other numbers?*

By an unreliable chain of logic puzzle enthusiasts the riddle reached us in a version with 'natural number' instead of 'positive integer'. That includes the number 0. In which case the riddle can no longer be solved. There is now exactly one other number pair that remains possible when Anne knows that she has 50.

Which other number pair?

The uncertainty about the formulation of the riddle and its solution did not deter the correspondents. Instead, it provoked their creativity and they redesigned the riddle.

⋆ Hans van Ditmarsch is contact author. We greatly acknowledge input from David Atkinson, Jan van Eijck, Wiebe van der Hoek, Barteld Kooi, and Rineke Verbrugge.

David Atkinson mentioned a solution for natural numbers, if Anne would have said that her number was 51. Strangely enough, we discovered later that this is also a solution when 0 is excluded. There is also (another) rather crucial difference between the 50 and the 51 version of the riddle...

What are the other numbers in the case 51?

In fact, there is an infinite number of x's such that the other numbers can be determined after Anne announces that she knows that her number is x.

In the mean time Hans formulated a version where the numbers must be between 0 and an upper bound max, and where after the three ignorance announcements of the original riddle the problem is:

What is the range of max*, if Anne now always knows her number?*

It is not possible to determine Anne's number in this case, nor what the numbers of the other agents are. This version is a great deal harder (we think) than the previous version and the original riddle. If no upper bound were given, Anne would certainly *not* always know her number.

Actually, Hans did the computation for one specific value of max—so that the question to the reader becomes "Show that Anne now always knows her number." He then enlisted the help of Ji Ruan, who explored the problem in the epistemic model checker DEMO recently developed by Jan van Eijck [2]. Ji confirmed, and corrected, the result and determined the requested range that constitutes the solution. This range includes (exactly) one value of max for which it (therefore) can be determined that Anne always knows the numbers, and *also* what Anne's number is (but again, not what the other numbers are).

Which value of max*?*

The 'what is my number' riddle(s) combines features from 'wisemen' or 'muddy children' puzzles [3] with features from another classic, the 'sum and product' riddle [4, 5]. A common feature in such riddles is that we are given a multi-agent interpreted system, and that successive announcements of ignorance finally result in its opposite, typically factual knowledge. An interpreted system [6] consists of a set of global states. A global state is (at least) a list of local states for each agent. There is common knowledge that each agent only knows its local state, and the extent of the domain—for ignorance statements to be truly informative the domain should be more restrictive than the full cartesian product of the sets of local state values. As in 'muddy children', we do not take the 'real' state of the agent (the number on its forehead) as its local state, but instead the information seen on the foreheads of others (the other numbers). 'Sum and product' is also about sums of numbers. Other epistemic riddles around involve cryptography and the verification of information security protocols [7].

The understanding of such riddles is facilitated by the availability of a suitable specification language, namely the logic of public announcements, wherein succinct descriptions in the logical language are combined with convenient relational structures on which to interpret them; and also by the availability of verification tools, such as DEMO, to aid interpreting such descriptions on such structures. It is harder to answer if and why a designed riddle is 'fun'. This par-

tially depends on its computational features. We corroborate this observation by some relevant comparisons with other riddles and games.

Section 2 provides an introduction into public announcement logic, and in Section 3 we analyze 'what is my number' in this logic—including solutions to all versions of the riddle here presented. In Section 4 we verify the results in the model checker DEMO. Section 5 shortly addresses the issue of computational complexity in game design.

2 Public Announcement Logic

Public announcement logic is a dynamic epistemic logic and is an extension of standard multi-agent epistemic logic. Intuitive explanations of the epistemic part of the semantics can be found in [6, 7]. We give a concise overview of, in that order, the language, the structures on which the language is interpreted, and the semantics.

Given are a finite set of agents N and a finite or countably infinite set of atoms P. The language of public announcement logic is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_n\varphi \mid C_B\varphi \mid [\varphi]\psi$$

where $p \in P$, $n \in N$, and $B \subseteq N$ are arbitrary. Other propositional and epistemic operators are introduced by abbreviation. For $K_n\varphi$, read 'agent n knows formula φ'. For example, if Anne knows that her number is 50, we can write $K_a 50_a$, where a stands for Anne and some set of atomic propositions is assumed that contains 50_a to represent 'Anne has the number 50.' For $C_B\varphi$, read 'group of agents B commonly know formula φ'. For example, we have that $C_{abc}(20_b \rightarrow K_a 20_b)$: it is common knowledge to Anne, Bill, and Cath, that if Bill's number is 20, Anne knows that (because she can see Bill's number on his forehead)—instead of $\{a, b, c\}$ we often write abc. For $[\varphi]\psi$, read 'after public announcement of φ, formula ψ (is true)'. For example, after Anne announces "(I know my number. It is 50.)" it is common knowledge that Bill's number is 20. This is formalized as $[K_a 50_a] C_{abc} 20_b$.

The basic structure is the epistemic model. This is a Kripke structure, or model, wherein all accessibility relations are equivalence relations. An *epistemic model* $M = \langle S, \sim, V\rangle$ consists of a *domain* S of (factual) *states* (or 'worlds'), *accessibility* $\sim : N \rightarrow \mathcal{P}(S \times S)$, where each $\sim(n)$ is an equivalence relation, and a *valuation* $V : P \rightarrow \mathcal{P}(S)$. For $s \in S$, (M, s) is an *epistemic state* (also known as a pointed Kripke model). For $\sim(n)$ we write $\sim_n$, and for $V(p)$ we write V_p. Accessibility $\sim$ can be seen as a set of equivalence relations $\sim_n$, and V as a set of valuations V_p. Given two states s, s' in the domain, $s \sim_n s'$ means that s is indistinguishable from s' for agent n on the basis of its information. For example, at the beginning of the riddle, triples $(2, 14, 16)$ and $(30, 14, 16)$ are indistinguishable for Anne but not for Bill nor for Cath. Therefore, assuming a domain of natural number triples, we have that $(2, 14, 16) \sim_a (30, 14, 16)$. The group accessibility relation $\sim_B$ is the transitive and reflexive closure of the union of all accessibility relations for the individuals in B: $\sim_B \equiv (\bigcup_{n\in B} \sim_n)^*$. This

relation is used to interpret common knowledge for group B. Instead of '$\sim_B$ equivalence class' ($\sim_n$ equivalence class) we write B-class (n-class).

For the semantics, assuming an epistemic model $M = \langle S, \sim, V\rangle$:

$$\begin{array}{ll}
M,s \models p & \text{iff } s \in V_p \\
M,s \models \neg\varphi & \text{iff } M,s \not\models \varphi \\
M,s \models \varphi \wedge \psi & \text{iff } M,s \models \varphi \text{ and } M,s \models \psi \\
M,s \models K_n\varphi & \text{iff for all } t \in S : s \sim_n t \text{ implies } M,t \models \varphi \\
M,s \models C_B\varphi & \text{iff for all } t \in S : s \sim_B t \text{ implies } M,t \models \varphi \\
M,s \models [\varphi]\psi & \text{iff } M,s \models \varphi \text{ implies } M|\varphi, s \models \psi
\end{array}$$

where epistemic model $M|\varphi = \langle S', \sim', V'\rangle$ is defined as

$$\begin{array}{l}
S' = \{s' \in S \mid M,s' \models \varphi\} \\
\sim'_n = \sim_n \cap (S' \times S') \\
V'_p = V_p \cap S'
\end{array}$$

The dynamic modal operator $[\varphi]$ is interpreted as an epistemic state transformer. Announcements are assumed to be truthful, and this is commonly known by all agents. Therefore, the model $M|\varphi$ is the model M restricted to all the states where φ is true, including access between states. The dual of $[\varphi]$ is $\langle\varphi\rangle$: $M,s \models \langle\varphi\rangle\psi$ iff $M,s \models \varphi$ and $M|\varphi, s \models \psi$. Formula φ is valid on model M, notation $M \models \varphi$, if and only if for all states s in the domain of M: $M,s \models \varphi$. Formula φ is valid, notation $\models \varphi$, if and only if for all models M: $M \models \varphi$.

A proof system for this logic is presented, and shown to be complete, in [8], with precursors—namely for public announcement logic *without* common knowledge—in [9, 10]. A concise completeness proof is given in [7]. The logic is decidable both with and without common knowledge [9, 8]. Results on the complexity of both logics can be found in [11].

In public announcement logic, not all formulas remain true after their announcement, in other words, $[\varphi]\varphi$ is *not* a principle of the logic. Some formulas involving epistemic operators become **false** after being announced! For a simple example, consider that Bill were to tell Anne (truthfully) at the initial setting of the riddle: "You don't know that your number is 50." Using a conversational implicature this means "(Your number is 50 and) You don't know that your number is 50." This is formalized as $50_a \wedge \neg K_a 50_a$. After the announcement, Anne knows that her number is 50: $K_a 50_a$. Therefore the announced formula, that was true before the announcement, has become false after the announcement. In the somewhat different setting that formulas of form $p \wedge \neg K_n p$ cannot be consistently known this phenomenon is called the Moore-paradox [12, 13]. In the underlying dynamic setting it has been described as an *unsuccessful update* [10, 7]. Similarly, ignorance statements in 'what is my number' such as Anne saying that she does not know her number may in due time lead to Anne knowing her number, the opposite of her ignorance.

3 Formalization of 'What is my number'

The set of agents $\{a, b, c\}$ represent Anne, Bill and Cath, respectively. Atomic propositions i_n represent that agent n has natural number i on its forehead. Therefore the set of atoms is $\{i_n \mid i \in \mathbb{N} \text{ and } n \in \{a, b, c\}\}$—or $\mathbb{N}^+$ instead of $\mathbb{N}$ and/or $i \leq \mathsf{max}$ in case there is an upper bound for the numbers. Given an upper bound, the property of, e.g., Anne seeing (and therefore knowing) the numbers of Bill and Cath is described as $\bigwedge_{y,z \leq \mathsf{max}}((y_b \wedge z_c) \rightarrow K_a(y_b \wedge z_c))$. The successive ignorance announcements, such as Anne saying: "I don't know my number," are quite simply described, namely as, in this case $\neg \bigvee_{x \leq \mathsf{max}} K_a x_a$ whereas the more general requirement that Anne now always knows her number corresponds to the validity of that formula on the epistemic model describing the problem or, viewed from the perspective of an actual number triple in that model, the truth of $C_{abc} \bigvee_{x \leq \mathsf{max}} K_a x_a$. Without an upper bound for the numbers, all these properties and announcements have *infinitary* descriptions which are not permitted in this (propositional) logic.

The epistemic model $\mathcal{T}^{\mathsf{max}} = \langle S, \sim, V \rangle$ is defined as follows, assuming a range $[0..\mathsf{max}]$ of natural numbers x, y, z. Models $\mathcal{T}^{\mathbb{N}}$, $\mathcal{T}^{\mathbb{N}+}$, and $\mathcal{T}^{\mathsf{max}+}$ are defined similarly for the corresponding other range; write $\mathcal{T}$ when range does not matter.

$$
\begin{array}{lll}
S & \equiv & \{(x, y, z) \mid x = y + z \text{ or } y = x + z \text{ or } z = x + y\} \\
(x, y, z) \sim_a (x', y', z') & \text{iff} & y = y' \text{ and } z = z' \\
(x, y, z) \sim_b (x', y', z') & \text{iff} & x = x' \text{ and } z = z' \\
(x, y, z) \sim_c (x', y', z') & \text{iff} & x = x' \text{ and } y = y' \\
(x, y, z) \in V_{x_a} & & \\
(x, y, z) \in V_{y_b} & & \\
(x, y, z) \in V_{z_c} & &
\end{array}
$$

The initial epistemic state of the puzzle can be described by a characteristic formula for finite interpreted systems [14] (this applies results for characterizing models formulated in [15, 16]). This presumes that the underlying system is seen as an interpreted system by regarding the number pair of the other agents as your own local state value. Up to bisimilarity, a finite model $\mathcal{T}$ ($\mathcal{T}^{\mathsf{max}}$, or $\mathcal{T}^{\mathsf{max}+}$) is now described by a theory $\mathcal{K}$ listing facts, and for each agent knowledge, and ignorance—we slightly abuse the language in the conjunctions, that are actually over all local state values, and in the atom names; note that ignorance and knowledge are given for Anne only; the operator $\hat{K}$ is the dual of K and stands for 'the agent considers it possible that':

$$
\begin{array}{l}
\bigvee_{(x,y,z)}(x_a \wedge y_b \wedge z_c) \\
\bigwedge_{(.,y,z)}((y_b \wedge z_c) \rightarrow K_a(y_b \wedge z_c)) \\
\bigwedge_{(.,y,z)}((y_b \wedge z_c) \rightarrow \hat{K}_a(y + z)_a \wedge \hat{K}(|y - z|)_a)
\end{array}
$$

One now has (as a slight adaptation and application of [14]) that arbitrary φ are valid in $\mathcal{T}$ iff they are entailed by $\mathcal{K}$, i.e. $\mathcal{T} \models \varphi$ iff $\mathcal{K} \models \varphi$; and that an arbitrary φ is true in an epistemic state for $\mathcal{T}$ iff it is entailed by an 'identifier' of that

state and common knowledge of $\mathcal{K}$: $\mathcal{T}, (x, y, z) \models \varphi$ iff $(x_a \wedge y_b \wedge z_c) \wedge C_{abc}\mathcal{K} \models \varphi$. The formula $(x_a \wedge y_b \wedge z_c) \wedge C_{abc}\mathcal{K}$ is the characteristic formula, or description, of the epistemic state $(\mathcal{T}, (x, y, z))$.

The fine-structure of the epistemic model $\mathcal{T}$ is not apparent from its formal definition nor from its characterization. A relevant question is what the background knowledge is that is available to the agents, i.e., what the *abc*-classes in the model are. Such a computation was performed by Panti for the 'sum and product' riddle, which revealed three classes.[3] A model for $\mathcal{T}$ for 'what is my number' has a different structure, with many more common knowledge classes. It is therefore quite informative to know what they are. The most elegant case is $\mathcal{T}^{\mathbb{N}+}$.

An *abc*-class in $\mathcal{T}^{\mathbb{N}+}$ is binary tree. The depth of the tree is according to the order: $(x, y, z) > (u, v, w)$ iff $(x > y$ and $y = v$ and $z = w)$ or $(x = u$ and $y > v$ and $z = w)$ or $(x = u$ and $y = v$ and $z > w)$. Modulo a permutation of agents and corresponding permutation of arguments in triples, every node except the root has one predecessor and two successors, as in Figure 1.

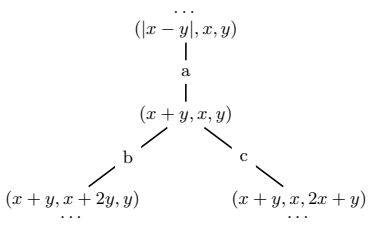

Fig. 1. Modulo agent symmetry, all parts of the model $\mathcal{T}^{\mathbb{N}+}$ branch as here. Arcs connecting nodes are labelled with the agent who cannot distinguish those nodes.

The root of each tree has label $(2x, x, x)$ or $(x, 2x, x)$ or $(x, x, 2x)$. An agent who sees two equal numbers, immediately infers that its own number must be their sum (twice the number that is seen), because otherwise it would have to be their difference 0 which is not a positive natural number. It will be obvious that: the structure truly is a forest (a set of trees), because each node only has a single parent; all nodes except roots are triples of three *different* numbers; and all trees are infinite. All *abc*-trees are isomorphic modulo (i) a multiplication factor for the numbers occurring in the arguments of the node labels, and modulo (ii) a permutation of arguments and a corresponding swap of agents, i.e., swap of arc labels. For example, the numbers occurring in the tree with root $(6, 3, 3)$ are thrice the corresponding numbers in the tree with root $(2, 1, 1)$; the tree with root $(2, 1, 1)$ is like the tree for root $(1, 2, 1)$ by applying permutations (213) to arguments and (alphabetically ordered) agent labels alike. For more details, see (the left side of) Figure 3.

[3] Either (in two of the three classes) the solution of the problem is already common knowledge in the initial state, or the agents commonly know that the sum of the numbers is at least 7 [17].

In case we start counting at 0, or have an upper bound for the numbers, the tree may be no longer binary. When starting from 0, each *abc*-equivalence class with root $(2x, x, x)$ is extended with one more node, the new root, $(0, x, x)$. An agent who sees a 0, infers that his number must be the other seen number. As this always applies to two out of three agents, the root has just one child $(2x, x, x)$. In this case $\mathcal{T}^{\mathbb{N}}$ we also have a singleton *abc*-equivalence class namely with root $(0, 0, 0)$.

When using an upper bound max, the tree is cut at the depth where nodes (x, y, z) occur such that the sum of two of the arguments x, y, z exceeds max. This explains possible other unary branching in a tree namely near the leaves. For example, node $(2, 5, 7)$ in the *abc*-class with root $(0, 1, 1)$ has only a *b*-child $(2, 9, 7)$ and a *c*-parent $(2, 5, 3)$, and not an *a*-child, as $5 + 7 = 12 > \mathsf{max}$. All roots $(0, x, x)$ with $x > \frac{1}{2}\mathsf{max}$ are singleton *abc*-classes in $\mathcal{T}^{\mathsf{max}}$ (and in $\mathcal{T}^{\mathsf{max}+}$). In such models it is no longer the case that all equivalence classes are 'similar' as in $\mathcal{T}^{\mathbb{N}+}$. We now have that $[(0, x, x)]_{\sim_{abc}} \subseteq [(0, y, y)]_{\sim_{abc}}$ if $x \geq y$, modulo a multiplication factor $\frac{y}{x}$, and also that $\mathcal{T}^{\mathsf{max}(+)} \subseteq \mathcal{T}^{\mathsf{max}'(+)}$ if $\mathsf{max} \leq \mathsf{max}'$.

For an example we describe the epistemic model $\mathcal{T}^{10}$ in detail. This model is also used later to illustrate the results of ignorance announcements. To simplify our notations, a triple (i, j, k) is written as ijk where the number 10 is written as A, as usual in combinatorics and base-16 arithmetic. All equivalence classes have a root containing one 0. The *abc*-equivalence classes with a root where Anne has a 0 (i.e., form $0xx$) are represented by roots 000–0AA; and similar for $x0x$ and $xx0$. From those, 000 and 066–0AA are singleton. The class with root 055 contains one other triple, namely A55, indistinguishable from 055 by Anne; 044 has some more structure, etc. The most complex class, with root 011, is displayed in Figure 2, on the left. All other classes in $\mathcal{T}^{10}$ are therefore similar to a subtree of this $[011]_{\sim_{abc}}$.

3.1 Processing Announcements

Announcements result in model restrictions. For each agent n, a n-class is either singleton or has two elements. An ignorance announcement by agent n in this riddle corresponds to removal of all singleton n-classes from the epistemic model $\mathcal{T}$ for the riddle, so that the remaining n-classes then consist of two states. Such ignorance statements therefore *never* result in common knowledge of all three numbers: either all states are removed, or multiples of two remain, so that at least one agent is uncertain about its number. Knowledge statements "I know my number" also cannot result in common knowledge: such a last remaining state would have been eliminated as well, as it is a singleton. The singletons typically occur together as non-branching subtrees that include the root or a leaf of an *abc*-class.[4] An ignorance announcement may have very different effects on *abc*-classes that are the same modulo agent permutations. For example, given

[4] We conjecture that they *only* take that form; i.e., a sequence of announcements in $\mathcal{T}$ *cannot* result in a model containing an *abc*-tree that has a node with one child only, a sibling, and a descendant with two children.

abc-classes with roots 121, 112, and 211, the effect of Anne saying that she does not know her number in $\mathcal{T}^{\mathbb{N}+}$ *only* results in elimination of 211, as only the first *abc*-class contains an *a*-singleton. Given 211, Anne knows that she has number 2 (as 0 is excluded). Triple 112 she cannot distinguish from 312, and 121 not from 321. Thus one proceeds with all three announcements. See also Figure 3.

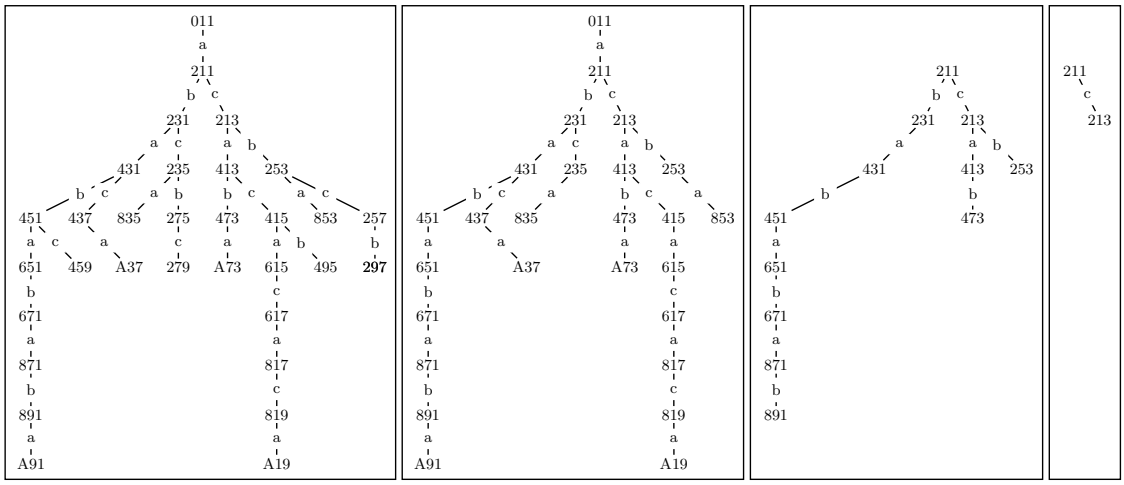

Fig. 2. Successive announcements in the *abc*-class with root 011 in model $\mathcal{T}^{10}$. The horizontal order of branches has no meaning. Symbol A represents 10.

If $\mathcal{T} \subseteq \mathcal{T}'$ and $\boldsymbol{\varphi}$ is a sequence of ignorance announcements executable in both models, then $\mathcal{T}|\boldsymbol{\varphi} \subseteq \mathcal{T}'|\boldsymbol{\varphi}$. In particular, if $\mathsf{max} \leq \mathsf{max}'$, then $\mathcal{T}^{\mathsf{max}}|\boldsymbol{\varphi} \subseteq \mathcal{T}^{\mathsf{max}'}|\boldsymbol{\varphi}$. For similar reasons, given a model $\mathcal{T}^{\mathsf{max}}$: if $x \leq y$, then $[0yy]_{\sim_{abc}}|\boldsymbol{\varphi} \subseteq [0xx]_{\sim_{abc}}|\boldsymbol{\varphi}$ (discounting a multiplication factor). Both properties facilitate systematic search for problem solutions.[5]

For the model $\mathcal{T}^{10}$, for the *abc*-class with root 011, the successive model restrictions by the three ignorance announcements are visualized in Figure 2. After those announcements, the triples 211 and 213 remain. This expresses that in case either of these is the actual number, Anne and Bill now know their numbers but Cath remains uncertain. But there are other equivalence classes to take into account, so this does not prove yet that Anne now *always* knows her number. We have now sufficient background to solve all versions of the riddle in quick succession.

[5] For arbitrary $M' \subseteq M$ and φ we do not have that $M'|\varphi \subseteq M|\varphi$. Given agents a, b and state variables p, q (in 10 p is true and q is false) consider the (two-state) model $M' =$ 11—a—10, which is a restriction of the (three state) model $M =$ 11—a—10—b—01. Consider $\varphi = K_b q \vee K_b \neg q$, for 'Bill knows whether q.' Then $M'|\varphi = M'$, whereas $M|\varphi$ is the singleton model consisting of state 11 wherein a and b have common knowledge of p and q. Therefore $M' \subseteq M$ but $M'|\varphi \not\subseteq M|\varphi$.

3.2 Solving the riddle

Anne knows $x = 50$, given $1 \leq x$ Three announcements cannot reduce a binary tree to a depth larger than three. We reduce the three classes with roots 011, 101, and 110, restricted to that depth, determine the nodes wherein Anne knows the numbers, and from which of those Anne's number divides 50. See Figure 3—note that triple 835 is indistinguishable for Anne from $(12, 7, 5)$, further down the tree. The unique triple wherein Anne's number divides 50 is 523. Therefore, the solution is that Bill has 20 and Cath has 30, as the triple $(50, 20, 30)$ remains after the three announcements in the *abc*-class with root $(10, 20, 10)$.

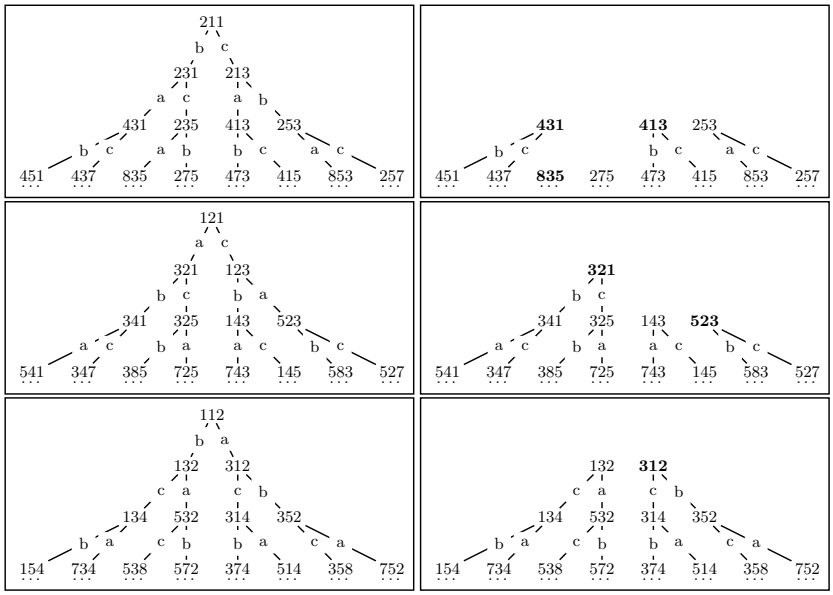

Fig. 3. The results of three ignorance announcements on the *abc*-classes of the model $\mathcal{T}^{\mathbb{N}+}$. The triples in bold are those where Anne knows her number.

Anne knows $x = 51$, given $1 \leq x$ Using Figure 3 again, we now determine the triples wherein Anne knows her number and it divides 51. This time, there is no unique such triple but both 321 and 312 satisfy the constraint. This means that, if Anne knows that she has 51, the other numbers must be 17 and 34 – but it is not known whether Bill has 17 and Cath 34, or the other way round. The original riddle could have been more restrictive: in the quoted version [1] it is *not* required to determine who holds which other number, but this can also be determined.

Wasn't the '51' version of the riddle for numbers $0 \leq x$, that is, including 0? Yes. But also in this case the other numbers must be 17 and 34. If we add root nodes 011, 101, and 110 to the *abc*-classes in Figure 3, the latter two result

in the same restrictions after the announcements, but the class with root 011 results in the top-left structure in the figure, as now only the b-announcement is informative, by eliminating root 011. And in triple 211, where Anne knows the numbers, 2 does not divide 51.

In the case of $\mathcal{T}^{\mathbb{N}+}$, the infinite number of x's wherein the other numbers can be determined after Anne announces that her number is x, are such that *only* one of $\{3, 4, 5\}$ divides x (excluding 1 and—unless in that set—x), and 8 does not divide x. Similarly, in the case of $\mathcal{T}^{\mathbb{N}}$, all x such that *only* one of $\{2, 3, 5\}$ divides x, and 4 does not divide x.

Anne knows the numbers, given $0 \leq x \leq 10$ The number 10 is in the range of max for which Anne always knows the numbers after the three announcements. We explain the solution for this case, and then proceed with the case of an arbitrary finite upper bound.

We have already seen, in Figure 2, that from *abc*-class with root 011 the triples 211 and 213 remain. Similar computations show that from the *abc*-classes with root 101 and 110 no triples remain. In other words, the announcements could not all three have been made (truthfully) if the number triple occurs in either of those two classes. Using the properties of inclusion for different *abc*-classes, we have now ruled out all classes of type $x0x$ and $xx0$ and only have to check other classes of type $0xx$. From class 022, the triples 242 and 246 remain after the three announcements. Therefore, whatever the numbers, Anne now knows her number. But the problem solver cannot determine what that number is (it may be 1, or it may be 2) and also cannot determine what the other numbers are, not even if it is also known what Anne's number is (if it is 1, the other numbers may be 2 and 1, or 2 and 3; and similarly if it is 2). Note that Bill also always knows the numbers after the three announcements, but not Cath.

Anne knows the numbers, given $0 \leq x \leq \mathsf{max}$ The further question was for which values of max, apparently including $\mathsf{max} = 10$, Anne always knows the numbers after the three announcements. The answer to this question is: $8 \leq \mathsf{max} \leq 13$. This means that if $\mathsf{max} = 7$, the three announcements cannot be made (without lying). And if $\mathsf{max} = 14$, it is not always the case that Anne knows her number: if Bill has 1 and Cath has 3, Anne cannot determine whether her number is 2 or 4; 213 and 413 are in that case the *only* two triples where Anne is still uncertain. If $\mathsf{max} > 11$, it is no longer the case that also Bill always knows his number. If $\mathsf{max} = 8$, and only for that value, the problem solver is also able to determine Anne's number: it is 2. The only remaining triples in this case are 211 and 213.

4 Verification of epistemic riddles

With epistemic model checkers we may verify properties of interpreted systems, knowledge-based protocols, and various other multi-agent systems. The model checkers MCK [18] and MCMAS [19] use the interpreted systems architecture, and exploration of the search space is based on ordered binary decision diagrams; other recent work includes [20]. The model checker DEMO [2] (written

```
module SUMXYZ
where
import DEMO
upb = 10
-- triples (x,y,z) with x,y,z <= upb, x = y+z or y = x+z or z = x+y
triplesx  = [(x,y,z)|x<-[0..upb], y<-[0..upb], z<-[0..upb], x==y+z]
triplesy  = [(x,y,z)|x<-[0..upb], y<-[0..upb], z<-[0..upb], y==x+z]
triplesz  = [(x,y,z)|x<-[0..upb], y<-[0..upb], z<-[0..upb], z==x+y]
triples = triplesx ++ triplesy ++ triplesz
-- associating states with number triples
numtriples = llength(triples)
llength [] =0
llength (x:xs) = 1+ llength xs
itriples = zip [0..numtriples-1] triples
-- initial multi-pointed epistemic model
three :: EpistM
three = (Pmod [0..numtriples-1] val acc [0..numtriples-1])
 where
 val = [(w,[P x, Q y, R z]) | (w,(x,y,z))<- itriples]
 acc = [(a,w,v)| (w,(x1,y1,z1))<-itriples, (v,(x2,y2,z2))<-itriples, y1==y2, z1==z2]++
       [(b,w,v)| (w,(x1,y1,z1))<-itriples, (v,(x2,y2,z2))<-itriples, x1==x2, z1==z2]++
       [(c,w,v)| (w,(x1,y1,z1))<-itriples, (v,(x2,y2,z2))<-itriples, x1==x2, y1==y2]
-- agents a,b,c say (respectively): I do not know my number x,y,z
fagxnot = Conj [(Disj[Neg (Prop (P x)), Neg (K a (Prop (P x))) ])|x <-[0..upb]]
aagxnot = public (fagxnot)
fagynot = Conj [(Disj[Neg (Prop (Q y)), Neg (K b (Prop (Q y))) ])|y <-[0..upb]]
aagynot = public (fagynot)
fagznot = Conj [(Disj[Neg (Prop (R z)), Neg (K c (Prop (R z))) ])|z <-[0..upb]]
aagznot = public (fagznot)
-- model restriction resulting from the three announcements
solution = showM (upds three [aagxnot, aagynot, aagznot])
```

Fig. 4. The DEMO program `SUMXYZ.hs`

in Haskell) implements the 'action model' logic of [8], wherein also more complex actions than public announcements can be described. Public announcements correspond to semantic objects called (singleton) action models. With the exception of dynamic modal operators, the syntax of DEMO is very much like public announcement logic. The inductive constructs `Top | Prop Prop | Neg Form | Conj [Form] | Disj [Form] | K Agent Form | CK [Agent] Form` correspond to counterparts $\top|p|\neg\varphi|\varphi \wedge \ldots \wedge \varphi|\varphi \vee \ldots \vee \varphi|K_n\varphi|C_B\varphi$.

The action model for a public announcement is created by a function `public` with the announced formula as its argument. An action model is a datatype `PoAM`, for pointed action model. The update operation is specified as a function `upd :: EpistM -> PoAM -> EpistM`. This corresponds to computing $(M|\varphi, s)$ from a given (M, s) and an announcement φ. The function `upds :: EpistM -> [PoAM] -> EpistM` performs a sequence of updates.

The DEMO program `SUMXYZ.hs`, displayed in Figure 4, implements the 'what is my number' problem for upper bound `max` $= 10$. The list `triples` corresponds to the set of possible triples (x, y, z) for the given bound 10. The next part of the program constructs the domain based on that list: this merely means that each member of the list must be associated with a state name. State names must be consecutive numbers, counting from 0. The initial model $\mathcal{T}^{10}$ is then represented as `three` in the program. The expression `(Pmod [0..numtriples-1] val acc`

`[0..numtriples-1])` defines `three` as an epistemic model (`Pmod`), with domain `[0..numtriples-1]`, valuation `val`, a set (list) of accessibility relations `acc` (and `[0..numpairs-1]` points—left unexplained here). In `val` we find for example `(67,[p6, q8, r2])` which says that state number 67 corresponds to triple $(6, 8, 2)$. Given `(43,[p10, q8, r2])` we now find `(a,43,67)` in `acc`.

Anne's announcement that she does not know her number is represented as the action model `aagxnot` constructed from the announcement formula `fagxnot` by the function `public`. The formula `fagxnot` specifies (disjunctively) that whatever x is—`x <-[0..upb]`—if Anne has it she does not know it—`(Disj[Neg (Prop (P x)), Neg (K a (Prop (P x))) ])` corresponds to $\neg x_a \vee \neg K_a x_a$. The final line in the program asks to display the results of the three ignorance announcements.

5 Design of epistemic puzzles

A puzzle such as 'what is my number' is 'fun' if it is not too easy, and not too complex. The complexity of Kripke models is typically a function of the number of possible worlds and the number of pairs in the relation. This does not take into account other available structure. For epistemic models, the number of equivalence classes in the partition seems more appropriate as a measure than the number of pairs in the corresponding equivalence relation, e.g., the universal relation contains the maximum number of pairs, but does not partition the domain—clearly it is less and not more complex than most other partitions even though these consist of fewer pairs. There is yet more structure to take into account: how do we represent that in 'what is my number' it is sufficient to consider only one of many equivalence classes? The focus of such investigations are somewhat different from those on the complexity of (real) imperfect information game tree search [21, 22], although there is overlap with epistemic games wherein both the complexity of the game state and of the game tree play a part. A relevant observation for the link with 'what is my number' is that in the card game of Cluedo the complexity of the epistemic models as measured in terms of states and accessibility does *not* increase—though *barely* so: the factor is $\frac{9}{21}$ [7]. Insight into the relation between game playability and game complexity may help to design new, exciting, epistemic puzzles.

6 Conclusions

We presented a new epistemic riddle, defined some versions, and solved it with the use of public announcement logic and epistemic model checking. Crucial in the analysis was to model the riddle as an interpreted system, and to focus on the description of the background knowledge, i.e., *abc*-equivalence classes of the epistemic model. We also specified and verified the riddle in the model checker DEMO. Logic puzzle design could benefit from similar future efforts, including results from complexity analysis.

References

1. Liu, A.: Problem section: Problem 182. Math Horizons **11** (2004) 324
2. van Eijck, J.: Dynamic epistemic modelling. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam (2004) CWI Report SEN-E0424.
3. Moses, Y.O., Dolev, D., Halpern, J.Y.: Cheating husbands and other stories: a case study in knowledge, action, and communication. Distributed Computing **1**(3) (1986) 167–176
4. Freudenthal, H.: (formulation of the sum-and-product problem). Nieuw Archief voor Wiskunde **3(17)** (1969) 152
5. McCarthy, J.: Formalization of two puzzles involving knowledge. In Lifschitz, V., ed.: Formalizing Common Sense : Papers by John McCarthy. Ablex Series in Artificial Intelligence. Ablex Publishing Corporation, Norwood, N.J. (1990).
6. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about Knowledge. MIT Press, Cambridge MA (1995)
7. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic. Manuscript (2006)
8. Baltag, A., Moss, L., Solecki, S.: The logic of common knowledge, public announcements, and private suspicions. In Gilboa, I., ed.: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 98). (1998) 43–56
9. Plaza, J.: Logics of public communications. In Emrich, M., Pfeifer, M., Hadzikadic, M., Ras, Z., eds.: Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems. (1989) 201–216
10. Gerbrandy, J.: Bisimulations on Planet Kripke. PhD thesis, University of Amsterdam (1999) ILLC Dissertation Series DS-1999-01.
11. Lutz, C.: Complexity and succinctness of public announcement logic. To appear in the proceedings of AAMAS 06 (2006)
12. Moore, G.: A reply to my critics. In Schilpp, P., ed.: The Philosophy of G.E. Moore. Northwestern University, Evanston IL (1942) 535–677
13. Hintikka, J.: Knowledge and Belief. Cornell University Press, Ithaca, NY (1962)
14. van Ditmarsch, H.: The logic of pit. Knowledge, Rationality & Action (Synthese) **149(2)** (2006) 343–375
15. Barwise, J., Moss, L.: Vicious Circles. CSLI Publications, Stanford (1996)
16. van Benthem, J.: Dynamic odds and ends. Technical report, ILLC, University of Amsterdam (1998) Report ML-1998-08.
17. Panti, G.: Solution of a number theoretic problem involving knowledge. International Journal of Foundations of Computer Science **2(4)** (1991) 419–424
18. Gammie, P., van der Meyden, R.: MCK: Model checking the logic of knowledge. In Alur, R., Peled, D., eds.: Proceedings of CAV 2004, Springer (2004) 479–483
19. Raimondi, F., Lomuscio, A.: Verification of multiagent systems via ordered binary decision diagrams: An algorithm and its implementation. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 04), IEEE Computer Society (2004) 630–637
20. Su, K.: Model checking temporal logics of knowledge in distributed systems. In McGuinness, D.L., Ferguson, G., eds.: Proceedings AAAI 04, (2004) 98–103
21. Sevenster, M.: Battleships as decision problem. ICGA Journal **27(3)** (2006) 142–149
22. Parker, A., Nau, D., Subrahmanian, V.: Game-tree search with combinatorially large belief states. In Kaelbling, L.P., Saffiotti, A., eds.: Proceedings of IJCAI-05, (2005) 254–259