Sleeping Beauty Reconsidered: Conditioning and Reflection in Asynchronous Systems

Joseph Y. Halpern*
Cornell University
Computer Science Department
Ithaca, NY 14853
halpern@cs.cornell.edu
http://www.cs.cornell.edu/home/halpern

Abstract

A careful analysis of conditioning in the *Sleeping Beauty* problem is done, using the formal model for reasoning about knowledge and probability developed by Halpern and Tuttle. While the Sleeping Beauty problem has been viewed as revealing problems with conditioning in the presence of imperfect recall, the analysis done here reveals that the problems are not so much due to imperfect recall as to *asynchrony*. The implications of this analysis for van Fraassen's *Reflection Principle* and Savage's *Sure-Thing Principle* are considered.

1 Introduction

The standard approach to updating beliefs in the probability literature is by conditioning. But it turns out that conditioning is somewhat problematic if agents have *imperfect recall*. In the economics community this issue was brought to the fore by the work of Piccione and Rubinstein [1997] (to which was dedicated a special issue of the journal *Games and Economic Behavior*). There has also been a recent surge of interest in the topic in the philosophy community, inspired by a re-examination by Elga [2000] of one of the problems considered by Piccione and Rubinstein, the so-called *Sleeping Beauty problem*.[1] (Some recent work on the problem includes [Arntzenius 2003; Dorr 2002; Lewis 2001; Monton 2002].)

I take the Sleeping Beauty problem as my point of departure in this paper too. I argue that the problems in updating arise not just with imperfect recall, but also in *asynchronous* systems, where agents do not know exactly what time it is, or do not share a global clock. Since both human and computer agents are resource bounded and forget, imperfect recall is the norm, rather than an unusual special case. Moreover, there are many applications where it is unreasonable to assume the existence of a global clock. Thus, understanding how to do updating in the presence of asynchrony and imperfect recall is a significant issue in knowledge representation.

The Sleeping Beauty problem is described by Elga as follows:

> Some researchers are going to put you to sleep. During the two days that your sleep will last, they will briefly wake you up either once or twice, depending on the toss of a fair coin (heads: once; tails: twice). After each waking, they will put you back to sleep with a drug that makes you forget that waking. When you are first awakened, to what degree ought you believe that the outcome of the coin toss is heads?

Elga argues that there are two plausible answers. The first is that it is $1/2$. After all, it was $1/2$ before you were put to sleep and you knew all along that you would be woken up. Thus, it should still be $1/2$ when you are actually woken up. The second is that it is $1/3$. Clearly if this experiment is carried out repeatedly, then in the long run, at roughly one third of the times that you are woken up, you are in a trial in which the coin lands heads.

Elga goes on to give another argument for $1/3$, which he argues is in fact the correct answer. Suppose you are put to sleep on Sunday, so that you are first woken on Monday and then possibly again on Tuesday if the coin lands tails. Thus, when you are woken up, there are three events that you consider possible:

- e_1: it is Monday and the coin landed heads;
- e_2: it is Monday and the coin landed tails;
- e_3: it is Tuesday and the coin landed tails.

Here is Elga's argument: Clearly if, after waking up, you learn that it is Monday, you should consider e_1 and e_2 equally likely. Since, conditional on learning that it is Monday, you consider e_1 and e_2 equally likely, you should consider them equally likely unconditionally. Now, conditional on the coin landing tails, it also seems reasonable that e_2 and e_3 should be equally likely; after all, you have no reason to think Monday is any more or less likely that Tuesday if the

*Work supported in part by NSF under grant CTC-0208535, by ONR under grants N00014-00-1-03-41 and N00014-01-10-511, by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the ONR under grant N00014-01-1-0795, and by AFOSR under grant F49620-02-1-0101.

[1] So named by Robert Stalnaker.

coin landed tails. Thus, unconditionally, e_2 and e_3 should be equally likely. But the only way for e_1, e_2, and e_3 to be equally likely is for them all to have probability $1/3$. So heads should have probability $1/3$.

Note that if the story is changed so that (1) heads has probability .99 and tails has probability .01, (2) you are woken up once if the coin lands heads, and (3) you are woken up 9900 times if the coin lands tails, then Elga's argument would say that the probability of tails is .99. Thus, although you know you will be woken up whether the coin lands heads or tails, and you are initially almost certain that the coin will land heads, when you are woken up (according to Elga's analysis) you are almost certain that the coin landed tails!

To analyze these arguments, I use a formal model for reasoning about knowledge and probability that Mark Tuttle and I developed [Halpern and Tuttle 1993] (HT from now on), which in turn is based on the "runs and systems" framework for reasoning about knowledge in computing systems, introduced in [Halpern and Fagin 1989] (see [Fagin, Halpern, Moses, and Vardi 1995] for motivation and discussion). Using this model, I argue that Elga's argument is not as compelling as it may seem. The analysis also reveals that, despite the focus of the economics community on imperfect recall, the real problem is not imperfect recall, but asynchrony: the fact that Sleeping Beauty does not know exactly what time it is.

Finally, I consider other arguments and desiderata traditionally used to justify probabilistic conditioning, such as frequency arguments, betting arguments, van Fraassen's [1984] *Reflection Principle*, and Savage's [1954] *Sure-Thing Principle*. I show that our intuitions for these arguments are intimately bound up with assumptions such as synchrony and perfect recall.

The rest of this paper is organized as follows. In the next section I review the basic runs and systems framework. In Section 3, I describe the HT approach to adding probability to the framework when the system is synchronous, and then consider two generalizations to the case that the system is asynchronous. In the Sleeping Beauty problem, these two generalizations give the two solutions discussed by Elga. In Section 4, I consider other arguments and desiderata.

2 The framework

2.1 The basic multi-agent systems framework

In this section, we briefly review the multiagent systems framework; see [Fagin, Halpern, Moses, and Vardi 1995] for more details.

A *multiagent system* consists of n agents interacting over time. At each point in time, each agent is in some *local state*. Intuitively, an agent's local state encapsulates all the information to which the agent has access. For example, in a poker game, a player's state might consist of the cards he currently holds, the bets made by the other players, any other cards he has seen, and any information he may have about the strategies of the other players (e.g., Bob may know that Alice likes to bluff, while Charlie tends to bet conservatively). In the Sleeping Beauty problem, we can assume that the agent has local states corresponding to "just woken up" and "sleeping". We could also include local states corresponding to "just before the experiment" and "just after the experiment".

Besides the agents, it is also conceptually useful to have an "environment" (or "nature") whose state can be thought of as encoding everything relevant to the description of the system that may not be included in the agents' local states. In many ways, the environment can be viewed as just another agent. For example, in the Sleeping Beauty problem, the environment state can encode the actual day of the week and the outcome of the coin toss. We can view the whole system as being in some *global state*, a tuple consisting of the local state of each agent and the state of the environment. Thus, a global state has the form $(s_e, s_1, \ldots, s_n)$, where s_e is the state of the environment and s_i is agent i's state, for $i = 1, \ldots, n$.

A global state describes the system at a given point in time. But a system is not a static entity. It is constantly changing over time. A *run* captures the dynamic aspects of a system. Intuitively, a run is a complete description of one possible way in which the system's state can evolve over time. Formally, a run is a function from time to global states. For definiteness, I take time to range over the natural numbers. Thus, $r(0)$ describes the initial global state of the system in a possible execution, $r(1)$ describes the next global state, and so on. A pair (r, m) consisting of a run r and time m is called a *point*. If $r(m) = (s_e, s_1, \ldots, s_n)$, then define $r_e(m) = s_e$ and $r_i(m) = s_i$, $i = 1, \ldots, n$; thus, $r_i(m)$ is agent i's local state at the point (r, m) and $r_e(m)$ is the environment's state at (r, m). I write $(r, m) \sim_i (r', m')$ if agent i has the same local state at both (r, m) and (r', m'), that is, if $r_i(m) = r'_i(m')$. Let $\mathcal{K}_i(r, m) = \{(r', m') : (r, m) \sim_i (r', m')\}$. Intuitively, $\mathcal{K}_i(r, m)$ is the set of points that i considers possible at (r, m). Sets of the form $\mathcal{K}_i(r, m)$ are sometimes called *information sets*.

In general, there are many possible executions of a system: there could be a number of possible initial states and many things that could happen from each initial state. For example, in a draw poker game, the initial global states could describe the possible deals of the hand by having player i's local state describe the cards held by player i. For each fixed deal of the cards, there may still be many possible betting sequences, and thus many runs. Formally, a *system* is a nonempty set of runs. Intuitively, these runs describe all the possible sequences of events that could occur in the system. Thus, I am essentially identifying a system with its possible behaviors.

The obvious system for the Sleeping Beauty problem consists of two runs, the first corresponding to the coin landing heads, and the second corresponding to the coin landing tails. However, there is some flexibility in how we model the global states. Here is one way: At time 0, a coin is tossed; the environment state encodes the outcome. At time 1, the agent is asleep (and thus is in a "sleeping" state). At time 2, the agent is woken up. If the coin lands tails, then at time 3, the agent is back asleep, and at time 4, is woken up again. Note that I have assumed here that time in both of these runs ranges from 0 to 5. Nothing would change if I allowed runs to have infinite length or a different (but sufficiently long) finite length.

Alternatively, we might decide that it is not important to model the time that the agent is sleeping; all that matters is the point just before the agent is put to sleep and the points where the agent is awake. Assume that Sleeping Beauty is in state b before the experiment starts, in state a after the experiment is over, and in state w when woken up. This leads to a model with two runs r_1 and r_2, where the first three global states in r_1 are (H, b), (H, w), and (H, a), and the first four global states in r_2 are (T, b), (T, w), (T, w), (T, a). Let $\mathcal{R}_1$ be the system consisting of the runs r_1 and r_2. This system is shown in Figure 1 (where only the first three global states in each run are shown). The three points where the agent's local state is w, namely, $(r_1, 1)$, $(r_2, 1)$, and $(r_2, 2)$, form what is traditionally called in game theory an *information set*. These are the three points that the agent considers possible when she is woken up. For definiteness, I use $\mathcal{R}_1$ in my analysis of Sleeping Beauty.

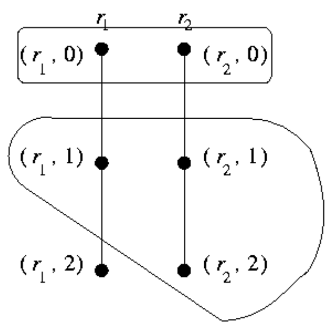

Figure 1: The Sleeping Beauty problem, captured using $\mathcal{R}_1$.

Notice that $\mathcal{R}_1$ is also compatible with a somewhat different story. Suppose that the agent is not aware of time passing. At time 0 the coin is tossed, and the agent knows this. If the coin lands heads, only one round passes before the agent is told that the experiment is over; if the coin lands tails, she is told after two rounds. Since the agent is not aware of time passing, her local state is the same at the points $(r_1, 2)$, $(r_2, 1)$, and $(r_2, 2)$. The same analysis should apply to the question of what the probability of heads is at the information set. The key point is that here the agent does not forget; she is simply unaware of the time passing.

Various other models are possible:

- We could assume (as Elga does at one point) that the coin toss happens only after the agent is woken up the first time. Very little would change, except that the environment state would be $\emptyset$ (or some other way of denoting that the coin hasn't been tossed) in the first two global states of both runs. Call the two resulting runs r_1' and r_2'.
- All this assumes that the agent knows when the coin is going to be tossed. If the agent doesn't know this, then we can consider the system consisting of the four runs r_1, r_1', r_2, r_2'.
- Suppose that we now want to allow for the possibility that, upon wakening, the agent learns that it is Monday (as in Elga's argument). To do this, the system must include runs where the agent actually learns that it is Monday. For example, we can consider the system $\mathcal{R}_2 = (r_1, r_2, r_1^*, r_2^*)$, where r_i^* is the same as r_i except that on Monday, the agent's local state encodes that it is Monday. Thus, the sequence of global states in r_1^* is (H, b), (H, M), (H, a), and the sequence in r_2^* is (T, b), (T, M), (T, w). $\mathcal{R}_2$ is described in Figure 2. Note that on Tuesday in r_2^*, the agent forgets whether she was woken up on Monday. She is in the same local state on Tuesday in r_2^* as she is on both Monday and Tuesday in r_2.

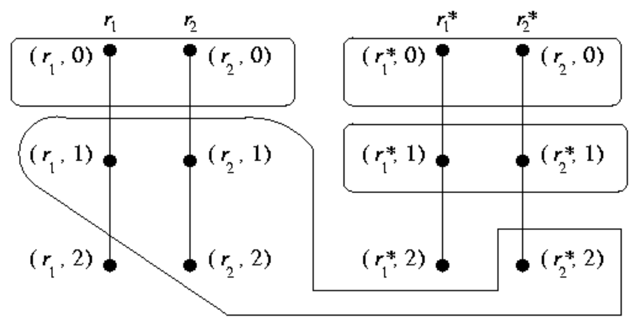

Figure 2: An alternate representation of the Sleeping Beauty problem, using $\mathcal{R}_2$.

Yet other representations of the Sleeping Beauty problem are also possible. The point that I want to emphasize here is that the framework has the resources to capture important distinctions about when the coin is tossed and what agents know.

2.2 Synchrony and perfect recall

One advantage of the runs and systems framework is that it can be used to easily model a number of important assumptions. I focus on two of them here: *synchrony*, the assumption that agents know the time, and *perfect recall*, the assumption that agents do not forget [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Vardi 1989]

Formally, a system $\mathcal{R}$ is *synchronous for agent* i if for all points (r, m) and (r', m') in $\mathcal{R}$, if $(r, m) \sim_i (r', m')$, then $m = m'$. Thus, if $\mathcal{R}$ is synchronous for agent i, then at time m, agent i knows that it is time m, because it is time m at all the points he considers possible. $\mathcal{R}$ is *synchronous* if it is synchronous for all agents. Note that the systems that model the Sleeping Beauty problem are not synchronous. When Sleeping Beauty is woken up on Monday, she does not know what day it is.

Consider the following example of a synchronous system, taken from [Halpern 2003]:

Example 2.1: Suppose that Alice tosses two coins and sees how the coins land. Bob learns how the first coin landed after the second coin is tossed, but does not learn the outcome of the second coin toss. How should this be represented as a multi-agent system? The first step is to decide what the local states look like. There is no "right" way of modeling the local states. What I am about to describe is one reasonable way of doing it, but clearly there are others.

The environment state will be used to model what actually happens. At time 0, it is $\langle\,\rangle$, the empty sequence, indicating that nothing has yet happened. At time 1, it is either $\langle H \rangle$ or $\langle T \rangle$, depending on the outcome of the first coin toss. At time 2, it is either $\langle H, H \rangle$, $\langle H, T \rangle$, $\langle T, H \rangle$, or $\langle T, T \rangle$, depending on the outcome of both coin tosses. Note that the environment state is characterized by the values of two random variables, describing the outcome of each coin toss. Since Alice knows the outcome of the coin tosses, I take Alice's local state to be the same as the environment state at all times.

What about Bob's local state? After the first coin is tossed, Bob still knows nothing; he learns the outcome of the first coin toss after the second coin is tossed. The first thought might then be to take his local states to have the form $\langle\,\rangle$ at time 0 and time 1 (since he does not know the outcome of the first coin toss at time 1) and either $\langle H \rangle$ or $\langle T \rangle$ at time 2. This choice would not make the system synchronous, since Bob would not be able to distinguish time 0 from time 1. If Bob is aware of the passage of time, then at time 1, Bob's state must somehow encode the fact that the time is 1. I do this by taking Bob's state at time 1 to be $\langle \textit{tick} \rangle$, to denote that one time tick has passed. (Other ways of encoding the time are, of course, also possible.) Note that the time is already implicitly encoded in Alice's state: the time is 1 if and only if her state is either $\langle H \rangle$ or $\langle T \rangle$.

Under this representation of global states, there are seven possible global states:

- $(\langle\,\rangle, \langle\,\rangle, \langle\,\rangle)$, the initial state,
- two time-1 states of the form $(\langle X_1 \rangle, \langle X_1 \rangle, \langle \textit{tick} \rangle)$, for $X_1 \in \{H, T\}$,
- four time-2 states of the form $(\langle X_1, X_2 \rangle, \langle X_1, X_2 \rangle, \langle \textit{tick}, X_1 \rangle)$, for $X_1, X_2 \in \{H, T\}$.

In this simple case, the environment state determines the global state (and is identical to Alice's state), but this is not always so.

The system describing this situation has four runs, $r^1, \ldots, r^4$, one for each of the time-2 global states. The runs are perhaps best thought of as being the branches of the computation tree described in Figure 3.

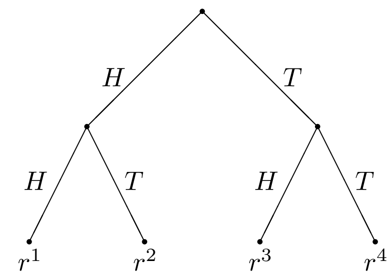

Figure 3: Tossing two coins.

Modeling perfect recall in the systems framework is not too difficult, although it requires a little care. In this framework, an agent's knowledge is determined by his local state. Intuitively, an agent has perfect recall if his local state is always "growing", by adding the new information he acquires over time. This is essentially how the local states were modeled in Example 2.1. In general, local states are not required to grow in this sense, quite intentionally. It is quite possible that information encoded in $r_i(m)$—i's local state at time m in run r—no longer appears in $r_i(m+1)$. Intuitively, this means that agent i has lost or "forgotten" this information. There is a good reason not to make this requirement. There are often scenarios of interest where it is important to model the fact that certain information is discarded. In practice, for example, an agent may simply not have enough memory capacity to remember everything he has learned. Nevertheless, although perfect recall is a strong assumption, there are many instances where it is natural to model agents as if they do not forget.

Intuitively, an agent with perfect recall should be able to reconstruct his complete local history from his current local state. To capture this intuition, let *agent i's local-state sequence at the point* (r, m) be the sequence of local states that she has gone through in run r up to time m, without consecutive repetitions. Thus, if from time 0 through time 4 in run r agent i has gone through the sequence $\langle s_i, s_i, s_i', s_i, s_i \rangle$ of local states, where $s_i \neq s_i'$, then her local-state sequence at $(r, 4)$ is $\langle s_i, s_i', s_i \rangle$. Agent i's local-state sequence at a point (r, m) essentially describes what has happened in the run up to time m, from i's point of view. Omitting consecutive repetitions is intended to capture situations where the agent has perfect recall but is not aware of time passing, so she cannot distinguish a run where she stays in a given state s for three rounds from one where she stays in s for only one round.

An agent has perfect recall if her current local state encodes her whole local-state sequence. More formally, *agent i has perfect recall in system $\mathcal{R}$* if, at all points (r, m) and (r', m') in $\mathcal{R}$, if $(r, m) \sim_i (r', m')$, then agent i has the same local-state sequence at both (r, m) and (r', m'). Thus, agent i has perfect recall if she "remembers" her local-state sequence at all times.[2] In a system with perfect recall, $r_i(m)$ encodes i's local-state sequence in that, at all points where i's local state is $r_i(m)$, she has the same local-state sequence. A system where agent i has perfect recall is shown in Figure 4.

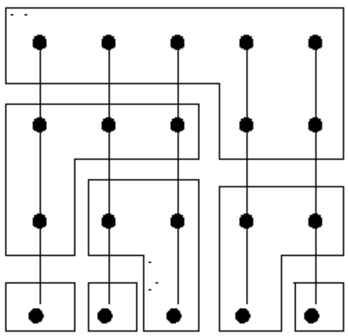

Figure 4: An asynchronous system where agent i has perfect recall.

The combination of synchrony and perfect recall leads to particularly pleasant properties. It is easy to see that if $\mathcal{R}$ is a synchronous system with perfect recall and $(r, m+1) \sim_i (r', m+1)$, then $(r, m) \sim_i (r', m)$. That is, if agent i considers run r' possible at the point $(r, m+1)$, then i must also consider run r' possible at the point (r, m). (Proof: since the system is synchronous and i has perfect recall, i's local state must be different at each point in r. For if i's local state were the same at two points (r, k) and (r, k') for $k \neq k'$, then agent i would not know that it was time k at the point (r, k). Thus, at the points $(r, m+1)$, i's local-state sequence must have length $m+1$. Since $(r, m+1) \sim_i (r', m+1)$, i has the same local-state sequence at $(r, m+1)$ and $(r', m+1)$. Thus, i must also have the same local-state sequence at the points (r, m) and (r', m), since i's local-state sequence at these points is just the prefix of i's local-state sequence at $(r, m+1)$ of length m. It is then immediate that $(r, m) \sim_i (r', m)$.) Thus, in a synchronous system with perfect recall, agent i's information set refines over time, as shown in Figure 5.

[2] This definition of perfect recall is not quite the same as that used in the game theory literature, where agents must explicitly recall the actions taken (see [Halpern 1997] for a discussion of the issues), but the difference between the two notions is not relevant here. In particular, according to both definitions, the agent has perfect recall in the "game" described by Figure 1.

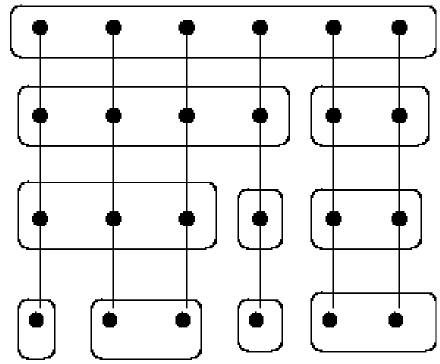

Figure 5: A synchronous system with perfect recall.

Note that whether the agent has perfect recall in the Sleeping Beauty problem depends in part on how we model the problem. In the system $\mathcal{R}_1$ she does; in $\mathcal{R}_2$ she does not. For example, at the point $(r_2^*, 2)$ in $\mathcal{R}_2$, where her local state is (T, w), she has forgotten that she was woken up at time 1 (because she cannot distinguish $(r_2, 2)$ from $(r_2^*, 2)$). (It may seem strange that the agent has perfect recall in $\mathcal{R}_1$, but that is because in $\mathcal{R}_1$, the time that the agent is asleep is not actually modeled. It happens "between the points". If we explicitly include local states where the agent is asleep, then the agent would not have perfect recall in the resulting model. The second interpretation of $\mathcal{R}_1$, where the agent is unaware of time passing, is perhaps more compatible with perfect recall. I use $\mathcal{R}_1$ here so as to stress that perfect recall is not really the issue in the Sleeping Beauty problem; it is the asynchrony.)

3 Adding probability

To add probability to the framework, I start by assuming a probability on the set of runs in a system. Intuitively, this should be thought of as the agents' common probability. It is not necessary to assume that the agents all have the same probability on runs; different agents may have use probability measures. Moreover, it is not necessary to assume that the probability is placed on the whole set of runs. There are many cases where it is convenient to partition the set of runs and put a separate probability measure on each cell in the partition (see [Halpern 2003] for a discussion of these issues). However, to analyze the Sleeping Beauty problem, it suffices to have a single probability on the runs. A *probabilistic system* is a pair $(\mathcal{R}, \mathrm{Pr})$, where $\mathcal{R}$ is a system (a set of runs) and Pr is a probability on $\mathcal{R}$. (For simplicity, I assume that $\mathcal{R}$ is finite and that all subsets of $\mathcal{R}$ are measurable.) In the case of the Sleeping Beauty problem, the probability on $\mathcal{R}_1$ is immediate from the description of the problem: each of r_1 and r_2 should get probability $1/2$. To determine a probability on the runs of $\mathcal{R}_2$, we need to decide how likely it is that the agent will discover that it is actually Monday. Suppose that probability is α. In that case, r_1 and r_2 both get probability $(1-\alpha)/2$, while r_1^* and r_2^* both get probability $\alpha/2$.

Unfortunately, the probability on runs is not enough for the agent to answer questions like "What is the probability that heads was tossed?" if she is asked this question at the point $(r_1, 1)$ when she is woken up in $\mathcal{R}_1$, for example. At this point she considers three points possible: $(r_1, 1)$, $(r_2, 1)$, and $(r_2, 2)$, the three points where she is woken up. She needs to put a probability on this space of three points to answer the question. Obviously, the probability on the points should be related to the probability on runs. But how?

3.1 The synchronous case

Tuttle and I suggested a relatively straightforward way of going from a probability on runs to a probability on points in synchronous systems. For all times m, the probability $\Pr$ on $\mathcal{R}$, the set of runs, can be used to put a probability $\Pr^m$ on the points in $\mathcal{R}^m = \{(r, m) : r \in \mathcal{R}\}$: simply take $\Pr^m(r, m) = \Pr(r)$. Thus, the probability of the point (r, m) is just the probability of the run r. Clearly, $\Pr^m$ is a well-defined probability on the set of time-m points. Since $\mathcal{R}$ is synchronous, at the point (r, m), agent i considers possible only time-m points. That is, all the points in $\mathcal{K}_i(r, m) = \{(r', m') : (r, m) \sim_i (r', m')\}$ are actually time-m points. Since, at the point (r, m), the agent considers possible only the points in $\mathcal{K}_i(r, m)$, it seems reasonable to take the agent's probability at the point (r, m) to the result of conditioning $\Pr^m$ on $\mathcal{K}_i(r, m)$, provided that $\Pr^m(\mathcal{K}_i(r, m)) > 0$, which, for simplicity, I assume here. Taking $\Pr_{(r,m,i)}$ to denote agent i's probability at the point (r, m), this suggests that $\Pr_{(r,m,i)}(r', m) = \Pr^m((r', m) \mid \mathcal{K}_i(r, m))$.

To see how this works, consider the system of Example 2.1. Suppose that the first coin has bias $2/3$, the second coin is fair, and the coin tosses are independent, as shown in Figure 6. Note that, in Figure 6, the edges coming out of each node are labeled with a probability, which is intuitively the probability of taking that transition. Of course, the probabilities labeling the edges coming out of any fixed node must sum to 1, since some transition must be taken. For example, the edges coming out of the root have probability $2/3$ and $1/3$. Since the transitions in this case (i.e., the coin tosses) are assumed to be independent, it is easy to compute the probability of each run. For example, the probability of run r^1 is $2/3 \times 1/2 = 1/3$; this represents the probability of getting two heads.

3.2 The general case

The question now is how the agents should ascribe probabilities in arbitrary (not necessarily synchronous) system, such as that of the Sleeping Beauty problem. The approach

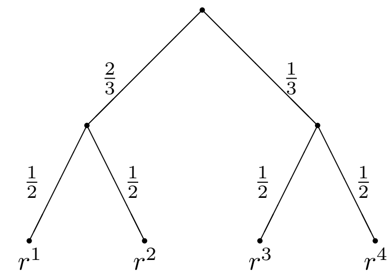

Figure 6: Tossing two coins, with probabilities.

suggested above does not immediately extend to the asynchronous case. In the asynchronous case, the points in $\mathcal{K}_i(r, m)$ are not in general all time-m points, so it does not make sense to condition on $\Pr^m$ on $\mathcal{K}_i(r, m)$. (Of course, it would be possible to condition on the time-m points in $\mathcal{K}_i(r, m)$, but it is easy to give examples showing that doing this gives rather nonintuitive results.)

I discuss two reasonable candidates for ascribing probability in the asynchronous case here, which are generalizations of the two approaches that Elga considers. I first consider these approaches in the context of the Sleeping Beauty problem, and then give the general formalization.

Consider the system described in Figure 1, but now suppose that the probability of r_1 is α and the probability of r_2 is $1-\alpha$. (In the original Sleeping Beauty problem, $\alpha = 1/2$.) It seems reasonable that at the points $(r_1, 0)$ and $(r_2, 0)$, the agent ascribes probability α to $(r_1, 0)$ and $1-\alpha$ to $(r_2, 0)$, using the HT approach for the synchronous case. What about at each of the points $(r_1, 1)$, $(r_2, 1)$, and $(r_2, 2)$? One approach (which I henceforth call the *HT approach*, since it was advocated in HT), is to say that the probability α of run r_1 is projected to the point $(r_1, 1)$, while the probability $1-\alpha$ of r_2 is projected to $(r_2, 1)$ and $(r_2, 2)$. How should the probability be split over these two points? Note that splitting the probability essentially amounts to deciding the relative probability of being at time 1 and time 2. Nothing in the problem description gives us any indication of how to determine this. HT avoid making this determination by making the singleton sets $\{(r_2, 1)\}$ and $\{(r_2, 2)\}$ nonmeasurable. Since they are not in the domain of the probability measure, there is no need to give them a probability. The only measurable sets in this space would then be $\emptyset$, $\{(r_1, 1)\}$, $\{(r_2, 1), (r_2, 2)\}$, and $\{(r_1, 1), (r_2, 1), (r_2, 2)\}$, which get probability 0, α, $1-\alpha$, and 1, respectively. An alternative is to take times 1 and 2 to be equally likely, in which case the probability of the set $\{((r_2, 1), (r_2, 2)\}$ is split over $(r_2, 1)$ and $(r_2, 2)$, and they each get probability $(1-\alpha)/2$. When $\alpha = 1/2$, this gives Elga's first solution. Although it is reasonable to assume that times 1 and 2 are equally likely, the technical results that I prove hold no matter how the probability is split between times 1 and 2.

The second approach, which I call the *Elga approach* (since it turns out to generalize what Elga does), is to require that for any pair of points (r,m) and (r',m') on different runs, the relative probability of these points is the same as the relative probability of r and r'. This property is easily seen to hold for the HT approach in the synchronous case. With this approach, the ratio of the probability of $(r_1,1)$ and $(r_2,1)$ is $\alpha : 1-\alpha$, as is the ratio of the probability of $(r_1,1)$ and $(r_2,2)$. This forces the probability of $(r_1,1)$ to be $\alpha/(2-\alpha)$, and the probability of each of $(r_1,1)$ and $(r_2,2)$ to be $(1-\alpha)/(2-\alpha)$. Note that, according to the Elga approach, if $\Pr$ is the probability on the runs of $\mathcal{R}_1$, $\alpha = 1/2$, so that $\Pr(r_1) = \Pr(r_2) = 1/2$, and $\Pr'$ is the probability that the agent assigns to the three points in the information set, then

$$\begin{array}{ll} & \Pr'((r_1,1) \mid \{(r_1,1),(r_2,1)\}) \\ = & \Pr'((r_1,1) \mid \{(r_1,1),(r_2,2)\}) \\ = & \Pr(r_1 \mid \{r_1,r_2\}) \\ = & 1/2. \end{array}$$

Thus, we must have $\Pr'((r_1,1)) = \Pr'((r_2,1)) = \Pr'((r_2,2))$, so each of the three points has probability $1/3$, which is Elga's second solution. Moreover, note that

$$\begin{array}{ll} & \Pr'((r_1,1) \mid \{(r_1,1),(r_2,1)\}) \\ = & \Pr'((r_2,1) \mid \{(r_1,1),(r_2,2)\}) \\ = & 1/2. \end{array}$$

This is one way of formalizing Elga's argument that $\Pr'$ should have the property that, conditional on learning it is Monday, you should consider "it is Monday and the coin landed heads" and "it is Monday and the coin landed tails" equally likely.

To summarize, the HT approach assigns probability among points in an information set I by dividing the probability of a run r among the points in I that lie on r (and then normalizing so that the sum is one), while the Elga approach proceeds by giving each and every point in I that is on run r the same probability as that of r, and then normalizing.

For future reference, I now give a somewhat more precise formalization of the HT and Elga approaches. To do so, it is helpful to have some notation that relates sets of runs to sets of points. If $\mathcal{S}$ is a set of runs and U is a set of points, let $\mathcal{S}(U)$ be the set of runs in $\mathcal{S}$ going through some point in U. and let $U(\mathcal{S})$ be the set of points in U that lie on some run in $\mathcal{S}$. That is,

$$\mathcal{S}(U) = \{r \in \mathcal{S} : (r,m) \in U \text{ for some } m\} \text{ and}$$
$$U(\mathcal{S}) = \{(r,m) \in U : r \in \mathcal{S}\}.$$

Note that, in particular, $\mathcal{K}_i(r,m)(r')$ is the set of points in the information set $\mathcal{K}_i(r,m)$ that are on the run r' and $\mathcal{R}(\mathcal{K}_i(r,m))$ is the set of runs in the system $\mathcal{R}$ that contain points in $\mathcal{K}_i(r,m)$. According to the HT approach, if $\Pr_i$ is agent i's probability on $\mathcal{R}$, the set of runs, then $\Pr^{HT}_{(r,m,i)}(\mathcal{K}_i(r,m)(r')) = \Pr_i(r' \mid \mathcal{R}(\mathcal{K}_i(r,m)))$. (Note that here I am using $\Pr^{HT}_{(i,r,m)}$ to denote agent i's probability at the point (r,m) calculated using the HT approach; I similarly will use $\Pr^{Elga}_{(i,r,m)}$ to denote agent i's preobability calculcated using the Elga approach.) That is, the probability that agent i assigns at the point (r,m) to the points in r' is just the probability of the run r' conditional on the probability of the runs going through the information set $\mathcal{K}_i(r,m)$. As I said earlier, Halpern and Tuttle do not try to assign a probability to individual points in $\mathcal{K}_i(r,m)(r')$ if there is more than one point on r' in $\mathcal{K}_i(r,m)$.

By way of contrast, the Elga approach is defined as follows:

$$\Pr^{Elga}_{(r,m,i)}(r',m') = \frac{\Pr_i(\{r'\} \cap \mathcal{R}(\mathcal{K}_i(r,m)))}{\sum_{r'' \in \mathcal{R}(\mathcal{K}_i(r,m))} \Pr_i(r'')|\mathcal{K}_i(r,m)(\{r''\})|}.$$

It is easy to check that $\Pr^{Elga}_{(r,m,i)}$ is the unique probability measure $\Pr'$ on $\mathcal{K}_i(r,m)$ such that $\Pr'((r_1,m_1))/\Pr'((r_2,m_2)) = \Pr_i(r_1)/\Pr_i(r_2)$ if $\Pr_i(r_2) > 0$. Note that $\Pr^{Elga}_{(r,m,i)}$ assigns equal probability to all points on a run r' in $\mathcal{K}_i(r,m)$. Even if $\Pr^{HT}_{(r,m,i)}$ is extended so that all points on a given run are taken to be equally likely, in general, $\Pr^{HT}_{(r,m,i)} \neq \Pr^{Elga}_{(r,m,i)}$. The following lemma characterizes exactly when the approaches give identical results.

Lemma 3.1: *$\Pr^{Elga}_{(r,m,i)} = \Pr^{HT}_{(r,m,i)}$ iff $|\mathcal{K}_i(r,m)(\{r_1\})| = |\mathcal{K}_i(r,m)(\{r_2\})|$ for all runs $r_1, r_2 \in \mathcal{R}(\mathcal{K}_i(r,m))$ such that $\Pr_i(r_j) \neq 0$ for $j = 1,2$.*

Note that, in the synchronous case, $|\mathcal{K}_i(r,m)(\{r'\})| = 1$ for all runs $r' \in \mathcal{R}(\mathcal{K}_i(r,m))$, so the two approaches are guaranteed to give the same answers.

4 Comparing the Approaches

I have formalized two approaches for ascribing probability in asynchronous settings, both of which generalize the relatively noncontroversial approach used in the synchronous case. Which is the most appropriate? I examine a number of arguments here.

4.1 Elga's Argument

Elga argued for the Elga approach, using the argument that if you discover it is Monday, then you should consider heads and tails equally likely. As I suggested above, I do not find this a compelling argument for the Elga approach. I agree that if you discover it is Monday, you should consider heads and tails equally likely. On the other hand, if you might actually learn that it is Monday, then there should be a run in the system reflecting this fact, as in the system $\mathcal{R}_2$ described Figure 2. In $\mathcal{R}_2$, even if the HT approach is used, if you discover it is Monday in run r_1^* or r_2^*, then you do indeed ascribe probability $1/2$ to heads. On the other hand, in r_1 and r_2, where you do *not* discover it is Monday, you also ascribe probability $1/2$ to heads when you are woken up, but conditional on it being Monday, you consider the probability of heads to be $2/3$. Thus, $\mathcal{R}_2$ allows us to model Elga's intuition that, after learning it is Monday (in r_1^* and r_2^*) you ascribe probability $1/2$ to heads, while still allowing you to ascribe probability $1/2$ to heads in r_1 and r_2 if you do not learn it is Monday. Moreover, in r_1 and r_2, conditional on it being Monday, you ascribe probability $2/3$ to heads.

The real issue here is whether, in r_1, the probability of heads conditional on it being Monday should be the same as the probability that you ascribe to heads in r_1^*, where you actually discover that it is Monday. We often identify the probability of V given U with the probability that you would ascribe to V if you learn U is true. What I am suggesting here is that this identification breaks down in the asynchronous case. This, of course, raises the question of what exactly conditioning means in this context. "The probability of V given U" is saying something like "if it were the case that U, then the probability of V would be ..." This is not necessarily the same as "if you were to learn that U, then the probability of V would be ..."[3]

Although $\mathcal{R}_2$ shows that Elga's argument for the $1/3$–$2/3$ answer is suspect, it does not follow that $1/3$–$2/3$ is incorrect. In the remainder of this section, I examine other considerations to see if they shed light on what should be the appropriate answer.

4.2 The Frequency Interpretation

One standard interpretation of probability is in terms of frequency. If the probability of a coin landing heads is $1/2$, then if we repeatedly toss the coin, it will land heads in roughly half the trials; it will also land heads roughly half the time. In the synchronous case, "half the trials" and "half the time" are the same. But now consider the Sleeping Beauty problem. What counts as a "trial"? If a "trial" is an experiment, then the coin clearly lands heads in half of the trials. But it is equally clear that the coin lands heads $1/3$ of the times that the agent is woken up. Considering "times" and "trials" leads to different answers in asynchronous systems; in the case of the Sleeping Beauty problem, these different answers are precisely the natural $1/2$–$1/2$ and $1/3$–$2/3$ answers.

4.3 Betting Games

Another standard approach to determining subjective probability, which goes back to Ramsey [1931] and De Finetti [1931], is in terms of betting behavior. For example, one way of determining the subjective probability that an agent ascribes to a coin toss landing heads is to compare the odds at which he would accept a bet on heads to one at which he would accept a bet on tails. While this seems quite straightforward, in the asynchronous case it is not. This issue was considered in detail in the context of the absented-minded driver paradox in [Grove and Halpern 1997]. Much the same comments hold here, so I just do a brief review.

Suppose that Sleeping Beauty is offered a \$1 bet on whether the coin landed heads or the coin landed tails every time she is woken up. If the bet pays off every time she answers the question correctly, then clearly she should say "tails". Her expected gain by always saying tails is \$1 (since, with probability $1/2$, the coin will land tails and she will get \$1 both times she is asked), while her expected gain by always saying heads is only $1/2$. Indeed, a risk-neutral agent should be willing to pay to take this bet. Thus, even though she considers heads and tails equally likely and ascribes probabilities using the HT approach, this betting game would have her act as if she considered tails twice as likely as heads: she would be indifferent between saying "heads" and "tails" only if the payoff for heads was \$2, twice the payoff for tails.

In this betting game, the payoff occurs at every time step. Now consider a second betting game, where the payoff is only once per trial (so that if the coin lands tails, the agent get \$1 if she says tails both times, and \$0.50 if she says tails only once). If the payoff is per trial, then the agent should be indifferent being saying "heads" and "tails"; the situation is analogous to the discussion in the frequency interpretation.

There is yet a third alternative. The agent could be offered a bet at only one point in the information set. If the coin lands heads, she must be offered the bet at $(r_1, 1)$. If the coin lands heads, an adversary must somehow choose if the bet will be offered at $(r_2, 1)$ or $(r_2, 2)$. The third betting game is perhaps more in keeping with the second story told for $\mathcal{R}_1$, where the agent is not aware of time passing and must assign a probability to heads and tails in the information set. It may seem that the first betting game, where the payoff occurs at each step, is more appropriate to the Sleeping Beauty problem—after all, the agent is woken up twice if the coin lands tails. Of course, if the goal of the problem is to maximize the expected number of correct answers (which is what this betting game amounts to), then there is no question that "tails" is the right thing to say. On the other hand, if the goal is to get the right answer "now", whenever now is, perhaps because this is the only time that the bet will be offered, then the third game is more appropriate. My main point here is that the question of the right betting game, while noncontroversial in the synchronous case, is less clear in the asynchronous case.

4.4 Conditioning and the Reflection Principle

To what extent is it the case that the agent's probability over time can be viewed as changing via conditioning? It turns out that the answer to this question is closely related to the question of when the Reflection Principle holds, and gives

[3] There are other reasons that "given U and "if you were to learn that U" should be treated differently; in the latter case, you must take into account how you came to learn that U is the case. Without taking this into account, you run into puzzles like the Monty Hall problem; see [Grünwald and Halpern 2003] for discussion of this point. I ignore this issue here, since it is orthogonal to the issues that arise in the Sleeping Beauty problem.

further support to using the HT approach to ascribing probabilities in the asynchronous case.

There is a trivial sense in which updating is never done by conditioning: At the point (r, m), agent i puts probability on the space $\mathcal{K}_i(r, m)$; at the point $(r, m+1)$, agent i puts probability on the space $\mathcal{K}_i(r, m+1)$. These spaces are either disjoint or identical (since the indistinguishability relation that determines $\mathcal{K}_i(r, m)$ and $\mathcal{K}_i(r, m+1)$ is an equivalence relation). Certainly, if they are disjoint, agent i cannot be updating by conditioning, since the conditional probability space is identical to the original probability space. And if the spaces are identical, it is easy to see that the agent is not doing any updating at all; her probabilities do not change.

To focus on the most significant issues, it is best to factor out time by considering only the probability ascribed to runs. Technically, this amounts to considering *run-based events*, that is sets U of points with the property that if $(r, m) \in U$, then $(r, m') \in U$ for all times m'. In other words, U contains all the points in a given run or none of them. Intuitively, we can identify U with the set of runs that have points in U. To avoid problems of how to assign probability in asynchronous systems, I start by considering synchronous systems. Given a set V of points, let $V^- = \{(r, m) : (r, m+1) \in V\}$; that is, V^- consists of all the points immediately preceding points in V. The following result, whose straightforward proof is left to the reader, shows that in synchronous systems where the agents have perfect recall, the agents do essentially update by conditioning. The probability that the agent ascribes to an event U at time $m+1$ is obtained by conditioning the probability he ascribes to U at time m on the set of points immediately preceding those he considers possible at time $m+1$.

Theorem 4.1: [Halpern 2003] *Let U be a run-based event and let $\mathcal{R}$ be a synchronous system where the agents have perfect recall. Then*

$$\Pr_{r,m+1,i}(U) = \Pr_{r,m,i}(U \mid \mathcal{K}_i(r, m+1)^-).$$

Theorem 4.1 does not hold without assuming perfect recall. For example, suppose that an agent tosses a fair coin and observes at time 1 that the outcome is heads. Then at time 2 he forgets the outcome (but remembers that the coin was tossed, and knows the time). Thus, at time 2, because the outcome is forgotten, the agent ascribes probability $1/2$ to each of heads and tails. Clearly, her time 2 probabilities are not the result of applying conditioning to her time 1 probabilities.

A more interesting question is whether Theorem 4.1 holds if we assume perfect recall and do not assume synchrony. Properly interpreted, it does, as I show below. But, as stated, it does not, even with the HT approach to assigning probabilities. The problem is the use of $\mathcal{K}_i(r, m+1)^-$ in the statement of the theorem. In an asynchronous system, some of the points in $\mathcal{K}_i(r, m+1)^-$ may still be in $\mathcal{K}_i(r, m+1)$, since the agent may not be aware of time passing. Intuitively, at time (r, m), we want to condition on the set of points in $\mathcal{K}_i(r, m)$ that are on runs that the agent considers possible at $(r, m+1)$. But this set is not necessarily $\mathcal{K}_i(r, m+1)^-$.

Let $\mathcal{K}_i(r, m+1)^{(r,m)} = \{(r', k) \in \mathcal{K}_i(r, m) : \exists m'((r, m+1) \sim_i (r', m'))\}$. Note that $\mathcal{K}_i(r, m+1)^{(r,m)}$ consists precisely of those points that agent considers possible at (r, m) that are on runs that the agent still considers possible at $(r, m+1)$. In synchronous systems with perfect recall, $\mathcal{K}_i(r, m+1)^{(r,m)} = \mathcal{K}_i(r, m+1)^-$ since, as observed above, if $(r, m+1) \sim_i (r', m+1)$ then $(r, m) \sim_i (r', m)$. In general, however, the two sets are distinct. Using $\mathcal{K}_i(r, m+1)^{(r,m)}$ instead of $\mathcal{K}^-_{r,m+1}$ gives an appropriate generalization of Theorem 4.1.

Theorem 4.2: [Halpern 2003] *Let U be a run-based event and let $\mathcal{R}$ be a system where the agents have perfect recall. Then,*

$$\Pr^{HT}_{r,m+1,i}(U) = \Pr^{HT}_{r,m,i}(U \mid \mathcal{K}_i(r, m+1)^{(r,m)}).$$

Thus, in systems with perfect recall, using the HT approach to assigning probabilities, updating proceeds by conditioning. Note that since the theorem considers only run-based events, it holds no matter how the probability among points on a run is distributed. For example, in the Sleeping Beauty problem, this result holds even if $(r_2, 1)$ and $(r_2, 2)$ are not taken to be equally likely.

The analogue of Theorem 4.2 does not hold in general for the Elga approach. This can already be seen in the Sleeping Beauty problem. Consider the system of Figure 1. At time 0 (in either r_1 or r_2), the event heads (which consists of all the points in r_1) is ascribed probability $1/2$. At time 1, it is ascribed probability $1/3$. Since $\mathcal{K}_{SB}(r_1, 1)^{(r_1,0)} = \{(r_1, 0), (r_2, 0)\}$, we have

$$\begin{array}{l} 1/3 = \Pr^{Elga}_{r_1,1,SB}(heads) \neq \\ \Pr^{Elga}_{r_1,0,SB}(heads) \mid \mathcal{K}_{SB}(r_1, 1)^{(r_1,0)}) = 1/2. \end{array}$$

The last equality captures the intuition that if Sleeping Beauty gets no additional information, then her probabilities should not change using conditioning.

Van Fraassen's [1995] *Reflection Principle* is a coherence condition connecting an agent's future beliefs and his current belief. Note that what an agent believes in the future will depend in part on what the agent learns. The *Generalized Reflection Principle* says that if we consider all the possible things that an agent might learn (or evidence that an agent might obtain) between the current time and time m—call these $E(1, k)$, …, $E(k, m)$—and $\Pr_1$, …, $\Pr_k$ are the agent's probability functions at time m (depending on which piece of evidence is obtained), then the agent's current probability should be a convex combination of $\Pr_1, \ldots, \Pr_k$. That is, there should exist coefficients $\alpha_1, \ldots, \alpha_k$ in the interval $[0, 1]$ such that $\Pr = \alpha_1 \Pr_1 + \ldots + \alpha_k \Pr_k$. Savage's [1954] *Sure-Thing Principle* is essentially a special case. It says that if the probability of A is α no matter what is learned at time m, then the probability of A should be α right now. This certainly seems like a reasonable criterion.

Van Fraassen [1995] in fact claims that if an agent changes his opinion by conditioning on evidence, that is, if $\Pr_j = \Pr(\cdot \mid E(j,m))$ for $j = 1, \ldots, k$, then the Generalized Reflection Principle must hold. The intuition is that the pieces of evidence $E(1,m), \ldots, E(k,m)$ must form a partition of underlying space (in each state, exactly one piece of evidence will be obtained), so that it becomes a straightfoward application of elementary probability theory to show that if $\alpha_j = \Pr(E(j,t))$ for $j = 1, \ldots, k$, then indeed $\Pr = \alpha_1 \Pr_1 + \cdots + \alpha_k \Pr_k$.

Van Fraassen was assuming that the agent has a fixed set W of possible worlds, and his probability on W changed by conditioning on new evidence. Moreover, he was assuming that the evidence was a subset of W. In the runs and systems framework, the agent is not putting probability on a fixed set of worlds. Rather, at each time k, he puts probability on the set of worlds (i.e., points) that he considers possible at time k. The agent's evidence is an information set—a set of points. If we restrict attention to run-based events, we can instead focus on the agent's probabilities on runs. That is, we can take W to be the set of runs, and consider how the agent's probability on runs changes over time. Unfortunately, agent i's evidence at a point (r,m) is not a set of runs, but a set of points, namely $\mathcal{K}_i(r,m)$. We can associate with $\mathcal{K}_i(r,m)$ the set of runs going through the points in $\mathcal{K}_i(r,m)$, namely, in the notation of Section 3.2, $\mathcal{R}(\mathcal{K}_i(r,m))$

In the synchronous case, for each time m, the possible information sets at time m correspond to the possible pieces of evidence that the agent has at time m. These information sets form a partition of the time-m points, and induce a partition on runs. In this case, van Fraassen's argument is correct. More precisely, if, for simplicity, "now" is taken to be time 0, and we consider some future time $m > 0$, the possble pieces of evidence that agent i could get at time m are all sets of the form $\mathcal{K}_i(r,m)$, for $r \in \mathcal{R}$. With this translation of terms, it is an immediate consequence of van Fraassen's observation and Theorem 4.1 that the Generalized Reflection Principle holds in synchronous systems with perfect recall. But note that the assumption of perfect recall is critical here. Consider an agent that tosses a coin and observes that it lands heads at time 0. Thus, at time 0, she assigns probability 1 to the event of that coin toss landing heads. But she knows that one year later she will have forgotten the outcome of the coin toss, and will assign that event probability $1/2$ (even though she will know the time). Clearly Reflection does not hold.

What about the asynchronous case? Here it is not straightforward to even formulate an appropriate analogue of the Reflection Principle. The first question to consider is what pieces of evidence to consider at time m. While we can consider all the information sets of form $\mathcal{K}_i(r,m)$, where m is fixed and r ranges over the runs, these sets, as we observed earlier, contain points other than time m points. While it is true that either $\mathcal{K}_i(r,m)$ is identical to $\mathcal{K}_i(r',m)$ or disjoint from $\mathcal{K}_i(r',m)$, these sets do *not* induce a partition on the runs. It is quite possible that, even though the set of points $\mathcal{K}_i(r,m)$ and $\mathcal{K}_i(r',m)$ are disjoint, there may be a run r'' and times m_1 and m_2 such that $(r'',m_1) \in \mathcal{K}_i(r,m)$ and $(r'',m_2) \in \mathcal{K}_i(r',m)$. For example, in Figure 4, if the runs from left to right are r_1–r_5, then $\mathcal{K}_{SB}(r_5,1) = \{r_1,\ldots,r_5\}$ and $\mathcal{K}_{SB}(r_1,1) = \{r_1,r_2,r_3\}$. However, under the assumption of perfect recall, it can be shown that for any two information sets $\mathcal{K}_i(r_1,m)$ and $\mathcal{K}_i(r_2,m)$, either (a) $\mathcal{R}(\mathcal{K}_i(r_1,m)) \subseteq \mathcal{R}(\mathcal{K}_i(r_2,m))$, (b) $\mathcal{R}(\mathcal{K}_i(r_2,m)) \subseteq \mathcal{R}(\mathcal{K}_i(r_1,m))$, or (c) $\mathcal{R}(\mathcal{K}_i(r_1,m)) \cap \mathcal{R}(\mathcal{K}_i(r_2,m)) = \emptyset$. From this it follows that there exist a collection $\mathcal{R}'$ of runs such that the sets $\mathcal{R}(\mathcal{K}_i(r',m))$ for $r' \in \mathcal{R}'$ are disjoint and the union of $\mathcal{R}(\mathcal{K}_i(r',m))$ taken over the runs $r' \in \mathcal{R}'$ consists of all runs in $\mathcal{R}$. Then the same argument as in the synchronous case gives the following result.

Theorem 4.3: *If $\mathcal{R}$ is a (synchronous or asynchronous) system with perfect recall and $\mathcal{K}_i(r_1,m),\ldots,\mathcal{K}_i(r_k,m)$ are the distinct information sets of the form $\mathcal{K}_i(r',m)$ for $r' \in \mathcal{R}(\mathcal{K}_i(r,0)$, then there exist $\alpha_1,\ldots,\alpha_k$ such that*

$$\Pr_i(\cdot \mid \mathcal{R}(\mathcal{K}_i(r,0))) = \sum_{j=1}^{k} \alpha_j \Pr_i(\cdot \mid \mathcal{R}(\mathcal{K}_i(r_j,m))).$$

The following corollary is immediate from Theorem 4.3, given the definition of $\Pr^{HT}_{(i,r,m)}$.

Corollary 4.4: *If $\mathcal{R}$ is a (synchronous or asynchronous) system with perfect recall and $\mathcal{K}_i(r_1,m),\ldots,\mathcal{K}_i(r_k,m)$ are the distinct information sets of the form $\mathcal{K}_i(r',m)$ for $r' \in \mathcal{R}(\mathcal{K}_i(r,0)$, then there exist $\alpha_1,\ldots,\alpha_k$ such that for all $\mathcal{R}' \subseteq \mathcal{R}$,*

$$\Pr^{HT}_{(i,r,0)}(\mathcal{K}_i(r,0)(\mathcal{R}')) = \sum_{j=1}^{k} \alpha_j \Pr^{HT}_{(i,r_j,m)}(\mathcal{K}_i(r_j,m))(\mathcal{R}')).$$

Corollary 4.4 makes precise the sense in which the Reflection Principle holds for the HT approach. Although the notation $\mathcal{K}_i(r,m)(\mathcal{R}')$ that converts sets of runs to sets of points makes the statement somewhat ugly, it plays an important role in emphasizing what I take to be an important distinction, that has largely been ignored. An agent assigns probability to points, not runs. At both time 0 and time m we can consider the probability that the agent assigns to the points on the runs in $\mathcal{R}'$, but the agent is actually assigning probability to quite different (although related) events at time 0 and time m.

The obvious analogue to Corollary 4.4 does not hold for the Elga approach. Indeed, the same example that shows conditioning fails in the Sleeping Beauty problem shows that the Reflection Principle does not hold. Indeed, this example shows that the sure-thing principle fails too. Using the Elga approach, the probability of heads (i.e., the probability of the points on the run where the coin lands heads) changes from $1/2$ to $1/3$ between time 0 and time 1, no matter what.

5 Conclusion

In this paper, I have tried to take a close look at the problem of updating in the presence of asynchronoy and imperfect recall. Let me summarize what I take to be the main points of this paper:

- It is important to have a good formal model that incorporates uncertainty, imperfect recall, and asynchrony in which probabilistic arguments can be examined. While the model I have presented here is certainly not the only one that can be used, it does have a number of attractive features.
- Whereas there seems to be only one reasonable approach to assigning (and hence updating) probabilities in the synchronous case, there are at least two such approaches in the asynchronous case. Both approaches can be supported using a frequency interpretation and a betting interpretation. However, only the HT approach supports the Reflection Principle in general. In particular, the two approaches lead to the two different answers in the Sleeping Beauty problem.
- We cannot necessarily identify the probability conditional on U with what the probability would be upon learning U. This identification is being made in Elga's argument; the structure $\mathcal{R}_2$ shows that they may be distinct.

One fact that seems obvious in light of all this discussion is that our intuitions regarding how to do updating in asynchronous systems are rather poor. Given how critical this problem is for KR, it clearly deserves further investigation.

Acknowledgments

Thanks to Moshe Vardi for pointing out Elga's paper, and to Oliver Board, Adam Elga, Sergiu Hart, Kevin O'Neill, Teddy Seidenfeld, Moshe Vardi, and two KR reviewers for a number of useful comments on an earlier draft of the paper.

References

Arntzenius, F. (2003). Some problems for conditionalization and reflection. *Journal of Philosophy 100*, 356–370.

de Finetti, B. (1931). Sul significato soggestivo del probabilità. *Fundamenta Mathematica 17*, 298–329.

Dorr, C. (2002). Sleeping Beauty: in defence of Elga. *Analysis 62*, 292–296.

Elga, A. (2000). Self-locating belief and the Sleeping Beauty problem. *Analysis 60*(2), 143–147.

Fagin, R., J. Y. Halpern, Y. Moses, and M. Y. Vardi (1995). *Reasoning about Knowledge*. Cambridge, Mass.: MIT Press.

Grove, A. J. and J. Y. Halpern (1997). On the expected value of games with absentmindedness. *Games and Economic Behavior 20*, 51–65.

Grünwald, P. D. and J. Y. Halpern (2003). Updating probabilities. *Journal of A.I. Research 19*, 243–278.

Halpern, J. Y. (1997). On ambiguities in the interpretation of game trees. *Games and Economic Behavior 20*, 66–96.

Halpern, J. Y. (2003). *Reasoning About Uncertainty*. Cambridge, Mass.: MIT Press.

Halpern, J. Y. and R. Fagin (1989). Modelling knowledge and action in distributed systems. *Distributed Computing 3*(4), 159–179.

Halpern, J. Y. and M. R. Tuttle (1993). Knowledge, probability, and adversaries. *Journal of the ACM 40*(4), 917–962.

Halpern, J. Y. and M. Y. Vardi (1989). The complexity of reasoning about knowledge and time, I: lower bounds. *Journal of Computer and System Sciences 38*(1), 195–237.

Lewis, D. (2001). Sleeping Beauty: reply to Elga. *Analysis 61*, 171–176.

Monton, B. (2002). Sleeping Beauty and the forgetful Bayesian. *Analysis 62*, 47–53.

Piccione, M. and A. Rubinstein (1997). On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior 20*(1), 3–24.

Ramsey, F. P. (1931). Truth and probability. In R. B. Braithwaite (Ed.), *The Foundations of Mathematics and Other Logical Essays*, pp. 156–198. London: Routledge and Kegan Paul.

Savage, L. J. (1954). *Foundations of Statistics*. New York: Wiley.

van Fraassen, B. C. (1984). Belief and the will. *Journal of Philosophy 81*, 235–245.

van Fraassen, B. C. (1995). Belief and the problem of Ulysses and the Sirens. *Philosophical Studies 77*, 7–37.