ON AMBIGUITIES IN THE INTERPRETATION OF GAME TREES

Joseph Y. Halpern*
Cornell University
Computer Science Department
4144 Upson Hall
Ithaca, NY 14853
halpern@cs.cornell.edu

Nov. 18, 1996

Abstract

Piccione and Rubinstein have pointed out ambiguities in the interpretation of games of imperfect recall. They focus on the notion of time consistency, and argue that a player in a game of imperfect recall may be time inconsistent, changing his strategy despite no new information and no change in his preferences. In this paper, it is argued that the apparent time inconsistency arises from implicit assumptions made in the definition about what the driver knows when he reconsiders his strategy and what he will remember if he changes his strategy, and about how the node at which reconsideration takes place is chosen. A model is proposed, based on earlier work in the computer science literature, that allows us—indeed, almost forces us—to make these issues explicit. Once these issues are made explicit, time inconsistency seems less inconsistent.

*Most of this work was carried out while I was at the IBM Almaden Research Center. I gratefully acknowledge IBM's support. This version of the paper is a slightly expanded version of one that will appear in *Games and Economic Behavior*; Section 3.5 and the proof of Theorem 4.2 do not appear in the journal version.

1 Introduction

In an early version of their fascinating paper, Piccione and Rubinstein [1997] (PR from now on) argue that "the model of extensive games with imperfect recall suffers from major ambiguities". They go on to say that "the difficulty in interpreting the model stems from questions concerning the knowledge described by the information partition and the restrictions that the information structure imposes on the set of strategies". To illustrate this issue, PR focus on *time consistency*: that is, the question of whether a player will want to change his strategy in the midst of playing a game. One of their primary examples is what they call the "absentminded driver paradox", which they describe as follows:

Example 1.1: An individual is sitting late at night in a bar planning his midnight trip home. In order to get home he has to take the highway and get off at the second exit. Turning at the first exit leads into a disastrous area (payoff 0). Turning at the second exit yields the highest reward (payoff 4). If he continues beyond the second exit he will reach the end of the highway and find a hotel where he can spend the night (payoff 1). The driver is absentminded and is aware of this fact. When reaching an intersection he cannot tell whether it is the first or the second intersection and he cannot remember how many he has passed. ∎

The situation is described by the game tree in Figure 1. Clearly the only decision

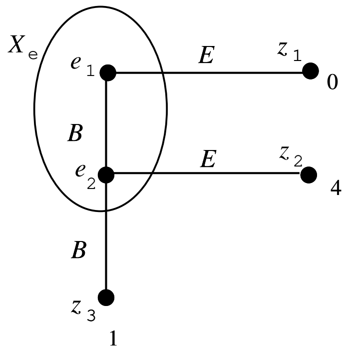

Figure 1: The absentminded driver game.

the driver has to make is whether to get off when he reaches an exit. He cannot plan on doing different things at each exit, since, by assumption, he does not know which exit he is at when he reaches an exit. Suppose we start by considering deterministic strategies. In this case, while sitting at the bar, there is one obviously best option: not to exit. For in this case, the driver will certainly reach the end of the highway, which has payoff 1. On the other hand, if the driver decides to get off when he reaches an exit, then he will do so at the first exit, which has payoff 0.

This is the driver's reasoning at time 0, while still in the bar. Now consider what happens when he reaches the first exit. Suppose that the driver remembers the strategy he chose at the bar. Since this strategy prescribes not exiting, he should, according to PR, ascribe subjective probability 1/2 to being at the first exit. Subjectively, he is equally likely to be at either exit. He thus concludes that it is optimal to get off, since the expected payoff of doing so is 2. This gives us time inconsistency: despite no new information and no change in his preferences, the driver is tempted to change his initial plan once he reaches an exit. PR go on to show that a similar time inconsistency arises even if the driver uses his optimal randomized (behavioral) strategy. (Their argument is reviewed in Section 3.) This certainly seems paradoxical!

At a high level, I agree completely with PR's claim that there are ambiguities in the standard approach to modeling of games of imperfect recall and, in particular, to modeling the knowledge of agents in such games. Moreover, I agree with their implicit claim that it is these ambiguities that lead to the apparently counterintuitive nature of the absentminded driver example. The goal of this paper is to try to make the ambiguities more explicit, and to propose a modeling methodology that helps us to avoid them.

I argue that in order to analyze this example (and time consistency in general), we must be very careful to specify (1) exactly what the driver knows and remembers (in particular, whether he knows his initial strategy and whether he will remember his new strategy if he switches), and (2) what causes the agent to reconsider. (Does the agent reconsider at just one node, or at every node in the information set? If it is just one node, how is that node chosen?)

The critical role of what the driver knows and remembers is perhaps best seen by considering the game described in Figure 2 (this is PR's Example 2). It is not hard

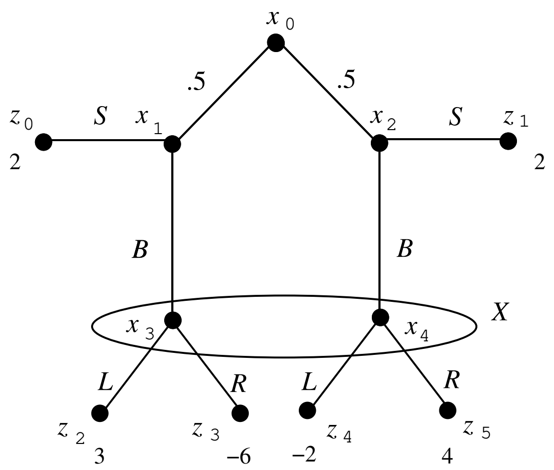

Figure 2: Another game with time inconsistency

to show that the strategy that maximizes expected utility chooses action S at node x_1, action B at node x_2, and action R at the information set X consisting of x_3 and x_4. Call this strategy f. Let f' be the strategy of choosing action B at x_1, action S at x_2, and L at X. PR argue that if node x_1 is reached, the agent should reconsider, and decide to switch from f to f'. *If the agent is able to remember that he switched strategies*, then this is correct; the agent is indeed better off (under any reasonable notion of "better off") if he switches.

The reason for the time inconsistency here is that an agent's strategy must dictate the same action at nodes x_3 and x_4, since they are in the same information set. Intuitively, since the agent cannot distinguish the nodes, he must do the same thing at both. If the agent had perfect recall, he could distinguish the nodes. In this case, the optimal strategy would essentially look like the result of switching from f to f' at x_1 without perfect recall: the agent plays L at x_3 (as he would with f') and R at x_4 (as he would with f). Switching strategies ends up simulating the optimal strategy here because, by having the ability to switch strategies and remember that he has switched, the agent is able to simulate perfect recall. Among other things, PR explicitly assume "that the decision maker is not allowed to employ an external device to assist him in keeping the information which he would otherwise lose". However, allowing an agent to know his strategy may act as just such an external device.

As this example suggests, an information set in a game tree may not adequately represent the information that the agent actually has. As a consequence, the standard restriction that a strategy must behave the same way at all nodes in an information set may be inappropriate. Of course, a strategy must be such that an agent does the same thing in all situations that he cannot distinguish. The point is that "situations that he cannot distinguish" and "nodes in the same information set" may be two quite different notions. Roughly speaking, they coincide if the agent knows his strategy, never changes it, and has perfect recall. Otherwise, they may differ.

So how can we capture an agent's information? I propose a solution based on the multi-agent system framework introduced by Halpern and Fagin [1989]—and discussed in detail in [Fagin, Halpern, Moses, and Vardi 1995]—and extended by Halpern and Tuttle [1993] to deal with randomized actions. The idea is to distinguish the "external world" from an agent's "internal world". The game tree is a useful representation of the external world. Nodes in a game tree can be viewed as describing possible states of the external world. Information sets then represent an upper bound on what the agent can know about the external world, even assuming that she has perfect recall. To represent an agent's internal world, I assume that the agent, like the external world, is always in some *(local) state*. Intuitively, this state describes all the relevant information the agent has—about the external world, about other agents (if there are other agents in the game), about her strategy, and so on. For example, as noted in [Fagin, Halpern, Moses, and Vardi 1995], to capture the fact that an agent knows her strategy, we can encode the strategy in the agent's local state. Similarly, to capture the fact that an agent remembers what actions she has performed thus far, these actions must be encoded in the agent's state. By making the agent's state explicit, there is no doubt about what the agent knows or does not know at any point.

The analogue to a strategy in this framework is a *protocol*, which is a function from local states to actions. Protocols are meant to capture the same intuitions as strategies: what an agent does can depend only on what he knows. But now, an agent's knowledge is captured, not by the information set, but by his local state. If the information sets characterize an agent's knowledge in a game, then protocols and strategies coincide. When they do not (as is the case in the two games discussed above), then I would argue that protocols are the right notion to consider, not strategies.

In the games of imperfect recall described in PR, what the agent's possible states are is left ambiguous. In the framework I present here, we are forced to be explicit about this. If we allow the agent to switch strategies, and the agent's state includes the last strategy chosen, the agent's local state will be different at nodes x_3 and x_4 in Figure 2. Since the agent's protocol is a function of his local state, not his information set, a protocol may perform different actions at these nodes. The optimal protocol in this game is indeed to choose to change strategies at node x_1 (which results in different actions being performed at x_3 and x_4), and it is indeed time consistent.

The rest of this paper is organized as follows. In Section 2, some basic definitions are given (which are mainly standard definitions of game theory). In Section 3, I take a closer look at the notion of time consistency, and consider four possible definitions of it and the assumptions underlying them: the original PR notion of time consistency, PR's notion of modified multi-self time consistency (and Aumann, Hart, and Perry's [1997] equivalent notion of *action optimality*), and two new notions called *gt* and *ms (multi-self)* consistency. As we shall see, the notions differ in important but subtle ways. In Section 4, I describe the multi-agent system framework and show how it can be used to capture information in games. In particular, I show that, given a game, each of the four notions of consistency gives rise to a particular system that captures in a precise sense the assumptions underlying the notion. I conclude with some discussion in Section 5. Proofs of some of the results can be found in the appendix.

2 Basic Definitions

A game Γ is described by a game tree consisting of a finite collection of nodes partially ordered by $\prec$ (where $\prec$ is a transitive and anti-symmetric relation). As usual, we write $x \preceq y$ if $x \prec y$ or $x = y$. Intuitively, $x \preceq y$ if there is a path from x to y in the tree. Game trees are assumed to be *rooted*; that is, there is a unique node x_0 such that for all nodes y in Γ, we have $x_0 \preceq y$. Finally, we assume that $\prec$ does define a tree, in that if $x \prec y$ and $x' \prec y$, then we must have either $x \preceq x'$ or $x' \preceq x$. For a k-player game, the nodes in a game tree can be partitioned into $k + 2$ sets denoted C, D_1, ..., D_k, and Z. Since in this paper I want to focus on single-agent games, I henceforth assume that $k = 1$, and refer to D rather than D_1. The set C consists of *chance nodes*, where nature moves. The edges coming out of nodes in C are labeled by the probability of nature taking that move. For example, in Figure 2, x_0 is a chance node; nature moves from x_0 to x_1 with probability .5. The set D consists of *decision nodes* where the player must move. D is further partitioned into *information sets*. (The information sets are described by ellipses in the game tree.) For example, in Figure 1, e_1 and e_2 are in the same information set. Intuitively, if two nodes are in the same information set, then the player cannot distinguish them. Exactly what "cannot distinguish" means is one of the major issues dealt with in this paper. At each node x in D, the player can choose among some set of actions, denoted $A(x)$. The edges coming out of x are labeled by these actions. We assume that the same set of actions can be performed at each node in a given information set. That is, if x and y are both in information set X, then $A(x) = A(y)$. For example, since e_1 and e_2 are both in information set X_e in Figure 1, we have $A(e_1) = A(e_2) = \{B, E\}$. Thus, we can write $A(X)$ to denote the actions that can be taken at an information set X. The set Z consists of the set of terminal nodes in Γ; that is, those nodes z for which there does not exist z' such that $z \prec z'$. With each node $z \in Z$ is associated a utility $u(z)$; $u(z)$ can be thought of as the payoff for reaching node z. For example, in Figure 1, $u(z_1) = 0$ and $u(z_2) = 4$.

Roughly speaking, an agent in game Γ has *perfect recall* if he always remembers what actions he has taken and what he knew previously. Formally, this is captured by associating with each node x in the game tree the sequence $exp(x)$ (for the *experience* of the agent at node x) of actions taken by the agent and information sets gone through by the agent in going from the root to x. For example, in Figure 1, $exp(e_2) = \langle X_e, B, X_e \rangle$, while in Figure 2, $exp(x_3) = \langle \{x_1\}, B, \{x_3, x_4\} \rangle$. I omit the formal definition here. A game of perfect recall is one where, for every information set X, we have $exp(x) = exp(x')$ for all nodes $x, x' \in X$. Thus, the game in Figure 1 is not a game of perfect recall, since $exp(e_1) \neq exp(e_2)$. The game in Figure 2 is not a game of perfect recall either, since $exp(x_3) \neq exp(x_4)$.

We can similarly associate with each node x two other sequences, denoted $exp'(x)$ and $exp''(x)$. The sequence $exp'(x)$ consists of the sequence of information sets gone through by the agent (but not the sequence of actions), while $exp''(x)$ consists of the sequence of information sets without consecutive repetitions. Thus, for example, in Figure 1, we have $exp'(e_2) = \langle X_e, X_e \rangle$, while $exp''(e_2) = \langle X_e \rangle$, with the second X_e omitted. PR say a game Γ has *perfect recall of information sets* if, for every information set X and $x, x' \in X$, we have $exp'(x) = exp'(x')$. Similarly, I say that Γ has *partial recall (of information sets)* if, for every information set X and $x, x' \in X$, we have $exp''(x) = exp''(x')$.[1] Note that the absentminded driver example exhibits partial recall, although not perfect recall of information sets. The game in Figure 2 does not exhibit partial recall (and, *a fortiori*, does not exhibit perfect recall of information sets either).

PR say that a game exhibits *absentmindedness* if there is an information set X with

[1]Interestingly, the notion of partial recall of information sets is actually closer in spirit to what is called perfect recall in the computer science literature [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Vardi 1986] than the standard definition of perfect recall (in terms of exp) used in the game theory literature. Roughly speaking, $exp''(x)$ is meant to capture the intuition that the agent is not aware of time passing. Discussion of and more motivation for the computer science definition is given in Footnote 7.

two nodes $x, x' \in X$ such that $x \prec x'$. Clearly a game that exhibits absentmindedness cannot be a game of perfect recall. The absentminded driver example of Figure 1 is a game that exhibits absentmindedness.

Intuitively, a *behavior strategy* specifies the action that an agent takes at each node $x \in D$. We allow the choice of action to be randomized. Formally, a behavior strategy b assigns to each node $x \in D$ a probability distribution $b(x)$ over the actions in $A(x)$. For notational convenience, this distribution is often denoted b_x. (Of course, if b is deterministic, b_x assigns probability 1 to some action in $a \in A(x)$; in this case, I write $b(x) = a$.) If nodes x and y are in the same information set X, we assume that $b_x = b_y$, since the agent is not supposed to be able to tell which of these nodes he is at. Thus, we can write b_X to denote the distribution on the actions in $A(X)$. We can think of b_x as a random device (e.g., a coin toss) that is activated when node x is reached. For example, if at node x an agent has two possible actions, L and R, and $b_x(L) = .6$, then at node x, the agent tosses a coin which lands heads with probability .6, and chooses action L if the coin lands heads, and action R otherwise. If x and y are distinct nodes such that $b_x = b_y$, then the coin is tossed independently at x and y; the outcome at x does not affect the outcome at y. Given a (generalized) behavior strategy b and a node x in the tree, let $p_b(y|x)$ denote the probability of reaching node y starting at node x and using strategy b. Clearly, if there is no path from x to y in Γ, we must have $p_b(y|x) = 0$. If there is a path, then $p_b(y|x)$ is just the probability of choosing the (unique) sequence of actions that lead from x to y, according to behavior strategy b. Finally, let $p_b(y)$ be an abbreviation for $p_b(y|r)$, where r is the root of the tree.

3 A Closer Look at Time Consistency

Roughly speaking, a behavior strategy b is said to be time consistent if, whenever an agent reaches an information set X in the game tree, then the agent does not consider some other strategy b' to be better than b, given that he has reached X. As usual, goodness is evaluated in terms of expected utility. However, the agent's subjective utility will depend on his beliefs. In this section, I consider four possible collections of beliefs that the agent may have, that lead to four notions of time consistency.

3.1 PR time consistency

In considering whether to switch from b to another strategy at information set X, PR seem to be implicitly assuming that the agent believes the following:

1. some process (external to the game) has picked a *unique* node x in X where he is reconsidering; he will not reconsider elsewhere,

2. at x, the agent remembers his initial strategy b,

3. if the agent switches to a new strategy b' at x, he will remember b' (and may then forget b).

The picture here is that the agent (believes that he) follows strategy b up to some point $x \in X$ chosen by some process, and then reconsiders at x. If he decides to continue with b, he does so for the rest of the game. If he switches to b', then he uses b' for the rest of the game, and remembers that he is using it.[2]

Keeping this picture in mind, let us consider the PR notion of time consistency. Suppose the agent is reconsidering at node x in information set X, and has been using strategy b. The agent does not know that he is at x; all he knows is that he is in X. His subjective belief that he is at x will depend on his belief that x is the node that the process chose (recall that Assumption 1 says that the process picks a *unique* node in X); let $\mu_b(x|X)$ denote this subjective probability. Notice that the agent's subjective probability may depend on the strategy b, which is why μ is subscripted by b, and why it is necessary for the agent to believe that he has been using b up to node x when calculating his belief that he is at x. (This follows from Assumptions 1 and 2.) For each node x in X, it is easy to evaluate the expected utility of a strategy b, starting at x; it is simply $EU(b; x) = \sum_{z \in Z} p_b(z|x)u(z)$. Given these assumptions, it is easy to compare the expected utility of sticking with strategy b to that of switching to strategy b', given our assumption that the agent will remember b' if he does switch to it (so he can play it even when is in a different information set) and will not make any further changes. This leads to PR's definition of time consistency.

Definition 3.1: A behavior strategy b is *PR time consistent* if for all information sets X such that $p_b(X) > 0$ and all strategies b', we have

$$\sum_{x \in X} \mu_b(x|X)EU(b; x) \geq \sum_{x \in X} \mu_b(x|X)EU(b'; x).$$

The agent's belief $\mu_b(x|X)$ that he is at x depends on his beliefs regarding how the node at which he reconsiders is chosen. Although, in principle, the definition makes sense for any choice of μ_b, I focus here on one particular choice. This choice is generated by a process that chooses a decision node uniformly at random before the game starts. If the agent reaches that node, he gets to reconsider there. Clearly, if there are N decision nodes, the probability that the agent reaches x and gets to reconsider there is $p_b(x)/N$. Thus, if the agent is reconsidering at information set X, then his subjective probability that he is reconsidering at x is $\mu_b(x|X) = p_b(x)/ \sum_{x' \in X} p_b(x')$.

This is the choice of μ_b that PR focus on in their paper. As pointed out in PR, we can think of this distribution as being determined by the following experiment: "Play the game over and over, using strategy b. As soon as a terminal node is reached, start over

[2] I do not mean to imply that this is the only story that one can tell characterizing PR's notion of time consistency. My claim is only that the technical analogues of the assumptions I have made do in fact characterize PR's notion of time consistency, and must follow from any other story that is told.

again." Then $\mu_b(x|X)$ is the expected fraction of times that the agent is at x, given that he is in X. We can also justify these beliefs in terms of a different process, which picks a time t in the interval $[0, T]$ uniformly at random, where T is the length of the longest path in the game tree, and lets the agent reconsider at time t if he is still playing the game at that time. It is easy to see that this again gives us $\mu_b(x|X) = p_b(x)/\sum_{x' \in X} p_b(x')$. (This choice of random process is in fact somewhat in the spirit of the discussion in the appendix of [Aumann, Hart, and Perry 1997].)

The assumption that there is some process external to the original game choosing the node at which the agent reconsiders introduces a significant new feature to the game. Although the particular process considered here, which chooses among the nodes uniformly at random, seems innocuous, it is not. To see this, consider the game tree described in Figure 3. It is easy to see that μ_b ascribes a belief of 1/3 to each node in the information

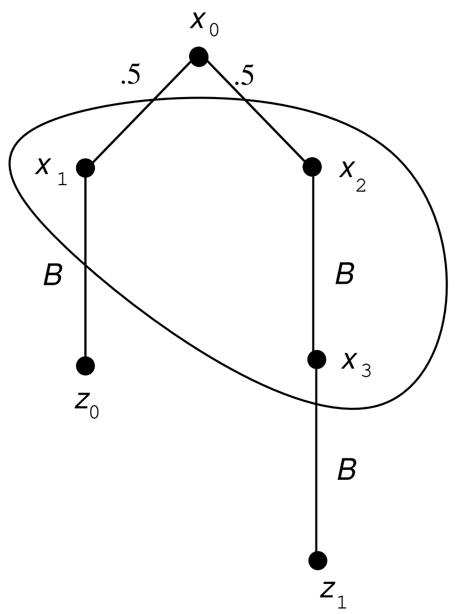

Figure 3: The impact of μ_b.

set. Thus, although nature was equally likely to go left as right, at the information set, the agent believes he is twice as likely to be on the left path as on the right path. If a node is picked at random, it is more likely to be on the longer path. Further discussion of the impact of the way the node is chosen can be found in [Grove and Halpern 1997].

The other assumptions that lead to PR time consistency are also quite strong. As was observed in the introduction, the ability to remember the last strategy chosen clearly gives extra information in the game of Figure 2. As we shall see in Section 4.4, being able to remember the last strategy chosen allows the agent to simulate perfect recall, and thus do better than the optimal strategy of exiting with probability 1/3 in the absentminded driver game. The assumption that the agent believes he will reconsider only once also plays a significant role. For example, if the agent believes he may have switched strategies before, then $\mu_b(x|X)$ is not the appropriate degree of belief for him to assign to being at x, given that he is in X. In general, the appropriate degree of belief depends on the agent's beliefs regarding what strategy (or strategies) he used to get to node x.[3] Similarly, if the agent believes that he might change strategies in the future, he should not use $EU(b';x)$ in the expression above, but rather should consider all possible changes to b'.

3.2 Modified multi-self time consistency

PR motivate their notion of modified multi-self consistency in terms of the *multi-self* view of decision making [Strotz 1956], where an "agent" is viewed as a "team", or a collection of "selves", one associated with each information set. Each "self" makes his choice independently. In Strotz's original approach, the selves may also have different payoff functions; PR consider a variant of this approach where the selves share the same payoff function (which makes sense if we think of the selves as members of the same team). Since the agents are independent, a change made by one agent will not be known to the other agents. This can be modeled equivalently (at least as far as time consistency is concerned) by removing the last of the three assumptions characterizing PR time consistency. With this change, all the agent can assume is that he may deviate from his strategy at the current node, but otherwise will follow it.

Definition 3.2: Let $a(x)$ denote the node that results from taking action $a \in A(x)$ at x. A strategy b is *modified multi-self consistent* if for every action $a \in A(X)$, we have

$$\sum_{x \in X} \mu_b(x|X) EU(b;x) \geq \sum_{x \in X} \mu_b(x|X) EU(b;a(x)).^4$$

As shown by PR, optimal strategies are modified multi-self consistent. Thus, an agent who starts out using an optimal strategy will not switch to another strategy when reconsidering, given the assumption that the choice of when to reconsider is made by an external choice that chooses a node for reconsideration uniformly at random. However, it is easy to see that a modified multi-self consistent strategy is not necessarily optimal. For example, in the simple game (with perfect recall) described in Figure 4, the strategy of playing L at every node is modified multi-self consistent, but not optimal. In retrospect,

[3] In games of perfect recall, it is not hard to show that $\mu_b(\cdot|X)$ is actually independent of b. This is not necessarily true for games of imperfect recall, and is one of the many complications that arise in considering such games.

[4] This definition is in the spirit of that of action optimality given in [Aumann, Hart, and Perry 1997]. PR's definition of modified multi-self consistency is somewhat different. According to their definition, b is modified multi-self consistent if for all actions $a \in A(X)$ and all actions $a' \in A(X)$ such that $b_X(a') > 0$, we have

$$\sum_{x \in X} \mu_b(x|X) EU(b;a'(x)) \geq \sum_{x \in X} \mu_b(x|X) EU(b;a(x)).$$

A straightforward argument shows that the two definitions are equivalent.

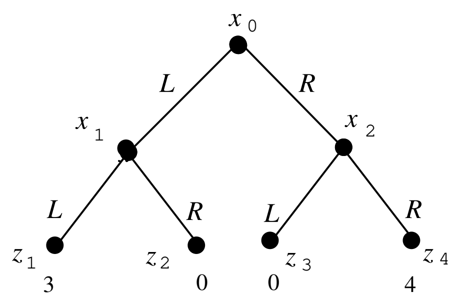

Figure 4: A game where an ms time consistent strategy is not optimal.

this observation is not too surprising. As observed by PR, a modified multi-self consistent strategy is an equilibrium, in the sense that the team does not gain by any team member unilaterally defecting from the strategy. As this example shows, there may well be equilibrium strategies that are not optimal.

3.3 Ms time consistency

In both PR time consistency and modified multi-self consistency, the choice of when to reconsider is made according to some process external to the game, which picks a node x in some information set X. Suppose instead we view the reconsideration process as depending only on the information that the agent has. Put another way, this means that the process is not picking a node at which the agent reconsiders; rather, it is picking the information set X at which the agent reconsiders. If the agent reconsiders at one node in X, then he reconsiders at all of them, since reconsideration depends only on his information.

This change leads to two further notions of time consistency, called *gt* (for *game tree*) time consistency and *ms* (for *multi-self* time consistency), which differ in the same way that PR and modified multi-self consistency differ, namely, in whether the agent remembers his new strategy.

In this section, I consider ms time consistency. In this case, the agent believes that

1. some process (external to the game) has picked a unique information set X where he is reconsidering; he will not reconsider elsewhere,

2. the agent remembers his initial strategy b.

Because the agent reconsiders at every point in X, if he does deviate from his original strategy b, he ends up following a strategy that may differ from b at X. Since the agent reconsiders only at X, and does not recall the fact that he has switched strategies, his new strategy will agree with b off X. Unlike PR and modified multi-self consistency, with ms time consistency, how the information set X at which the agent can reconsider is chosen is irrelevant, just as long as all information sets (at least, all information sets X such that $p_b(X) > 0$) are chosen with positive probability.

Definition 3.3: The expected utility of b, denoted $EU(b)$, is $\sum_{z \in Z} p_b(z)u(z)$. A behavior strategy b is *ms time consistent* if for all information sets X and all strategies b' that agree with b off X (i.e., such that $b'(X') = b(X')$ for $X' \neq X$), we have

$$EU(b) \geq EU(b').$$

It follows immediately from the definition that an optimal strategy is ms time consistent. The example in Figure 4 shows that the converse may not hold.

How do ms and modified multi-self consistency compare? It is easy to see that in games without absentmindedness, the two notions agree.

Observation 3.4: *If Γ is a game without absentmindedness, then b is modified multi-self consistent iff b is ms time consistent.*

In games with absentmindedness, the two notions may differ. The following technical characterization of modified multi-self consistency helps to delineate the differences.

Definition 3.5: Given a strategy b, an information set X, and two actions $a_1, a_2 \in A(X)$, we say that b' is an (a_1, a_2, X)*-variant* of b if b agrees with b' off X, and $b_X(a) = b'_X(a)$ for all actions $a \notin \{a_1, a_2\}$. For $\alpha \leq b_X(a_1) + b_X(a_2)$, let b_α be the (a_1, a_2, X)-variant of b such that $b_\alpha(a_1) = \alpha$, and let $V_{a_1,a_2}(\alpha)$ be the expected utility of b_α. ∎

As mentioned earlier, PR show that an optimal strategy is modified multi-self consistent. Their proof actually shows much more. Indeed, it shows the following.

Proposition 3.6: *A strategy b is modified multi-self consistent iff for each information set X such that $p_b(X) > 0$ and for all $a_1, a_2 \in A(X)$, we have (1) if $b_X(a_1) = 0$, then $V'_{a_1,a_2}(b_X(a_1)) \leq 0$, (2) if $b_X(a_2) = 0$, then $V'_{a_1,a_2}(b_X(a_1)) \geq 0$, and (3) if $b_X(a_1) > 0$ and $b_X(a_2) > 0$, then $V'_{a_1,a_2}(b_X(a_1)) = 0$.*

Corollary 3.7: *If b is ms time consistent then it is modified multi-self consistent.*

The converse to Corollary 3.7 does not hold. Consider the modification of the absent-minded driver example where the payoffs at z_1, z_2, and z_3 are 1, 0, and 2 respectively, instead of 0, 4, and 1. An easy computation shows that if b_α is the strategy of not exiting (that is, taking action B) with probability α, then $EU(b_\alpha) = 2\alpha^2 - \alpha + 1$. Clearly, the optimal strategy is b_1, never exiting, which gives a payoff of 2. In the notation of Definition 3.5, $EU(b_\alpha)$ is $V_{B,E}(\alpha)$. Note that $V'_{B,E}(\alpha) = 4\alpha - 1$. It follows from Proposition 3.6 that each of b_0, $b_{1/4}$, and b_1 is modified multi-self consistent. On the other hand, since there is only one information set in this game, it follows that b_1 is the only ms time consistent strategy.

To summarize, we have shown that the set of optimal strategies is a subset of the ms time consistent strategies, which is a subset of the modified multi-self consistent strategies. In general, the containment is strict.

3.4 Gt time consistency

Gt (game tree) time consistency is like PR time consistency in that the agent is assumed to remember his new strategy if he switches strategies and like ms time consistency in that the question of whether the agent reconsiders his strategy depends only on the agent's information. There is one more twist to the definition of gt time consistency: the agent is assumed to switch strategies only once. Intuitively, the motivation for this is that if the agent switches, then he is switching to his optimal strategy, so there is no need for him to switch again. This intuition is flawed: As we shall see in Section 4.4, if the agent can remember that he has switched strategies, then he may be able to do better by not switching immediately to the optimal strategy. Rather, he may want to use the switch as a way of encoding information to allow him to simulate perfect recall. Thus, it is perhaps better to view the restriction that only one switch is allowed as a way to prevent the use of the strategy as an encoding device.

To summarize, gt time consistency makes sense if we assume that the agent believes that

1. some process external to the game has picked a unique information set X where he is reconsidering; he will not reconsider elsewhere, and he will only reconsider once in X,
2. the agent remembers his initial strategy b up to the time that he reconsiders,
3. after reconsidering, if the agent switches to a new strategy b', then he remembers b'.

Let the *upper frontier* of an information set X, denoted $\hat{X}$, consist of all those nodes $x \in X$ such that there is no node $x' \in X$ that precedes x on some path from the root. For example, in the absentminded driver game, $\hat{X}_e = \{e_1\}$. It is easy to see that, given the assumptions of gt time consistency, if the agent switches strategies at all in an information set X, then he does so at the upper frontier of X: Whatever considerations would have led the agent to switch strategies at some node $x \in X$ would have led him to switch strategies at the (unique) node $x' \in \hat{X}$ such that $x' \preceq x$. (This is precisely where the assumption that whether the agent switches depends only on his information comes into play.)

These considerations lead to the following definition.

Definition 3.8: A behavior strategy b is *gt ("game tree") time consistent* if for all information sets X such that $p_b(X) > 0$ and all strategies b', we have

$$\sum_{x \in \hat{X}} \mu_b(x|\hat{X}) EU(b; x) \geq \sum_{x \in \hat{X}} \mu_b(x|\hat{X}) EU(b'; x).$$

Thus, the difference between gt time consistency and PR time consistency is the use of $\hat{X}$ rather than X.

How different are PR and gt time consistency? As is shown below, they differ only in games that exhibit absentmindedness. These are precisely the games where $\hat{X}$ may differ from X. For example, in the absentminded driver game of Figure 1, let d be the deterministic strategy where the driver never exits, and d' the deterministic strategy where he always exits. It is easy to see that $\mu_d(e_1) = \mu_d(e_2) = 1/2$, while $\mu_{d'}(e_1) = 1$ and $\mu_{d'}(e_2) = 0$. According to the PR definition, the driver who starts by using strategy d should switch to d' at information set X_e, since the expected utility of switching to d' is 2 $(0 \times 1/2 + 4 \times 1/2)$ while the expected utility of sticking with d is only 1. This is a correct calculation if the choice regarding the node at which reconsideration takes place is made by a random process, that chooses according to μ_d. On the other hand, if we assume that the decision as to whether to switch is based on the information the driver has, then whatever decision the driver makes at e_2, he knows he would have made the same decision at e_1. Since the expected utility of switching at e_1 is 0, he should not switch.

In the case of perfect recall, it is well known that a strategy is optimal iff it is PR time consistent. In fact, PR prove that this is true in games with perfect recall of information sets, provided that there is no absentmindedness.[5] PR time consistency and gt time consistency coincide in games with no absentmindedness (since $X = \hat{X}$ for all information sets X in games with no absentmindedness), so in such games, it is also the case that a strategy is optimal iff it is gt time consistent. In fact, an even stronger result holds for gt time consistency.

Theorem 3.9: *If Γ is a game of partial recall, then b is optimal iff b is gt time consistent.*

Proof: See the appendix. ∎

Since the absentminded driver game is one of partial recall, it follows from Theorem 3.9 that the only gt time consistent strategy in that game is the optimal strategy of exiting with probability 1/3. On the other hand, as PR show, the only PR time consistent strategy in that game is to exit with probability 5/9.

Once we move beyond games of partial recall, the optimal strategy may not be gt time consistent. Consider the game in Figure 2 (which is not a game of partial recall). In this game, PR time consistency and gt time consistency coincide, and they do not agree with optimality. At the node x_1, it is clear that the agent should switch from f to f'. He can do better than his optimal strategy by switching. In fact, even in games of partial recall, if the agent can switch strategies, he may be able to do better than his optimal strategy. In particular, the absentminded driver can improve on his optimal strategy if he is allowed to switch strategies (see Section 4.4). What is going on here?

[5]Actually, they show it for a condition weaker than perfect recall of information sets, but they do restrict to games with no absentmindedness.

As I observed in the Introduction, an agent who (quite rationally) decides to switch from f to f' at node x_1 ends up playing the optimal strategy for the game where x_3 and x_4 are distinguishable: that is, he chooses action B at both x_1 and x_2, L at x_3, and R at x_4. This suggests that the ability to switch strategies at x_1 gives the agent the ability to pass on information that allows him to distinguish x_3 from x_4. The information sets are simply not capturing the agent's knowledge in this game. This intuition is made precise in the Section 4.1, where an arguably clearer model of the agent's information is presented.

3.5 Other notions of time consistency

I have considered four notions of time consistency here. It is clearly possible to define others; I briefly mention two.

Battigalli [1995] considers a notion he calls *constrained time consistency*. Roughly speaking, while modified multi-self time consistency allows only deviations at a single node, constrained time consistency also allows deviations at information sets that are *irrelevant*, in that they are reached with probability 0. (See [Battigalli 1995] for a formal definition.) It is also possible to modify ms time consistency in a way analogous to the way modified multi-self time consistency is modified to get constrained time consistency. (See an earlier version of this paper [1995] for the formal definition and a comparison of the resulting notions.) However, it is not clear what the motivation would be for allowing reconsideration at irrelevant nodes in the context of time consistency (although it may be appropriate in other contexts).

Another possibility is to drop the assumption that reconsideration takes place only at one node, or at one information set. This starts to get complicated, since, in general, we must then model the agent's beliefs regarding how likely he is never to switch strategies again, or to switch strategies only once more, and so on. Any particular set of assumptions regarding these issues can, of course, be modeled.

No matter what notion of time consistency we consider, we must carefully model the underlying assumptions. In the next section, I present a framework for doing so.

4 Representing Games as Systems

PR and Aumann, Hart, and Perry also discuss the assumptions that underlie their notions of consistency. Their assumptions seem quite different from the ones here. In particular, the assumption that the agent believes that he will reconsider only once, which I have argued plays a key role in PR time consistency and modified multi-agent time consistency, is not mentioned explicitly in either paper. English is notoriously slippery. Given the ambiguity that we have seen in making sense of time consistency in games with absentmindedness, it becomes particularly important to express these assumptions within the model, not outside it.

Each of the assumptions considered in the previous section can in fact be captured by an appropriate expanded game tree. There is no difficulty capturing the assumption that a random process chooses a node or an information set at which the agent gets to reconsider. This just amounts to adding an extra move for nature at the beginning of the game. Capturing the assumption that the agent does or does not remember his strategy is more subtle. Since an information set consists of a set of nodes in the game tree, at best, it can model only what an agent knows about the moves that have been made. Thus, if we want to use an agent's information set to model information about strategies that other agents are using (or information about the strategy the agent herself is using), it is generally necessary to include strategy choices as moves in the game.[6]

Information sets are awkward in other contexts too (that do not arise in analyzing time consistency). For example, it is difficult to use information sets to model future moves or, indeed, anything about the future. (As Rubinstein [1991] observes, it is difficult to model considerations such as "I know that if I reach the second intersection I will have doubt of 10% that I am at the first intersection".) Moreover, traditionally, an agent's information sets consist only of nodes where the agent is about to move. (This is not a universal assumption; for example, it is not made by Rasmusen [1989].) Nodes where agent 1 does not move are not part of any information set for agent 1. The traditional notion of knowledge says that an agent knows a fact if it is true at all nodes in his information set. This means that we cannot make sense out of statements such as "agent 1 knows that agent 2 knows that agent 1 has moved left", let alone "it is common knowledge that agent 1 has moved left", because a node that is in some information set for agent 1 is not in any of agent 2's information sets. Battigalli and Bonnano [1996] propose one particular approach for making sense of such statements in games of perfect recall, but this is clearly not a general solution to the problem.

So how can we model such considerations? There is no difficulty doing so using the standard states-of-the-world approach first used in the economics literature by Aumann [1976] (which actually is a variant of the standard possible-worlds model for knowledge in the philosophical literature that goes back to Hintikka [1962]; see [Fagin, Halpern, Moses, and Vardi 1995] for discussion). In this approach, each state in a state space Ω is a complete description of the world, which includes what happened in the past and what will happen in the future, the agents' beliefs and knowledge, and so on. The trouble with this approach is that, while it does a good job of capturing an agent's knowledge, it does not do such a good job of describing the play of the game—who moves when, and what the possible moves are. Moreover, because time is not explicit in this approach, it becomes difficult to model statements such as "I know now that after I move my opponent will not know ...".

I describe in the next subsection (a slightly simplified version of) an approach that goes back to [Halpern and Fagin 1989] and can be viewed as combining features of both

[6]Of course, in a given information set X, an agent does have some minimal information regarding strategies: She knows that a strategy that did not lead to X is impossible.

game trees and the states-of-the-world approach. It has been used quite successfully in dealing with computer science problems [Fagin, Halpern, Moses, and Vardi 1995]. In this approach, a game is represented as a multi-agent *system*. In the description of the system, the actual play of the game is distinguished from what goes on in the agent's mind.

4.1 The framework

To describe the agent's state of mind, we assume that, at any point in time, the agent is in some *state*. Occasionally this is called a *local state*, to distinguish it from a *global state*, which is defined below. The local state is essentially what is called an agent's *type* in the game theory literature. Intuitively, it encapsulates all the information to which the agent has access. Deciding how to model the state can be quite a nontrivial issue. 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 has about the strategies of the other players. Alternatively, a forgetful player may not remember all the details of the bets made by the other players; his state would reflect this.

To describe the external world, we use an *environment*, which is also in some state at any point in time. Roughly speaking, the environment's state describes everything relevant to the system that is not part of the agents' states. For example, when describing a game, we can take the environment's state to consist of the sequence of actions that have been taken up to a certain point. If we do this, we can essentially identify the possible environment states with the nodes in the game tree.

The system as a whole can be described by a *global state*, a tuple of the form $(\ell_e, \ell_1, \ldots, \ell_n)$, where ℓ_e is the environment's state, and ℓ_i is agent i's state, $i = 1, \ldots, n$. A global state describes the system at any given point in time. We are typically interested in dynamic systems that change over time. A *run* is a function from time (which is taken for simplicity to range over the natural numbers) to global states. Intuitively, a run is a complete description of how the system's global state evolves over time. For example, when analyzing a game, a run could be a particular play of a game. Thus, if r is a run, $r(0)$ describes the initial global state of the system, $r(1)$ describes the next global state, and so on. A *point* is a pair (r, m) consisting of a run r and time m. If $r(m) = (\ell_e, \ell_1, \ldots, \ell_n)$, let $r_i(m) = \ell_i$. Thus, $r_i(m)$ is agent i's local state at the point (r, m).

Finally, a *system* is formally defined to be a set of runs. Intuitively, a system is being identified with its set of possible behaviors. Thus, for example, the game of bridge can be identified with all the possible games of bridge that can be played (where a run describes a particular game, by describing the deal of the cards, the bidding, and the play of the hand).

Notice that information sets are conspicuously absent from this definition. Information sets in fact do not have to be specified exogenously; they can be reconstructed from the local states. Given a system, that is, a set $\mathcal{R}$ of runs, we can define an equivalence relation on the points in $\mathcal{R}$. The point (r, m) is *indistinguishable from* (r', m') *by agent* i, denoted $(r, m) \sim_i (r', m')$, if $r_i(m) = r'_i(m)$. Thus, two points are indistinguishable by agent i if agent i has the same local state at each point. Clearly $\sim_i$ is an equivalence relation. The $\sim_i$ relations can be viewed as defining information sets. However, note that even a point where agent i does not move is in an information set for agent i.

Actions play an important role in generating systems. Typically we think of them as being generated by a protocol. In [Fagin, Halpern, Moses, and Vardi 1995], there is a notion of a *context* in which the agents are embedded (i.e., the context in which the game is played), which includes the actions that can be taken. Formally, an action is just a global state transformer, that is, a function from global states to global states. A *protocol* for an agent in this setting is a function from that agent's local states to a probability distribution over actions. A protocol is the analogue of a strategy, and captures the intuition that what an agent does can depend only on his information. At two points where the agent has the same information, the agent must do the same thing.

Actions do not appear in the formal definition of a system. Nevertheless, they often do appear (either implicitly or explicitly) in the global state of the system. That is because we typically think of the runs of the system as being generated by the agents and the environment performing some actions. For example, in modeling the game of bridge, we may well put actions such as bidding into the global state. If the agent knows what action was taken, it would be encoded in his local state. Otherwise, we can encode it in the environment's state if the action is considered relevant to the description of the system. Thus, we can ensure that the actions appear in the global state whenever they are relevant to the description of the system.[7]

Given a protocol and a set of initial global states, we can construct a *computation tree* for that protocol. The successors of a given global state in the tree are the results of actions that can be taken according to the protocol. Each transition has a probability, according to the probability assigned by the protocol to the action corresponding to that transition. The paths in the tree become the runs of the system, and (given a distribution on initial states) we can put a distribution on the runs in the system in the obvious way. Rather than going through all the formal definitions here (they can be found in [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Tuttle 1993]), I shall

[7] With this background, I can explain how perfect recall is defined in the computer science literature [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Vardi 1986]. Borrowing the notation from Section 2, define $exp''(r, m)$ to be the sequence of local states that the agent has gone through, with consecutive repetitions omitted. An agent is said to exhibit perfect recall if, whenever $(r', m') \sim_i (r, m)$, then $exp''(r, m) = exp''(r', m')$. Intuitively, this definition of perfect recall says that the agent remembers everything he ever knew. It does not say that he remembers everything he ever did. He may not remember his actions if his actions were not encoded in his local state to begin with. The reason that consecutive repetitions are omitted is that if an agent has the same local state at a sequence of consecutive states, this just means that time has passed without the agent being aware of it (or of anything else that may have happened, including actions that he has performed). Again, this is not counted against the agent having perfect recall of everything he knew.

focus on some examples of applying these ideas to modeling games.

4.2 From game trees to systems

When modeling a game, we must first decide how to model the states of the agent and the environment. Of course, there are many ways to do this. Here is one approach that should be useful for many games of interest: Given a game Γ, it is typically useful to let the environment state encode (among other things) the current node in the game (or, equivalently, the sequence of actions taken to reach that node). Thus, I assume that environment states have the form $(x, \ldots)$, where $x \in \Gamma$. The "…" includes whatever else might be relevant to the system; we shall see some examples shortly. Similarly, it is typically useful to let the agent's state encode (among other things), what nodes the agent considers possible. Thus, I assume that the agent's local state has the form $(X, \ldots)$, where X is a set of nodes in the tree (intuitively, this is the set of nodes that the agent considers possible) and, again, the "…" incorporates whatever other relevant information the agent may have (for example, about his protocol, or protocols that other agents may be using). Although for some applications it may be useful to take X to be an arbitrary set of nodes, here, for simplicity, I further assume that in a global state of the form $((x, \ldots), (X, \ldots))$, if $x \in D$, then X is the information set containing x.

Just as we associate a set of allowed actions $A(x)$ with each node x in the tree, we associate a set of allowed actions $A(g)$ with each global state g. Actually, it is convenient to take $A(g)$ to be a set of allowed *probability distributions* over actions. By being able to stipulate that only certain distributions over actions are allowed, we can capture cleanly nature's moves, as well as certain constraints on the agents' moves. Suppose g has the form $((x, \ldots), \ell)$; then $A(g)$ depends on whether x is in D, C, or Z.

- If $x \in D$, we could identify $A(g)$ with all possible distributions on the actions in $A(x)$. However, as we shall see, it is sometimes useful to allow more constraints to be imposed, and to allow more general actions. For example, if the agent's local state includes his strategy, then we may want $A(g)$ to consist only of the distribution on $A(x)$ that is determined by the strategy. On the other hand, if the agent can switch strategies, then we may have distributions over actions that can both change the node in the tree and change the agent's strategy. Thus, we assume that the elements in $A((x, \ldots), \ell)$ are distributions over actions of the form $(\ldots, a)$, where $a \in A(x)$. The "…" in $(\ldots, a)$ allows for a general action that may, for example, result in the agent changing his strategy, as well as changing the node in the tree. After performing action $(\ldots, a)$, the global state becomes $((a(x), \ldots), \ell')$, for some appropriate local state ℓ'. I further assume that $A((x, \ldots), \ell)$ depends only on the agent's local state ℓ, so that $A((x, \ldots), \ell) = A((x', \ldots), \ell)$ if $x, x' \in D$. This is the analogue of the assumption that the agent's set of possible actions must be the same at all nodes in the same information set.

- If $x \in C$, then $A((x, \ldots), \ell)$ consists of a single distribution over the possible transitions at node x, as defined by the game tree. More precisely, assume that k is the outdegree of the node x in the game tree. Associate an action c_j, $j = 1, \ldots, k$, with each of the edges leading out from x. Performing action c_j in $((x, \ldots), \ell)$ results in a global state of the form $((c_j(x), \ldots), \ell')$, where $c_j(x)$ is the jth successor of x in the tree, in some fixed ordering of x's successors. If the transition to $c_j(x)$ has probability p_j, then $A((x, \ldots), \ell)$ consists of the unique distribution over $c_1, \ldots, c_k$ that gives c_j probability p_j.

- If $x \in Z$, then $A((x, \ldots), \ell) = \emptyset$; no actions are possible at terminal nodes.

A *context* is a triple $(\mathcal{G}, \mathcal{G}_0, A)$ consisting of a set $\mathcal{G}$ of global states as above, a subset $\mathcal{G}_0$ of $\mathcal{G}$ of initial states, and a function A associating with each global state $g \in \mathcal{G}$ a set $A(g)$ of allowed distributions over actions. Given a protocol π, we say that a run r is *consistent with* π in context $(\mathcal{G}, \mathcal{G}_0, A)$ if (a) $r(0) \in \mathcal{G}_0$ (so that the initial state is a legal initial state), and (b) $r(m) = ((x, \ldots), \ell)$, then (i) if $x \in D$, then $r(m+1) = a(r(m))$ for some a such that $\pi(\ell)$ gives a positive probability, (ii) if $x \in C$, then $r(m+1) = c(r(m))$ for some action c corresponding to a transition out of x in the tree, (iii) if $x \in Z$, then $r(m+1) = r(m)$.[8] The *system generated by* π (in context $(\mathcal{G}, \mathcal{G}_0, A)$) consists of all the runs consistent with π. Given a distribution on the global states in $\mathcal{G}_0$, we can get a distribution on the runs in the system generated by π in a straightforward way, and thus compute the expected utility of protocol π.

4.3 Modeling time consistency

Given a game Γ, I now construct four contexts corresponding to the game, one for each of the four notions of time consistency considered in Section 3. These systems attempt to capture formally the assumptions underlying the four notions.

The environment state is used to capture when reconsideration takes place. Thus, I take the environment state to have the form (x, Y, b), where x is a node in the tree, $Y \subseteq D$, and b is a strategy. Intuitively, x is the current node in the tree, Y is the set of nodes where reconsideration takes place (if a node in Y is reached in the game), and b is the agent's initial strategy. For PR time consistency and modified multi-self consistency, Y is always a singleton; for ms and gt time consistency, Y is an information set.[9]

The agent's local state has the form (X, b, t), where X is a set of nodes in the tree, b is a strategy, and t is either 0 or 1, depending on whether the agent reconsiders his strategy at that state. Note that the local state makes the fact that the agent knows his strategy explicit, by making the strategy part of the local state.

[8] For consistency with [Fagin, Halpern, Moses, and Vardi 1995; Halpern and Fagin 1989], where runs are taken to be infinite, I have assumed that once the system reaches a terminal node, it stays there forever.

[9] Only in the case of gt time consistency is it necessary to encode the agent's initial strategy in the environment state. However, I do it in all cases for uniformity of presentation.

The set $\mathcal{G}$ of global states consists of all global states of the form $((x, Y, b_0), (X, b, t))$ where $t = 0$ if $x \notin Y$ or $b \neq b_0$, $t = 1$ otherwise, and X is the information set containing x if $x \in D$. The choice of X if $x \notin D$ is unimportant for the current discussion; any fixed choice will do. For definiteness, I take $X = \{x\}$ in this case.

Next we must define the allowed actions. This is where PR and gt time consistency differs from ms and modified multi-self consistency. In the case of PR and gt, the agent remembers his new strategy if he chooses to switch strategies; this fact must be encoded in his local state. In the case of ms and modified multi-self consistency, the agent does not remember the change. Thus, I consider two possible sets of distributions on actions, denoted A^1 and A^2, depending on which notion of consistency is being considered. Suppose $g = ((x, Y, b_0), (X, b, t))$. If x is in C or Z, then $A^1(g) = A^2(g)$ is determined as discussed earlier. If $x \in D$, then $A^1(g)$ and $A^2(g)$ depend on whether $t = 0$ or $t = 1$. If $t = 0$, then the agent just performs an action according to strategy b. Thus, $A^1(g) = A^2(g)$, and consists of the unique distribution over the actions in $A(x)$ determined by b. Performing action $a \in A(x)$ in global state $((x, Y, b_0), (X, b, 0))$ results in the unique global state in $\mathcal{G}$ of the form $((a(x), Y, b_0), (X', b, t))$. Note that the strategy b in the agent's local state remains unchanged. The interesting case is if $t = 1$; this is when the agent can reconsider his strategy. There are now two subcases, depending on the notion of time consistency:

- For PR and gt time consistency, the agent can actually switch strategies. If the agent switches to strategy b', he then performs an action according to the distribution determined by b'. Thus, for each strategy b', the set $A^1(g)$ contains a unique distribution over actions of the form (b', a), for $a \in A(x)$. The probability of the action (b', a) is just the probability that b' assigns to a at x, that is, $b'_x(a)$. Performing action (b', a) in global state $((x, Y, b_0), (X, b, 1))$ results in the unique global state in $\mathcal{G}$ of the form $((a(x), Y, b_0), (X', b', 0))$. We encode the fact that the agent remembers his strategy after switching by placing b' in the agent's local state.

- For modified multi-self time consistency and ms time consistency, the agent does not switch strategies when reconsidering, but may deviate from the strategy in his local state when $t = 1$. Thus, $A^2(g)$ consists of all distributions over actions in $A(x)$.[10] The effect of performing an action $a \in A(x)$ in global state g is just as was defined above. In particular, note that the strategy does *not* change.

Given a strategy b, let $\mathcal{G}_b$ consist of all the states in $\mathcal{G}$ of the form $((x, \{y\}, b), (X, b, t))$, where x is the root of the tree. We can capture the intuition that nature chooses a node y at which the agent may reconsider at random by putting the uniform distribution on $\mathcal{G}_b$. Let $\mathcal{G}'_b$ consist of all states in $\mathcal{G}$ of the form $((x, Y, b), (X, b, t))$, where x is the root of the tree and Y is an information set.

[10]The definition of modified multi-self consistency allows only the deviation of performing a particular action, with probability 1. Thus, it may seem that to capture modified multi-self consistency, we should take $A(g)$ to consist only of distributions that give probability 1 to an action in $A(x)$. However, it is easy to see that nothing would change if we allowed a deviation by some distribution over actions in the definition of modified multi-self consistency.

Theorem 4.1:

(a) Strategy b is PR time consistent if and only if the protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}_b, A^1)$, assuming a uniform distribution on the initial states in $\mathcal{G}_b$.

(b) Strategy b is modified multi-self consistent if and only if the protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}_b, A^2)$, assuming a uniform distribution on the initial states in $\mathcal{G}_b$.

(c) The following are equivalent:

(i) Strategy b is ms time consistent.

(ii) The protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}'_b, A^2)$, assuming a uniform distribution on the states in $\mathcal{G}'_b$.

(iii) The protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}'_b, A^2)$ for every distribution on the initial states that assigns positive probability to every state in $\mathcal{G}'_b$.

(d) The following are equivalent:

(i) Strategy b is gt time consistent.

(ii) The protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}'_b, A^1)$, assuming a uniform distribution on the states in $\mathcal{G}'_b$.

(iii) The protocol that never switches strategies is optimal in context $(\mathcal{G}, \mathcal{G}'_b, A^1)$ for every distribution on the initial states that assigns positive probability to every state in $\mathcal{G}'_b$.

The proof of Theorem 4.1 is completely straightforward. The theorem essentially translates the definitions of these notions of time consistency into the systems framework. Several points are worth mentioning though.

- Although the use of the uniform distribution is crucial in parts (a) and (b) of the theorem, any distribution that assigns positive probability to all the initial states suffices for parts (c) and (d). This result stresses the role of the uniform distribution in PR and modified multi-self time consistency. As I observed earlier, the use of the uniform distribution is far from innocuous here.

- The fact that the agent performs the same action whenever he deviates in an information set X follows from the fact that we are using a protocol and the agent's local state is the same at all the global states in X where he deviates.

- The assumptions underlying all the notions of time consistency are all of the form "The agent believes ...". In the systems constructed here, these beliefs become knowledge, in the sense that they are true at all the points the agent considers possible. Thus, these systems can be viewed as encoding the agent's beliefs. Although these systems could be embedded in systems where the agent's beliefs might be false, nothing in the analysis would change if this were done. It is worth stressing that, in any case, the use of local states makes the agent's beliefs explicit.

- There is no difficulty in finding systems corresponding to the notion of constrained time consistency and the other notions alluded to in Section 3.5, and proving a result analogous to Theorem 4.1 for them.

4.4 The effect of being able to recall strategies

To capture all the notions of time consistency considered here, I assumed that the agent (believes that he) remembers his initial strategy up to the point that he reconsiders.[11] In addition, one of the assumptions underlying PR time consistency (and gt time consistency) is that the agent remembers his new strategy if he reconsiders and changes. As the game in Figure 2 illustrates, the latter assumption is very powerful; it can allow us to simulate perfect recall. As I now show, if we drop the assumption made in PR and gt time consistency that the agent reconsiders only once, and allow the agent to reconsider arbitrarily often and remember the most recent strategy used, then we can come arbitrarily close to simulating perfect recall in any game. Intuitively, the ability to remember the most recent strategy used gives the agent a significant amount of encoding power.

One advantage of the framework presented here is that the distinction between strategies and protocols allows us to make precise the intuition that the agent is using a protocol that makes use of his ability the remember the most recent strategy used. Given a game Γ, let $\mathcal{G}_\Gamma$ consist of all global states of the form $(x, (X, b))$, where x is a node in the tree, X is the information set containing x if $x \in D$, and $\{x\}$ otherwise, and b is a strategy. (The environment state no longer has a set Y where the agent reconsiders, since the agent can always reconsider; for similar reasons, there is no need to encode the agent's original strategy in the environment state, nor is there a need for the bit t in the agent's local state.) At a global state $(x, (X, b))$, the agent can perform any action of the form (b', a), where b' is a strategy and a is given positive probability according to b_x. Performing action (b', a) in global state $(x, (X, b))$ results in the unique global state in $\mathcal{G}_\Gamma$ of the form $(a(x), (X', b'))$. Formally, we take $A^\Gamma(x, (X, b)) = A^1((x, \{x\}, b), (X, b, 1))$, where A^1 is as defined in the previous section.

With this setup, consider the absentminded driver example. Let b_α be the strategy where the driver continues (i.e., does not exit) with probability α. Suppose the driver

[11] Actually, it would have sufficed to assume that the agent's beliefs were consistent with his strategy.

uses the following protocol: If his local state is (X_e, b_α), $\alpha \neq 1$, he performs the action (b_1, B). That is, he switches to the strategy of always playing B and performs action B (the only one allowed according to this strategy). On the other hand, if his state is (X_e, b_1), he performs the action (b_0, E). That is, he switches to strategy b_0 and performs action E (again, the only choice). Unless he starts with initial state (X_e, b_1), this protocol is guaranteed to get him home, with payoff 4, just as if he had perfect recall. Essentially what is going on here is that, as long as the driver does not start with strategy b_1, then the driver always knows exactly where he is: at node e_1, he knows he is at node e_1 (from the fact that his current strategy is not b_1), and at node e_2, he knows he is at node e_2 (from the fact that his current strategy is b_1). This is also what is going on in the analysis of the game in Figure 2.

This situation essentially holds in general. To make this precise, we need a few definitions. Given a game Γ, let Γ^{pr} be the game that is identical to Γ except for the agent's information sets. These are given as the coarsest refinement of his information sets in Γ that gives him perfect recall. Of course, if Γ is a game where the agent has perfect recall, then $\Gamma^{pr} = \Gamma$.

An information set X in a game Γ is *nontrivial* if there are at least two actions in $A(X)$. Let g_b be the unique global state $(x, (X, b))$ such that x is the root of Γ.

Theorem 4.2:

(a) If Γ is a game with at least two nontrivial information sets, or if Γ has only one nontrivial information set, but no path in Γ has more than two nodes in that information set, then there is a deterministic protocol π^* and a strategy b such that the expected utility of π^* in context $(\mathcal{G}_\Gamma, \{g_b\}, A_\Gamma)$ is equal to that of the optimal strategy in Γ^{pr}.

(b) If Γ has only one nontrivial information set, then there is a strategy b such that for all $\epsilon > 0$, there is a protocol π^ϵ such that the expected utility of π^ϵ in context $(\mathcal{G}_\Gamma, \{g_b\}, A_\Gamma)$ is within ϵ of that of the optimal strategy in Γ^{pr}.

Proof: See the appendix. ∎

The proof of Theorem 4.2 just generalizes the arguments given earlier for the absentminded driver example. The ability to recall the last strategy used gives the agent encoding power sufficient to simulate perfect recall. This theorem is not terribly interesting insofar as telling us how games should be played. However, it does show the power of the assumption of being able to switch strategies while still recalling the last strategy used.[12]

[12] In [Halpern 1995], a variant of this result is proved: Roughly speaking, define a *level-0 protocol* to be a strategy, and a *level-$(k+1)$ protocol* to be one which allows changes in level-k protocols. It is shown that, for a sufficiently large k, there is a level-k protocol that simulates the optimal protocol in Γ^{pr}. Under the assumptions of part (a) of Theorem 4.2, we can take $k = 1$, so the two results coincide in this case. Under the assumptions of part (b) of Theorem 4.2, we may have $k > 1$, but the k-level protocol is deterministic.

On the other hand, if we assume that the agent can remember his initial strategy, is it so unreasonable to assume that he can also remember his later choices? Perhaps it might be more reasonable to assume that the agent can remember neither his initial choice nor his later choices. This would truly correspond to absentmindedness. However, if we do not assume that the agent can recall his initial choice, then none of the notions of time consistency considered in the literature seem appropriate.

Ambrus-Lakatos [1995] introduced what he called *extended strategies*—which are essentially protocols where the local state encodes the previous strategy used, and the agent has actions which allow him to switch strategy—and used them to as a tool to investigate strategy recall in games of imperfect recall. Notice that the notion of protocol used here is far more general than his notion of extended strategy, since the local state can do much more than just encode the previous strategy used. Ambrus-Lakatos independently proved a theorem similar in spirit to Theorem 4.2.[13]

5 Discussion

What should we make of the various definitions of time consistency? How reasonable are they? That depends on how appropriate the underlying assumptions are; this is up to the modeler to decide. However, the results of the earlier sections shed some light on the impact of the various choices.

The results of Section 4.3 make it clear that the various notions of time consistency involve playing a game different from the original game. This "reconsideration" game involves assuming that the agent remembers the last strategy he chose (in the case of PR and gt time consistency) and some assumptions about when the agent is able to reconsider. In the case of PR time consistency and modified multi-self consistency, the reconsideration game is quite sensitive to the nature of the process that chooses where reconsideration may occur. Here I have focused on the process that may be characterized as choosing the node where reconsideration occurs uniformly at random among all the decision nodes; this process leads to the same subjective beliefs μ_b as considered by PR and Aumann, Hart, and Perry. As the game in Figure 2 and the modification of the absentminded driver game considered in Section 3.3 show, this choice results in reconsideration games for which the optimal strategy may be different from that of the original game, and may even result in a higher payoff.[14] Only in the case of ms consistency are the optimal strategies in the reconsideration game the same as the optimal strategies in the original game.

It should not be surprising that changing the game may result in possibly different

[13]His theorem is not quite correct as stated, since it only has an analogue to part (a) of Theorem 4.2, and thus does not apply to games with only one nontrivial information set and a path that includes more than two nodes from that information set.

[14]Grove and I [1997] provide a characterization of a class of random processes for which the optimal strategies in the reconsideration game and the original game are the same.

behavior. Perhaps what makes these observations surprising is that in the case of perfect recall, all the reconsideration games are in fact equivalent to the original game.

The other moral that should come out of these results is the importance of carefully modeling what is going on. In principle, (the games corresponding to) all the notions of time consistency could have been modeled using (rather complicated) game trees. However, it is not clear that they would have been the best tool. To quote Myerson [1991], "the general form or structure of the models we use to describe games must be carefully considered. A model structure that is too simple may force us to ignore vital aspects of the real games we want to study." It is all too easy, using game trees, to draw information sets without thinking about whether they really are appropriate representations of an agent's knowledge. Typically, issues such as what an agent knows about his strategy are not modeled by information sets.

In the systems framework I have described here, an agent's information is described explicitly at each point in time by his local state, and his behavior is described by means of a protocol, a function from local states to actions. The framework forces on the modeler the discipline of making clear exactly what an agent knows at any time. Thus, it allows a modeler to make an independent argument as to when a game tree *is* an adequate representation of an agent's knowledge.

That leads to an obvious question: Just when is the standard game tree model adequate? The pat answer is that it is adequate when the information set is an adequate description of the agent's information and we do not want to allow actions that can modify the agent's state as well as the environment. In general, adequacy depends on what we are trying to analyze. Clearly the game tree model is perfectly adequate for many analyses involving games of perfect recall, otherwise it would not have survived so long. On the other hand, PR's examples show that when it comes to analyzing time consistency in games of imperfect recall, information sets are inadequate. The difficulty of using game trees to analyze games of imperfect recall may be one reason that they have received relatively little attention in the game theory literature. (By way of contrast, games of imperfect recall are the norm in the computer science literature. Note that any game played by finite automata that is sufficiently long (in particular, that goes on for more steps than the automata have local states), will be a game of imperfect recall, indeed, one that exhibits absentmindedness. Thus, games of imperfect recall arise quite often in practice.) Even in games of perfect recall, game trees cannot be used easily to represent information that players have about other players' strategies. I believe the systems representation could prove useful here as well, especially when it comes to analyzing issues of rationality. I hope to return to this issue in future work.

Although it is beyond the scope of this paper to go into extensive discussion about the systems approach to modeling games, it has some other advantages that are worth pointing out.

- There is no need to assume that agents move sequentially. An "action" may well consist of a tuple of actions, one for each of the agents. (Indeed, this is precisely what is assumed in the more general framework presented in [Fagin, Halpern, Moses, and Vardi 1995].) For example, if we are considering prisoners' dilemma, where a prisoner may either cooperate (C) or defect (D), and we view these actions as happening simultaneously, then the allowed actions could be (C, C), (C, D), (D, C), or (D, D).

- Using standard semantics of knowledge [Fagin, Halpern, Moses, and Vardi 1995], there is no difficulty in this model of making sense out of one agent knowing that another agent knows something, or there being common knowledge, at any point in the system.

- I have largely ignored the question here of how an agent should assign probabilities to events in systems. This issue is discussed and formalized in some detail in [Halpern and Tuttle 1993], at least for systems that correspond to games with perfect recall. (The philosophical issues of what is an "appropriate" distribution in systems corresponding to games of imperfect recall is also briefly discussed.) In any case, once we associate with each agent a probability distribution that characterizes his beliefs, there is no difficulty in making sense out of statements such as "I know now that if I reach the second intersection, I will place probability 1/10 on being at the first intersection".

- There is no difficulty capturing the multiself approach in this framework. Each "self" just becomes another agent.

Of course, given a systems representation of a finite game, it can be viewed as a collection of game trees. By adding a dummy root node, we get a single game tree back. Thus, at some level, it can be argued that the standard game tree model can deal with all the issues I have raised (once we allow an agent's information sets to form a partition of *all* the nodes in a game tree, not just the nodes where the agent moves). As I tried to suggest earlier, this does not always seem to be the best way of looking at things. There are times when it seems useful to distinguish the external game from what is going on in the agent's head. For such situations, the systems representation can be a useful addition to a modeler's arsenal. To back up this argument requires further examples, especially ones showing that the system approach really helps in the analysis of games. This too is something I hope to explore further.

A Appendix: Proofs

In this section, I prove the results stated in Sections 3 and 4. The results are restated here for the reader's convenience.

To prove Theorem 3.9, the following characterization of gt time consistency is useful.

Definition A.1: If Y is a set of nodes in D, then a strategy b' is *identical to* b *off* Y, written $b \approx_Y b'$, if $b(x) = b'(x)$ for all $x \in D - Y$. A strategy b is *optimal on* Y if $EU(b) \geq EU(b')$ for all b' such that $b \approx_Y b'$.

Lemma A.2: *A strategy b is gt time consistent in Γ iff for each information set X such that $p_b(X) > 0$, b is optimal on $R(X)$, where $R(X)$ consists of all the agent's nodes reachable from some node in X.*

Proof: The proposition follows immediately once we show that, for all information sets X with $p_b(X) > 0$ and for all strategies b_0 and b_1 that are identical to b off $R(X)$, we have

$$EU(b_0) \leq EU(b_1) \tag{1}$$

iff

$$\sum_{x \in \hat{X}} \mu_b(x|\hat{X}) EU(b_0; x) \leq \sum_{x \in \hat{X}} \mu_b(x|\hat{X}) EU(b_1; x). \tag{2}$$

Given an information set X with $p_b(X) > 0$, let Z_X consist of all nodes in Z that are preceded by some node in X; i.e., $Z_X = \{z \in Z : x \preceq z \text{ for some } x \in X\}$. Since b_0 and b_1 differ only in the probability of reaching terminal nodes in Z_X, (1) holds iff

$$\sum_{z \in Z_X} p_{b_0}(z) u(z) \leq \sum_{z \in Z_X} p_{b_1}(z) u(z). \tag{3}$$

For each node $z \in Z_X$, there is a unique node $x_z \in \hat{X}$ such that $x_z \leq z$. Since b_i is identical to b off of $R(X)$, it is immediate that $p_{b_i}(z) = p_{b_i}(z|x_z) p_b(x_z)$, for $i = 0, 1$. Thus, for $i = 0, 1$,

$$\sum_{z \in Z_X} p_{b_i}(z) u(z) = \sum_{x \in \hat{X}} p_b(x) EU(b_i; x). \tag{4}$$

It easily follows from the definitions that $\mu_b(x|\hat{X}) = p_b(x)/p_b(\hat{X})$, so $p_b(x) = \mu_b(x|\hat{X}) p_b(\hat{X})$. The equivalence of (1) and (2) now follows. ∎

Theorem 3.9: *If Γ is a game of partial recall, then b is optimal iff b is gt time consistent.*

Proof: Let $X_1, \ldots, X_k$ be the information sets in Γ such that there is no decision node preceding some node in X_i, $i = 1, \ldots, k$. Since Γ is a game of partial recall, it follows that the sets $R(X_i)$ are pairwise disjoint, and that their union is all of D. It now easily follows that b is optimal iff b is optimal on each $R(X_i)$ such that $p_b(X_i) > 0$. The result now follows immediately from Proposition A.2. (Notice that this argument fails if the sets $R(X_i)$ are not disjoint, and hence fails for arbitrary games.) ∎

Theorem 4.2:

(a) If Γ is a game with at least two nontrivial information sets, or if Γ has only one nontrivial information set, but no path in Γ has more than two nodes in that information set, then there is a deterministic protocol π^* and a strategy b such that the expected utility of π^* in context $(\mathcal{G}_\Gamma, \{g_b\}, A_\Gamma)$ is equal to that of the optimal strategy in Γ^{pr}.

(b) If Γ has only one nontrivial information set, then there is a strategy b such that for all $\epsilon > 0$, there is a protocol π^ϵ such that the expected utility of π^ϵ in context $(\mathcal{G}_\Gamma, \{g_b\}, A_\Gamma)$ is within ϵ of that of the optimal strategy in Γ^{pr}.

Proof: Let b^* be the optimal strategy in Γ^{pr}. Since Γ^{pr} has perfect recall, we can assume without loss of generality that b^* is deterministic. For ease of exposition, assume that there are no chance nodes in Γ. This means that each information set in Γ^{pr} is a singleton. Let $x_1, \ldots, x_N$ be the decision nodes on the path of the unique play of the b^*, where x_1 is the root of Γ^{pr}. Suppose $b^*(x_j) = a_j$, for $j = 1, \ldots, N$.

For part (a), first suppose that there are at least two nontrivial information sets in Γ. Let Y_0 and Y_1 be nontrivial information sets. Suppose $a' \in A(Y_0)$ and $a'' \in A(Y_1)$. I now show how to construct a protocol π^* that simulates b^*.

Let b^j, $j = 1, \ldots, N$, be a strategy such that (1) if x_j is in information set X, then $b^j_X(a_j) = 1$, (2) if $X \neq Y_0$, then $b^j_{Y_0}(a') = j/N$ and, if $X \neq Y_1$, $b^j_{Y_1}(a'') = 1$, (3) if $X = Y_0$, then $b^j_{Y_1}(a'') = j/N$. Intuitively, b^j is used to encode of the fact that the agent is at node x_{j+1}. If the agent is at node x_j, then he should use strategy b^{j+1}, since it agrees with b^* at node x_{j+1}. For the purposes of this part of the proof, a strategy is *special* if it is in $\{b_1, \ldots, b_N\}$.

Bearing this intuition in mind, the protocol π^* is defined as follows. Since π^* is deterministic, I take $\pi^*(X, b)$ to be an action (rather than the probability distribution that assigns probability 1 to an action). There are three cases to consider in defining $\pi^*(X, b)$:

1. If b is not special and X does not contain x_1 (the root node), then $\pi^*(X, b) = (b, a)$, where a is some fixed action in $b(X)$ (it does not matter which one).

2. If b is not special and X contains x_1, then $\pi^*(X, b) = (b^1, a_1)$,

3. If $b = b^j$, then $\pi^*_{(X,b)} = (b^{j+1}, a_{j+1})$.

This completes the definition of π^*.

Let b be any nonspecial strategy. I leave it to the reader to show that in any run r such that $r(0) = g_b$, the actions performed by π^* at any global state $(x, (X, b'))$ that arises in r are precisely those performed by b^* at the node x. That is, π^* started in g_b simulates b^*, and thus has the same expected utility as b^*.

Next suppose that there is only one nontrivial information set in Γ, say Y. If there is only one node of Y on the path $x_1, \ldots, x_N$, say y, then an agent a can trivially follow

b^* in Γ, by performing the unique action possible at each information set other than Y, and performing the action $b^*(y)$ at Y.

If there are two nodes of Y on the path $x_1, \ldots, x_N$, then we can proceed much as in the discussion of the absentminded driver example in Section 4.4. Suppose that the nodes of Y on the path are y_1 and y_2, and y_1 precedes y_2. Let b^j, $j = 1, 2$, be the unique strategy such that b^j_Y assigns probability 1 to $b^*(y_j)$. (Note that b^j is determined at all other information sets, since every other information set is trivial.) We now define π as follows

- for all $X \neq Y$, $\pi^*(X, b) = (b, a)$, where a is the unique action possible at x;
- if $b \neq b^1$, then $\pi^*(Y, b) = (b^1, b^*(y_1))$;
- $\pi^*(Y, b^1) = (b^2, b^*(y_2))$.

Again, it is easy to see that $b \neq b_1$, then in any run r such that $r(0) = g_b$, the actions performed by π^* at any global state $(x, (X, b'))$ that arises in r are precisely those performed by b^* at the node x.

If there are more than two nodes of Y on the path $x_1, \ldots, x_N$, then we are in case (b). Let $y_0, \ldots, y_k$ be the nodes in Y that occur on this path, in that order, and suppose $y_j = x_{i_j}$, so that $b^*(y_j) = a_{i_j}$, $j = 1, \ldots, k$. Let a' and a'' be two of the actions in $A(Y)$. (Since Y is nontrivial, we know that $A(Y)$ has at least two actions.) Fix $N > 2k$ and $0 \in \{0, \ldots, k\}$. Let $b^{j,N}$ be the unique strategy such that (1) $b^{j,N}_Y(a_{i_j}) = 1 - (j/N)$, (2) $b^{j,N}_Y(a') = j/N$ if $a' \neq a_{i_j}$, and (iii) $b^{j,N}_Y(a'') = j/N$ if $a' = a_{i_j}$. Essentially, we want to use strategy $b^{j,N}$ if the agent is at y_j, and use $b^{j-1,N}$ to encode the fact that the last node in Y that the agent was in was y_{j-1}. Note that $b^{j,N}$ does not exactly simulate b^* at y_j, since according to $b^{j,N}$, the action a_{i_j} is performed with probability only $1 - (j/N)$, not with probability 1, as it is by b^*. However, we can make the outcomes as close as we like by taking N sufficiently large. A strategy is *N-special* if it is of the form $b^{j,N}$, for some j.

Define protocol π_N as follows:

- For all $X \neq Y$, $\pi_N(X, b)$ assigns probability 1 to (b, a), where a is the unique action possible at X;
- If b is not N-special, then $\pi_N(Y, b)$ assigns probability 1 to $(b^{0,N}, a_{i_0})$,
- If $b = b^{j,N}$, then $\pi_N(Y, b) = (b^{j+1,N}, a_{i_{j+1}})$.

Choose b such that b is not N-special for any N. Fix ϵ. I leave it to the reader to check that we can choose N sufficiently large so that π_N approximates b^* sufficiently well so as to make the expected utility of π_N in context $(\mathcal{G}_\Gamma, \{g_b\}, A_\Gamma)$ be within ϵ of that of b^*. ∎

Acknowledgments

I would like to thank Robert Aumann, Adam Grove, Sergiu Hart, David Kreps, Bart Lipman, Ariel Rubinstein, and Moshe Vardi for useful discussion and their comments on an earlier version of this paper.

References

Ambrus-Lakatos, L. (1995). Decision theory, imperfect recall, and the concept of strategy. mimeo, Princeton University.

Aumann, R. J. (1976). Agreeing to disagree. *Annals of Statistics 4*(6), 1236–1239.

Aumann, R. J., S. Hart, and M. Perry (1997). The absent-minded driver. *Games and Economic Behavior*. This issue.

Battigalli, P. (1995). Dynamic consistency and imperfect recall. Mimeo, Princeton University.

Battigalli, P. and G. Bonanno (to appear, 1996). Synchronic information, knowledge and common knowledge in extensive games. In M. Bacharach, L. A. Gerard-Varet, P. Mongin, and H. Shin (Eds.), *Epistemic Logic and the Theory of Games and Decisions*. Dordrecht, Netherlands: Kluwer.

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 absent-mindedness. *Games and Economic Behavior*. This issue.

Halpern, J. Y. (1995). On ambiguities in the interpretation of game trees. Research Report RJ 9995, IBM.

Halpern, J. Y. and R. Fagin (1989). Modelling knowledge and action in distributed systems. *Distributed Computing 3*(4), 159–179. A preliminary version appeared in *Proc. 4th ACM Symposium on Principles of Distributed Computing*, 1985, with the title "A formal model of knowledge, action, and communication in distributed systems: preliminary report".

Halpern, J. Y. and M. R. Tuttle (1993). Knowledge, probability, and adversaries. *Journal of the ACM 40*(4), 917–962. A preliminary version appears in *Proceedings of the 8th ACM Symposium on Principles of Distributed Computing*, 1989, pp. 103–118.

Halpern, J. Y. and M. Y. Vardi (1986). The complexity of reasoning about knowledge and time. In *Proc. 18th ACM Symp. on Theory of Computing*, pp. 304–315.

Hintikka, J. (1962). *Knowledge and Belief*. Ithaca, N.Y.: Cornell University Press.

Myerson, R. B. (1991). *Game Theory.* Cambridge, Mass.: Harvard University Press.

Piccione, M. and A. Rubinstein (1997). On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*. This issue.

Rasmusen, E. (1989). *Games and Information: An Introduction to Game Theory.* Oxford, U.K. and Cambridge, Mass.: Basil Blackwell.

Rubinstein, A. (1991). Comments on the interpretation of game theory. *Econometrica 59*, 909–924.

Strotz, R. H. (1956). Myopia and inconsistency in dynamic utility maximization. *Review of Economic Studies 23*, 165–180.