Formalizing and Enforcing Purpose Restrictions in Privacy Policies (Full Version)

Michael Carl Tschantz **Anupam Datta**
Jeannette M. Wing

March 22, 2012*
CMU-CS-12-106

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was supported by the U.S. Army Research Office grants DAAD19-02-1-0389 and W911NF-09-1-0273 to CyLab, by the National Science Foundation (NSF) grants CCF0424422 and CNS1064688, and by the U.S. Department of Health and Human Services grant HHS 90TR0003/01. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

[*] On May 10, 2012, the above funding acknowledgements were corrected. No other changes were made.

This technical report is the full version of a conference paper presented at the 33rd IEEE Symposium on Security and Privacy (San Francisco, May 2012) and published in the conference proceedings as "Formalizing and Enforcing Purpose Restrictions in Privacy Policies" [71]. The authors reported some of the material in this technical report in an earlier technical report [70].

Keywords: Privacy, Formal Methods, Auditing, Compliance Checking, Planning, MDPs

Abstract

Privacy policies often place restrictions on the purposes for which a governed entity may use personal information. For example, regulations, such as the Health Insurance Portability and Accountability Act (HIPAA), require that hospital employees use medical information for only certain purposes, such as treatment, but not for others, such as gossip. Thus, using formal or automated methods for enforcing privacy policies requires a semantics of *purpose restrictions* to determine whether an action is *for* a purpose or not. We provide such a semantics using a formalism based on *planning*. We model planning using a modified version of Markov Decision Processes (MDPs), which exclude redundant actions for a formal definition of *redundant*. We argue that an action is for a purpose if and only if the action is part of a plan for optimizing the satisfaction of that purpose under the MDP model. We use this formalization to define when a sequence of actions is *only for* or *not for* a purpose. This semantics enables us to create and implement an algorithm for automating auditing, and to describe formally and compare rigorously previous enforcement methods. To validate our semantics, we conduct a survey to compare our semantics to how people commonly understand the word "purpose".

1 Introduction

Purpose is a key concept for privacy policies. For example, the European Union requires that [69]:

> Member States shall provide that personal data must be [. . .] collected for specified, explicit and legitimate purposes and not further processed in a way incompatible with those purposes.

The United States also has laws placing purpose restrictions on information in some domains such as the Health Insurance Portability and Accountability Act (HIPAA) [54] for medical information and the Gramm-Leach-Bliley Act [73] for financial records. These laws and best practices motivate organizations to discuss in their privacy policies the purposes for which they will use information.

Some privacy policies warn users that the policy provider may use certain information for certain purposes. For example, the privacy policy of a medical provider states, "We may disclose your [protected health information] for public health activities and purposes [. . .]" [74]. Such warnings do not constrain the behavior of the policy provider.

Other policies that prohibit using certain information for a purpose do constrain the behavior of the policy provider. Examples include the privacy policy of Yahoo! Email, which states that "Yahoo!'s practice is *not* to use the content of messages stored in your Yahoo! Mail account *for* marketing purposes" [78, emphasis added].

Some policies even limit the use of certain information to an explicit list of purposes. The privacy policy of The Bank of America states, "Employees are authorized to access Customer Information *for* business purposes *only*." [9, emphasis added]. The HIPAA Privacy Rule requires that health care providers only use protected health information about a patient with that patient's authorization or for a fixed list of allowed purposes, such as treatment and billing [54].

These examples show that verifying that an organization obeys a privacy policy requires a semantics of *purpose restrictions*. In particular, enforcement requires the ability to determine that the organization obeys at least two classes of purpose restrictions. Yahoo!'s privacy policy shows an example of the first class: a rule requiring that an organization does *not* use certain information *for* a purpose. HIPAA provides an example of the second class: a rule requiring that an organization use certain information *only for* a given list of purposes. We call the first class of restrictions *prohibitive rules* (not-for) and the second class *exclusivity rules* (only-for). A prohibitive rule disallows an action for a particular purpose. An exclusivity rule disallows an action for every purpose other than the exceptions the rule lists. Each class of rule requires determining whether the organization's behavior is *for* a purpose, but they differ in whether this determination indicates a violation or compliance.

Manual enforcement of privacy policies is labor intensive and error prone [30]. Thus, to reduce costs and build trust, organizations should automate the enforcement of their privacy policies; tool support for this activity is emerging in the market. For example, Fair Warning sells automated services to hospitals for detecting privacy breaches [30]. Meanwhile, previous research has proposed formal methods to enforce purpose restrictions [2, 20, 37, 3, 57, 38, 53, 29].

However, each of these endeavors starts by assuming that actions or sequences of actions are labeled with the purposes they are *for*. They avoid analyzing the meaning of *purpose* and provide no method of performing this labeling other than through intuition alone. The absence of a formal semantics to guide this determination has hampered the development of methods for ensuring policy compliance. Such a definition would provide insights into how to develop tools that identify suspicious accesses in need of detailed auditing and algorithms for determining whether an action could be for a purpose. It would also show which enforcement methods are most accurate. More fundamentally, it could frame the scientific basis of a societal and legal understanding of purpose and of privacy policies. Such a foundation can, for example, guide implementers as they codify in software an organization's privacy policies.

The goal of this work is to study the meaning of *purpose* in the context of enforcing privacy policies. We aim to provide formal definitions suitable for automating the enforcement of purpose restrictions. We focus on automated auditing since we find that post-hoc auditing by a trusted auditor provides the perspective often required to determine the purpose of an action. However, we believe our semantics is applicable to other enforcement mechanisms and may also clarify informal reasoning. For example, in Section 5.3, we use it to create an operating procedure that encourages compliance with a purpose restriction.

We find that *planning* is central to the meaning of purpose. We see the role of planning in the definition of the sense of the word "purpose" most relevant to our work [1]:

> The object for which anything is done or made, or for which it exists; the result or effect intended or sought; end, aim.

Similarly, work on cognitive psychology calls purpose "the central determinant of behavior" [27, p. 19]. In Section 2, we present an example making this relationship between planning and purpose explicit. We (as have philosophers [68]) conclude that if an auditee (the person or organization being audited) chooses to perform an action a while planning to achieve the purpose p, then the auditee's action a is *for the purpose* p. Our goal is to make these notions formal in a manner useful for the automation of auditing.

In Section 3, we present a formalism based upon these intuitions. We formalize planning using Markov Decision Processes (MDPs) and provide semantics to purpose restrictions based upon planning with MDPs. Section 4 provides an auditing method and discusses the ramifications of the auditor observing only the behaviors of the auditee and not the underlying planning process of the auditee that resulted in these behaviors. We characterize circumstances in which the auditor can still acquire enough information to determine that the auditee violated the privacy policy. To do so, the auditor must first use our MDP model to construct all the possible behaviors that the privacy policy allows and then compare it with all the behaviors of the auditee that could have resulted in the observed auditing log. Section 5 presents an implemented algorithm for auditing based on our formal definitions and also shows how to use it to create an operating procedure that encourages compliance with a purpose restriction.

To validate our semantics, we perform an empirical study. In Section 6, we present the results of a survey testing how people understand the word "purpose". The survey compares our planning based method to the prior method based on whether an action improves the satisfaction of a purpose. We find that our method matches the survey participants' responses much more closely than the prior method.

In Section 7, we use our formalism to discuss the strengths and weaknesses of each previous method. In particular, we find that each method enforces the policy given the set of all possible allowed behaviors, which is a set that our method can construct. We also compare the previous auditing methods, which differ in their trade-offs between auditing complexity and accuracy of representing this set of behaviors. Section 8 discusses other related work.

Our work makes the following contributions:

1. The first semantic formalism of when a sequence of actions is for a purpose;

2. Empirical validation that our formalism closely corresponds to how people understand the word "purpose";

3. An algorithm employing our formalism and its implementation for auditing; and

4. The characterization of previous policy enforcement methods in our formalism and a comparative study of their expressiveness.

The first two contributions illustrate that planning can formalize purpose restrictions. The next two illustrate that our formalism may aid automated auditing and analysis. While we view these results as a significant step towards enforcement of practical privacy policies with purpose restrictions, we recognize that further work is needed before we will have audit tools that are ready for use in organizations that must comply with complex policies. We outline concrete directions for future work towards this goal in Section 9.

Although motivated by our goal to formalize the notions of *use* and *purpose* prevalently found in privacy policies, our work is more generally applicable to a broad range of policies, such as fiscal policies governing travel reimbursement or statements of ethics proscribing conflicts of interest.

2 Motivation of Our Approach

We start with an informal example that suggests that *an action is for a purpose if the action is part of a plan for achieving that purpose.* Consider a physician working at a hospital who, as a specialist, also owns a private practice that tests for bone damage using a novel technique for extracting information from X-ray images. After seeing a patient and taking an X-ray, the physician forwards the patient's medical record including the X-ray to his private practice to apply this new technology. As this action entails the transmission of protected health information, the physician will have violated HIPAA if this transmission is not for one of the purposes HIPAA allows. The physician would also run afoul of the hospital's own policies governing when outside consultations are permissible unless this action was for a legitimate purpose. Finally, the patient's insurance will only reimburse the costs associated with this consultation if a medical reason (purpose) exists for them. The physician claims that this consultation was for reaching a diagnosis. As such, it is for the purpose of treatment and, therefore, allowed under each of these policies. The hospital auditor, however, has selected this action for investigation since the physician's making a referral to his own private practice makes possible the alternate motivation of profit.

Whether or not the physician violated these policies depends upon details not presented in the above description. For example, we would expect the auditor to ask questions such as:

1. Was the test relevant to the patient's condition?

2. Did the patient benefit medically from having the test?

3. Was this test the best option for the patient?

We will introduce these details as we introduce each of the factors relevant to the purposes behind the physician's actions.

States and Actions. Sometimes the purposes for which an agent takes an action depend upon the previous actions and the state of the system. In the above example, whether or not the test is relevant depends upon the condition of the patient, that is, the state that the patient is in.

While an auditor could model the act of transmitting the record as two (or more) different actions based upon the state of the patient, modeling two concepts with one formalism could introduce errors. A better approach is to model the state of the system. The state captures the context in which the physician takes an action and allows for the purposes of an action to depend upon the actions that precede it.

The physician's own actions also affect the state of the system and, thus, the purposes for which his actions are. For example, had the physician transmitted the patient's medical record before taking the X-ray, then the transmission could not have been for treatment since the physician's private practice only operates on X-rays and would have no use for the record without the X-ray.

The above example illustrates that when an action is for a purpose, the action is part of a sequence of actions that can lead to a state in which some goal associated with the purpose is achieved. In the example, the goal is reaching a diagnosis. Only when the X-ray is first added to the record is this goal reached.

Non-redundancy. Some actions, however, may be part of such a sequence without actually being for the purpose. For example, suppose that the patient's X-ray clearly shows the patient's problem. Then, the physician can reach a diagnosis without sending the record to the private practice. Thus, while both taking the X-ray and sending the medical record might be part of a sequence of actions that leads to achieving a diagnosis, the transmission does not actually contribute to achieving the diagnosis: the physician could omit it and the diagnosis could still be reached.

From this example, it may be tempting to conclude that an action is *for* a purpose only if that action is *necessary* to achieve that purpose. However, consider a physician who, to reach a diagnosis, must either send the medical record to a specialist or take an MRI. In this scenario, the physician's sending the record to the specialist is not necessary since he could take an MRI. Likewise, taking the MRI is not necessary. Yet, the physician must do one or the other and that action will be for the purpose of diagnosis. Thus, an action may be for a purpose without being necessary for achieving the purpose.

Rather than *necessity*, we use the weaker notion of *non-redundancy* found in work on the semantics of *causation* (e.g., [48]). Given a sequence of actions that achieves a goal, an action in it is *redundant* if that sequence with that action removed (and otherwise unchanged) also achieves the goal. An action is *non-redundant* if removing that action from the sequence would result in the goal no longer being achieved. Thus, non-redundancy may be viewed as necessity under an otherwise fixed sequence of actions.

For example, suppose the physician decides to send the medical record to the specialist. Then, the sequence of actions modified by removing this action would not lead to a state in which a diagnosis is reached. Thus, the transmission of the medical record to the specialist is non-redundant. However, had the X-ray revealed to the physician the diagnosis without needing to send it to a specialist, the sequence of actions that results from removing the transmission from the original sequence would still result in a diagnosis. Thus, the transmission would be redundant.

Quantitative Purposes. Above we implicitly presumed that the diagnosis from either the specialist or an MRI had equal quality. This need not be the case. Indeed, many purposes are actually fulfilled to varying degrees. For example, the purpose of marketing is never completely achieved since there is always more marketing to do. Thus, we model a purpose by assigning to each state-action pair a number that describes how well that action fulfills that purpose when performed in that state. We require that the physician selects the test that maximizes the quality of the diagnosis as determined by the total purpose score accumulated over all his actions.

We must adjust our notion of non-redundancy accordingly. An action is non-redundant if removing that action from the sequence would result in the purpose being satisfied less. Now, even if the physician can make a diagnosis himself, sending the record to a specialist would be non-redundant if getting a second opinion improves the quality of the diagnosis.

Probabilistic Systems. The success of many medical tests and procedures is probabilistic. For example, with some probability the physician's test may fail to reach a diagnosis. The physician would still have transmitted the medical record for the purpose of diagnosis even if the test failed to reach one. This possibility affects our semantics of purpose: now an action may be for a purpose even if that purpose is never achieved.

To account for such probabilistic events, we model the outcome of the physician's actions as probabilistic. For an action to be for a purpose, we require that there be a non-zero probability of the purpose being achieved and that the physician attempts to maximize the expected reward. In essence, we require that the physician attempts to achieve a diagnosis. Thus, the auditee's *plan* determines the purposes behind his actions.

3 Planning for a Purpose

In this section, we present a formalism for planning that accounts for quantitative purposes, probabilistic systems and non-redundancy. We first review Markov Decision Processes (MDPs)—a natural model for planning with probabilistic systems. In general, an agent planning for some purpose constructs an MDP to help select its actions. The MDP models the agent's environment and how the agent's actions affect the environment's state. We use the reward function of the MDP to quantify the degree of satisfaction of a purpose upon taking an action from a state. The agent selects a plan that determines for each state, the action that the agent will perform if the agent reaches that state. The plan the agent selects optimizes the expected total discounted reward (degree of purpose satisfaction) under the MDP.

We then develop a stricter definition of optimal than used with standard MDPs. We use this definition to create models we call "NMDPs" for *Non-redundant MDPs*. In addition to requiring that strategies optimize the expected total discounted reward, NMDPs exclude strategies that employ redundant actions that neither decrease nor increase the total reward. We end with an example illustrating the use of an NMDP to model an audited environment.

3.1 Markov Decision Processes

An MDP may be thought of as a probabilistic automaton where each transition is labeled with a reward in addition to an action. Rather than having accepting or goal states, the "goal" of an MDP is to maximize the total reward over time. Furthermore, we distinguish between the MDP, which is a model of an environment, and the agent, which is an entity using the model to select its actions. Thus, while it is convenient to speak informally of actions arising from an MDP, strictly speaking actions are performed by an agent because of the agent's use of the MDP model to select these actions.

To define partially observable MDPs, let $\mathsf{Dist}(X)$ denote the space of all distributions over the set X. That is, $f \in \mathsf{Dist}(X)$ is a function from X to the reals between 0 and 1 that obeys the standard of axioms of probability theory making it a distribution over X. An MDP is a tuple $m = \langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle$ where

- $\mathcal{S}$ is a set of states;
- $\mathcal{A}$ is a set of actions;
- $t : \mathcal{S} \times \mathcal{A} \to \mathsf{Dist}(\mathcal{S})$, a transition function from a state and an action to a distribution over states;
- $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a reward function; and
- γ, a discount factor such that $0 < \gamma < 1$.

where $\mathbb{R}$ is the set of real numbers. For each state s in $\mathcal{S}$, the agent using the MDP to plan selects an action a from $\mathcal{A}$ to perform. Upon performing the action a in the state s, the agent receives the reward $r(s, a)$. The environment then transitions to a new state s' with probability $\mu(s')$ where μ is the distribution provided by $t(s, a)$. The goal of the agent is to select actions to maximize its expected total discounted reward $\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \rho_i\right]$ where $i \in \mathbb{N}$ (the set of natural numbers) ranges over time modeled as discrete steps, ρ_i is the reward at time i, and the expectation is taken over the probabilistic transitions. The discount factor γ accounts for the preference of people to receive rewards sooner than later. It may be thought of as similar to inflation. We require that $\gamma < 1$ to ensure that the expected total discounted reward is bounded.

We formalize the agent's plan as a *stationary strategy* (commonly called a "policy", but we reserve that word for privacy policies). A stationary strategy is a function σ from the state space $\mathcal{S}$ to the set $\mathcal{A}$ of actions (i.e., $\sigma : \mathcal{S} \to \mathcal{A}$) such that at a state s in $\mathcal{S}$, the agent always selects to perform the action $\sigma(s)$. The value of a state s under a strategy σ is

$$V_m(\sigma, s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \sigma(s_i))\right]$$

The Bellman equation [11] shows that

$$V_m(\sigma, s) = r(s, \sigma(s)) + \gamma \sum_{s' \in \mathcal{S}} t(s, \sigma(s))(s') * V_m(\sigma, s')$$

A strategy σ^* is optimal if and only if for all states s, $V_m(\sigma^*, s) = \max_\sigma V_m(\sigma, s)$. At least one optimal policy always exists (see, e.g., [62]). Furthermore, if σ^* is optimal, then

$$\sigma^*(s) \in \operatorname*{\mathsf{argmax}}_{a \in \mathcal{A}} \left[r(s, a) + \gamma \sum_{s' \in \mathcal{S}} t(s, \sigma(s))(s') * V_m(\sigma, s')\right]$$

We denote this set of optimal strategies as $\mathsf{opt}(\langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle)$, or when the transition system is clear from context, as $\mathsf{opt}(r)$. Such strategies are sufficient to maximize the agent's expected total discounted reward despite only depending upon the current state of the MDP.

Under this formalism, the auditee plays the role of the agent optimizing the MDP to plan. We presume that each purpose may be modeled as a reward function. That is, we assume the degree to which a purpose is satisfied may be captured by a function from states and actions to a real number. The higher the number, the higher the degree to which that purpose is satisfied. When the auditee wants to plan for a purpose p, it uses a reward function, r^p, such that $r^p(s, a)$ is the degree to which taking the action a from state s aids the purpose p. We also assume that the expected total discounted reward can capture the degree to which a purpose is satisfied over time. We say that the auditee plans *for* the purpose p when the auditee adopts a strategy σ that is optimal for the MDP $\langle \mathcal{S}, \mathcal{A}, t, r^p, \gamma \rangle$.

Executions and Behaviors. Given the strategy σ and the actual results of the probabilistic transitions yielded by t, the agent exhibits an *execution*. We represent this execution as an infinite sequence $e = [s_1, a_1, s_2, a_2, \ldots]$ of alternating states and actions starting with a state, where s_i is the ith state that the agent was in and a_i is the ith action the agent took, for all i in $\mathbb{N}$. We call a finite prefix b of an execution e a *behavior*.

Not every sequence of states and actions is a possible execution of the agent under an MDP. For an execution to be possible under an MDP, it must be consistent with some strategy and the transitions relation t. We say an execution e is *consistent* with a strategy σ if and only if $a_i = \sigma(s_i)$ for all i in $\mathbb{N}$ where a_i is the ith action in e and s_i is the ith state in e. A behavior is consistent with a strategy if it can be extended to an execution consistent with that strategy.

To determine whether an execution is possible under t, let a *contingency* κ be a function from $\mathcal{S} \times \mathcal{A} \times \mathbb{N}$ to $\mathcal{S}$ such that $\kappa(s, a, i)$ is the state that results from taking the action a in the state s as the ith action. We say that a contingency κ is *consistent* with an MDP if and only if κ only picks states to which the transition function t of the MDP assigns a non-zero probability to (i.e., for all s in $\mathcal{S}$, a in $\mathcal{A}$, and i in $\mathbb{N}$, $t(s, a)(\kappa(s, a, i)) > 0$).

Given an MDP m, let $m(s, \kappa)$ be the possibly infinite state model that results of having κ resolve all the probabilistic choices in m and having the model start in state s. Let $m(s, \kappa, \sigma)$ denote the execution that results from using the strategy σ and state s in the non-probabilistic model $m(s, \kappa)$. Formally, $m(s, \kappa, \sigma) = [s_1, a_1, s_2, a_2, \ldots]$ where $s_1 = s$ and for all $i \in \mathbb{N}$, $a_i = \sigma(s_i)$ and $s_{i+1} = \kappa(s_i, a_i, i)$.

Consistent contingencies capture the idea of possible executions. Formally, we say that an execution $e = [s_1, a_1, s_2, a_2, \ldots]$ is *possible* for m if and only if there exists a state s of m, a contingency κ consistent with m, and a strategy σ for m such that $e = m(s, \kappa, \sigma)$ Similarly, we say that a behavior $b = [s_1, a_1, \ldots, s_n, a_n]$ is *possible* for m if and only if there exists a state s of m, a contingency κ consistent with m, and a strategy σ for m such that $b \sqsubset m(s, \kappa, \sigma)$ where $\sqsubset$ denotes the proper-prefix relation. The following lemma reduces the global property of a behavior being possible for an MDP to local properties of the MDP.

Lemma 1. *For all MDPs m and behaviors $b = [s_1, a_1, \ldots, s_n, a_n] \in (\mathcal{S} \times \mathcal{A})^*$, b is possible for m if and only if for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$ and for all $i \leq n$ and $j \leq n$, $s_i = s_j$ implies that $a_i = a_j$.*

Proof. Suppose that b is possible for m. Then, there exists a state s of m, a contingency κ consistent with m, and a strategy σ for m such that $b \sqsubset m(s, \kappa, \sigma)$. Since $b \sqsubset m(s, \kappa, \sigma)$, for all $i < n$, $\kappa(s_i, a_i, i) = s_{i+1}$. Since κ is consistent with m, for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$. Since σ is stationary, $a_i = \sigma(s_i) = \sigma(s_j) = a_j$ for all $i, j \leq n$ such that $s_i = s_j$.

Suppose that for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$ and for all $i \leq n$ and $j \leq n$, $s_i = s_j$ implies that $a_i = a_j$. Let $s = s_1$. Let σ be some strategy such $\sigma(s_i) = a_i$ for all $i \leq n$. Such a σ exists since $s_i = s_j$ implies that $a_i = a_j$ for all $i \leq n$ and $j \leq n$. Let κ be some contingency consistent with m such that for all $i < n$, $\kappa(s_i, a_i, i) = s_{i+1}$. Such a κ exists since for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$. $b \sqsubset m(s, \kappa, \sigma)$. □

3.2 Non-redundancy

MDPs do not require that strategies be non-redundant. Even given that the auditee had an execution e from using a strategy σ in $\mathsf{opt}(r^p)$, some actions in e might not be *for* the purpose p. The reason is that some actions may be redundant despite being costless. The MDP optimization criterion behind opt prevents redundant actions from delaying the achievement of a goal as the reward associated with that goal would be further discounted making such redundant actions sub-optimal. However, the optimization criterion is not affected by redundant actions when they appear after all actions that provide non-zero rewards. Intuitively, the hypothetical agent planning only for the purpose in question would not perform such unneeded actions even if they have zero reward. Thus, to create our formalism of non-redundant MDPs (NMDPs), we replace opt with a new optimization criterion nopt that prevents these redundant actions while maintaining the same transition structure as a standard MDP.

To account for redundant actions, we must first contrast such actions with doing nothing. Thus, we introduce a distinguished action stop that stands for stopping and doing nothing. For all states s, stop labels a transition with zero reward (i.e., $r(s, \mathsf{stop}) = 0$) that is a self-loop (i.e., $t(s, \mathsf{stop})(s) = 1$). (We could put stop on only the subset of states that represent possible stopping points by slightly complicating our formalism.) Since we only allow deterministic stationary strategies and stop only labels self-loops, this decision is irrevocable: once the agent stops and does nothing, the agent does nothing forever. As selecting to do nothing results in only zero rewards henceforth, it may be viewed as stopping with the previously acquired total discounted reward.

Lemma 2. *For all NMDPs m, strategies σ for m, and states s, if $\sigma(s) = \mathsf{stop}$, then $V_m(\sigma, s) = 0$.*

Proof.

$$V_m(\sigma, s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \sigma(s_i))\right]$$

where s_i is the ith state that the environment modeled by the NMDP enters starting with $s = s_0$.

Proof by induction shows that for all i, $s_i = s$. The base case follows from the definition of s_0. For the inductive case, the inductive hypothesis shows that $s_i = s$. $s_{i+1} = s'$ with probability $t(s_i, \sigma(s_i))(s') = t(s, \sigma(s))(s') = t(s, \mathsf{stop})(s') = \mathsf{degen}(s)$ by the definition of NMDPs where $\mathsf{degen}(s)(s'') = 1$ if and only if $s'' = s$ and is equal to 0 for all other s''. Thus, with certainty, $s_{i+1} = s$.

Thus,

$$V_m(\sigma, s) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s_i, \sigma(s_i))\right] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(s, \mathsf{stop})\right] = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i 0\right] = 0$$

□

We use the idea of stopping and doing nothing to make formal when one execution contains more actions than another despite both being of infinite length. Given an execution e, let $\mathsf{active}(e)$ denote the prefix of e before the first instance of stop. $\mathsf{active}(e)$ will be equal to e in the case where e does not contain stop. An execution e_1 is a *proper sub-execution* of an execution e_2 if and only if $\mathsf{active}(e_1) \sqsubset \mathsf{active}(e_2)$ where $\sqsubset$ is the proper prefix relation. (We also use $\sqsubseteq$ for the prefix-or-equal relation.) Note if e_1 does not contain the stop, it cannot be a proper sub-execution of any execution.

We use contingencies to compare strategies. Given two strategies σ and σ', we write $\sigma' \prec \sigma$ if and only if for all contingencies κ and states s, $m(s, \kappa, \sigma')$ is a proper sub-execution of or equal to $m(s, \kappa, \sigma)$, and for at least one contingency κ' and state s', $m(s', \kappa', \sigma')$ is a proper sub-execution of $m(s', \kappa', \sigma)$. Intuitively, σ' proves that σ produces a redundant execution under κ' and s'. As we would expect, $\prec$ is a strict partial ordering on strategies.

Proposition 1. *$\prec$ is a strict partial order.*

Proof. The proper sub-execution relation is a strict partial order. This follows directly from the proper-prefix relation $\sqsubset$ being a strict partial order. We write $\lhd$ for *proper sub-execution* and $\unlhd$ for *proper sub-execution or equal.*

Now, we show that $\prec$ is also a strict partial ordering.

- Irreflexivity: for no σ is $\sigma \prec \sigma$. For $\sigma \prec \sigma$ to be true, there would have to exist a $\sigma \in \mathsf{opt}$ such that for at least one contingency κ' and s', $m(s', \kappa', \sigma')$ is a proper sub-execution of itself. However, this is impossible since the sub-execution relation is strict partial order.

- Asymmetry: for all σ_1 and σ_2, if $\sigma_1 \prec \sigma_2$, then it is not the case that $\sigma_2 \prec \sigma_1$. To show a contradiction, suppose $\sigma_1 \prec \sigma_2$ and $\sigma_2 \prec \sigma_1$ are both true. It would have to be the case that for all contingencies κ and states s, $m(s, \kappa, \sigma_1) \unlhd m(s, \kappa, \sigma_2)$ and $m(s, \kappa, \sigma_2) \unlhd m(s, \kappa, \sigma_1)$. Since $\lhd$ is a strict partial order, this implies that for all s and κ, $m(s, \kappa, \sigma_1) = m(s, \kappa, \sigma_2)$. Thus, there cannot exist a contingency κ' and state s' such that $m(s', \kappa', \sigma_2) \lhd m(s', \kappa', \sigma_1)$. Then $\sigma_2 \prec \sigma_1$ cannot be true, a contradiction.

- Transitivity: for all σ_1, σ_2, and σ_3, if $\sigma_1 \prec \sigma_2$ and $\sigma_2 \prec \sigma_3$, then $\sigma_1 \prec \sigma_3$. Suppose $\sigma_1 \prec \sigma_2$ and $\sigma_2 \prec \sigma_3$. Then for all for all contingencies κ and states s, $m(s, \kappa, \sigma_1) \unlhd m(s, \kappa, \sigma_2)$ and $m(s, \kappa, \sigma_2) \unlhd m(s, \kappa, \sigma_3)$. Since $\unlhd$ has transitivity, this implies that $m(s, \kappa, \sigma_1) \unlhd m(s, \kappa, \sigma_3)$ for all κ and s.

  Furthermore, it must be the case that there exists a contingency κ' and state s' such that $m(s', \kappa', \sigma_1) \lhd m(s', \kappa', \sigma_2)$. From above, $m(s', \kappa', \sigma_2) \unlhd m(s', \kappa', \sigma_3)$. Thus, by the transitivity of $\lhd$, $m(s', \kappa', \sigma_1) \lhd m(s', \kappa', \sigma_3)$ as needed. This implies that $\sigma_1 \prec \sigma_3$.

□

We define $\mathsf{nopt}(m)$ to be the subset of $\mathsf{opt}(m)$ holding only strategies σ such that for no $\sigma' \in \mathsf{opt}(m)$ does $\sigma' \prec \sigma$. $\mathsf{nopt}(m)$ is the set of non-redundant optimal policies.

The next lemma converts the requirements for being non-redundant from being about the executions of an MDP to being a local property. It uses the definition that $Q^*_m(s, a) = r(s, a) + \gamma \sum_{s'} t(s, a)(s') * V^*_m(s')$ and the proof uses that $Q_m(\sigma, s, a) = r(s, a) + \gamma \sum_{s'} t(s, a)(s') * V_m(\sigma, s')$.

Lemma 3. *For all NMDPs m and σ in $\mathsf{opt}(m)$, σ is in $\mathsf{nopt}(m)$ if and only if for all states s such that $\sigma(s) \neq \mathsf{stop}$, $Q^*_m(s, \sigma(s)) > 0$.*

Proof. *If Direction.* Suppose that for all s such that $\sigma(s) \neq \mathsf{stop}$, $Q^*_m(s, \sigma(s)) > 0$. For the purposes of showing a contradiction, assume that $\sigma \notin \mathsf{nopt}(m)$. Then there exists σ' such that $\sigma' \in \mathsf{opt}(m)$ and $\sigma' \prec \sigma$. This implies that there exists κ' and s' such that $\mathsf{active}(m(s', \kappa', \sigma'))$ is a strict prefix of $\mathsf{active}(m(s', \kappa', \sigma))$. $m(s', \kappa', \sigma')$ must have the form $[s_1, a_1, s_2, \ldots, s_n, \mathsf{stop}, \ldots]$ and $m(s', \kappa', \sigma)$ must have the form $[s_1, a_1, s_2, \ldots, s_n, a_n, \ldots]$ for some n where $a_n \neq \mathsf{stop}$. Since $\sigma(s_n) = a_n \neq \mathsf{stop}$, $Q^*_m(s, \sigma(s)) > 0$. Since both σ and σ' are in $\mathsf{opt}(m)$, $0 < Q^*_m(s_n, \sigma(s)) = Q^*_m(s_n, \sigma'(s)) = Q^*_m(s_n, \mathsf{stop}) = Q_m(\sigma, s_n, \mathsf{stop})$. However, by Lemma 2, $Q_m(\sigma, s_n, \mathsf{stop}) = V_m(\sigma, s_n) = 0$, a contradiction. Thus, our assumption that $\sigma \notin \mathsf{nopt}(m)$ is false and σ is $\mathsf{nopt}(m)$.

Only-If Direction. Suppose σ is in $\mathsf{nopt}(m)$. Consider a state s such that $\sigma(s) \neq \mathsf{stop}$. Since σ is in $\mathsf{nopt}(m)$, there exists no σ' in $\mathsf{opt}(m)$ such that $\sigma' \prec \sigma$. That is, there exists no σ' such that σ' is in $\mathsf{opt}(m)$; for all contingencies κ' consistent with m, states s', $\mathsf{active}(m(s', \kappa', \sigma')) \sqsubseteq \mathsf{active}(m(s', \kappa', \sigma))$; and there exists a contingency κ'' and s'' such that $\mathsf{active}(m(s'', \kappa'', \sigma')) \sqsubset \mathsf{active}(m(s'', \kappa'', \sigma))$. That is, for all σ', either (1) σ' is not in $\mathsf{opt}(m)$; (2) it is not the case that for all contingencies κ' consistent with m, states s', $\mathsf{active}(m(s', \kappa', \sigma')) \sqsubseteq \mathsf{active}(m(s', \kappa', \sigma))$; or (3) it is not the case that there exists a contingency κ'' and a state s'' such that $\mathsf{active}(m(s'', \kappa'', \sigma')) \sqsubset \mathsf{active}(m(s'', \kappa'', \sigma))$.

We consider each of those three possibilities for σ' such that σ' is equal to σ except $\sigma'(s) = \mathsf{stop}$.

1. Case: σ' is not in $\mathsf{opt}(m)$. Since σ' is not in $\mathsf{opt}(m)$, there must exist $s^\dagger$ such that $\sigma'(s^\dagger) \notin \mathsf{argmax}_a\, Q^*_m(s^\dagger, a)$. Since σ is in $\mathsf{opt}(m)$, for all $s' \neq s$, $\sigma'(s') = \sigma(s') \in \mathsf{argmax}_a\, Q^*_m(s', a)$. Thus, $s^\dagger$ must be s. Since $\sigma'(s) \notin \mathsf{argmax}_a\, Q^*_m(s, a)$ and $Q^*_m(s, \sigma'(s)) = Q^*_m(s, \mathsf{stop}) = 0$, $\max_a Q^*_m(s, a) = V^*_m(s) > 0$. Since σ is in $\mathsf{opt}(m)$, $Q^*_m(s, \sigma(s)) = V^*_m(s) > 0$.

2. Case: It is not the case that for all contingencies κ' consistent with m, and for all states s', $\mathsf{active}(m(s', \kappa', \sigma')) \sqsubseteq \mathsf{active}(m(s', \kappa', \sigma))$. For all κ' and s', $m(s', \kappa', \sigma)$ and $m(s', \kappa', \sigma')$ only differ if they reach the state s since σ and σ' only differ at the state s. If s is never reached, then $\mathsf{active}(m(s', \kappa', \sigma')) = \mathsf{active}(m(s', \kappa', \sigma))$. If s is reached, then $m(s', \kappa', \sigma')$ has the form $[s', a_1, s_2, a_2, \ldots, s, \mathsf{stop}, \ldots]$ and $m(s', \kappa', \sigma)$ has the form $[s', a_1, s_2, a_2, \ldots, s, \sigma(s), \ldots]$. Thus, either way, $\mathsf{active}(m(s', \kappa', \sigma')) \sqsubseteq \mathsf{active}(m(s', \kappa', \sigma))$. Thus, it is the case that for all contingencies κ' consistent with m, states s', $\mathsf{active}(m(s', \kappa', \sigma')) \sqsubseteq \mathsf{active}(m(s', \kappa', \sigma))$. Since this is a contradiction, the result trivially holds.

3. Case: There does not exist a contingency κ'' and a state s'' such that $\mathsf{active}(m(s'', \kappa'', \sigma')) \sqsubset \mathsf{active}(m(s'', \kappa'', \sigma))$. Let s'' be s. Then for all κ'', $m(s'', \kappa'', \sigma') = m(s, \kappa'', \sigma')$ has the form $[s, \mathsf{stop}, \ldots]$. $m(s, \kappa'', \sigma)$ has the form $[s, \sigma(s), \ldots]$ for some $\sigma(s) \neq \mathsf{stop}$. Thus, there exists a contingency κ'' and s'' such that $\mathsf{active}(m(s'', \kappa'', \sigma')) \sqsubset \mathsf{active}(m(s'', \kappa'', \sigma))$. Since this is a contradiction, the result trivially holds.

Thus, the result holds under all three possible cases. □

One of the reasons that the MDP model is useful is that an optimal strategy is guaranteed to exist. Fortunately, we can prove that $\mathsf{nopt}(m)$ is also guaranteed to be non-empty. One way to prove this result would use reasoning about well-ordered sets, Proposition 1, and the fact that space of all possible strategies is finite for NMDPs with finite state and action spaces. However, we provide a proof that depends more upon the structure of NMDPs since it can extend to NMDPs with infinite state spaces, which becomes important in the next chapter.

Theorem 1. *For all MDPs m,* $\mathsf{nopt}(m)$ *is not empty.*

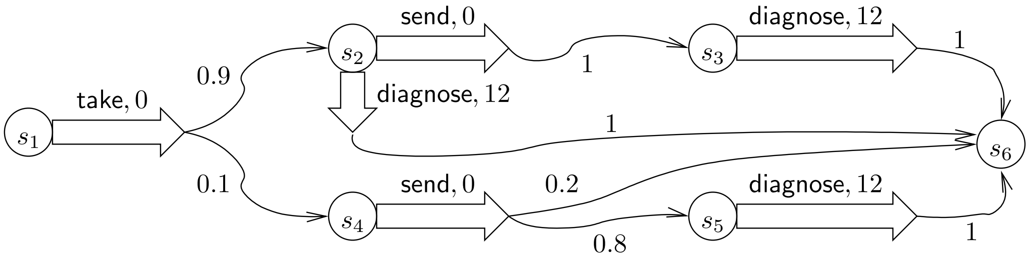

Figure 1: The environment model m_{ex1} that the physician used. Circles represent states, block arrows denote possible actions, and squiggly arrows denote probabilistic outcomes. Self-loops of zero reward under all actions, including the special action stop, are not shown.

Proof. $\mathsf{opt}(m)$ is non-empty (see, e.g., [62]). Let σ be some element of $\mathsf{opt}(m)$. Let σ' be σ except whenever $Q^*_m(s, \sigma(s)) \leq 0$, $\sigma'(s) = \mathsf{stop}$. For any such state s, $Q^*_m(s, \sigma(s)) \leq 0 = V_m(\sigma', s) = Q_m(\sigma', s, \sigma'(s))$ by Lemma 2. For all other states $Q^*_m(s, \sigma'(s)) = Q^*_m(s, \sigma(s))$ since $\sigma'(s) = \sigma(s)$. In either case, $V^*_m(s) = Q^*_m(s, \sigma(s))$ since s is optimal. Thus, for all states s, $V^*_m(s) \leq Q^*_m(s, \sigma'(s))$. Thus, σ' is in $\mathsf{opt}(m)$. Furthermore, by construction, for all s, $\sigma'(s) \neq \mathsf{stop}$ implies that $Q^*_m(s, \sigma'(s)) = Q^*_m(s, \sigma(s)) > 0$. Thus, σ' is in $\mathsf{nopt}(m)$ by Lemma 3. □

3.3 Example: Modeling the Physician's Environment

Suppose an auditor is inspecting a hospital and comes across a physician referring a medical record to his own private practice for analysis of an X-ray as described in Section 2. As physicians may only make such referrals for the purpose of treatment (treat), the auditor may find the physician's behavior suspicious. To investigate, the auditor may formally model the hospital using our formalism.

After studying the hospital and how the physician's actions affect it, the auditor would construct the NMDP $m_{\mathsf{ex1}} = \langle \mathcal{S}_{\mathsf{ex1}}, \mathcal{A}_{\mathsf{ex1}}, t_{\mathsf{ex1}}, r^{\mathsf{treat}}_{\mathsf{ex1}}, \gamma_{\mathsf{ex1}} \rangle$ shown in Figure 1. The figure conveys all components of the NMDP except γ_{ex1}. For instance, the block arrow from the state s_1 labeled take and the squiggly arrows leaving it denote that after the agent performs the action take from state s_1, the environment will transition to the state s_2 with probability 0.9 and to state s_4 with probability of 0.1 (i.e., $t_{\mathsf{ex1}}(s_1, \mathsf{take})(s_2) = 0.9$ and $t_{\mathsf{ex1}}(s_1, \mathsf{take})(s_4) = 0.1$). The number over the block arrow further indicates the degree to which the action satisfies the purpose of treat. In this instance, it shows that $r^{\mathsf{treat}}_{\mathsf{ex1}}(s_1, \mathsf{take}) = 0$. This transition models the physician taking an X-ray. With probability 0.9, he is able to make a diagnosis right away (from state s_2); with probability 0.1, he must send the X-ray to his practice to make a diagnosis. Similarly, the transition from state s_4 models that his practice's test has a 0.8 success rate of making a diagnosis; with probability 0.2, no diagnosis is ever reached. For simplicity, we assume that all diagnoses have the same quality of 12 and that second opinions do not improve the quality; the auditor could use a different model if these assumptions are false.

Using the model, the auditor computes $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$, which consists of those strategies that maximizes the expected total discounted degree of satisfaction of the purpose of treatment where the expectation is over the probabilistic transitions of the model. $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ includes the appropriate strategy σ_1 where $\sigma_1(s_1) = \mathsf{take}$, $\sigma_1(s_4) = \mathsf{send}$, $\sigma_1(s_2) = \sigma_1(s_3) = \sigma_1(s_5) = \mathsf{diagnose}$, and

$\sigma_1(s_6) = \mathsf{stop}$. Furthermore, $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ excludes the redundant strategy σ_2 that performs a redundant send where σ_2 is the same as σ_1 except for $\sigma_2(s_2) = \mathsf{send}$. Performing the extra action send delays the reward of 12 for achieving a diagnosis resulting in its discounted reward being $\gamma^2_{\mathsf{ex1}} * 12$ instead of $\gamma_{\mathsf{ex1}} * 12$ and, thus, the strategy is not optimal.

However, $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ does include the redundant strategy σ_3 that is the same as σ_1 except for $\sigma_3(s_6) = \mathsf{send}$. $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ includes this strategy despite the send actions from state s_6 being redundant since no positive rewards follow the send actions. Fortunately, $\mathsf{nopt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ does not include σ_3 since σ_1 is both in $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ and $\sigma_1 \prec \sigma_3$. To see that $\sigma_1 \prec \sigma_3$ note that for every contingency κ and state s, the $m_{\mathsf{ex1}}(s, \kappa, \sigma_1)$ has the form b followed by an finite sequence of stop (interleaved with the state s_6) for some finite prefix b. For the same κ, $m_{\mathsf{ex1}}(s, \kappa, \sigma_3)$ has the form b followed by an infinite sequence of send actions (interleaved with the state s_6) for the same b. Thus, $m_{\mathsf{ex1}}(s, \kappa, \sigma_1)$ is a proper sub-execution of $m_{\mathsf{ex1}}(s, \kappa, \sigma_3)$.

The above modeling implies that the strategy σ_1 can be for the purpose of treatment but σ_2 and σ_3 cannot be.

4 Auditing

In the above example, the auditor constructed a model of the environment in which the auditee operates. The auditor must use the model to determine if the auditee obeyed the policy. We first discuss this process for auditing exclusivity policy rules and revisit the above example. Then, we discuss the process for prohibitive policy rules. In the next section, we provide an auditing algorithm that automates comparing the auditee's behavior, as recorded in a log, to the set of allowed behaviors.

4.1 Auditing Exclusivity Rules

Suppose that an auditor would like to determine whether an auditee performed some logged actions *only for* the purpose p. The auditor can compare the logged behavior to the behavior that a hypothetical agent would perform when planning for the purpose p. In particular, the hypothetical agent selects a strategy from $\mathsf{nopt}(\langle \mathcal{S}, \mathcal{A}, t, r^p, \gamma \rangle)$ where $\mathcal{S}$, $\mathcal{A}$, and t models the environment of the auditee; r^p is a reward function modeling the degree to which the purpose p is satisfied; and γ is an appropriately selected discounting factor. If the logged behavior of the auditee would never have been performed by the hypothetical agent, then the auditor knows that the auditee violated the policy.

In particular, the auditor must consider all the possible behaviors the hypothetical agent could have performed. For a model m, let $\mathsf{nbehv}(r^p)$ represent this set where a finite prefix b of an execution is in $\mathsf{nbehv}(r^p)$ if and only if there exists a strategy σ in $\mathsf{nopt}(r^p)$, a contingency κ, and a state s such that b is a prefix of $m(s, \kappa, \sigma)$.

The auditor must compare $\mathsf{nbehv}(r^p)$ to the set of all behaviors that could have caused the auditor to observe the log that he did. We presume that the log ℓ was created by a process log that records features of the current behavior. That is, $\mathsf{log}\colon B \to L$ where B is the set of behaviors and L the set of logs, and $\ell = \mathsf{log}(b)$ where b is the prefix of the actual execution of the environment available at the time of auditing. The auditor must consider all the behaviors in $\mathsf{log}^{-1}(\ell) = \{\, b \in B \mid \mathsf{log}(b) = \ell \,\}$ as possible where log^{-1} is the inverse of the logging function. In the best case for the auditor, the log records the whole prefix b of the execution that transpired until the time of auditing, in which case $\mathsf{log}^{-1}(\ell) = \{\ell\}$. However, the log may be incomplete by missing actions, or may include only partial information about an action such as that it was one of a set of actions.

If $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^p)$ is empty, then the auditor may conclude that the auditee did not plan for the purpose p, and, thus, violated the rule that auditee must only perform the actions recorded in ℓ for the purpose p; otherwise, the auditor must consider it possible that the auditee planned for the purpose p.

If $\mathsf{log}^{-1}(\ell) \subseteq \mathsf{nbehv}(r^p)$, the auditor might be tempted to conclude that the auditee surely obeyed the policy rule. However, as illustrated in the second example below, this is not necessarily true. The problem is that $\mathsf{log}^{-1}(\ell)$ might have a non-empty intersection with $\mathsf{nbehv}(r^{p'})$ for some other purpose p'. In this case, the auditee might have been actually planning for the purpose p' instead of p. Indeed, given the likelihood of such other purposes for non-trivial scenarios, we consider proving compliance practically impossible. However, this incapability is of little consequence: $\mathsf{log}^{-1}(\ell) \subseteq \mathsf{nbehv}(r^p)$ does imply that the auditee is behaving as though he is obeying the policy. That is, in the worse case, the auditee is still doing the right things even if for the wrong reasons.

4.2 Example: Auditing the Physician

Below we revisit the example of Section 3.3. We consider two cases. In the first, the auditor shows that the physician violated the policy. In the second, auditing is inconclusive.

Violation Found. Suppose after constructing the model as above in Section 3.3, the auditor maps the actions recorded in the access log ℓ_1 to the actions of the model m_{ex1}, and finds $\mathsf{log}^{-1}(\ell_1)$ holds only a single behavior: $b_1 = [s_1, \mathsf{take}, s_2, \mathsf{send}, s_3, \mathsf{diagnose}, s_6, \mathsf{stop}, s_6, \mathsf{stop}]$. Next, using $\mathsf{nopt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$, as computed above, the auditor constructs the set $\mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ of all behaviors an agent planning for treatment might exhibit. The auditor would find that b_1 is not in $\mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$.

To see this, note that every execution e_1 that has b_1 as a prefix is generated from a strategy σ such that $\sigma(s_2) = \mathsf{send}$. The strategy σ_2 from Section 3.3 is one such strategy. None of these strategies are members of $\mathsf{opt}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ for the same reason as σ_2 is not a member. Thus, b_1 cannot be in $\mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$. As $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ is empty, the audit reveals that the physician violated the policy.

Inconclusive. Now suppose that the auditor sees a different log ℓ_2 such that $\mathsf{log}^{-1}(\ell_2) = \{b_2\}$ where $b_2 = [s_1, \mathsf{take}, s_4, \mathsf{send}, s_5, \mathsf{diagnose}, s_6, \mathsf{stop}, s_6, \mathsf{stop}]$. In this case, our formalism would not find a violation since b_2 is in $\mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$. In particular, the strategy σ_1 from above produces the behavior b_2 under the contingency that selects the bottom probabilistic transition from state s_1 to state s_4 under the action take.

Nevertheless, the auditor cannot be sure that the physician obeyed the policy. For example, consider the NMDP m'_{ex1} that is m_{ex1} altered to use the reward function $r^{\mathsf{profit}}_{\mathsf{ex1}}$ instead of $r^{\mathsf{treat}}_{\mathsf{ex1}}$. $r^{\mathsf{profit}}_{\mathsf{ex1}}$ assigns a reward of zero to all transitions except for the send actions from states s_2 and s_4, to which it assigns a reward of 9. σ_1 is in $\mathsf{nopt}(r^{\mathsf{profit}}_{\mathsf{ex1}})$ meaning that not only the same actions (those in b_2), but even the exact same strategy can be either for the allowed purpose treat or the disallowed purpose profit. Thus, if the physician did refer the record to his practice for profit, he cannot be caught as he has tenable deniability of his ulterior motive of profit.

4.3 Auditing Prohibitive Rules

In the above example, the auditor was enforcing the rule that the physician's actions be *only for* treatment. Now, consider auditing to enforce the rule that the physician's actions are *not for* personal profit. To obey this purpose restriction, the auditee need not have attempted to minimize the degree of satisfaction of the purpose. Rather the auditee, need merely to have ignored the prohibited purpose.

To audit for compliance with a rule prohibiting the purpose p, after seeing the log ℓ, the auditor could check whether $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^p)$ is empty. If so, then the auditor knows that the policy was obeyed because the auditee could not have been planning for the purpose p. If not, then the auditor cannot prove nor disprove a violation. In the above example, just as the auditor is unsure whether the actions were *for* the required purpose of treatment, the auditor is unsure whether the actions are *not for* the prohibited purpose of profit.

Leveraging Multiple Purposes. An auditor might decide to investigate some of the cases where $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^p)$ is not empty. In this case, the auditor could limit his attention to only those possible violations of a prohibitive rule that cannot be explained away by some allowed purpose. For example, in the inconclusive example above, the physician's actions can be explained with the allowed purpose of treatment. As the physician has tenable deniability, it is unlikely that investigating his actions would be a productive use of the auditor's time. Thus, the auditor should limit his attention to those logs ℓ such that both $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^{\mathsf{profit}}_{\mathsf{ex1}})$ is non-empty and $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(r^{\mathsf{treat}}_{\mathsf{ex1}})$ is empty.

A similar additional check using disallowed purposes could be applied to enforcing exclusivity rules. However, for exclusivity rules, this check would identify cases where the auditee's behavior could have been either for the allowed purpose or a disallowed purpose. Thus, it would serve to find additional cases to investigate and increase the auditor's workload rather than reduce it. Furthermore, the auditee would have tenable deniability for these possible ulterior motives, making these investigations a poor use of the auditor's time.

5 Auditing Algorithm

5.1 Algorithm

We would like to automate the auditing process described above. To this end, we present in Figure 2 an algorithm AuditNMDP that aids the auditor in comparing the log to the set of allowed behaviors. Since we are not interested in the details of the logging process and would like to focus on the planning aspects of our semantics, we limit our attention to the case where $\mathsf{log}(b) = b$ (i.e., the log is simply the behavior of the auditee). However, future work could extend our algorithm to handle incomplete logs by constructing the set of all possible behaviors that could give rise to that log.

The algorithm presumes that the MDP m is finite. That is, both $\mathcal{S}$ and $\mathcal{A}$ are finite. As proved below (Theorem 2), AuditNMDP(m, b) returns *true* if and only if $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty. In the case of an exclusivity rule, the auditor may conclude that the policy was violated when AuditNMDP returns *true*. In case of a prohibitive rule, the auditor may conclude the policy was obeyed when AuditNMDP returns *true*.

```
AuditNMDP($m = \langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle,\ b = [s_1, a_1, s_2, a_2, \ldots, s_n, a_n]$):
if (ImpossibleMDP($m, b$))
    return true    //  behavior impossible for NMDP
$V^*_m :=$ SolveMDP($m$)
for ($i := 1$; $i \leq n$; $i{+}{+}$):
    if ($\mathtt{Q}^*(m, V^*_m, s_i, a_i) < V^*_m(s_i)$):
        return true    //  action suboptimal
    if ($\mathtt{Q}^*(m, V^*_m, s_i, a_i) \leq 0$ and $a_i \neq \mathsf{stop}$):
        return true    //  action redundant
return false
```

Figure 2: The algorithm AuditNMDP. SolveMDP may be any MDP solving algorithm. Figure 3 shows ImpossibleMDP.

```
ImpossibleMDP($m = \langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle,\ b = [s_1, a_1, s_2, a_2, \ldots, s_n, a_n]$):
for ($i := 1$; $i \leq n$; $i{+}{+}$):
    if ($s_i \notin \mathcal{S}$):
        return true    //  $s_i$ is not a state
    if ($a_i \notin \mathcal{A}$):
        return true    //  $a_i$ is not an action
for ($i := 1$; $i < n$; $i{+}{+}$):
    if ($t(s_i, a_i)(s_{i+1}) \leq 0$):
        return true    //  $s_{i+1}$ unreachable from $s_i$
    for ($j := i + 1$; $j \leq n$; $j{+}{+}$):
        if ($s_i = s_j$ and $a_i \neq a_j$):
            return true    //  no stationary strategy could have produced the behavior
return false
```

Figure 3: The algorithm ImpossibleMDP. Returns whether the given behavior is possible for the given MDP.

The algorithm operates by checking a series of local conditions of the NMDP m and behavior b that are equivalent to the global property of whether $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty (as proved by Lemma 4). First, AuditNMDP checks whether the behavior b is possible for m using the sub-routine ImpossibleMDP shown in Figure 3. ImpossibleMDP checks whether every state and action is valid (Lines 12 and 14), every state is reachable by the state proceeding it (Line 17), and that the same action is performed from equal states in b (Line 20).

Next, the AuditNMDP checks whether the behavior b is optimal (Line 05) and non-redundant (Line 07). To do so, AuditNMDP uses a sub-routine SolveMDP to compute V_m^*, which for each state s records $V_m^*(s)$, the optimal value for s. The fact that NMDPs are a type of MDP allows AuditNMDP to use any MDP optimization algorithm for SolveMDP, such as reducing the optimization to a system of linear equations [28].

AuditNMDP uses a function Q^* that computes Q^* from V^*:

$$\mathsf{Q}^*(m, V_m^*, s, a) = r(s_i, a_i) + \gamma \sum_{s' \in \mathcal{S}} t(s_i, a_i)(s') * V_m^*(s')$$

Thus, $\mathsf{Q}^*(m, V_m^*, s, a)$ is equal to $Q_m^*(s, a)$.

The essence of the algorithm is checking whether $\mathsf{log}^{-1}(\ell) \cap \mathsf{nbehv}(m)$ is empty. For simplicity, our algorithm presumes that $\mathsf{log}^{-1}(\ell)$ holds only one behavior. This restriction manifests itself in that each of the local checks (Lines 01, 05, and 07) only considers a single sequence of states and actions.

If $\mathsf{log}^{-1}(\ell)$ holds more than a single behavior but is a small set, then the auditor may run the algorithm for each behavior in $\mathsf{log}^{-1}(\ell)$. Alternatively, in some cases the set $\mathsf{log}^{-1}(\ell)$ may have structure that a modified algorithm could leverage. For example, if $\mathsf{log}^{-1}(\ell)$ is missing what action is taken at some states of the execution or only narrows down the taken action to a set of possible alternatives, a conjunction of constraints on the action taken at each state may identify the set.

5.1.1 Correctness

To prove correctness, we use the following lemma that allows us to reduce checking for violations to local properties of the NMDP and the auditee's behavior.

Lemma 4. *For an NMDP m, the behavior $b = [s_1, a_1, \ldots, s_n, a_n]$ is in $\mathsf{nbehv}(m)$ if and only if b is a possible behavior of m, and for all $i \leq n$, $Q_m^*(s_i, a_i) = V_m^*(s_i)$ and $a_i \neq \mathsf{stop}$ implies that $Q_m^*(s_i, a_i) > 0$.*

Proof. First, for the only-if direction, suppose $b \in \mathsf{nbehv}(m)$. Since b is in $\mathsf{nbehv}(m)$, there exists a state s, a contingency κ consistent with m, and strategy σ in $\mathsf{nopt}(m)$ such that $b \sqsubset m(s, \kappa, \sigma)$. Thus, b is possible since κ is consistent with with m. Since $b \sqsubset m(s, \kappa, \sigma)$, for all $i \leq n$, $\sigma(s_i) = a_i$. Since σ is in $\mathsf{nopt}(m)$, for all $i \leq n$, $Q_m^*(s_i, a_i) = Q_m^*(s_i, \sigma(s_i)) = V_m^*(s_i)$. Since σ is in $\mathsf{nopt}(m)$, by Lemma 3, for all s such that $\sigma(s) \neq \mathsf{stop}$, $Q_m^*(s, \sigma(s)) > 0$. Thus, for all $i \leq n$, $\sigma(s_i) \neq \mathsf{stop}$, $Q_m^*(s, \sigma(s)) > 0$.

Second, for the if direction, suppose b is a possible behavior of m, and for all $i \leq n$, $Q_m^*(s_i, a_i) = V_m^*(s_i)$ and $a_i \neq \mathsf{stop}$ implies that $Q_m^*(s_i, a_i) > 0$. By Theorem 1, $\mathsf{nopt}(m)$ is not empty. Let σ be some element of $\mathsf{nopt}(m)$. Let σ' be identical to σ except for all i, $\sigma'(s_i) = a_i$, which is well defined since b is possible. For all i, $Q_m^*(s_i, \sigma'(s_i)) = Q_m^*(s_i, a_i) = V_m^*(s_i)$ and $\sigma'(s_i) = a_i \neq \mathsf{stop}$ implies that $Q_m^*(s_i, \sigma'(s_i)) = Q_m^*(s_i, a_i) > 0$. For all other states s, $Q_m^*(s_i, \sigma'(s)) = Q_m^*(s, \sigma(s)) = V_m^*(s)$

and $\sigma'(s) \neq \mathsf{stop}$ implies that $Q^*_m(s, \sigma'(s)) > 0$ by Lemma 3 since $\sigma'(s) = \sigma(s)$. Thus, for all s, $Q^*_m(s, \sigma'(s)) = V^*_m(s)$, which implies that σ' is in $\mathsf{opt}(m)$. Furthermore, for all s, $\sigma'(s) \neq \mathsf{stop}$ implies that $Q^*_m(s, \sigma'(s)) > 0$, which implies that σ' is in $\mathsf{nopt}(m)$ by Lemma 3.

By Lemma 1, b being possible implies that for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$. Thus, there exists a contingency κ that is consistent with m such that $\kappa(s_i, a_i, i) = s_{i+1}$. Furthermore, $b \sqsubset m(s, \kappa, \sigma')$ for $s = s_1$. Thus, since σ' is in $\mathsf{nopt}(m)$, b is in $\mathsf{nbehv}(m)$. □

The above lemma combines with reasoning about the actual code of the program to yield its correctness. First, we prove the correctness of ImpossibleMDP as a lemma.

Lemma 5. *For all MDPs m and behaviors b,* ImpossibleMDP(m, b) *is a decision procedure for whether b is not a possible behavior of m.*

Proof. To show that ImpossibleMDP is a decision procedure, we must show that it always terminates, that b is not possible for m if and only if ImpossibleMDP(m, b) returns true, and that b is possible for m if and only if ImpossibleMDP(m, b) returns false.

To show that ImpossibleMDP terminates note that all the for loops involve a monotonically increasing counter (i or j) and that they all terminate after the counter reaches finite number (n or $n + 1$).

ImpossibleMDP returns true if and only if one of the following is true: (1) there exists $i \leq n$ such that s_i is not a state of m, (2) there exists $i \leq n$ such that a_i is not an action of m, (3) there exists $i < n$ such that $t(s_i, a_i)(s_{i+1}) \leq 0$, (4) there exists $i < n$ and j where $i < j \leq n$ such that $s_i = s_j$ and $a_i \neq a_j$. ImpossibleMDP returns false if and only if all of the conditions (1), (2), (3), and (4) are false. Conditions (1) and (2) are both false if and only if b is in $(\mathcal{S} \times \mathcal{A})^*$. Condition (3) is false if and only if for all $i < n$, $t(s_i, a_i)(s_{i+1}) > 0$. Condition (4) is false if and only if all $i \leq n$ and $j \leq n$, $s_i = s_j$ implies that $a_i = a_j$.

Thus, by Lemma 1, b is possible for m if and only if the conditions (1), (2), (3), and (4) are all false, which is exactly when ImpossibleMDP returns false. Furthermore, b is not possible for m if and only if one of the conditions (1), (2), (3), and (4) is true, which is exactly when ImpossibleMDP returns true. □

Theorem 2. *For all finite NMDPs m and behaviors b,* AuditNMDP *is a decision procedure for whether $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty.*

Proof. To show that AuditNMDP is a decision procedure, we must show that it always terminates, that $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty if and only if AuditNMDP(m, b) returns true, and that $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is non-empty if and only if AuditNMDP(m, b) returns false.

To show that AuditNMDP terminates, note that SolveMDP is also guaranteed to terminate because m is finite. Thus, each iteration of the for loop terminates. Furthermore, n is a finite number and i monotonically increases toward it. Thus, the loop will execute only a finite number of times. Furthermore Q^* will terminate since $\mathcal{S}$ is finite.

Now, we show that $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty if and only if AuditNMDP(m, b) returns true. AuditNMDP(m, b) returns true if and only if at least one of the following is true: (1) b is not possible (see Lemma 5), (2) there exists $i \leq n$ such that $\mathsf{Q}^*(m, V^*_m, s_i, a_i) < V^*_m(s_i)$, (3) there exists $i \leq n$ such that $\mathsf{Q}^*(m, V^*_m, s_i, a_i) \leq 0$ and $a_i \neq \mathsf{stop}$. At least one of the Conditions (1), (2), or (3) is true if and only if the following is false: b is a possible behavior of m, for all $i \leq n$, $Q^*_m(s_i, a_i) = V^*_m(s_i)$ and $a_i \neq \mathsf{stop}$ implies that $Q^*_m(s_i, a_i) > 0$. Thus, by Lemma 4, AuditNMDP(m, b) returns true if and only if b is not in $\mathsf{nbehv}(m)$. Since $\mathsf{log}^{-1}(b) = \{b\}$, AuditNMDP$(m, b)$ returns true if and only if $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty.

Since AuditNMDP(m, b) always terminates and can only return true or false, and returns true if and only if $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty, AuditNMDP(m, b) returns false if and only if $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is non-empty. □

5.1.2 Running Time

The running time of the algorithm is dominated by the MDP optimization conducted by SolveMDP. SolveMDP may be done exactly by reducing the optimization to a system of linear equations [28]. Such systems may be solved in polynomial time [43, 41]. However, in practice, large systems are often difficult to solve. Fortunately, a large number of algorithms for making iterative approximations exist whose running time depends on the quality of the approximation. (See [47] for a discussion.) In the next section, we discuss an implementation using such a technique.

5.2 Approximation Algorithm and Implementation

Rather than implement the exact algorithm AuditNMDP found in Section 5, we implemented an approximation algorithm using the standard value iteration algorithm to solve MDPs (see, e.g., [62]). The value iteration algorithm starts with an arbitrary guess of an optimal strategy for an MDP and the value of each state under that policy. With each iteration, the algorithm improves its estimation of the optimal strategy and its value. It continues to make successively more accurate estimations until the improvement between one iteration and next is below some threshold ϵ. At this point, the algorithm returns its estimations. The difference between its estimation of the value of each state under the optimal policy and the true value is bounded by $2\epsilon\gamma/(1 - \gamma)$ where γ is the discount factor of the MDP [76, 77]. Each iteration takes $O(|\mathcal{S}|^2 * |\mathcal{A}|)$ time. The number of iterations needed to reach convergence grows quickly in γ making the algorithm pseudo-polynomial time in γ and polynomial time in $|\mathcal{A}|$ and $|\mathcal{S}|$ [72]. Despite the linear programming approach having better worst-case complexity, value iteration tends to perform well in practice. Using value iteration in our algorithm results in it having the same asymptotic running time of pseudo-polynomial in γ.

To maintain soundness, the approximate auditing algorithm differs from the exact algorithm to account for the approximations made by the value-iteration algorithm. Figure 4 shows a general framework for auditing with approximation algorithms. SolveMDPApprox is an approximation algorithm for solving MDPs. It returns lower and upper bounds on the value of $V^*_m(s, a)$ for each s and a. AuditNMDPApprox uses these bounds to soundly audit.

For example, the auditor may select to use value iteration for SolveMDPApprox. In this case, $\mathtt{V}^*_{\mathsf{low}}(s, a) = \mathtt{V}^*_{\mathsf{app}}(s, a) - 2\epsilon\gamma/(1 - \gamma)$ and $\mathtt{V}^*_{\mathsf{up}}(s, a) = \mathtt{V}^*_{\mathsf{app}}(s, a) + 2\epsilon\gamma/(1 - \gamma)$ where $\mathtt{V}^*_{\mathsf{app}}(s, a)$ is the value of the approximation returned by value iteration using ϵ for the accuracy parameter.

With these changes, the approximation algorithm is sound in that it will return *true* only when the original algorithm AuditNMDP solving the MDPs exactly would return *true*.

Theorem 3. *For all finite NMDPs m and behaviors b, if* AuditNMDPApprox(m, b) *returns true, then* $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ *is empty.*

Proof. If AuditNMDPApprox(m, b) returns true, then one of the following is true: (1) the subroutine ImpossibleMDP returns true, (2) there exists $i \leq n$ such that $\mathtt{Q}^*(m, \mathtt{V}^*_{\mathsf{up}}, s_i, a_i) < \mathtt{V}^*_{\mathsf{low}}(s_i)$, or (3) there exists $i \leq n$ such that $\mathtt{Q}^*(m, \mathtt{V}^*_{\mathsf{up}}, s_i, a_i) \leq 0$ and $a_i \neq \mathsf{stop}$. If (1) is true, then b is not

```
AuditNMDPApprox($m = \langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle,\ b = [s_1, a_1, s_2, a_2, \ldots, s_n, a_n]$):
if (ImpossibleMDP($m, b$))
  return true     //  behavior impossible for NMDP
$\langle \mathsf{V}^*_{\mathsf{low}}, \mathsf{V}^*_{\mathsf{up}} \rangle :=$ SolveMDPApprox($m$)
for ($i := 1$; $i \leq n$; $i{+}{+}$):
  if ($\mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) < \mathsf{V}^*_{\mathsf{low}}(s_i)$):
    return true     //  action suboptimal
  if ($\mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) \leq 0$ and $a_i \neq \mathsf{stop}$):
    return true     //  action redundant
return false
```

Figure 4: The algorithm AuditNMDPApprox. SolveMDPApprox is an MDP approximation algorithm. Figure 3 shows ImpossibleMDP.

a possible behavior of m by Lemma 5. If (2) is true, then for that i, $Q^*_m(s_i, a_i) \neq V^*_m(s_i)$ since $Q^*_m(s_i, a_i) \leq \mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) < \mathsf{V}^*_{\mathsf{low}}(s_i) \leq V^*_m(s_i)$. If (3) is true, then for that i, $a_i \neq \mathsf{stop}$ does not imply that $Q^*_m(s_i, a_i) > 0$ since $a_i \neq \mathsf{stop}$ and $Q^*_m(s_i, a_i) \leq \mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) \leq 0$. Thus, under each of these cases, Lemma 4 shows that $b = [s_1, a_1, s_2, a_2, \ldots, s_n, a_n]$ is not in $\mathsf{nbehv}(m)$. This fact implies that $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty since $\mathsf{log}^{-1}(b) = \{b\}$. □

AuditNMDPApprox is not complete: it may return *false* in cases where AuditNMDP would return *true*. These additional results of *false* mean that additional violations of exclusivity rules might go uncaught and additional compliance with prohibitive rules might go unproven. However, since *false* indicates an inconclusive audit, they do not alter soundness of the implementation.

When AuditNMDPApprox returns *false*, the auditor may use a more accurate approximation algorithm for SolveMDPApprox in hopes that improving accuracy of the approximations will produce the conclusive response of *true*. For the value iteration algorithm, the auditor just needs to rerun the algorithm with a lower value for ϵ. There always exists a value of ϵ small enough to show that $\mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) < \mathsf{V}^*_{\mathsf{low}}(s_i)$ when it is actually the case that $Q^*_m(s_i, a_i) < V^*_m(s_i)$. However, when $Q^*_m(s_i, a_i) = 0$, there will be no value of ϵ small enough to make $\mathsf{Q}^*(m, \mathsf{V}^*_{\mathsf{up}}, s_i, a_i) \leq 0$ true. Thus, AuditNMDPApprox using value iteration will never catch when $\mathsf{log}^{-1}(b) \cap \mathsf{nbehv}(m)$ is empty because an action of b is redundant but otherwise optimal ($V^*_m(s_i) = Q^*_m(s_i, a_i) = 0$ but $a_i \neq \mathsf{stop}$ for some a_i).

We programmed our implementation in the Racket dialect of Scheme [32]. The implementation is available at:

`http://www.cs.cmu.edu/~mtschant/purpose/`

The implementation uses an explicit representation of the state and actions spaces. The transition and reward functions are represented using hash maps. Since we did not optimize the implementation, we did not benchmark its performance. However, in Section 5.3, we use it to aid understanding a complex example and report its performance in that section.

5.3 Example: Creating an Operating Procedure

In some environments, an auditee may have difficulty determining whether an action is allowed under a policy. For example, Regional Health Information Organizations (RHIOs) store and make available medical records for a region. Since RHIOs are a new technology and do not directly provide treatment, arguments may arise over what actions are allowed under the exclusivity restriction that records may only be used for the purpose of treatment.

A physician considering reading such a record may find the circumstances too complex to understand without help, but we cannot expect the physician to perform the modeling required to use our auditing algorithm. However, an RHIO may use our algorithm to audit simulated logs of possible future uses and determine which actions the restriction allows. The RHIO may generalize these quantitative results to a qualitative operating procedure, such as *the physician may read records of patients with whom he does not have a current relationship only when seeing that patient in the future is highly likely.* Below, we show an example of reasoning that could lead to this procedure. After constructing a model of an RHIO, we approach this problem by using the implementation of AuditNMDPapprox algorithm discussed in Section 5.2.

5.3.1 Modeling

Our model of an RHIO stores the records of various patients. Let p_i represent the probability that the physician will see patient i next. Let $p_\varnothing$ represent the probability that the physician does not have a next patient. The action read_i represents reading the ith patient's record. The reward ρ_i^{t} is a reward measuring how well the physician treats patient i without seeing that patient's record beforehand. (t stands for "treat".) Let δ_i^{r} represent the improvement in how well the physician may treat patient i with seeing the record beforehand. (r stands for "read".) The model also includes the action study. Studying yields an improvement of δ_i^{s} in the level of treatment the physician may provide to patient i. (s stands for "study".) In general, reading a patient records improves treatment for that patient whereas studying improves the treatment of all patients.

We assume that the effects of reading a medical record or studying wears off over time. Not only does this assumption model the limited nature of human memory, but also allows us to model an infinite number of time steps using a finite model. In particular, we encode the last h actions (reading, studying, and treating) of the physician in the states of the model. The reward for treating a patient depends upon this history.

We model this example as a family of NMDPs that depend upon the parameter h, the number of steps before a physician forgets something. For simplicity, we assume that the number of patients is equal to h as well. (Having more patients than the physician can remember cannot change his behavior.) In the case, where more than h patients are stored in an RHIO, we consider the subset of the RHIO that holds the patients that would benefit the most from having their records read (those that maximizes $p_i * \delta_i^{\mathsf{r}}$).

Formally, let m_{ex2}^h be the model for the parameter h (and others introduced below). $m_{\mathsf{ex2}}^h = \langle \mathcal{S}, \mathcal{A}, t, r, \gamma \rangle$ where the action space $\mathcal{A}$ is equal to $\{\mathsf{stop}, \mathsf{treat}, \mathsf{study}, \mathsf{read}_1, \ldots, \mathsf{read}_h\}$. The state space $\mathcal{S}$ is equal to $\{\mathsf{treat}, \mathsf{study}, \mathsf{read}_1, \ldots, \mathsf{read}_h\}^h \times \mathcal{C}$ where $\{\mathsf{treat}, \mathsf{study}, \mathsf{read}_1, \ldots, \mathsf{read}_h\}^h$ encodes a h-step history of the physician's actions and $\mathcal{C}$ is the set of possible conditions in which the physician may currently find himself. The history records, in order, which action the physician made in each of the h most recent steps before the current step unless the physician has taken the do-nothing action stop. Once the physician performs stop, the history is frozen at its current value and does not record the current or future stop actions. The history does not record the stop actions since they always result in returning to the current state making updating the history impossible. However, this failure to record the stop action does not alter the optimal strategy of the NMDP m_{ex2}^h since stop is of zero reward and results in a self-loop at every state. The set of conditions $\mathcal{C}$ is equal to $\{\emptyset, \mathsf{o}, 1, \ldots, h\}$ where $\emptyset$ represents no patients currently wanting to see the physician, o

form of state	action	reward
$\langle\langle a_1, \ldots a_h\rangle, c\rangle$ (any state)	study	0
$\langle\langle a_1, \ldots a_h\rangle, c\rangle$ (any state)	read_i	0
$\langle\langle a_1, \ldots a_h\rangle, \emptyset\rangle$	treat	0
$\langle\langle a_1, \ldots a_h\rangle, \mathsf{o}\rangle$	treat	$\rho^{\mathsf{t}}_{\mathsf{o}} + n * \delta^{\mathsf{s}}_{\mathsf{o}}$
$\langle\langle a_1, \ldots, \mathsf{read}_i, \ldots a_h\rangle, i\rangle$	treat	$\rho^{\mathsf{t}}_{\mathsf{o}} + \delta^{\mathsf{r}}_i + n * \delta^{\mathsf{s}}_i$
$\langle\langle a_1, \ldots a_h\rangle, i\rangle$ where read_i is not in $\langle a_1, \ldots a_h\rangle$	treat	$\rho^{\mathsf{t}}_i + n * \delta^{\mathsf{s}}_i$

Table 1: The rewards for m^h_{ex2}. In the last three rows, n stands for the number of instances of study in $\langle a_1, \ldots a_h\rangle$.

(short for "other") represents a patient not in the RHIO attempting to see the physician, and i in $\{1, \ldots, h\}$ represents the ith patient of the RHIO attempting to see the physician.

At each time step, the physician chooses whether to treat the current patient (if any), read a patient's record, study, or do nothing. This updates his history by replacing the oldest of the h entries with this choice. The condition also probabilistically updates to a value of $\mathcal{C}$. The transition function t is such that $t(\langle\langle a_1, a_2, \ldots, a_h\rangle, c\rangle, a)$ is equal to a distribution d over these possible next states. In particular, d depends upon additional parameters p_c for each c in $\mathcal{C}$. p_c provides the probability of c being the next condition in which the physician finds the hospital. The distribution d assigns the probability of p_c to the next state $\langle\langle a_2, \ldots, a_h, a\rangle, c\rangle$ for each c and the probability of 0 for all other states where a is the current action of the physician.

Table 1 lists the rewards for each state and action. The reward for the actions read_i or study is always 0. The reward for treat depends upon the state. For the state $\langle\langle a_1, \ldots a_h\rangle, \emptyset\rangle$, the reward is also 0. For the state $\langle\langle a_1, \ldots a_h\rangle, \mathsf{o}\rangle$, the reward is $\rho^{\mathsf{t}}_{\mathsf{o}} + n * \delta^{\mathsf{s}}_{\mathsf{o}}$ where $\rho^{\mathsf{t}}_{\mathsf{o}}$ is the base reward for treating a patient not in the database, n is the number of instances of study in $\langle a_1, \ldots, a_h\rangle$, and $\delta^{\mathsf{s}}_{\mathsf{o}}$ is the additional reward achieved per studying action. For the state $\langle\langle a_1, \ldots a_h\rangle, i\rangle$, if there exists j in $\{1, \ldots, h\}$ such that $a_j = \mathsf{read}_i$, the reward will be $\rho^{\mathsf{t}}_i + \delta^{\mathsf{r}}_i + n * \delta^{\mathsf{s}}_i$ where ρ^{t}_i is base reward for treating patient i, δ^{r}_i is the additional reward for having read the patient's record, and n and δ^{s}_i are as before. For the state $\langle\langle a_1, \ldots a_h\rangle, i\rangle$, if there does not exist j in $\{1, \ldots, h\}$ such that $a_j = \mathsf{read}_i$, the reward will be $\rho^{\mathsf{t}}_i + n * \delta^{\mathsf{s}}_i$. ρ^{t}_i, δ^{r}_i, δ^{s}_i, $\rho^{\mathsf{t}}_{\mathsf{o}}$, and $\delta^{\mathsf{s}}_{\mathsf{o}}$ are all additional parameters to the model. We also treat the discounting factor γ as a parameter.

The number of actions is $|\{\mathsf{stop}, \mathsf{treat}, \mathsf{study}, \mathsf{read}_1, \ldots, \mathsf{read}_h\}| = 3 + h$. The number of states is

$$|\{\mathsf{treat}, \mathsf{study}, \mathsf{read}_1, \ldots, \mathsf{read}_h\}^h \times \mathcal{C}| = (2 + h)^h * (2 + h) = (h + 2)^{h+1}$$

For every state s and action a except stop, each of the possible $h + 2$ conditions in $\mathcal{C}$ could arise in the next state from performing action a in state s. Presuming all the probability parameters p_c are non-zero, the resulting number of non-zero transitions is

$$|\mathcal{S}| * |\mathcal{A} - \{\mathsf{stop}\}| * |\mathcal{C}| + |\mathcal{S}| = (h+2)^{h+1} * (h+3-1) * (h+2) + (h+2)^{h+1} = (h+2)^{h+3} + (h+2)^{h+1}$$

where the second summand accounts for the self-loop under stop at each state.

Since m^2_{ex2} has 64 states and 1088 non-zero transitions, we cannot easily represent the whole model in a diagram. Thus, in Figure 5, we show just part of m^2_{ex2}. It shows only the part of the NMDP relevant to transitions from the states $\langle\langle \mathsf{s}, 2\rangle, \emptyset\rangle$ or $\langle\langle 2, \mathsf{s}\rangle, 2\rangle$. The part of m^2_{ex2} is sufficient to illustrate the possibility of multi-state cycles. In particular, it shows the possibility of executions of the following form

$$[\langle\langle \mathsf{study}, \mathsf{read}_2\rangle, \emptyset\rangle, \quad \mathsf{study}, \quad \langle\langle \mathsf{read}_2, \mathsf{study}\rangle, 2\rangle, \quad \mathsf{read}_2, \quad \langle\langle \mathsf{study}, \mathsf{read}_2\rangle, \emptyset\rangle, \quad \ldots]$$

5.3.2 Methodology

We conducted experiments with our implementation to gain a feel for how the values of the parameters affects the allowed behavior. For simplicity, in our experiments, we treat all patients in the RHIO as identical and use the same rewards in the case of a patient not in the RHIO (i.e., $p_i = p_j$, $\rho^{\mathsf{t}}_i = \rho^{\mathsf{t}}_j = \rho^{\mathsf{t}}_{\mathsf{o}}$, $\delta^{\mathsf{r}}_i = \delta^{\mathsf{r}}_j$, and $\delta^{\mathsf{s}}_i = \delta^{\mathsf{s}}_j = \delta^{\mathsf{s}}_{\mathsf{o}}$ for all i and j in $\{1, \ldots, h\}$). Thus, we simply write p_{i} in the place of p_i for all i in $\{1, \ldots, h\}$. We also write δ^{r} in the place of δ^{r}_i, ρ^{t} for ρ^{t}_i or $\rho^{\mathsf{t}}_{\mathsf{o}}$, and δ^{s} for δ^{s}_i or $\delta^{\mathsf{s}}_{\mathsf{o}}$ for all i in $\{1, \ldots, h\}$.

Our experiments study how large the improvement δ^{r} must be compared to the improvement δ^{s} for the obeying the policy means that the physician must read a patient's record instead of studying. Given fixed values for all other parameters, we call this lowest value of δ^{r} for which the physician may read the record of a patient without violating the policy the *reading threshold*.

We use our implementation to estimate the reading threshold using simulations. We call this estimation the *simulatively estimated reading threshold* (SERT). Each simulation corresponds to setting the value of δ^{r} to some value v and testing with the AUDITNMDPAPPROX implementation whether reading is allowed at the value v. In particular, we test whether studying (as opposed to reading a record) at the state $\langle\langle \mathsf{treat}, \ldots, \mathsf{treat}\rangle, \emptyset\rangle$ is a violation of the policy. If so, then v is an upper bound on the reading threshold; if not, then v is a lower bound. The algorithm establishes the initial lower bound as 1 and finds an initial upper bound by exponentially increasing the value of v until AUDITNMDPAPPROX returns *true*. After establishing initial lower and upper bounds, the estimation algorithms iteratively uses their average for the next value of v tested by AUDITNMDPAPPROX to find either a tighter lower or upper bound. The estimation algorithm continues until the bounds are within 1% of one another and uses their average as the SERT. If we were using AUDITNMDP, then this procedure would guarantee that the resulting SERT is within 0.5% of the true reading threshold. However, since we use the approximate AUDITNMDPAPPROX, the SERT may be further from the true reading threshold.

We implemented these estimation techniques in the Racket dialect of Scheme to use our implementation of the AUDITNMDPAPPROX algorithm. They may be downloaded from:

`http://www.cs.cmu.edu/~mtschant/purpose/`

We ran our implementations on a Lenovo U110 laptop computer with 3GB of memory and a 1.60 GHz Intel Core 2 Duo CPU running the DrRacket interpreter in Windows Vista.

5.3.3 Results

We compare the SERTs across several models in the family m^h_{ex2}. Table 2 summarizes the results for each model we studied. The table also reports the running time required to compute the SERT for each model.

For all experiments, we use δ^{s} equal to 1. Most of the experiments used $h = 2$. For each of these experiments, we used three different values for the discounting factor γ: 0.9, 0.1, and 0.01. We ran three experiments with $h = 3$.

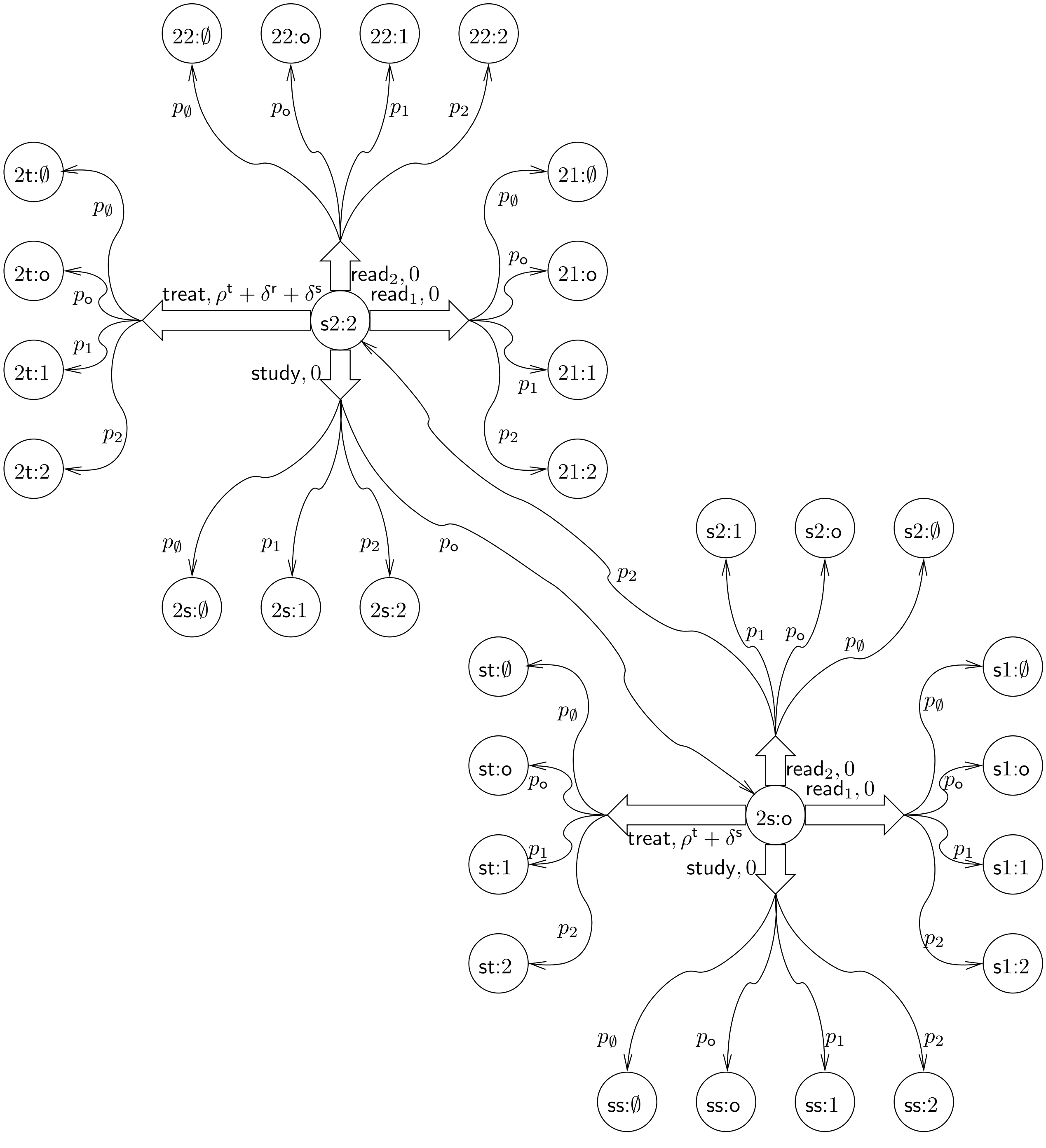

Figure 5: Part of the NMDP m^2_{ex2}. The figure only shows transitions and rewards originating at either state $\langle\langle \mathsf{s}, 2\rangle, \mathsf{none}\rangle$ or $\langle\langle 2, \mathsf{s}\rangle, 2\rangle$. It only shows states involved in one of these transitions. It abbreviates the state $\langle\langle a_1, a_2\rangle, c\rangle$ as $\bar{a}_1\bar{a}_2{:}c$ where $\bar{a}_1$ and $\bar{a}_2$ abbreviations for actions: read_1 becomes 1; read_2 becomes 2; and treat, t.

s	p_{i}	p_{o}	$p_{\emptyset}$	ρ^{t}	γ	SERT	time
2	0.01	0.95	0.03	1000	0.01	97.539	15 sec
2	0.01	0.95	0.03	1000	0.1	97.539	20 sec
2	0.01	0.95	0.03	1000	0.9	98.945	234 sec
2	0.01	0.95	0.03	10000	0.01	97.539	16 sec
2	0.01	0.95	0.03	10000	0.1	97.539	22 sec
2	0.01	0.95	0.03	10000	0.9	98.945	272 sec
2	0.01	0.95	0.03	100	0.01	97.539	16 sec
2	0.01	0.95	0.03	100	0.1	97.539	19 sec
2	0.01	0.95	0.03	100	0.9	98.945	196 sec
2	0.01	0.95	0.03	10	0.01	97.539	15 sec
2	0.01	0.95	0.03	10	0.1	97.539	16 sec
2	0.01	0.95	0.03	10	0.9	98.945	162 sec
2	0.01	0.95	0.03	1	0.01	97.539	13 sec
2	0.01	0.95	0.03	1	0.1	97.539	16 sec
2	0.01	0.95	0.03	1	0.9	**74.336**	128 sec
2	0.0001	0.9698	0.03	1000	0.01	9753.906	18 sec
2	0.0001	0.9698	0.03	1000	0.1	9753.906	24 sec
2	0.0001	0.9698	0.03	1000	0.9	9894.531	283 sec
2	0.0001	0.9698	0.03	10	0.01	9753.906	18 sec
2	0.0001	0.9698	0.03	10	0.1	9753.906	20 sec
2	0.0001	0.9698	0.03	10	0.9	9894.531	194 sec
2	0.0001	0.9698	0.03	1	0.01	9753.906	16 sec
2	0.0001	0.9698	0.03	1	0.1	9753.906	20 sec
2	0.0001	0.9698	0.03	1	0.9	**7363.281**	157 sec
2	0.0001	0.95	0.05	1000	0.01	9542.969	18 sec
2	0.0001	0.95	0.05	1000	0.1	9542.969	24 sec
2	0.0001	0.95	0.05	1000	0.9	9683.594	283 sec
2	0.01	0.8	0.18	1000	0.01	82.07	15 sec
2	0.01	0.8	0.18	1000	0.1	82.07	20 sec
2	0.01	0.8	0.18	1000	0.9	84.18	234 sec
3	0.01	0.94	0.03	1000	0.01	97.539	45 min
3	0.01	0.94	0.03	1000	0.1	97.539	63 min
3	0.01	0.94	0.03	1000	0.9	98.242	12 hours

Table 2: Results of experiments on m^{h}_{ex2}. In all cases $\delta^{\mathsf{s}} = 1$. The values for the estimations are rounded to three decimal places. Two outliers are in boldface.

The table does not report upon the running time for a single call to AuditNMDPapprox. For the $h = 2$ cases, the running time for a single call of AuditNMDPapprox varied between 1.3 and 27 seconds. For the $h = 3$ cases, the running time varied between 261 seconds and 70 minutes. Each computation of sert took between 10 and 12 calls to AuditNMDPapprox.

Examining Table 2, we see two outliers (in boldface). The outliers use the very low value of 1 for ρ^{t}. We compared the optimal strategies for these outliers with similar models. Only in the case of the outliers does the physician read patient records rather than provide treatment even when a patient is present. Intuitively, the physician shows this behavior since the reward ρ^{t} for treating a patient without having either read the patient's record or studying is less than the expect increase in rewards for studying or reading a record. This effect disappears for lower values of γ since increases in future rewards become more heavily discounted.

As expected from known complexity results [72], increasing γ increases the running time.

5.3.4 Discussion

The complexity of the above calculations highlights how our model of planning does not correspond to how humans plan (further discussed in Section 8). We cannot expect physicians to perform complex modeling and analysis let alone to use computer simulations before deciding whether to read a record or study. However, compliance officers at hospitals may find these results helpful while drafting policy manuals.

For example, consider a large hospital where the probability of a physician seeing a typical patient in the RHIO is less than 1 in $10,000$. At such a hospital, the reading threshold of about 9700 holds across various of values for ρ^{t} and γ. Extrapolating from the results for the tests using $h = 3$ with $p_{\mathsf{i}} = 0.01$ instead of 0.0001, one may conclude that the value is likely to remain around 9700 for larger values of h. In many settings, managers may find an improvement from reading a patient's record of 9700 times the improvement from studying inconceivable. In this case, a policy manual may quantitatively summarize the quantitative results shown in Table 2 as prohibiting a physician from reading patient records unless the physician has a reason to believe that the patient is much more likely than average to be seeking care. Such a prohibition may not make sense at a small practice where the probability of seeing an average patient is 1 in 100 since reading a record could conceivably produce an improvement of 97 times the improvement from studying.

6 Empirical Study of Semantics

6.1 Goals

Both previous work and this work offer methods for enforcing privacy policies that feature purpose restrictions. These methods test whether a sequence of actions violates a clause of a privacy policy that restricts certain actions to be only for certain purposes. By providing a test for whether the purpose restriction is violated, these methods implicitly provide a semantics for these restrictions.

To ensure that these methods correctly enforce the privacy policy, one must show that the semantics employed by a method matches the intended meaning of the policy. Unfortunately, determining the intended meaning of a policy from its text is often impossible. Furthermore, these policies often act as agreements among multiple parties who may differ in their interpretation of the policy. For these reasons, we compare the semantics proposed by these methods of policy enforcement to the most common interpretations of policies.

While previous works have not provided a formal semantics, it appears that many works (e.g., [3, 38]) flag actions as a violation if they do not further the purpose in question. (See Section 7 for a description of past works.) In particular, these works make assumptions about how people think about *purpose* in the context of enforcing a privacy policy that restricts an agent to only performing a certain class of actions for a certain purpose. The following hypothesis characterizes these assumptions:

H1. The agent obeys the restriction if and only if the action furthered the purpose.

This hypothesis entails the following hypothesis about how people interpret the meaning of *purpose*:

H1'. An action is for a purpose if and only if that action furthers that purpose.

Our work instead asserts that an action may be for a purpose even if that purpose is never furthered. In particular, we assert that the action merely has to be part of a plan for furthering that purpose. Thus, our formalism assumes the following hypothesis (in the same context as above):

H2. The auditee obeys the restriction if and only if the auditee performed that action as part of a plan for furthering that purpose.

(We do not construct our algorithms directly from Hypothesis H2. Rather they are approximations using only observable information.) Similarly, this hypothesis entails the following:

H2'. An action is for a purpose if and only if the auditee performed that action as part of a plan for furthering that purpose.

To show that our work provides a method of enforcing purpose restrictions more faithful to their common meaning, we would like to disprove Hypotheses H1 and H1' while proving Hypotheses H2 and H2'.

As Hypothesis H1 is a bi-implication, we can disprove it by disproving either of the following hypotheses (here and henceforth, in the same context as above):

H1a. If the action furthers a purpose, then the auditee obeys the restriction.

H1b. If the auditee obeys the restriction, then the action furthers a purpose.

We will attempt to disprove both Hypotheses H1a and H1b.

Similarly, Hypothesis H2 breaks into two sub-hypotheses:

H2a. If the auditee performed an action as part of a plan for furthering a purpose, then the auditee obeyed the restriction.

H2b. If the auditee obeyed the restriction, then the auditee performed the action as part of a plan for furthering that purpose.

We will test both of these hypotheses by providing example scenarios of an auditee performing actions with descriptions of his plans. However, these tests will not prove either of these hypotheses as doing so would require testing them under all scenarios. Indeed, given that some tests could be carefully crafted to bring about success for reasons unrelated to planning, such testing does not necessarily provide good evidence in favor of these hypotheses. To provide better evidence for the truth of Hypothesis H2, we will also test the following related hypothesis:

	Furthered purpose	Did not further purpose
Planned for purpose	C_{pf}	$\mathrm{C}_{\mathsf{p\bar{f}}}$
Not planned for purpose	$\mathrm{C}_{\mathsf{\bar{p}f}}$	$\mathrm{C}_{\mathsf{\bar{p}\bar{f}}}$

Table 3: Classes of Scenarios for Survey Questionnaire. Each position in the grid identifies the scenario class associated with the values of the two factors given on each axis.

H2c. Describing an action as being part of a plan for furthering purpose as opposed to not being part of such a plan in a scenario causes people to think that the auditee obeyed the restriction.

H2c may be viewed a causal or directional version of H2. Unlike H2a and H2b, which may be tested with unrelated scenarios, H2c must be tested with scenarios that only differ from one another in whether or not the action is part of a plan for the purpose in question.

For completeness we also test the causal version of H1:

H1c. Describing an action as furthering a purpose as opposed to not furthering a purpose in a scenario causes people to think that the auditee obeyed the restriction.

As Hypothesis H1 leads to Hypotheses H1a, H1b, and H1c, Hypothesis H1' leads to corresponding hypotheses H1a', H1b', and H1c'. Similarly, H2' leads to H2a', H2b', H2c'. We also test these hypotheses to provide additional evidence for our formalism.

6.2 Methodology

Approach. We may disprove Hypothesis H1a by exhibiting a scenario in which an action of an auditee furthers a purpose, but people feel that the auditee did not obey a purpose restriction stating that the action may only be performed for that purpose. We may disprove Hypothesis H1b by exhibiting an scenario in which an action does not further a purpose, but people feel that the auditee obeyed the restriction. To test Hypothesis H1c, we construct a pair of scenarios that differs only in whether the action furthered the purpose in question, and show that people's feelings about whether the auditee obeyed the restriction is unchanged across the two scenarios.

Testing Hypotheses H2a, H2b, and H2c is similar to testing the corresponding hypothesis for H1. However, we expect the opposite results. For example, to test Hypothesis H2c, we construct a pair of scenarios that differs only in whether the auditee performed that action as part of a plan for furthering that purpose. We expect to show that people feel that the auditee obeyed the restriction only in the scenario in which the action is part of a plan for furthering that purpose.

To these ends, we use four classes of scenarios: Classes C_{pf}, $\mathrm{C}_{\mathsf{p\bar{f}}}$, $\mathrm{C}_{\mathsf{\bar{p}f}}$, and $\mathrm{C}_{\mathsf{\bar{p}\bar{f}}}$. Each class is determined by two factors: (1) whether or not the action furthers the purpose in question in the scenario and (2) whether or not the auditee performs the action as part of a plan for furthering the purpose. Table 3 identifies these classes along these two axes. (E.g., $\mathrm{C}_{\mathsf{\bar{p}f}}$ stands for the scenario that was *not* **p**lanned ($\mathsf{\bar{p}}$) for the purpose but **f**urthered (f) it.)

Showing that people think the auditee does not obey the restriction in Scenario Class $\mathrm{C}_{\mathsf{\bar{p}f}}$ is sufficient for disproving Hypothesis H1 by disproving Hypothesis H1a. Showing that people think the auditee obeys the restriction in Class $\mathrm{C}_{\mathsf{p\bar{f}}}$ provides additional evidence that previous approaches are insufficient by disproving the other direction, H1b, of the bi-implicational Hypothesis H1.

$\mathbf{S}_{\mathsf{pf}}$**.** A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of treating the patient. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist succeeds in treating the patient.

$\mathbf{S}_{\mathsf{p}\bar{\mathsf{f}}}$**.** A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of treating the patient. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist did *not* succeed in treating the patient.

$\mathbf{S}_{\bar{\mathsf{p}}\mathsf{f}}$**.** A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of reducing costs for the hospital. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist succeeds in treating the patient.

$\mathbf{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$**.** A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of reducing costs for the hospital. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist did *not* succeed in treating the patient.

Table 4: Questionnaire Scenarios. For each scenario class, the scenario used on the questionnaire.

Comparing Class C_{pf} against $\mathrm{C}_{\mathsf{p}\bar{\mathsf{f}}}$ tests Hypothesis H1c. Comparing Class $\mathrm{C}_{\bar{\mathsf{p}}\mathsf{f}}$ against $\mathrm{C}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ also tests Hypothesis H1c.

For Hypothesis H2, showing that people think the auditee obeyed the restriction in Classes C_{pf} and $\mathrm{C}_{\mathsf{p}\bar{\mathsf{f}}}$ each provides evidence for Hypothesis H2a. Showing that people think the auditee does not obey the restriction in Classes $\mathrm{C}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathrm{C}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ each provides evidence for Hypothesis H2b by way of the contrapositive. Comparing Class C_{pf} against $\mathrm{C}_{\bar{\mathsf{p}}\mathsf{f}}$ and comparing Class $\mathrm{C}_{\mathsf{p}\bar{\mathsf{f}}}$ against $\mathrm{C}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ test Hypothesis H2c.

Questionnaire Construction. We constructed a questionnaire with four scenarios, one from each of the four scenario classes above. The auditee in these four scenarios is subject to a privacy policy that states that the auditee may only use a type of information for a single purpose. The policy we used for the questionnaire is as follows:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

Table 4 presents the scenarios where Scenario S_{xy} is the scenario in Scenario Class C_{xy}.

For each scenario, we ask the participant Question Q1: whether the auditee obeyed the policy. The participant may select among *yes*, *no*, and *I don't know*. We conjecture that the majority of participants will answer this question with *yes* for the scenarios in classes C_{pf} and $\mathrm{C}_{\mathsf{p}\bar{\mathsf{f}}}$, and with *no* for $\mathrm{C}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathrm{C}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$.

To help determine the reasoning used by the participants, which would be especially useful if our survey results deviated from the expected, we included the free form Question Q2 asking why the participant chose the answer he did to the first question. We also asked the closely related

Q1. Did the case worker obey the above privacy policy?

Q2. Why did you answer Question 4 as you did?

Q3. Did the case worker share the record with the specialist for the purpose of treatment?

Q4. Was the goal of the case worker's plan to treat the patient?

Q5. Did the specialist succeed in treating the patient?

Table 5: Questionnaire Questions. Note that Question Q1 is numbered as Question 4 on the questionnaire.

Question Q3 of whether or not the action was for the allowed purpose of treatment. We expected this question to be answered identically to the first Question Q1. We included it to help determine whether the questionnaire was well worded and to test the Hypotheses H1' and H2'.

Lastly, we included two simple questions, Questions Q4 and Q5, about each scenario. These questions have objectively correct answers that the participant can easily find by reading the scenarios. Checking that the participant chose the correct answer allowed us to ensure that the participants actually read the scenario and answered accordingly rather than arbitrarily. Table 5 shows the wording of these questions.

We presented these scenarios and questions as a questionnaire. For each survey participant, we randomly ordered the scenarios to reduce the effects that scenario ordering may have. For each scenario, we ordered the questions with the objective questions, Q4 and Q5, first to encourage the participant to read the scenario before answering the subjective questions in which we are interested. The subjective questions are ordered as follows: Q3, Q1, and, lastly, Q2. Appendix A shows a sample questionnaire.

Pilot Study. Before running the main survey we conducted a small scale pilot study of ten participants. The participants were recruited on Amazon Mechanical Turk (`www.mturk.com`) using a small payment of $1.50 (USD). (Appendix A.3 shows the advertisement.) Participants took the survey online using Mechanical Turk's survey functionality without randomly ordering the scenarios.

The goal of this pilot study was to ensure that our recruitment and survey mechanisms worked. We also closely examined the responses to determine whether the participants were seriously answering the questions or not, and whether Questions Q4 and Q5 identified arbitrary responses. As the goal of this study was not to collect data on our hypotheses, we did not statistically analyze the data. However, we will qualitatively describe the results below.

In the pilot study, seven of the ten respondents matched our predictions perfectly. One respondent deviated for a single answer in a manner inconsistent with the other answers provided by the respondent. Thus, we suspect that his response is most likely an error in selecting the answer.

A second respondent said that the action was not for the purpose of treatment in Scenarios $\mathrm{S}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$, but that, nevertheless, the case worker obeyed the policy since the specialist would try to provide treatment. This response suggests that Hypotheses H2 and H2' are more than trivially different.

The third respondent to deviate from our hypothesis claimed that the action was for the purpose of treatment and the case worker obeyed the policy in all of the scenarios including Scenarios $\mathsf{S}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathsf{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ where goal of the case worker was cost reduction. This respondent's answer to Question Q2 suggests that the case worker did not violate the policy as the scenarios provide evidence that the specialist provided treatment whereas they provide no evidence that any of the actions reduced costs. For example, this respondent provided the following for Question Q2 given Scenario $\mathsf{S}_{\bar{\mathsf{p}}\mathsf{f}}$:

> Though the case worker's goal was cost-reduction, the medical records were still provided for the purpose of treating the patient; simply giving medical records to outside specialists, with no further actions, would not be a way to reduce costs for a hospital.

This response highlights that our scenarios discuss treatment in more detail than cost reduction, which could have unintended effects on people's analysis of them.

Interestingly, while these two deviations do not match our Hypothesis H2, they are consistent with the approximations our algorithm makes. While these deviations suggest interesting directions for future studies, we decided that these issues did not warrant rewriting the scenarios to include more information on cost reduction or to examine more carefully the differences between Hypotheses H2 and H2'.

None of the respondents said that the policy was violated in Scenario $\mathsf{S}_{\mathsf{p}\bar{\mathsf{f}}}$, providing evidence against Hypothesis H1. None of the respondents answered Questions Q4 or Q5 incorrectly and none of their responses appeared arbitrary.

Survey Protocol. The main survey consisted of two hundred participants. We conducted the survey in the same manner as the pilot study but with three changes. First, given the ease with which we recruited participants for the pilot study, we reduced the payment to \$0.50.

Second, while still using Mechanical Turk to recruit and pay participants, we used Survey Gizmo (`www.surveygizmo.com`) to conduct the survey. This change allowed us to randomly order the scenarios for each participant.

Third, given the success of Questions Q4 and Q5, we decided before the survey to exclude from the results any participants who got more than one of them wrong in total across all four scenarios. The odds of correctly guessing either all the answers or all but one is less than 4% presuming the participant knows that *I don't know* is never a correct answer.[1]

We analyzed the survey responses according to the statistical model presented in the next section.

6.3 Statistical Modeling

In this section, we provide a detailed description of the statistical tests we employ in the next section. Those with a background in hypothesis testing and statistics may find the following summary sufficient.

[1] The odds of guessing correctly one of the questions is $\frac{1}{2}$ since there are two possible answers (ruling out *I don't know*). Each of the four scenarios have two questions meaning that seven or eight would have to be correctly guessed for a guessing participant to avoid rejection. We model these guesses using the binomial distribution, which has the cumulative distribution function $F(x; n, p) = \Pr[X \leq x] = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}$ where x is the number of successes, n the number of trails, and p is the probablity of success. In particular, we find that odds of getting 7 or more success is $1 - F(6; 8, \frac{1}{2})$ where $F(6; 8, \frac{1}{2})$ is the odds of getting 6 or fewer successes. This is $1 - F(6; 8, \frac{1}{2}) = 1 - \sum_{i=0}^{6} \binom{8}{i} \frac{1}{2}^i (1 - \frac{1}{2})^{8-i} < 1 - 0.96 = 0.04$. Without ruling out the option of *I don't know*, the odds of successfully avoiding detection would be $1 - F(6; 8, \frac{1}{3}) < 0.01$.

Summary. Each of the hypotheses H1a, H1b, H2a, and H2b makes predictions about whether Question Q1 will be answered with *yes* or *no*. We model these answers as a draw from a binomial distribution and we interpret these predictions as predictions about probability of success for the binomial distribution. For Hypotheses H1a and H1b, we treat their predictions as the null hypotheses about the probability of success and attempt to reject them to disprove H1. We treat the predictions of H2a and H2b as the alternative hypotheses and attempt to reject their negations as null hypotheses to provide evidence in favor of H2. Table 7 presents how to convert these predictions in testable hypotheses. In short, we interpret a prediction that a question will be answered with a certain response as an assertion that the probability of success (seeing that response) is at least 0.5.

To test Hypothesis H1c, we use McNemar's Test to test whether or not an action furthering a purpose has a statistically significant effect on how people answer Question Q1. We test Hypothesis H2c using McNemar's Test across scenarios that only differ in the goal of the auditee's plan.

We test Hypotheses H1' and H2' analogously using Question Q3 in the place of Q1. For all statistical tests, we use $\alpha = 0.05$ for the threshold of statistical significance.

6.3.1 Hypothesis Testing

An underlying presumption of this work is that *purpose* has an objective definition on which people generally agree. However, even under this presumption, we cannot expect that, for each question, every participant will respond with the same answer. Some participants might misread the question or hold non-standard views. Thus, we model each response to Question Q1 as a trial of a distribution over the three possible responses: *yes*, *no*, and *I don't know*.

The hypotheses H1a, H1b, H2a, and H2b each make predictions about how people will answer Question Q1 in various scenarios. For example, Hypothesis H1a predicts that people will answer Question Q1 with *yes* rather than *no* when given a scenario of Class $\mathrm{C}_{\bar{\mathsf{p}}\mathsf{f}}$. Literally interpreted, Hypothesis H1a predicts that the probability of answering *yes* under Scenario $\mathrm{S}_{\bar{\mathsf{p}}\mathsf{f}}$, which we denote as $p_{\bar{\mathsf{p}}\mathsf{fy}}$, will be 1. However, as discussed above, we would expect to see the probability $p_{\bar{\mathsf{p}}\mathsf{fy}}$ being somewhat less than 1 even if Hypothesis H1a is true. The lower the probability, the more questionable the truth of the hypothesis becomes. The lower limit at which we reject the hypothesis as false depends upon how one formalizes the hypothesis. We choose to set this limit at the probability 0.5 since a hypothesis that does not correctly predict the majority of outcomes appears clearly false to us. Thus, we formalize this prediction as:

H1a$_{\mathsf{0y}}$. $p_{\bar{\mathsf{p}}\mathsf{fy}} \geq 0.5$

As we hope to disprove Hypothesis H1a, we would like to cast doubt on the Hypotheses H1a$_{\mathsf{0y}}$, which makes it a *null hypothesis* we hope to reject. Rejecting the null hypothesis provides evidence in favor of the *alternative hypothesis* we hope to show:

H1a$_{\mathsf{ay}}$. $p_{\bar{\mathsf{p}}\mathsf{fy}} < 0.5$

Since H1a$_{\mathsf{0y}}$ predicts a large number of *yes* responses and H1a$_{\mathsf{ay}}$ predicts a small number, the smaller the number of *yes* responses observed among the survey responses, the more likely H1a$_{\mathsf{ay}}$ seems relative to H1a$_{\mathsf{0y}}$. That is, seeing a small number y or fewer *yes* responses is more unlikely under the assumption of H1a$_{\mathsf{0y}}$ than under the assumption H1a$_{\mathsf{ay}}$. As this small number y decreases the probability of seeing y or fewer *yes* responses under the assumption of H1a$_{\mathsf{0y}}$ decreases. This probability is called the *p-value*. It is convenient to represent the p-value as $\Pr[Y \le y \mid \mathrm{H1a}_{\mathsf{0y}}]$ where Y is a random variable over the number of observed *yes* responses and y is the actual number of observed *yes* responses. However, the hypothesis H1a$_{\mathsf{0y}}$ is a composite hypothesis asserting that $p_{\bar{\mathsf{p}}\mathsf{fy}} = p$ for some $p \ge 0.5$. Since we would like to disprove the null hypothesis for all these possible values of p, we use the upper bound as the p-value: $\max_{p:0.5 \le p \le 1} \Pr[Y \le y \mid p_{\bar{\mathsf{p}}\mathsf{fy}} = p]$.

If the number y of observed *yes* responses is small enough, then the p-value may become so small that we may confidently reject the null hypothesis H1a$_{\mathsf{0y}}$ in favor of the alternative H1a$_{\mathsf{ay}}$. Since we are looking for a low value for the number of *yes* responses to reject the null hypothesis, we are using a *lower-tail rejection region*.

We must decide how unlikely the observation must be before we are willing to reject the null hypothesis. This choice must balance the risk of incorrectly rejecting a null hypothesis that is actually true (called *Type I Error*) with the risk of incorrectly accepting a null hypothesis that is false (called *Type II Error*). Following convention, we choose the level of Type I Error to be $\alpha = 0.05$. That is, we reject H1a$_{\mathsf{0y}}$ in favor of H1a$_{\mathsf{ay}}$ if the p-value (the probability of seeing observed number of *yes* responses or fewer under the assumption that H1a$_{\mathsf{0y}}$ is true) is less than 0.05.

Hypothesis H1a also produces another prediction: that the number of *no* responses to Question Q1 will be low for Scenario S$_{\bar{\mathsf{p}}\mathsf{f}}$. We can also formalize this prediction as a null hypothesis that we hope to reject:

H1a$_{\mathsf{0n}}$**.** $p_{\bar{\mathsf{p}}\mathsf{fn}} \le 0.5$

The alternative hypothesis we hope to accept in favor of the null hypothesis is

H1a$_{\mathsf{an}}$**.** $p_{\bar{\mathsf{p}}\mathsf{fn}} > 0.5$

In this case, we become more willing to reject the null hypothesis H1a$_{\mathsf{0n}}$ as the number of *no* responses *increases*. This creates an *upper-tail rejection region* in which we are interested in the probability of seeing the observed number of *no* responses or more under the assumption that the null hypothesis is true. As before this quantity is called the p-value. We will again reject if the p-value is less than $\alpha = 0.05$.

We can also formalize the predictions made by Hypothesis H2a. However, as we hope to provide evidence in favor of Hypothesis H2 instead of disproving it, we treat its predictions as alternative hypotheses rather than null hypotheses. For the null hypotheses we use the negations of its predictions and attempt to disprove them. For example, Hypothesis H2a predicts that the number of *yes* responses to Question Q1 for Scenario S$_{\mathsf{p}\bar{\mathsf{f}}}$ will be high. Thus, we attempt to provide evidence for the following alternative hypothesis:

H2a$_{\mathsf{ay}}$**.** $p_{\mathsf{p}\bar{\mathsf{f}}\mathsf{y}} > 0.5$

We do so by showing the probability of seeing the observed number of *yes* responses or more (the p-value using an upper tail rejection region) is unlikely (less than $\alpha = 0.05$) under the assumption that the following null hypothesis is true:

H2a$_{\mathsf{0y}}$**.** $p_{\mathsf{p}\bar{\mathsf{f}}\mathsf{y}} \le 0.5$

We may similarly, formalize other predictions of Hypotheses H1a and H2a as well as the predictions of Hypotheses H1b and H2b. We show each of these formalizations in Table 7 in the next section while presenting the survey results.

These formalizations are, however, only useful if we can compute the value of the p-value under each of them. That is, we must have a formal model of the survey responses that allows us to compute the probability of seeing the responses we observe under the null hypothesis. We now turn to describing such a model.

6.3.2 Binomial Model of the Survey

Each null hypotheses that we test is an assertion about the probability of observing either a *yes* or a *no* response. In the case that the null hypothesis is an assertion about the probability of observing *yes*, we consider the response of *yes* to be a *success* outcome representing successfully observing the response about which the assertion is. We may collapse the responses of *no* and *I don't know* into a single *failure* outcome that represents failure to see *yes*. Likewise, in the case where the null hypothesis is an assertion about the probability of observing *no*, we may treat *no* as a success outcome while treating *yes* and *I don't know*, jointly, as a failure outcome.

By using only two outcomes (success and failure), we may model each survey response as a Bernoulli trial, which models the flipping of a possibly biased coin. The degree of bias determines the probability of success, which models the probability of a respondent answering the question in the manner we are testing.

We model all the responses to a single question of our survey collectively as a series of identical independent Bernoulli trials with each respondent corresponding to one trial. For a given number of trials and probability of success for each trial, the binomial distribution provides the probability of seeing each possible number of successes. (As we do not allow the same individual to take the survey more than once, the assumption of identical independent trials is not completely satisfied since later responses are from a smaller pool of possible respondents that does not include the previous respondents. This factor results in the hypergeometric distribution being a more accurate model. However, since we are drawing our participants from a pool much larger than the sample size, the binomial distribution provides a good approximation.) In particular, the binomial distribution has the cumulative distribution function $F(x; n, p) = \Pr[X \leq x] = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}$ where x is the number of successes, n the number of trails, and p is the probability of success.

Our null hypotheses are assumptions about the value of the success probability p (not to be confused with the idea of a p-value). Using the binomial distribution, we may determine the probability of seeing the responses observed under the null hypothesis. However, we are actually interested in the p-value: the probability of seeing a set of responses at least as extreme as the observed one where the meaning of *extreme* depends upon whether we are using a lower-tail or an upper-tail rejection region.

For example, consider the null hypothesis $\mathrm{H1a}_{\mathsf{0y}}$ that $p_{\bar{\mathsf{p}}\mathsf{fy}} \geq 0.5$. We will reject $\mathrm{H1a}_{\mathsf{0y}}$ using a lower-tail rejection region if its p-value is less than $\alpha = 0.05$ where the p-value is the probability of seeing the observed number of *yes* responses (success outcomes) or fewer. Under our binomial model, the p-value for $\mathrm{H1a}_{\mathsf{0y}}$ is

$$\max_{p:0.5\leq p\leq 1} \Pr[Y{\leq}y \mid p_{\bar{\mathsf{p}}\mathsf{fy}} = p] = \max_{p:0.5\leq p\leq 1} \Pr[Y{\leq}y \mid Y \sim \mathsf{B}(n,p)] = \max_{p:0.5\leq p\leq 1} F(x; n, p)$$

where $Y \sim \mathsf{B}(n, p)$ asserts that Y is a random variable obeying the binomial distribution with a sample size of n and success probability of p.

We may use $F(x; n, 0.5)$ in the place of $\max_{p:0.5\leq p\leq 1} F(x; n, p)$ since we will reject the null hypothesis under the first value if and only if we reject it under the second value. The reason for this equivalence is that $F(x;n,p)$ is an decreasing function in p and is always maximized at $p = 0.5$ when $0.5 \leq p$.

For Hypothesis H2a, we are interested in the null hypothesis that $p_{\mathsf{p\bar{f}y}} \leq 0.5$ using an upper-tail rejection region. For this null hypothesis, the p-value equals $\max_{p:0 \leq p \leq 0.5} 1 - F(x;n,p) = 1 - \min_{p:0 \leq p \leq 0.5} F(x;n,p)$. Similar to the case with the lower-tail rejection region, we may replace $\min_{p:0 \leq p \leq 0.5} F(x;n,p)$ with $F(x;n,0.5)$ since $F(x;n,p)$ is minimized at the largest available value of p, that is, 0.5.

The ability to use $F(x;n,0.5)$ in computing the p-value for both lower-tail and upper-tail rejection regions justifies the convention of writing the null hypotheses using an equality rather than an inequality relation. Whether or not the equality is short hand for a greater-than-equal or a less-than-equal relation may be inferred from the alternative hypothesis paired with the null hypothesis. We will adopt this convention for the remainder of this work.

6.3.3 McNemar's Test

To test hypotheses H1c and H2c, we must compare the responses across scenarios. These responses are not independent since the same respondent produces responses for both scenarios. That is, the responses are produced as *matched-pairs*. McNemar's test provides a method of determining from these matched-pairs the effects of switching between the two scenarios [51]. In particular, McNemar's test examines the number of pairs where the response switches either from *yes* to *no* or from *no* to *yes*. The test approximates the probability of the number of switches being produced by two dependent draws from one distribution. If this probability is small, then one may reject the null hypothesis that switching between the two scenarios had no effect. By rejecting this null hypothesis, one provides evidence for the alternative hypothesis that the difference between the two scenarios affected the responses.

For example, for hypothesis H2c, we compare the responses to Question Q1 across the Scenarios S_{pf} and $\mathrm{S}_{\mathsf{\bar{p}f}}$. We use the null hypothesis that whether or not the case worker employed a plan for treating the patient has no effect on whether or not survey participants think the case worker violated the policy. If we find that a large number of respondents have different responses across the two scenarios, then we would reject the null hypothesis and conclude that case worker's planning does have an effect.

We test Hypothesis H1a' in a manner similar to how we test Hypothesis H1a. However, we use Question Q3 instead of Question Q1. Analogously, we test Hypotheses H1b', H1c', H2a', H2b', and H2c' in a manner similar to Hypotheses H1b, H1c, H2a, H2b, and H2c, respectively, using Question Q3 in place of Question Q1.

6.4 Results

While we only offered to pay the first 200 respondents, we received 207 completed surveys. The extra surveys may have resulted from people misunderstanding the instructions and not collecting payment.

Of these completed surveys, we excluded 20 respondents for missing two or more of the objective questions. All of the statistics shown in this section are calculated from the remaining 187 respondents. Appendix C shows the same statistics for all 207 respondents. Including the 20 excluded respondents does not change the significance of any of our hypothesis tests.

Table 6 shows the distributions of responses for each question. Informally examining the tables shows that the vast majority of the respondents conform to Hypothesis H2. For example, 177 (95%) of the respondents answered Question Q1 for Scenario $\mathrm{S}_{\mathsf{p}\bar{\mathsf{f}}}$ with the answer of *yes* as predicted by Hypothesis H2, whereas only eight (4%) answered with *no* as predicted by Hypothesis H1. However, the difference is less pronounced for Scenario $\mathrm{S}_{\bar{\mathsf{p}}\mathsf{f}}$ where 133 (71%) match Hypothesis H2's prediction of *no* and 45 (24%) matches H1's prediction of *yes*. Interestingly, 31 (17%) answered *yes* for Scenario $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ despite both hypotheses predicting *no*.

Table 7 shows the hypothesis tests we conducted using the binomial model. The top half of the table shows tests intended to disprove Hypothesis H1 while the bottom half shows tests attempting to confirm Hypothesis H2. Every test in favor of Hypothesis H2 obtains statistical significance. Eight of the 16 tests against Hypothesis H1 obtain statistical significance. The eight that do not obtain significance are the cases where the two hypotheses agree. In every case where the two disagree, both the test confirming Hypothesis H2 and the one against Hypothesis H1 obtains significance.

Since the results of the hypothesis testing where so strongly in favor of Hypothesis H2 using the probability of 0.5 as the null hypothesis, we decided to calculate the most extreme probabilities that still obtains significance. For testing that a probability is less than a value (lower tail rejection region), the most extreme value is the minimal value, whereas it is the maximum value for testing that a probability is greater than a value (upper tail rejection region). Table 8 shows these probabilities conservatively calculated up to 0.01 away from the true extreme probability. For example, the bottom row shows $p'_{4\mathsf{y}}$ is less than 0.26 with statistical significance but not less than 0.25 with statistical significance. (This does not imply that $p'_{4\mathsf{y}} > 0.25$ with statistical significance.) As these probabilities are more extreme for Hypotheses H2a and H2a' than Hypotheses H2b and H2b', H2a and H2a' appear to be more accurate. However, as we added these statistics to the analysis after having conducted the survey, they may suffer from confirmation bias.

Table 9 shows the results of using McNemar's Test to compare the distribution of responses to one question across two scenarios. For example, the last row compares the distribution producing responses to Question Q3 for Scenario $\mathrm{S}_{\mathsf{p}\bar{\mathsf{f}}}$ to that producing responses for Scenario $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$. McNemar's Test shows that the differences in the observed responses are statistically significant. This result indicates that the two distributions differ as predicted by Hypothesis H2c'. On the other hand, the fourth line of Table 9 shows that the responses for Question Q3 do not differ significantly across Scenarios $\mathrm{S}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$. This result differs from Hypothesis H1, which predicts that people would answer the question differently across the two scenarios. McNemar's Test validates all four predictions of Hypothesis H2. It validates one of the predictions of Hypothesis H1. The statistic could not be computed in one case as the data was too sparse for the calculation.

We were surprised to see the degree of difference between how people answered Questions Q1 and Q3. For example, for Scenario $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$, 79% of respondents answered Question Q2 with *no* whereas only 74% answered Q3 with *no* despite our belief that both questions should be answered identically (see Table 6). To test whether these differences are statistically significant, we used McNemar's test to compare the responses to these two questions within a single scenario. Table 10 shows the results. None of the tests showed a statistically significant difference in how the questions were answered, but two of the tests failed to produce a numeric p-value. (Appendix B shows the matched pairs used by McNemar's test for Tables 9 and 10.)

Scenario	Yes	I don't know	No
S_{pf}	182 (97%)	2 (01%)	3 (02%)
$S_{p\bar{f}}$	177 (95%)	2 (01%)	8 (04%)
$S_{\bar{p}f}$	45 (24%)	9 (05%)	133 (71%)
$S_{\bar{p}\bar{f}}$	31 (17%)	9 (05%)	147 (79%)

Q1: Was the policy obeyed?

Scenario	Yes	I don't know	No
S_{pf}	185 (99%)	2 (01%)	0 (00%)
$S_{p\bar{f}}$	183 (98%)	1 (01%)	3 (02%)
$S_{\bar{p}f}$	43 (23%)	6 (03%)	138 (74%)
$S_{\bar{p}\bar{f}}$	38 (20%)	10 (05%)	139 (74%)

Q3: Was the action for the purpose?

Scenario	Yes	I don't know	No
S_{pf}	186 (99%)	0 (00%)	1 (01%)
$S_{p\bar{f}}$	184 (98%)	1 (01%)	2 (01%)
$S_{\bar{p}f}$	12 (06%)	1 (01%)	174 (93%)
$S_{\bar{p}\bar{f}}$	6 (03%)	0 (00%)	181 (97%)

Q4: Was the goal treatment?

Scenario	Yes	I don't know	No
S_{pf}	187 (100%)	0 (00%)	0 (00%)
$S_{p\bar{f}}$	2 (01%)	0 (00%)	185 (99%)
$S_{\bar{p}f}$	179 (96%)	0 (00%)	8 (04%)
$S_{\bar{p}\bar{f}}$	3 (02%)	0 (00%)	184 (98%)

Q5: Was the treatment successful?

Table 6: Survey Responses. In Scenario S_{pf}, the case worker's goal was treatment and the treatment was successful; in $S_{p\bar{f}}$, the goal was treatment and it failed; in $S_{\bar{p}f}$, the goal was cost reduction and the treatment succeeded; and in $S_{\bar{p}\bar{f}}$, the goal was cost reduction and the treatment failed.

Testing	Alternative Hypothesis	Null Hypothesis	p-Value	Significant?
Against H1a	$p_{\mathsf{pfy}} < 0.5$	$p_{\mathsf{pfy}} = 0.5$	1	No
Against H1a	$p_{\mathsf{pfn}} > 0.5$	$p_{\mathsf{pfn}} = 0.5$	1	No
Against H1a	$p_{\mathsf{\bar{p}fy}} < 0.5$	$p_{\mathsf{\bar{p}fy}} = 0.5$	3.28889e-013	Yes
Against H1a	$p_{\mathsf{\bar{p}fn}} > 0.5$	$p_{\mathsf{\bar{p}fn}} = 0.5$	3.527326e-009	Yes
Against H1a'	$p'_{\mathsf{pfy}} < 0.5$	$p'_{\mathsf{pfy}} = 0.5$	1	No
Against H1a'	$p'_{\mathsf{pfn}} > 0.5$	$p'_{\mathsf{pfn}} = 0.5$	1	No
Against H1a'	$p'_{\mathsf{\bar{p}fy}} < 0.5$	$p'_{\mathsf{\bar{p}fy}} = 0.5$	3.08316e-014	Yes
Against H1a'	$p'_{\mathsf{\bar{p}fn}} > 0.5$	$p'_{\mathsf{\bar{p}fn}} = 0.5$	2.662347e-011	Yes
Against H1b	$p_{\mathsf{p\bar{f}n}} < 0.5$	$p_{\mathsf{p\bar{f}n}} = 0.5$	1.699463e-043	Yes
Against H1b	$p_{\mathsf{p\bar{f}y}} > 0.5$	$p_{\mathsf{p\bar{f}y}} = 0.5$	6.090736e-041	Yes
Against H1b	$p_{\mathsf{\bar{p}\bar{f}n}} < 0.5$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	1	No
Against H1b	$p_{\mathsf{\bar{p}\bar{f}y}} > 0.5$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1	No
Against H1b'	$p'_{\mathsf{p\bar{f}n}} < 0.5$	$p'_{\mathsf{p\bar{f}n}} = 0.5$	5.556827e-051	Yes
Against H1b'	$p'_{\mathsf{p\bar{f}y}} > 0.5$	$p'_{\mathsf{p\bar{f}y}} = 0.5$	2.570485e-049	Yes
Against H1b'	$p'_{\mathsf{\bar{p}\bar{f}n}} < 0.5$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	1	No
Against H1b'	$p'_{\mathsf{\bar{p}\bar{f}y}} > 0.5$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1	No
For H2a	$p_{\mathsf{pfy}} > 0.5$	$p_{\mathsf{pfy}} = 0.5$	9.461645e-048	Yes
For H2a	$p_{\mathsf{pfn}} < 0.5$	$p_{\mathsf{pfn}} = 0.5$	5.556827e-051	Yes
For H2a	$p_{\mathsf{p\bar{f}y}} > 0.5$	$p_{\mathsf{p\bar{f}y}} = 0.5$	6.090736e-041	Yes
For H2a	$p_{\mathsf{p\bar{f}n}} < 0.5$	$p_{\mathsf{p\bar{f}n}} = 0.5$	1.699463e-043	Yes
For H2a'	$p'_{\mathsf{pfy}} > 0.5$	$p'_{\mathsf{pfy}} = 0.5$	8.961588e-053	Yes
For H2a'	$p'_{\mathsf{pfn}} < 0.5$	$p'_{\mathsf{pfn}} = 0.5$	5.097894e-057	Yes
For H2a'	$p'_{\mathsf{p\bar{f}y}} > 0.5$	$p'_{\mathsf{p\bar{f}y}} = 0.5$	2.570485e-049	Yes
For H2a'	$p'_{\mathsf{p\bar{f}n}} < 0.5$	$p'_{\mathsf{p\bar{f}n}} = 0.5$	5.556827e-051	Yes
For H2b	$p_{\mathsf{\bar{p}fn}} > 0.5$	$p_{\mathsf{\bar{p}fn}} = 0.5$	3.527326e-009	Yes
For H2b	$p_{\mathsf{\bar{p}fy}} < 0.5$	$p_{\mathsf{\bar{p}fy}} = 0.5$	3.28889e-013	Yes
For H2b	$p_{\mathsf{\bar{p}\bar{f}n}} > 0.5$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	7.078408e-016	Yes
For H2b	$p_{\mathsf{\bar{p}\bar{f}y}} < 0.5$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1.479279e-021	Yes
For H2b'	$p'_{\mathsf{\bar{p}fn}} > 0.5$	$p'_{\mathsf{\bar{p}fn}} = 0.5$	2.662347e-011	Yes
For H2b'	$p'_{\mathsf{\bar{p}fy}} < 0.5$	$p'_{\mathsf{\bar{p}fy}} = 0.5$	3.08316e-014	Yes
For H2b'	$p'_{\mathsf{\bar{p}\bar{f}n}} > 0.5$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	9.252051e-012	Yes
For H2b'	$p'_{\mathsf{\bar{p}\bar{f}y}} < 0.5$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	4.896385e-017	Yes

Table 7: Binomial Hypothesis Tests

Testing	Alternative Hypothesis	Null Hypothesis
Proving H2a	$p_{\mathsf{pfy}} > 0.94$	$p_{\mathsf{pfy}} = 0.94$
Proving H2a	$p_{\mathsf{pfn}} < 0.05$	$p_{\mathsf{pfn}} = 0.05$
Proving H2a	$p_{\mathsf{p\bar{f}y}} > 0.91$	$p_{\mathsf{p\bar{f}y}} = 0.91$
Proving H2a	$p_{\mathsf{p\bar{f}n}} < 0.08$	$p_{\mathsf{p\bar{f}n}} = 0.08$
Proving H2a'	$p'_{\mathsf{pfy}} > 0.96$	$p'_{\mathsf{pfy}} = 0.96$
Proving H2a'	$p'_{\mathsf{pfn}} < 0.02$	$p'_{\mathsf{pfn}} = 0.02$
Proving H2a'	$p'_{\mathsf{p\bar{f}y}} > 0.95$	$p'_{\mathsf{p\bar{f}y}} = 0.95$
Proving H2a'	$p'_{\mathsf{p\bar{f}n}} < 0.05$	$p'_{\mathsf{p\bar{f}n}} = 0.05$
Proving H2b	$p_{\mathsf{\bar{p}fn}} > 0.65$	$p_{\mathsf{\bar{p}fn}} = 0.65$
Proving H2b	$p_{\mathsf{\bar{p}fy}} < 0.3$	$p_{\mathsf{\bar{p}fy}} = 0.3$
Proving H2b	$p_{\mathsf{\bar{p}\bar{f}n}} > 0.73$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.73$
Proving H2b	$p_{\mathsf{\bar{p}\bar{f}y}} < 0.22$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.22$
Proving H2b'	$p'_{\mathsf{\bar{p}fn}} > 0.67$	$p'_{\mathsf{\bar{p}fn}} = 0.67$
Proving H2b'	$p'_{\mathsf{\bar{p}fy}} < 0.29$	$p'_{\mathsf{\bar{p}fy}} = 0.29$
Proving H2b'	$p'_{\mathsf{\bar{p}\bar{f}n}} > 0.68$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.68$
Proving H2b'	$p'_{\mathsf{\bar{p}\bar{f}y}} < 0.26$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.26$

Table 8: Extreme Binomial Hypothesis Tests. This table shows the hypothesis test using the most extreme probability for which statistical significance is still achieved and is accurate up to two places after the decimal point.

Testing	Question	Scenarios	p-Value	Significant?
For H1c	Q1	S_{pf} vs. $S_{\mathsf{p\bar{f}}}$	NaN	No
For H1c	Q1	$S_{\mathsf{\bar{p}f}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	0.02674664	Yes
For H1c'	Q3	S_{pf} vs. $S_{\mathsf{p\bar{f}}}$	0.3916252	No
For H1c'	Q3	$S_{\mathsf{\bar{p}f}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	0.3951831	No
For H2c	Q1	S_{pf} vs. $S_{\mathsf{\bar{p}f}}$	1.020173e-029	Yes
For H2c	Q1	$S_{\mathsf{p\bar{f}}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	3.112267e-031	Yes
For H2c'	Q3	S_{pf} vs. $S_{\mathsf{\bar{p}f}}$	5.186851e-031	Yes
For H2c'	Q3	$S_{\mathsf{p\bar{f}}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	8.40055e-031	Yes

Table 9: McNemar's Tests Across Scenarios

Scenario	Questions	p-Value	Significant?
S_{pf}	Q1 vs. Q3	NaN	No
$S_{\mathsf{p\bar{f}}}$	Q1 vs. Q3	NaN	No
$S_{\mathsf{\bar{p}f}}$	Q1 vs. Q3	0.3843414	No
$S_{\mathsf{\bar{p}\bar{f}}}$	Q1 vs. Q3	0.2239329	No

Table 10: McNemar's Tests Across Questions

6.5 Limitations of Study

Various factors affect the validity of our conclusions. We discuss each of them below.

By mentioning whether or not the auditee is performing the action as part of a plan, it forces the participant to consider the relationship between purposes and plans. It is possible that participants not primed to think about planning would substantiate H1.

The use of Mechanical Turk raises questions about how representative our population sample is. Ross et al. look at the demographics of Mechanical Turk workers and find that among U.S. workers, a disproportionate number are female [60]. However, Berinsky, Huber, and Lenz find that Mechanical Turk studies are as representative, if not more representative, than convenience samples commonly used in research [12]. While we attempted to limit our sample to adults in the United States, Mechanical Turk's ability to verify the qualification criteria is limited. Even given a representative pool of Mechanical Turk workers for our sampling frame, our sample may be biased as the participants selected to take our survey rather than us having randomly selected them from the pool.

The use of paid but unmonitored participants, also raises concerns that participants might provide arbitrary answers to speed through the questionnaire. Kittur, Chi, and Suh present experimental results of using Mechanical Turk for user studies [44]. They conclude that Mechanical Turk can be useful if one eliminates such spurious submissions by including questions with known answers and rejecting participants who fail to correctly answer these questions. We follow this protocol by using Questions Q4 and Q5 to force the participant to read the scenarios and by notifying survey participants that we may withhold payment if they answer arbitrarily. Answering the remaining questions (Q1, Q2, and Q3) becomes fairly easy after having correctly answered Questions Q4 and Q5. By making the additional work required for meaningful participation small, we hope to have reduced arbitrary responses. However, by threatening to withhold payment, we may have increased the *demand effect*, the tendency of participants to provide the answers they believe the surveyor would like to observe as opposed to their honest opinions (see, e.g., [55]).

Some respondents might answer later questions in a manner consistent with their answers to earlier questions despite having differing opinions. This bias could arise since some of the differences between questions may appear trivial, especially since we made each question similar to the others to reduce confounding factors. As no scenario has the same answers to both Questions Q3 and Q4 together as any other scenario, we hope to have reduced this bias.

Nonattitudes occur when a participant arbitrarily selects a response since they do not have an opinion on a question. To reduce the effect of nonattitudes, we included the option of a *I don't know* response.

We do not claim that the questionnaire tests all relevant factors (i.e., we do not claim high content validity). Indeed, we did not test some factors that we suspect may affect respondents such as whether the policy is perceived as good or bad.

Another concern is that respondents may change their opinions over time. We did not perform a follow-up study to determine how reliable our survey is over time.

It is also possible that our survey questions are not understood by the respondents in a manner consistent with testing the meaning of *purpose*. The various forms of validity discussed below attempt to determine whether our survey actually measured the concepts in which we are interested.

We believe that our survey has face validity. That is, we believe that our questions are, on their face, well worded for testing our hypotheses.

Including both Questions Q1 and Q3 not only allowed us to compare the truth of Hypothesis H1a to H1a' (and likewise with the other unprimed-primed pairs of hypotheses), but also to see the effects of the changing the wording of the questions. As the respondents typically answered these two questions in the same manner, we believe that our results are not overly influenced by the wording of the questions and pertain to the underlying concepts. That is, we believe our survey has convergent validity. However, that some respondents varied their responses across Questions Q1 and Q3 within a single scenario deserves further investigation.

As we know of no previous empirical research addressing the issues tested by our study, we cannot compare our results to those already proved about the meaning of *purpose*. Thus, we cannot that argue that our survey has construct validity by showing that it agrees with previous results.

A survey respondent may confuse the concepts we are testing with related ones reducing the divergent validity of our survey. For example, rather than actually answer Question Q1, they may instead provide the answer to the following question: "Was the case worker's action consistent with someone seeking treatment for the patient?" Such confusion may explain some of the unexpected variation in responses between Questions Q1 and Q3.

The ultimate goal of our work is to determine how people think policies involving the concept of *purpose* should be enforced. Our survey is detached from any actual enforcement. Respondents might behave differently than their responses suggest given the task of actually enforcing a policy. They may also differ from their responses in their feelings if they were actually subject to such a policy. Our survey is most similar to the respondent acting as a neutral third-party or judge in a dispute over the meaning of a policy. However, even in such a role, the respondent's behavior may differ from that suggested from his responses. Ideally, our survey will predict with a high degree of accuracy how the respondents would behave in each of these three roles (policy enforcer, policy subject, or neutral third-party) establishing that our survey actually corresponds to the behavior we wish to study (i.e., has criterion validity). However, we have not established this form of validity.

6.6 Discussion

The results shown above provide evidence in favor of defining an action to be for a purpose if and only if an agent performed the action as part of a plan for furthering that purpose (Hypothesis H2). The binomial tests provide strong evidence against defining an action to be for a purpose if and only if that action furthered the purpose (Hypothesis H1). McNemar's test provides some support for Hypothesis H1. Indeed, informally examining the response distributions (Table 6), it appears Hypothesis H1 does accurately model a small minority of respondents. However, Hypothesis H2 appears to accurately model a much larger number of respondents. For these reasons, we conclude that Hypothesis H2 provides a superior model to that of Hypothesis H1.

Nevertheless, the relative strength of Hypothesis H2a compared to Hypothesis H2b suggests that some people feel that an action being for a purpose is sufficient but not necessary for an action to be for a purpose. Examining free-form responses to Question Q2 suggests that some people feel that the action of sharing a record is for the purpose of treatment since it is the same action that would be taken had the case worker been planning for treatment. This suggests a third class of hypotheses:

H3 The auditee obeys the (purpose) restriction if and only if the auditee performed an action that a hypothetical agent would take had it planned for the purpose.

H3' An action is for a purpose if and only if that action is the action a hypothetical agent would take had it planned for the purpose.

These hypotheses place strictly weaker restrictions on the auditee's behavior consistent with the idea that H2 is sufficient but not necessary. Interestingly, they match the approximations our algorithm makes in attempting to enforce Hypothesis H2. Unfortunately, by not mentioning whether the case worker's choice to forward the record in Scenarios $\mathrm{S}_{\bar{\mathsf{p}}\mathsf{f}}$ and $\mathrm{S}_{\bar{\mathsf{p}}\bar{\mathsf{f}}}$ is consistent with the actions of a hypothetical agent planning for treatment, we cannot test these hypotheses using the conducted survey.

7 Applying our Formalism to Prior Methods

Past methods of enforcing purpose restrictions have not provided a means of assigning purposes to sequences of actions. Rather, they presume that the auditor (or someone else) already has a method of determining which behaviors are for a purpose. In essence, these methods presuppose that the auditor already has the set of allowed behaviors $\mathsf{nbehv}(r^p)$ for the purpose p that he is enforcing. These methods differ in their intensional representations of the set $\mathsf{nbehv}(r^p)$. Thus, some may represent a given set exactly while others may only be able to approximate it. These differences mainly arise from the different mechanisms they use to ensure that the auditee only exhibits behaviors from $\mathsf{nbehv}(r^p)$. We use our semantics to study how reasonable these approximations are.

Byun et al. use role-based access control [63] to present a methodology for organizing privacy policies and their enforcement [20, 19, 53]. They associate purposes with sensitive resources and with roles, and their methodology only grants the user access to the resource when the purpose of the user's role matches the resource's purpose. The methodology does not, however, explain how to determine which purposes to associate with which roles. Furthermore, a user in a role can perform actions that do not fit the purposes associated with his role allowing him to use the resource for a purpose other than the intended one. Thus, their method is only capable of enforcing policies when there exists some subset A of the set of actions $\mathcal{A}$ such that $\mathsf{nbehv}(r^p)$ is equal to the set of all interleavings of A with $\mathcal{S}$ of finite but unbounded length (i.e., $\mathsf{nbehv}(r^p) = (\mathcal{S} \times A)^*$). The subset A corresponds to those actions that use a resource with the same purpose as the auditee's role. Despite these limitations, their method can implement the run-time enforcement used at some organizations, such as a hospital that allows physicians access to any record to avoid denying access in time-critical emergencies. However, it does not allow the fine-grain distinctions used during post-hoc auditing done at some hospitals to ensure that physicians do not abuse their privileges. *Group-centric access control* has similar advantages and limitations [46].

Al-Fedaghi uses the work of Byun et al. as a starting point but concludes that rather than associating purposes with roles, one should associate purposes with sequences of actions [3]. Influenced by Al-Fedaghi, Jafari et al. adopt a similar position calling these sequences *workflows* [38]. The set of workflows allowed for a purpose p corresponds to $\mathsf{nbehv}(r^p)$. They do not provide a formal method of determining which workflows belong in the allowed set leaving this determination to the intuition of the auditor. They do not consider probabilistic transitions and the intuition they supply suggests that they would only include workflows that successfully achieve or improve the purpose. Thus, our approach appears more lenient by including some behaviors that fail to improve the purpose. As shown in Chapter 6, this leniency is key to capturing the semantics of purpose restrictions. An auditor could encode a workflow in the state of the environment to get results similar to Al-Fedaghi's or Jafari et al.'s results while using *Contextual Role-Based Access Control* [52] or *Situation-Based Access Control* [56].

Others have adopted a hybrid approach allowing the roles of an auditee to change based on the state of the system [57, 29]. These dynamic roles act as a level of indirection assigning an auditee to a state. This indirection effectively allow role-based access control to simulate the workflow methods to be just as expressive.

Agrawal et al. propose a methodology called *Hippocratic databases* for protecting the privacy of subjects of a database [2]. They propose to use a *query intrusion model* to enforce privacy polices governing purposes. Given a request for access and the purpose for which the requester claims the request is made, the query intrusion model compares the request to previous requests with the same purpose using an approach similar to intrusion detection. If the request is sufficiently different from previous ones, it is flagged as a possible violation. While the method may be practical, it lacks soundness and completeness. Furthermore, by not being semantically motivated, it provides no insight into the semantics of purpose. To avoid false positives, the set of allowed behaviors $\mathsf{nbehv}(r^p)$ would have to be small or have a pattern that the query intrusion model could recognize.

Jif is a language extension to Java designed to enforce requirements on the flows of information in a program [22]. Hayati and Abadi explain how to reduce purpose restrictions to information flow properties that Jif can enforce [37]. Their method requires that inputs are labeled with the purposes for which the policy allows the program to use them and that each unit of code be labeled with the purposes for which that code operates. If information can flow from an input statement labeled with one purpose to code labeled for a different purpose, their method produces a compile-time type error. (For simplicity, we ignore their use of sub-typing to model sub-purposes.) In essence, their method enforces the rule *if information i flows to code c, then i and c must be labeled with the same purpose*. The interesting case is when the code c uses the information i to perform some observable action $a_{c,i}$, such as producing output. Under our semantics, we treat the program as the auditee and view the policy as limiting these actions. By directly labeling code, their method does not consider the contexts in which these actions occur. Rather the action $a_{c,i}$ is aways either allowed or not based on the purpose labels of c and i. By not considering context, their method is subject to the same limitations as the method of Byun et al. with the subset A being equal to the set of all actions $a_{c,i}$ such that c and i have the same label. However, using more advanced type systems (e.g., typestate [67]), they might be able extend their method to consider the context in which code is executed and increase the method's expressiveness.

8 Related Works

8.1 Related Problems in Policy Enforcement

We have already covered the most closely related work in Section 7. Below we discuss work on related problems in computer science.

Minimal Disclosure. The works most similar to ours in approach have been on *minimal disclosure*, which requires that the amount of information used in granting a request for access should be as little as possible while still achieving the purpose behind the request. Massacci, Mylopoulos, and Zannone define minimal disclosure for Hippocratic databases [50]. Barth, Mitchell, Datta, and Sundaram study minimal disclosure in the context of workflows [10]. They model a workflow as meeting a utility goal if it satisfies a temporal logic formula. Minimizing the amount of information disclosed is similar to an agent maximizing his reward and thereby not performing actions that have costs but no benefits. However, we consider several factors that these works do not, including quantitative purposes that are satisfied to varying degrees and probabilistic behavior resulting in actions being for a purpose despite the purpose not being achieved, which is necessary to capture the semantics of purpose restrictions (Section 6).

Expressing Privacy Policies with Purpose. Work on understanding the components of privacy policies has shown that *purpose* is a common component of privacy rules (see, e.g., [15, 16]).

Some languages for specifying access-control policies allow the purpose of an action to partially determine if access is granted. For example, EPAL is a language in which privacy policies are expressed by listing all the conditions under which a system should grant a request for access to sensitive resources [58]. These conditions may depend upon four factors: the identity of the requester for access, the resource requested, the action the requester would like to perform on the resource, and the purpose for which the requester would like to perform the action. However, EPAL lacks a formal semantics that describes when an action is for a purpose and treats purposes as syntactic labels. Rather, it depends on the system making use of the language to determine what actions are for what purposes and provides no formal guidance as to how the system should make this determination.

The Platform for Privacy Preferences (P3P) offers a language for specifying the privacy policies of websites [25]. These policies must state the purposes for which the website collects information. The policy may either reference one of the predefined purposes that the language offers or provide a custom purpose. The specification of the language provides a description of each of the predefined purposes in natural language [24]. The policy author must provide such a description for any custom purposes he uses. We hope our work will provide a method of formalizing when information use meets the requirements of these descriptions.

SPARCLE is a system for authoring and examining privacy policies [17, 18]. The system consumes policies written in a restricted form of natural language and parses them into standard components. The system then allows the user to examine the policy by focusing on different components, edit the policy, and translate the policy into machine readable formats (e.g., EPAL). One of the standard components SPARCLE considers is purpose. While SPARCLE is capable of identifying restrictions on purpose in a policy, it does not assign a semantics to these restrictions.

Hanson et al. provide an algebra for tracking the permissible uses of data as it is transferred from system to system and is combined with other information [36]. However, this work is not concerned with the meaning of *purpose* or *for*.

8.2 Works from Philosophy and Psychology

Philosophy concerns defining the meaning of words. Philosophers typically proceed by iteratively refining a definition to match their intuitions about each new example of the word's use. The experimental methods of psychology (defined broadly to include linguistics and cognitive science) have given raise to experimental philosophy. This hybrid methodology studies the meaning of the words by looking at the most common view of a population rather than the intuitions of experts. Our work uses intuition until Section 6, which presents a survey. Both philosophy and psychology apply their methodologies to understanding the nature of human planning. We discuss these efforts below.

Philosophical Foundations. Philosophical works usually use *purpose* in the sense of *the purpose of life*, which differs from the sense in which privacy policies use the word. These works use *desire*, *motivation*, and *intention* to refer to notions similar to the sense of purpose we are interested in. Generally, *desires* and *motivations* refer to the reasons that cause an agent to act. That is, in this sense, motivations are purposes in our formalism. While philosophers disagree more over the meaning of *intention*, it typically refers to the modifications the agent hopes to make to the state of the world. That is, intentions are actions the agent plans to take in our formalism. For example, then prevention of hunger motivates the intention to go grocery shopping.

The modern philosophical work in this area starts with Anscombe who argues that the intention of an action is the answer offered to the question Why did you perform that action? [5].

Taylor provides a detailed explanation of the importance of planning to the meaning of *purpose*, but does not provide any formalism [68]. Taylor concludes that one must distinguish the purpose of actions from their effects: the effects are the actual results of the actions whereas the purpose is merely the desired effects (page 216). Our model formalizes this distinction by allowing an action to be for a purpose despite that purpose not being achieved.

Bratman builds on Anscombe's work by emphasizing the importance of agent planning in determining intentions to create the Belief-Desire-Intention (BDI) model [14]. In Bratman's work, an intention is an action an agent plans to take where the plan is formed while attempting to maximize the satisfaction of the agent's desires. To some extent Section 3 may be viewed a formalization of a simplification of Bratman's view. (The plans of Bratman are more complex than our strategies to account for the limited reasoning abilities of humans.)

Using Bratman's work as a starting point, Cohen and Levesque present a logical formalization of when an agent intends to perform an action or intents to bring about a state of affairs [23]. Roughly speaking, under their formalism, an agent intends to satisfy a predict p over states if and only if the agent has knowingly performing a sequence of actions that makes p true as a goal that it believes it can achieve and will continue to attempt to make p true until it believes it is impossible to do so. These predicates are related to binary purpose scores, and our formalism produces strategies that roughly correspond to the intentional actions of Cohen and Levesque. However, our formalism also handles quantitative purposes. Cohen and Levesque comment on the existence of such purposes and propose to model them as a series of intentions, but do not provide a formalism to do so.

Intentions also affect planning and will become important as we search for more accurate models of human planning. Roy use logics and game theory to formalize how intentions can affect an agent's planning [61]. He uses his formalism to study when an agent's plan is rational given the agent's intentions. Given the auditee's intentions, we could replace our MDP formalism of planning with Roy's intention-driven formalism.

Causality. Our treatment of *for* in Section 3 is motivated by the counterfactual definition of *causality*. This definition requires that for an action to cause an effect that both the effect actually occurs and that the effect might not have occurred if the action did not occur. For example, Mackie defines a *cause* to be insufficient and non-redundant parts of unnecessary but sufficient causes (INUS conditions) for an effect [48]. Mackie models causes and effects as facts. Working with sets of causes, this means that a fact c is a cause of an effect e if there exists a set C such that C is sufficient to entail e (sufficiency) and no subset of C is sufficient to entail e (non-redundancy).

We borrow the notion of *non-redundancy* from Mackie's definition of causality. Roughly speaking, we replace the causes with actions and the effect with a purpose.

Experimental Philosophy. Experimental philosophy has found some inconsistencies in how people tend to use the word "intent" called the *Knobe effect* [45]. When it comes to benefits for purposes that are good, people tend to only say that the actor intended for the benefits if the actor selected his action taking the purpose into consideration, which agrees with our model. However, when it comes to bad purposes, people tend to say that the actor intended for the (bad) benefits even if the actor did not select his action with the goal of achieving the bad purpose in mind, which disagrees with our model. (See [31] for a survey.)

Human Planning. Psychological studies have produced models of human thought (see, e.g., [4]). However, these are too low-level and incomplete for our needs [27]. The GOMS (Goals, Operators, Methods, and Selection rules) formalism provides a higher level model, but is limited to selecting behavior using simple planning approaches [21, 39]. Simon's approach of *bounded rationality* [66] and related heuristic-based approaches [34] model more complex planning, but with less precise predictions.

8.3 Related Algorithms

Plan Recognition. Attempting to infer the plan that an agent has while performing an action is *plan recognition* [64]. Plan recognition may predict the future actions of agents allowing systems to anticipate them. Often, plan recognition algorithms model how "low-level" actions contribute to achieving a "top-level" action that is done for its own sake (see, e.g., [42]). These top-level actions are similar to purposes. However, out auditing algorithm checks whether a sequence of actions are consistent with given a purpose rather than attempting to predict the most likely purpose motivating the actions.

The work most closely related to ours is that of Baker, Saxe, and Tenenbaum [7, 8]. They use an MDP model similar to ours to predict the most likely explanation for a sequence of actions. Ramírez and Geffner extend this work to partially observable MDPs (POMDPs) for modeling an agent that cannot directly observe the state it is in [59]. Rather than having a reward function, under these models, the agent attempts to reduce the *costs* of reaching a *goal* state. For each possible goal state, their algorithms use the degree to which the agent's actions minimizes the costs of reaching the goal state to assign a probability to that goal state being the one pursued by the agent. Our reward functions are similar to the negation of their cost functions, but these works predict which goal state the agent is pursuing rather than which cost function it is using. They do not consider non-redundancy. Our algorithm for auditing is similar to their algorithms. However, to maintain soundness, our algorithm accounts for the error of approximate MDP solving. Furthermore, their algorithms may assign a non-zero probability to a goal state even if the agent's actions are inconsistent with pursuing that goal under our strict definition.

Also related is the work of Mao and Gratch [49]. While it differs from our work in the same ways as the work of Baker et al., it also differs in that rewards track how much the agent wants to achieve the goal rather than the degree of satisfaction of the goal.

Most work on plan recognition assumes that the agent is not attempting to mislead the plan recognizer since they are designed to aid cooperation with the agent. Our work is related to work on *adversarial* plan detection [6].

Particularly related is the work of Geib and Goldman, who use adversarial plan recognition to aid intrusion detection [33]. Similar to standard works, they model plans as a graph that represents a space of possible plans. Nodes of the graph represent actions and directed edges represent the order in which the adversary must perform the actions. Intrusions are paths in the graph from an initial node to a goal node. However, unlike most work on plan recognition, owing to the hostile nature of the actor, they do not assume that all relevant actions are observable. Thus, rather than simply comparing the observed actions to paths in the graph to determine possible plans, their recognition algorithm also considers unobserved actions consistent with the state of the system that the adversary might have performed.

Relatedly, Cuppens, Autrel, Miège, and Benferhat attempt to recognize malicious intentions for intrusion detection [26]. They model attacks as consisting of multiple actions each with pre-conditions and post-conditions. An adversary attempts to perform a *malicious* action by first performing all the *suspicious* actions needed to enable the pre-condition of the malicious action. Their approach is to observe these suspicious actions and predict from their model what other actions the adversary might have performed or will be performing. In particular, they try to predict which (if any) malicious action the adversary is attempting to perform using a shortest path heuristic. The distinction between suspicious and malicious actions does not apply our work since we consider purposes, not actions, to be malicious. Indeed, in our setting many actions, such as looking up a medical record, could be either acceptable or malicious depending upon the context.

The models of planning used in both of these works differ from ours in two ways. First, we model purposes quantitatively instead of qualitatively. Second, our work considers probabilistic effects of the environment that might cause the agent to fail to achieve its plan.

Automated Planning. *Decision-theoretic planning* is planning to optimize some criteria, such as a purpose. (Blythe provides a survey [13].) Optimizing MDPs or POMDPs to create plans are just two instances of decision-theoretic planning. Other instances may be more accurate, convenient, or general models of human planning.

For example, due to uncertainty the auditor may have about the model used by the auditee, we are interested in environment models that are like MDPs but without fixed probabilities assigned to transitions. Discrete-time Markov chains without fixed probabilities are known as *interval-valued discrete-time Markov chains* (IDTMCs). The form of IDTMC most similar to our model is the Uncertain Markov Chain (UMC) model [40]. We hope the algorithm of Sen et al. [65] for a model checking problem related to UMCs may shed light on how to generalize our algorithm found in Section 5.

9 Summary and Future Work

We use planning to create the first formal semantics for determining when a sequence of actions is for a purpose. In particular, our formalism uses models similar to MDPs for planning, which allows us to automate auditing for both exclusivity and prohibitive purpose restrictions. We have provided an auditing algorithm and implementation based on our formalism. We have illustrated the use of our algorithm to create operating procedures.

We validate that our method based on planning accurately captures the meaning of purpose restrictions with intuitive examples (Sections 3.3, 4.2, 4.3, and 5.3) and an empirical study of how people understand the word "purpose" in the context of privacy policy enforcement.

We use our formalism to explain and compare previous methods of policy enforcement in terms of a formal semantics. Our formalism highlights that an action can be for a purpose even if that purpose is never achieved, a point present in philosophical work on the subject (e.g., [68]), but whose ramifications on policy enforcement had been unexplored. Fundamentally, our work shows the difficulties of enforcement due to issues such as the tenable deniability of ulterior motives (Sections 4.2 and 4.3).

However, we recognize the limitations of our formalism. While MDPs are useful for automated planning, they are not specialized for modeling planning by humans. While this concern does not apply to creating operating procedures, it holds human auditees to unrealistically high standards leading to the search for models reflecting the bounded abilities of humans to plan. However, "[a] comprehensive, coherent theory of bounded rationality is not available" [34, p. 14]. Nevertheless, we believe the essence of our work is correct: an action is for a purpose if the actor selects to perform that action while planning for the purpose. Future work will instantiate our semantic framework with more complete models of human planning.

Additionally, future work will make our formalism easier to use. To use our auditing algorithm, an auditor must not only log the auditee's behavior but also know how the auditee *could* have behaved with an environment model. Given the difficulty of this task, we desire methods for finding policy violations that do not require a completely accurate model of the environment.

For example, Experience-Based Access Management (EBAM) is an informal methodology for managing access-control policies and related models that has the auditor iteratively refine these models using observations from audit logs [35]. Each iteration uses observations of false positives and negatives in an effort to improve the accuracy of the models enabling more accurate enforcement of the policy. To make this methodology formal, which allows for automation, an auditor must formally define the policies and models. For example, Zhang et al. apply EBAM to Role-Based Access Control (RBAC) by focusing on RBAC policies and modeling individuals subject to the policy using a role hierarchy [79]. Using this formalism, Zhang et al. provide an algorithm to refine the role hierarchy over time.

To apply EBAM to purpose restrictions requires a formal semantics for them, which we provide. Furthermore, future work will have to provide a refinement algorithm consistent with our semantics. Reinforcement learning (e.g., Q-learning [75]) provide strategies for optimizing the total reward using observations of auditee behavior (often available from logs) instead of a single fixed MDP model. By learning the model from observations, reinforcement learning may provide a means of refining the environment models used by our auditing algorithm. Combining these methods, an auditor may use our formal semantics and auditing algorithm to determine when an auditee violates the policy under the current model of the environment and use reinforcement learning to improve the model in response to false positives and negatives in a manner consistent with the EBAM methodology.

Acknowledgments. We thank Lorrie Faith Cranor, Joseph Y. Halpern, Dilsun Kaynar, Divya Sharma, and Manuela M. Veloso for many helpful comments on this work. We also thank the anonymous reviewers on our submission to the 33rd IEEE Symposium on Security and Privacy, the conference version [71] of this report.

References

[1] purpose, n. In *The Oxford English Dictionary.* Oxford University Press, 2nd edition, 1989.

[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. Hippocratic

databases. In *VLDB '02: Proceedings of the 28th International Conference on Very Large Data Bases*, pages 143–154. VLDB Endowment, 2002.

[3] Sabah S. Al-Fedaghi. Beyond purpose-based privacy access control. In *ADC '07: Proceedings of the Eighteenth Australasian Database Conference*, pages 23–32. Australian Computer Society, Darlinghurst, Australia, 2007.

[4] John R. Anderson, Daniel Bothell, Michael D. Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin. An integrated theory of the mind. *Psychological Review*, 111:1036–1060, 2004.

[5] G.E.M. Anscombe. *Intention*. Harvard University Press, Cambridge, MA, USA, 1957.

[6] Jerome Azarewicz, Glenn Fala, Ralph Fink, and Christof Heithecker. Plan recognition for airborne tactical decision making. In *National Conference on Artificial Intelligence*, pages 805–811. 1986.

[7] Chris L. Baker, Joshua B. Tenenbaum, and Rebecca R. Saxe. Bayesian models of human action understanding. In *Advances in Neural Information Processing Systems 18*, pages 99–106. MIT Press, 2006.

[8] Chris L. Baker, Rebecca Saxe, and Joshua B. Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

[9] Bank of America Corporation. Bank of America privacy policy for consumers, September 2005. Available at `http://www.bankofamerica.com/privacy/pdf/eng-boa.pdf`. Accessed Feb. 4, 2011.

[10] Adam Barth, John Mitchell, Anupam Datta, and Sharada Sundaram. Privacy and utility in business processes. In *CSF '07: Proceedings of the 20th IEEE Computer Security Foundations Symposium*, pages 279–294. IEEE Computer Society, Washington, DC, USA, 2007.

[11] Richard Bellman. On the theory of dynamic programming. *Proceedings of the National Academy of Sciences*, 38:716–719, 1952.

[12] Adam J. Berinsky, Gregory A. Huber, and Gabriel S. Lenz. Using Mechanical Turk as a subject recruitment tool for experimental research. Submitted for review, June 2011.

[13] Jim Blythe. Decision-theoretic planning. *AI Magazine*, 20(2):37–54, 1999.

[14] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge, MA, USA, 1987.

[15] Travis D. Breaux and Annie I. Antón. Analyzing goal semantics for rights, permissions, and obligations. In *RE '05: Proceedings of the 13th IEEE International Conference on Requirements Engineering*, pages 177–188. IEEE Computer Society, Washington, DC, USA, 2005.

[16] Travis D. Breaux and Annie I. Antón. Analyzing regulatory rules for privacy and security requirements. *IEEE Trans. Softw. Eng.*, 34(1):5–20, 2008.

[17] Carolyn Brodie, Clare-Marie Karat, John Karat, and Jinjuan Feng. Usable security and privacy: a case study of developing privacy management tools. In *SOUPS '05: Proceedings of the 2005 Symposium on Usable Privacy and Security*, pages 35–43. ACM, New York, NY, USA, 2005.

[18] Carolyn A. Brodie, Clare-Marie Karat, and John Karat. An empirical study of natural language parsing of privacy policy rules using the sparcle policy workbench. In *SOUPS '06: Proceedings of the Second Symposium on Usable Privacy and Security*, pages 8–19. ACM, New York, NY, USA, 2006.

[19] Ji-Won Byun and Ninghui Li. Purpose based access control for privacy protection in relational database systems. *The VLDB Journal*, 17(4):603–619, 2008.

[20] Ji-Won Byun, Elisa Bertino, and Ninghui Li. Purpose based access control of complex data for privacy protection. In *SACMAT '05: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, pages 102–110. ACM, New York, NY, USA, 2005.

[21] Stuart Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, 1983.

[22] Stephen Chong, Andrew C. Myers, K. Vikram, and Lantian Zheng. *Jif Reference Manual*, February 2009. Available at `http://www.cs.cornell.edu/jif`.

[23] Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3):213–261, March 1990.

[24] Lorrie Cranor, Marc Langheinrich, Massimo Marchiori, Martin Presler-Marshall, and Joseph Reagle. The Platform for Privacy Preferences 1.0 (P3P1.0) specification. W3C Recommendation, April 2002.

[25] Lorrie Faith Cranor. *Web Privacy with P3P*. O'Reilly, 2002.

[26] Frédéric Cuppens, Fabien Autrel, Alexandre Miège, and Salem Benferhat. Recognizing malicious intention in an intrusion detection process. In Ajith Abraham, Javier Ruiz del Solar, and Mario Köppen, editors, *Second International Conference on Hybrid Intelligent Systems*, volume 87 of *Frontiers in Artificial Intelligence and Applications*, pages 806–817. IOS Press, 2002.

[27] Jagannath Prasad Das, Binod C. Kar, and Rauno K. Parrila. *Cognitive Planning: The Psychological Basis of Intelligent Behavior*. Sage, 1996.

[28] F. d'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10 (1):98–108, October 1963.

[29] Md. Enamul Kabir, Hua Wang, and Elisa Bertino. A conditional purpose-based access control model with dynamic roles. *Expert Syst. Appl.*, 38(3):1482–1489, March 2011.

[30] FairWarning. Privacy breach detection for healthcare. White Paper, 2010. Available at `http://www.fairwarning.com/documents/2010-privacy-breach-detection-fairwarning.pdf`.

[31] Adam Feltz. The knobe effect: A brief overview. *Journal of Mind and Behavior*, 28:265–278, 2008.

[32] Matthew Flatt and PLT. Reference: Racket. Technical Report PLT-TR-2010-1, PLT Inc., 2010. Available at `http://racket-lang.org/tr1/`. Revised August 2011.

[33] Christopher W. Geib and Robert P. Goldman. Plan recognition in intrusion detection systems. In *DARPA Information Survivability Conference and Exposition (DISCEX)*. 2001.

[34] Gerd Gigerenzer and Reinhard Selten, editors. *Bounded Rationality: The Adaptive Toolbox*. Dahlem Workshop Reports. MIT Press, 2002.

[35] Carl A. Gunter, David M. Liebovitz, and Bradley Malin. Experience-based access management: A life-cycle framework for identity and access management systems. *IEEE Security and Privacy*, 9(5):48–55, 2011.

[36] Chris Hanson, Tim Berners-Lee, Lalana Kagal, Gerald Jay Sussman, and Daniel Weitzner. Data-purpose algebra: Modeling data usage policies. In *POLICY '07: Proceedings of the Eighth IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 173–177. IEEE Computer Society, Washington, DC, USA, 2007.

[37] Katia Hayati and Martín Abadi. Language-based enforcement of privacy policies. In *PET 2004: Workshop on Privacy Enhancing Technologies*, pages 302–313. Springer-Verlag, 2005.

[38] Mohammad Jafari, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard. Enforcing purpose of use via workflows. In *WPES '09: Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society*, pages 113–116. ACM, New York, NY, USA, 2009.

[39] Bonnie E. John and David E. Kieras. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Trans. Comput.-Hum. Interact.*, 3(4):320–351, December 1996.

[40] B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of Sixth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 266–277. IEEE Press, July 1991.

[41] N. Karmarkar. A new polynomial-time algorithm for linear programming. In *STOC '84: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 302–311. ACM, New York, NY, USA, 1984.

[42] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 32–37. AAAI, 1986.

[43] L. G. Khachian. A polynomial algorithm in linear programming. *Dokl. Akad. Nauk SSSR*, 244:1093–1096, 1979. English translation in Soviet Math. Dokl. 20, 191-194, 1979.

[44] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with Mechanical Turk. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 453–456. ACM, New York, NY, USA, 2008.

[45] J. Knobe. Intentional action and side effects in ordinary language. *Analysis*, 63:190–193, 2003.

[46] Ram Krishnan, Ravi Sandhu, Jianwei Niu, and William H. Winsborough. A conceptual framework for group-centric secure information sharing. In *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 384–387. ACM, New York, NY, USA, 2009.

[47] Michael L. Littman, Thomas L. Dean, and Leslie P. Kaelbling. On the complexity of solving Markov decision problems. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–95)*, pages 394–402. Morgan Kaufmann, 1995.

[48] John L. Mackie. *The Cement of the Universe: A Study of Causation*. Oxford University Press, 1974.

[49] Wenji Mao and Jonathan Gratch. A utility-based approach to intention recognition. In *AAMAS 2004 Workshop on Agent Tracking: Modeling Other Agents from Observations*. July 2004.

[50] Fabio Massacci, John Mylopoulos, and Nicola Zannone. Hierarchical hippocratic databases with minimal disclosure for virtual organizations. *The VLDB Journal*, 15(4):370–387, 2006.

[51] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

[52] Gustavo H.M.B. Motta and Sergio S. Furuie. A contextual role-based access control authorization model for electronic patient record. *Information Technology in Biomedicine, IEEE Transactions on*, 7(3):202–207, sept 2003.

[53] Qun Ni, Elisa Bertino, Jorge Lobo, Carolyn Brodie, Clare-Marie Karat, John Karat, and Alberto Trombetta. Privacy-aware role-based access control. *ACM Trans. Inf. Syst. Secur.*, 13(3):24:1–24:31, July 2010.

[54] Office for Civil Rights. Summary of the HIPAA privacy rule. OCR Privacy Brief, U.S. Department of Health and Human Services, 2003.

[55] Martin T. Orne. Demand characteristics and the concept of quasi-controls. In Robert Rosenthal and Ralph L. Rosnow, editors, *Artifacts in Behavioral Research: Robert Rosenthal and Ralph L. Rosnow's Classic Books*, page 110. Oxford University Press, 2009.

[56] Mor Peleg, Dizza Beimel, Dov Dori, and Yaron Denekamp. Situation-based access control: Privacy management via modeling of patient data access scenarios. *J. of Biomedical Informatics*, 41(6):1028–1040, December 2008.

[57] Huanchun Peng, Jun Gu, and Xiaojun Ye. Dynamic purpose-based access control. In *International Symposium on Parallel and Distributed Processing with Applications*, pages 695–700. IEEE Computer Society, Los Alamitos, CA, USA, 2008.

[58] Calvin Powers and Matthias Schunter. Enterprise privacy authorization language (EPAL 1.2). W3C Member Submission, November 2003.

[59] Miquel Ramírez and Hector Geffner. Goal recognition over POMDPs: Inferring the intention of a POMDP agent. In Toby Walsh, editor, *IJCAI*, pages 2009–2014. IJCAI/AAAI, 2011.

[60] Joel Ross, Lilly Irani, M. Six Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers? Shifting demographics in Mechanical Turk. In *Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, CHI EA '10, pages 2863–2872. ACM, New York, NY, USA, 2010.

[61] Olivier Roy. *Thinking before Acting: Intentions, Logic, Rational Choice.* PhD thesis, Institute for Logic, Language and Computation; Universiteit van Amsterdam, 2008.

[62] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2nd edition, 2003.

[63] Ravi S. Sandhu. Role hierarchies and constraints for lattice-based access controls. In *ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security*, pages 65–79. Springer-Verlag, London, UK, 1996.

[64] C.F. Schmidt, N.S. Sridharan, and J.L. Goodson. The plan recognition problem: An intersection of psychology and artificial intelligence. *Artificial Intelligence*, 11(1-2):45 – 83, 1978.

[65] Koushik Sen, Mahesh Viswanathan, and Gul Agha. Model-checking markov chains in the presence of uncertainties. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 3920 of *Lecture Notes in Computer Science*, pages 394–410. Springer-Verlag Berlin Heidelberg, 2006.

[66] Herbert A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 69:99–118, 1955.

[67] R E Strom and S Yemini. Typestate: A programming language concept for enhancing software reliability. *IEEE Trans. Softw. Eng.*, 12(1):157–171, January 1986.

[68] Richard Taylor. *Action and Purpose.* Prentice-Hall, 1966.

[69] The European Parliament and the Council of the European Union. Directive 95/46/EC. *Official Journal of the European Union*, L 281:31–50, November 1995.

[70] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. On the semantics of purpose requirements in privacy policies. Technical Report CMU-CS-11-102, School of Computer Science, Carnegie Mellon University, February 2011. Also available at `http://arxiv.org/abs/1102.4326`.

[71] Michael Carl Tschantz, Anupam Datta, and Jeannette M. Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *Proceedings of the IEEE Symposium on Security and Privacy.* IEEE Computer Society, Los Alamitos, CA, USA, 2012. To appear.

[72] Paul Tseng. Solving h-horizon stationary Markov decision process in time proportional to log(h). *Operations Research Letters*, 9(5):287–297, 1990.

[73] United States Congress. Financial services modernization act of 1999. Title 15, United States Code, Section 6802, February 2010.

[74] Washington Radiology Associates, P.C. Notice of privacy practices, April 2003. Available at `http://www.washingtonradiology.com/office-guide/privacy.asp`. Accessed Feb. 4, 2011.

[75] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards.* PhD thesis, Cambridge University, 1989.

[76] Ronald Williams and Leemon C. Baird. Tight performance bounds on greedy policies based on imperfect value functions. Technical Report NU-CCS-93-14, Northeastern University, November 1993.

[77] Ronald Williams and Leemon C. Baird. Tight performance bounds on greedy policies based on imperfect value functions. In *Proceedings of the Tenth Yale Workshop on Adaptive and Learning Systems*. Yale University, June 1994.

[78] Yahoo! Privacy policy: Yahoo Mail, 2010. Available at `http://info.yahoo.com/privacy/us/yahoo/mail/details.html`.

[79] Wen Zhang, Carl A. Gunter, David Liebovitz, Jian Tian, and Bradley Malin. Role prediction using electronic medical record system audits. In *AMIA 2011 Annual Symposium*, pages 858–867. American Medical Informatics Association, October 2011.

A Questionnaire

Below is the content of the questionnaire. The formatting differed in that it was broken up into multiple webpages. Initial instructions were shown on Mechanical Turk's website (Appendix A.1). The additional instructions, questions, and payment information were shown on Survey Gizmo's website (Appendix A.2). Survey Gizmo always showed the additional instructions first and the payment information last. For each participant, Survey Gizmo presented the scenarios in a random order and on its own webpage. Survey Gizmo numbers the questions dynamically based upon the order in which Survey Gizmo presents the scenarios.

A.1 Mechanical Turk

If you choose to participate, you will be asked a series questions [sic] about when an action is for a purpose. If you fill out the survey reasonably (do not just randomly select answers), you will be paid for your participation. The risks of taking this survey are equivalent to every day computer use. Your participation is voluntary.
If you choose to participate, then fill out the survey at SurveyGizmo using the following link:
`http://edu.surveygizmo.com/s3/621146/Hospital-Survey`
Upon completion enter the last four digits of your phone number here:

□

We ask for this number so we can track who successfully completed the survey. We will ask you to enter the same number at SurveyGizmo.

A.2 Survey Gizmo

Instructions. Metropolis General Hospital has the following privacy policy:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

For each scenario below, please answer the following questions based on your understanding of the above policy.

Scenario 1. Metropolis General Hospital has the following privacy policy:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

Please answer the following questions based on your understanding of the above policy and the following scenario:
A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of treating the patient. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist succeeds in treating the patient.

1. Was the goal of the case worker's plan to treat the patient?

   (a) Yes
   (b) No
   (c) I don't know

2. Did the specialist succeed in treating the patient?

   (a) Yes
   (b) No
   (c) I don't know

3. Did the case worker share the record with the specialist for the purpose of treatment?

   (a) Yes
   (b) No
   (c) I don't know

4. Did the case worker obey the above privacy policy?

   (a) Yes
   (b) No
   (c) I don't know

5. Why did you answer Question 4 as you did?

Scenario 2. Metropolis General Hospital has the following privacy policy:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

Please answer the following questions based on your understanding of the above policy and the following scenario:
A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of treating the patient. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist did *not* succeed in treating the patient.

1. Was the goal of the case worker's plan to treat the patient?
   (a) Yes
   (b) No
   (c) I don't know

2. Did the specialist succeed in treating the patient?
   (a) Yes
   (b) No
   (c) I don't know

3. Did the case worker share the record with the specialist for the purpose of treatment?
   (a) Yes
   (b) No
   (c) I don't know

4. Did the case worker obey the above privacy policy?
   (a) Yes
   (b) No
   (c) I don't know

5. Why did you answer Question 4 as you did?

Scenario 3. Metropolis General Hospital has the following privacy policy:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

Please answer the following questions based on your understanding of the above policy and the following scenario:
A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of reducing costs for the hospital. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist succeeds in treating the patient.

1. Was the goal of the case worker's plan to treat the patient?

   (a) Yes
   (b) No
   (c) I don't know

2. Did the specialist succeed in treating the patient?

   (a) Yes
   (b) No
   (c) I don't know

3. Did the case worker share the record with the specialist for the purpose of treatment?

   (a) Yes
   (b) No
   (c) I don't know

4. Did the case worker obey the above privacy policy?

   (a) Yes
   (b) No
   (c) I don't know

5. Why did you answer Question 4 as you did?

Scenario 4. Metropolis General Hospital has the following privacy policy:

> Metropolis General Hospital and its employees will share a patient's medical record with an outside specialist only for the purpose of providing that patient with treatment.

Please answer the following questions based on your understanding of the above policy and the following scenario:
A case worker employed by Metropolis General Hospital meets with a patient. The case worker develops a plan with the sole goal of reducing costs for the hospital. The plan includes sharing the patient's medical record with an outside specialist. Upon receiving the record, the specialist did *not* succeed in treating the patient.

1. Was the goal of the case worker's plan to treat the patient?

   (a) Yes
   (b) No
   (c) I don't know

2. Did the specialist succeed in treating the patient?

   (a) Yes

(b) No

(c) I don’t know

3. Did the case worker share the record with the specialist for the purpose of treatment?

   (a) Yes

   (b) No

   (c) I don’t know

4. Did the case worker obey the above privacy policy?

   (a) Yes

   (b) No

   (c) I don’t know

5. Why did you answer Question 4 as you did?

Payment Information. To receive payment on Mechanical Turk, please enter the last four digits of your phone number here:

A.3 Mechanical Turk Advertisement

```
Research survey on the meaning of privacy

Requester: Michael Carl Tschantz    HIT Expiration Date: Jul 20, 2011 (2 weeks 5 days)   Reward: $0.50
                                    Time Allotted: 10 minutes                            HITs Available: 200
Description: Take a short survey about how you interpret a privacy policy to help
             research on the topic taking place at Carnegie Mellon University.

Keywords: Survey, Research

Qualifications Required:
HIT approval rate (%) is greater than 95
Location is US
```

B Tables of Matched Pairs

Question Q1		$S_{p\bar{f}}$		
		Yes	I don't know	No
	Yes	176	0	6
S_{pf}	I don't know	0	2	0
	No	1	0	2

Question Q1		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	26	3	16
$S_{\bar{p}f}$	I don't know	1	5	3
	No	4	1	128

Question Q3		$S_{p\bar{f}}$		
		Yes	I don't know	No
	Yes	182	1	2
S_{pf}	I don't know	1	0	1
	No	0	0	0

Question Q3		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	25	2	16
$S_{\bar{p}f}$	I don't know	0	4	2
	No	13	4	121

Question Q1		$S_{\bar{p}f}$		
		Yes	I don't know	No
	Yes	45	8	129
S_{pf}	I don't know	0	1	1
	No	0	0	3

Question Q1		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	30	8	139
$S_{p\bar{f}}$	I don't know	0	1	1
	No	1	0	7

Question Q3		$S_{\bar{p}f}$		
		Yes	I don't know	No
	Yes	43	6	136
S_{pf}	I don't know	0	0	2
	No	0	0	0

Question Q3		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	37	9	137
$S_{p\bar{f}}$	I don't know	0	0	1
	No	1	1	1

Scenario S_{pf}		Q3		
		Yes	I don't know	No
	Yes	181	1	0
q1	I don't know	1	1	0
	No	3	0	0

Scenario $S_{p\bar{f}}$		Q3		
		Yes	I don't know	No
	Yes	176	1	0
q1	I don't know	2	0	0
	No	5	0	3

Scenario $S_{\bar{p}f}$		Q3		
		Yes	I don't know	No
	Yes	32	2	11
q1	I don't know	1	3	5
	No	10	1	122

Scenario $S_{\bar{p}\bar{f}}$		Q3		
		Yes	I don't know	No
	Yes	22	0	9
q1	I don't know	2	5	2
	No	14	5	128

C Results Using All Respondents

Scenario	Yes	I don't know	No
S_{pf}	201 (97%)	2 (01%)	4 (02%)
$\mathrm{S}_{\mathsf{p\bar{f}}}$	195 (94%)	3 (01%)	9 (04%)
$\mathrm{S}_{\mathsf{\bar{p}f}}$	61 (29%)	11 (05%)	135 (65%)
$\mathrm{S}_{\mathsf{\bar{p}\bar{f}}}$	44 (21%)	12 (06%)	151 (73%)

Q1: Was the policy obeyed?

Scenario	Yes	I don't know	No
S_{pf}	205 (99%)	2 (01%)	0 (00%)
$\mathrm{S}_{\mathsf{p\bar{f}}}$	202 (98%)	2 (01%)	3 (01%)
$\mathrm{S}_{\mathsf{\bar{p}f}}$	59 (29%)	9 (04%)	139 (67%)
$\mathrm{S}_{\mathsf{\bar{p}\bar{f}}}$	51 (25%)	14 (07%)	142 (69%)

Q3: Was the action for the purpose?

Scenario	Yes	I don't know	No
S_{pf}	205 (99%)	1 (00%)	1 (00%)
$\mathrm{S}_{\mathsf{p\bar{f}}}$	202 (98%)	2 (01%)	3 (01%)
$\mathrm{S}_{\mathsf{\bar{p}f}}$	25 (12%)	5 (02%)	177 (86%)
$\mathrm{S}_{\mathsf{\bar{p}\bar{f}}}$	18 (09%)	2 (01%)	187 (90%)

Q4: Was the goal treatment?

Scenario	Yes	I don't know	No
S_{pf}	206 (100%)	1 (00%)	0 (00%)
$\mathrm{S}_{\mathsf{p\bar{f}}}$	3 (01%)	1 (00%)	203 (98%)
$\mathrm{S}_{\mathsf{\bar{p}f}}$	196 (95%)	3 (01%)	8 (04%)
$\mathrm{S}_{\mathsf{\bar{p}\bar{f}}}$	5 (02%)	0 (00%)	202 (98%)

Q5: Was the treatment successful?

Table 11: Survey Results for All Respondents

Testing	Alternative Hypothesis	Null Hypothesis	p-Value	Significant?
Against H1a	$p_{\mathsf{pfy}} < 0.5$	$p_{\mathsf{pfy}} = 0.5$	1	No
Against H1a	$p_{\mathsf{pfn}} > 0.5$	$p_{\mathsf{pfn}} = 0.5$	1	No
Against H1a	$p_{\mathsf{\bar{p}fy}} < 0.5$	$p_{\mathsf{\bar{p}fy}} = 0.5$	1.59774e-009	Yes
Against H1a	$p_{\mathsf{\bar{p}fn}} > 0.5$	$p_{\mathsf{\bar{p}fn}} = 0.5$	7.097797e-006	Yes
Against H1a'	$p'_{\mathsf{pfy}} < 0.5$	$p'_{\mathsf{pfy}} = 0.5$	1	No
Against H1a'	$p'_{\mathsf{pfn}} > 0.5$	$p'_{\mathsf{pfn}} = 0.5$	1	No
Against H1a'	$p'_{\mathsf{\bar{p}fy}} < 0.5$	$p'_{\mathsf{\bar{p}fy}} = 0.5$	2.606856e-010	Yes
Against H1a'	$p'_{\mathsf{\bar{p}fn}} > 0.5$	$p'_{\mathsf{\bar{p}fn}} = 0.5$	4.514694e-007	Yes
Against H1b	$p_{\mathsf{p\bar{f}n}} < 0.5$	$p_{\mathsf{p\bar{f}n}} = 0.5$	8.206618e-048	Yes
Against H1b	$p_{\mathsf{p\bar{f}y}} > 0.5$	$p_{\mathsf{p\bar{f}y}} = 0.5$	4.833563e-044	Yes
Against H1b	$p_{\mathsf{\bar{p}\bar{f}n}} < 0.5$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	1	No
Against H1b	$p_{\mathsf{\bar{p}\bar{f}y}} > 0.5$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1	No
Against H1b'	$p'_{\mathsf{p\bar{f}n}} < 0.5$	$p'_{\mathsf{p\bar{f}n}} = 0.5$	7.187894e-057	Yes
Against H1b'	$p'_{\mathsf{p\bar{f}y}} > 0.5$	$p'_{\mathsf{p\bar{f}y}} = 0.5$	1.503496e-053	Yes
Against H1b'	$p'_{\mathsf{\bar{p}\bar{f}n}} < 0.5$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	1	No
Against H1b'	$p'_{\mathsf{\bar{p}\bar{f}y}} > 0.5$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1	No
For H2a	$p_{\mathsf{pfy}} > 0.5$	$p_{\mathsf{pfy}} = 0.5$	5.08808e-052	Yes
For H2a	$p_{\mathsf{pfn}} < 0.5$	$p_{\mathsf{pfn}} = 0.5$	3.684324e-055	Yes
For H2a	$p_{\mathsf{p\bar{f}y}} > 0.5$	$p_{\mathsf{p\bar{f}y}} = 0.5$	4.833563e-044	Yes
For H2a	$p_{\mathsf{p\bar{f}n}} < 0.5$	$p_{\mathsf{p\bar{f}n}} = 0.5$	8.206618e-048	Yes
For H2a'	$p'_{\mathsf{pfy}} > 0.5$	$p'_{\mathsf{pfy}} = 0.5$	1.046682e-058	Yes
For H2a'	$p'_{\mathsf{pfn}} < 0.5$	$p'_{\mathsf{pfn}} = 0.5$	4.861731e-063	Yes
For H2a'	$p'_{\mathsf{p\bar{f}y}} > 0.5$	$p'_{\mathsf{p\bar{f}y}} = 0.5$	1.503496e-053	Yes
For H2a'	$p'_{\mathsf{p\bar{f}n}} < 0.5$	$p'_{\mathsf{p\bar{f}n}} = 0.5$	7.187894e-057	Yes
For H2b	$p_{\mathsf{\bar{p}fn}} > 0.5$	$p_{\mathsf{\bar{p}fn}} = 0.5$	7.097797e-006	Yes
For H2b	$p_{\mathsf{\bar{p}fy}} < 0.5$	$p_{\mathsf{\bar{p}fy}} = 0.5$	1.59774e-009	Yes
For H2b	$p_{\mathsf{\bar{p}\bar{f}n}} > 0.5$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	1.443359e-011	Yes
For H2b	$p_{\mathsf{\bar{p}\bar{f}y}} < 0.5$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	1.440142e-017	Yes
For H2b'	$p'_{\mathsf{\bar{p}fn}} > 0.5$	$p'_{\mathsf{\bar{p}fn}} = 0.5$	4.514694e-007	Yes
For H2b'	$p'_{\mathsf{\bar{p}fy}} < 0.5$	$p'_{\mathsf{\bar{p}fy}} = 0.5$	2.606856e-010	Yes
For H2b'	$p'_{\mathsf{\bar{p}\bar{f}n}} > 0.5$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.5$	4.581869e-008	Yes
For H2b'	$p'_{\mathsf{\bar{p}\bar{f}y}} < 0.5$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.5$	7.161858e-014	Yes

Table 12: Binomial Hypothesis Tests for All Respondents

Testing	Alternative Hypothesis	Null Hypothesis
Proving H2a	$p_{\mathsf{pfy}} > 0.94$	$p_{\mathsf{pfy}} = 0.94$
Proving H2a	$p_{\mathsf{pfn}} < 0.05$	$p_{\mathsf{pfn}} = 0.05$
Proving H2a	$p_{\mathsf{p\bar{f}y}} > 0.9$	$p_{\mathsf{p\bar{f}y}} = 0.9$
Proving H2a	$p_{\mathsf{p\bar{f}n}} < 0.08$	$p_{\mathsf{p\bar{f}n}} = 0.08$
Proving H2a'	$p'_{\mathsf{pfy}} > 0.96$	$p'_{\mathsf{pfy}} = 0.96$
Proving H2a'	$p'_{\mathsf{pfn}} < 0.02$	$p'_{\mathsf{pfn}} = 0.02$
Proving H2a'	$p'_{\mathsf{p\bar{f}y}} > 0.94$	$p'_{\mathsf{p\bar{f}y}} = 0.94$
Proving H2a'	$p'_{\mathsf{p\bar{f}n}} < 0.04$	$p'_{\mathsf{p\bar{f}n}} = 0.04$
Proving H2b	$p_{\mathsf{\bar{p}fn}} > 0.59$	$p_{\mathsf{\bar{p}fn}} = 0.59$
Proving H2b	$p_{\mathsf{\bar{p}fy}} < 0.36$	$p_{\mathsf{\bar{p}fy}} = 0.36$
Proving H2b	$p_{\mathsf{\bar{p}\bar{f}n}} > 0.67$	$p_{\mathsf{\bar{p}\bar{f}n}} = 0.67$
Proving H2b	$p_{\mathsf{\bar{p}\bar{f}y}} < 0.27$	$p_{\mathsf{\bar{p}\bar{f}y}} = 0.27$
Proving H2b'	$p'_{\mathsf{\bar{p}fn}} > 0.61$	$p'_{\mathsf{\bar{p}fn}} = 0.61$
Proving H2b'	$p'_{\mathsf{\bar{p}fy}} < 0.35$	$p'_{\mathsf{\bar{p}fy}} = 0.35$
Proving H2b'	$p'_{\mathsf{\bar{p}\bar{f}n}} > 0.62$	$p'_{\mathsf{\bar{p}\bar{f}n}} = 0.62$
Proving H2b'	$p'_{\mathsf{\bar{p}\bar{f}y}} < 0.31$	$p'_{\mathsf{\bar{p}\bar{f}y}} = 0.31$

Table 13: Extreme Binomial Hypothesis Tests for All Respondents. This table shows the hypothesis test using the most extreme probability for which statistical significance is still achieved and is accurate up to two places after the decimal point.

Question Q1		$S_{p\bar{f}}$		
		Yes	I don't know	No
	Yes	194	1	6
S_{pf}	I don't know	0	2	0
	No	1	0	3

Question Q1		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	39	4	18
$S_{\bar{p}f}$	I don't know	1	7	3
	No	4	1	130

Question Q3		$S_{p\bar{f}}$		
		Yes	I don't know	No
	Yes	201	2	2
S_{pf}	I don't know	1	0	1
	No	0	0	0

Question Q3		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	38	3	18
$S_{\bar{p}f}$	I don't know	0	7	2
	No	13	4	122

Question Q1		$S_{\bar{p}f}$		
		Yes	I don't know	No
	Yes	61	10	130
S_{pf}	I don't know	0	1	1
	No	0	0	4

Question Q1		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	43	10	142
$S_{p\bar{f}}$	I don't know	0	2	1
	No	1	0	8

Question Q3		$S_{\bar{p}f}$		
		Yes	I don't know	No
	Yes	59	9	137
S_{pf}	I don't know	0	0	2
	No	0	0	0

Question Q3		$S_{\bar{p}\bar{f}}$		
		Yes	I don't know	No
	Yes	50	12	140
$S_{p\bar{f}}$	I don't know	0	1	1
	No	1	1	1

Scenario S_{pf}		Q3		
		Yes	I don't know	No
	Yes	200	1	0
q1	I don't know	1	1	0
	No	4	0	0

Scenario $S_{p\bar{f}}$		Q3		
		Yes	I don't know	No
	Yes	194	1	0
q1	I don't know	2	1	0
	No	6	0	3

Scenario $S_{\bar{p}f}$		Q3		
		Yes	I don't know	No
	Yes	47	3	11
q1	I don't know	1	5	5
	No	11	1	123

Scenario $S_{\bar{p}\bar{f}}$		Q3		
		Yes	I don't know	No
	Yes	34	1	9
q1	I don't know	2	8	2
	No	15	5	131

Table 14: Matched Pairs for All Respondents

Testing	Question	Scenarios	p-Value	Significant?
For H1c	Q1	S_{pf} vs. $S_{\mathsf{p\bar{f}}}$	NaN	No
For H1c	Q1	$S_{\mathsf{\bar{p}f}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	0.008449127	Yes
For H1c'	Q3	S_{pf} vs. $S_{\mathsf{p\bar{f}}}$	0.3430301	No
For H1c'	Q3	$S_{\mathsf{\bar{p}f}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	0.2147006	No
For H2c	Q1	S_{pf} vs. $S_{\mathsf{\bar{p}f}}$	2.300576e-030	Yes
For H2c	Q1	$S_{\mathsf{p\bar{f}}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	2.598558e-032	Yes
For H2c'	Q3	S_{pf} vs. $S_{\mathsf{\bar{p}f}}$	7.115157e-032	Yes
For H2c'	Q3	$S_{\mathsf{p\bar{f}}}$ vs. $S_{\mathsf{\bar{p}\bar{f}}}$	4.269341e-032	Yes

Table 15: McNemar's Test Across Scenarios for All Respondents

Scenario	Questions	p-Value	Significant?
S_{pf}	Q1 vs. Q3	NaN	No
$S_{\mathsf{p\bar{f}}}$	Q1 vs. Q3	NaN	No
$S_{\mathsf{\bar{p}f}}$	Q1 vs. Q3	0.2997806	No
$S_{\mathsf{\bar{p}\bar{f}}}$	Q1 vs. Q3	0.3736321	No

Table 16: McNemar's Test Across Questions for All Respondents