

# **Sherlock Holmes goes Cyber**

## **Deception Detection on the Basis of Actions**

James D. Jones  
Hemant Joshi  
Umit Topaloglu  
Eric Nelson

Department of Computer Science  
University of Arkansas at Little Rock  
james.d.jones@acm.org

### **Abstract**

In work very relevant to national defense and homeland security, this paper describes software that detects deception on the basis of actions. This is in sharp contrast with present approaches that detect deception based on physiological factors, as well as on verbal and non-verbal cues. Our approach attempts to model agents and their actions. This is achieved in a logic programming framework using a theory of agents, a theory of actions, and a theory of reasoning with respect to time. As a test case, a children's mystery is analyzed and implemented. The software correctly reasons about who the potential suspects are, and ultimately, correctly identifies the chief culprit. Further, it can correctly introspect with regard to previously held beliefs.

### **Introduction**

Present approaches to automatically identifying deception, detect deception on the basis of physiological factors, and on the basis of verbal and nonverbal queues ([10], [11], [31]). An example of how software can use physiological factors to detect deception would include measuring vocal stress. Another example would be to measure heat released from the body. An example of using a verbal queue would be a situation where the speaker consistently speaks in first person plural rather than in first person singular. The aim of such a tactic is to avoid accountability. For instance, the speaker may say "we wrote the memo", instead of "I wrote the memo". Another tactic to avoid accountability is to completely disassociate one's self from the action, speaking in the third person as in "it was decided that ...", rather than in the first person "I decided that ...".

In contrast to these approaches, we have developed software that identifies deception on the basis of actions, using a well defined theory of actions ([6], [7], [26], [28]). Software based on this approach should be able to predict consequences of actions, and predict future actions. A more detailed discussion of our paradigm is at the end of this introduction. The program we have developed is able to mimic the thought processes and conclusions of a police investigation.

To demonstrate our approach, we have created a scenario loosely based upon a child's mystery<sup>1</sup>. The basic idea of the story is that a bat is missing, and the goal is to identify who stole the bat. The basic flow of the story is as follows. There is a practice at the beginning of the story. A particular baseball bat is missing, and presumed stolen. Everyone present at that practice is a suspect in the theft of the bat. There is a subsequent practice in which a glove becomes missing. Everyone present at that subsequent practice is a suspect in the theft of the glove. At a later time, someone is discovered in possession of the glove. That person is assumed to have stolen the glove, and hence, the most likely person to have stolen the bat, since he was a suspect in the theft of both items. The events of our story occur over time. Our program can correctly represent and reason about these events.

### *Background: A-Prolog*

The main technical tool to be used for reasoning about deception is *A-Prolog* -- a language of logic programs under the answer set (stable model) semantics ([29], [30]). *A-Prolog* can be viewed as a purely declarative language with roots in logic programming ([35], [36]), syntax and semantics of standard Prolog ([17], [18]), and in the work on nonmonotonic logic ([48], [42]). The inference engine used is SMOBELS. This inference engine is aimed at computing answer sets (stable models) of programs of *A-Prolog* ([45], [44], [16]).

## **Execution**

The events in the story are sequential with respect to time and so the experiments were performed in incremental fashion. They are presented here as four distinct executions. However, each execution completely subsumes the previous execution. As such, the final execution contains all the results of the earlier executions.

### *Execution 1*

Execution 1 is trivial, but is necessary to demonstrate that our program reasons correctly, and does not enter the arena with unfair predispositions. The results of this execution tell us that there are no items missing, and hence no suspects.

### *Execution 2*

At a subsequent time (time period 2), a practice occurs. Those present at the practice are: Jill, Marshall, Ben, and Gwen. Jill's bat becomes missing, and is assumed stolen. Those present at the practice are assumed suspects in the theft of the bat. Obviously, Jill is not a suspect, since she is the owner of the bat. The relevant facts and inferences output from the program are presented in figure 1.

---

<sup>1</sup> The story used as a basis for our scenario is "Something Queer at the Ball Park", by Elizabeth Levy. We have greatly altered the story in order to provide interesting results.

```
holds(missing(bat),2,1)
holds(suspect(marshall,bat),2,1)
holds(suspect(ben,bat),2,1)
holds(suspect(gwen,bat),2,1)
```

Figure 1 - results of execution 2

To explain the meaning of these formulae, consider the first statement

```
holds(missing(bat),2,1)
```

This is cast in the language of “situation calculus” ([40]). The word “holds” means “our system believes”. In formalisms dealing with situation calculus, and theories of actions, such a formula is of the form

```
holds(fluent, time_period, truth_value)
```

A *fluent* is a time varying variable, and an adequate discussion would unduly complicate this paper. Nonetheless, the “variable” we have chosen to represent the fact that this bat is missing is *missing(bat)*. (It is completely circumstantial that there are parenthesis with an argument, *bat*. We could have just used the text *missing* without any arguments to represent the missing bat. However, since we will later have another object which is missing, we chose this approach.)

Our program execution will occur over a span of time. That time will be divided into distinct *time periods*. The length of these periods is not a concern. The key point is that we have separately identifiable time periods, as in “the first time period”, “the second time period”, etc. Our formula states that we are stating something about time period #2.

The final argument of our formula is a *truth\_value*, which will be either “1” (true), or “0” (false.) The formula we are considering states that at time period 2, our system believes that it is true that the bat is missing.

On the other hand, the formula

```
holds(missing(bat),2,0)
```

means that at time period 2, our system believes that it is false that the bat is missing.

Looking at the remainder of figure 1, it further states that our system believes at time period 2 that Marshall, Ben, and Gwen are suspects in the stealing of the bat. This is in complete agreement with our intuition. We do not have knowledge of any other persons having opportunity to steal the bat.

### Execution 3

Our third execution is when a subsequent practice takes place at time value 4. The persons present at this practice are Jill, Marshall, Ben, Gwen, and Erica. It is important to note that Erica is present at this practice and was not present at the earlier practice. The significant event that happens at this practice is that Gwen's glove is missing. Figure 2 shows us the pertinent facts and conclusions.

```
holds(missing(bat),4,1)
holds(missing(glove),4,1)

holds(suspect(marshall,bat),4,1)
holds(suspect(ben,bat) ,4,1)
holds(suspect(gwen,bat),4,1)

holds(suspect(marshall,glove),4,1)
holds(suspect(ben,glove),4,1)
holds(suspect(jill,glove),4,1)
holds(suspect(eric,glove),4,1)
```

Figure 2 - results of execution 3

A little explaining is in order, before we explain the significance of these results. You will notice that all these formulae have time period 4. You will notice in particular that “the system believes that it is true at time period 4 that the bat is missing.” Figure 1 had a very similar formula, stating that fact as of time period 2. Neither of these formula state WHEN the bat was stolen, but merely reflects the system's beliefs as of each of these time periods. This is very much akin to saying “when I woke up this morning, I thought it was going to rain, but by the time I ate lunch, it was apparent that it would not rain.” Humans readily *introspect* about their beliefs, and their beliefs at one time may be counter to their beliefs at another time. Although, in this case, the system's beliefs at time period 2 and time period 4 about the bat missing are consistent with each other.

Let us now discuss figure 2. Immediately, we notice that the bat as well as the glove are missing. Note that we now have two groups of suspects. We have one group of suspects for stealing the bat, and another group of suspects for stealing the glove. Notice that both Ben and Marshall are suspects for the theft of both items. This is because both were present at each practice and neither owns either object. Also Gwen is a suspect only in the theft of the bat. That is because even though she was present at both practices, she cannot be a suspect for stealing the glove since she owns it. Similar reasoning applies to Jill and the bat. It is interesting to note that Erica is a suspect for stealing the glove and not for stealing the bat. This is because she was absent from the first practice, and therefore, could not have stolen the bat.

### Execution 4

For our final execution, Marshall is caught in possession of the glove (time value 5). He is therefore presumed to have stolen the glove, and further presumed to be the chief suspect in stealing the bat. Our primary conclusions are illustrated in figure 3.

```
holds(stolen(marshall,glove),5,1)
holds(suspect(marshall,bat),5,1)
```

Figure 3 - results of execution 4

This figure focuses only on the primary conclusions: that we conclude that Marshall stole the glove, and that we conclude that he is the main suspect (and the only main suspect) in the stealing of the bat. In reality, our program tells us all its beliefs. What is missing from this figure is that the former suspects for stealing the bat (namely, Ben and Gwen) are STILL suspects. If something happened such that Marshall was no longer considered the chief suspect, then Ben and Gwen would resurface as primary suspects.

Figure 4 gives more detail of the results of execution 4. It is easy to see that we can introspect about the system's beliefs. In particular, we can see that certain persons were suspects at one time, and are no longer suspects. Further, as mentioned in the preceding paragraph, if circumstances changed such that Marshall was no longer a suspect (such as, he has an air tight alibi), then the earlier suspects would re-emerge as suspects again.

```
holds(missing(bat),2,1)           // At time=2, the bat is missing, Ben
holds(suspect(marshall,bat),2,1) // Marshall, and Gwen are suspects
holds(suspect(ben,bat),2,1)
holds(suspect(gwen,bat),2,1)

holds(missing(bat),3,1)           // no change at time=3
holds(suspect(marshall,bat),3,1)
holds(suspect(ben,bat),3,1)
holds(suspect(gwen,bat),3,1)

holds(missing(bat),4,1)           //same as before, PLUS glove is
holds(missing(glove),4,1)         //missing, has its own suspects
holds(suspect(marshall,bat),4,1)
holds(suspect(ben,bat),4,1)
holds(suspect(gwen,bat),4,1)
holds(suspect(marshall,glove),4,1)
holds(suspect(ben,glove),4,1)
holds(suspect(jill,glove),4,1)
holds(suspect(eric,glove),4,1)

holds(stolen(marshall,glove),5,1) // Marshall caught stealing the glove
holds(suspect(marshall,bat),5,1) // and is the chief suspect for the bat
```

Figure 4 - more details of the result of execution 4

### Future Work

We have seen the ability of the program to reason with available, incomplete information. It correctly models our intuition. However, there are three very significant avenues by which this software could be enhanced. First, we could more closely employ an already well established theory of actions. Following this theory more closely, our actions could be more complicated. In addition, our actions could have prerequisites, and consequences. Certain actions could happen in parallel, and other actions could be mutually exclusive. We could predict the consequences of actions, and we could predict future actions.

Another significant enhancement would be to follow the tri-axis of police investigations. That is, that suspects should have the *means*, *motive*, and *opportunity*. In our scenario here, we ignored the first two (means and motive), and we trivialized the latter (opportunity.) In our case, we considered that those who were present at practice had opportunity. What if someone was at practice, but was in the concession stand the entire time (meaning that they were nowhere near the bat)? Or, what about the opportunity a car rider may have had? (That is, someone who rode in the car with Jill and her bat, but who did not attend practice.)

A final enhancement would be to apply this software to solve other mysteries. This pursuit would highlight other considerations. Further, the overlap between scenarios may identify opportunities for more general approaches.

### BIBLIOGRAPHY

- [1] M. Balduccini, J. Galloway, M. Gelfond. (2001, September). Diagnosing physical systems in A-Prolog. In *Proceedings of the 6th international conference on logic programming and nonmonotonic reasoning*, 213-225.
- [2] M. Balduccini, M. Barry, M. Gelfond, M. Nogueira, R. Watson. (2001). An A-Prolog decision support system for the space shuttle. *Lecture Notes in Computer Science - Proceedings of Practical Aspects of Declarative Languages '01*, pp. 169-183.
- [3] M. Balduccini, M. Gelfond, M. Nogueira, R. Watson. (2001, September). The USA-Advisor: A case study in answer set planning. In *Proceedings of the 6th international conference on logic programming and nonmonotonic reasoning*, 439-442.
- [4] M. Balduccini, M. Gelfond, M. Nogueira. (2000, September). A-Prolog as a tool for declarative programming. In *Proceedings of the 12th international conference on software engineering and knowledge engineering*.
- [5] C. Baral, M. Gelfond. (2000). Reasoning agents in dynamic domains. In J. Minker (Ed.), *Logic based artificial intelligence*, Kluwer.
- [6] C. Baral, M. Gelfond, A. Provetti. (1997) Reasoning about actions: laws, observations and hypotheses. *Journal of logic programming*. 31, 201-244.
- [7] C. Baral, M. Gelfond. (1997) Reasoning about effects of concurrent actions. *Journal of logic programming*, 31, 85—118.
- [8] C. Baral, M. Gelfond. (1994). Logic programming and knowledge representation (Survey paper). *Journal of logic programming*, 19, 73-148.

- [9] T. Bickmore, J. Cassell. (2001, March). Relational agents: a model and implementation of building user trust. In *Proceedings of the SIGCHI conference on human factors in computing systems*.
- [10] S. Breban, J. Vassileva. (2002, July). Trust and reputation: A coalition formation mechanism based on inter-agent trust relationships. In *Proceedings of the first international joint conference on autonomous agents and multi-agent systems*.
- [11] D. Buller, J. Burgoon. (1996). Interpersonal deception theory. *Communication Theory*, 6, 203-242.
- [12] J. Burgoon, J. Bonito, K. Kam. (In press). Communication and trust under face-to-face and mediated conditions: implications for leading from a distance. In S. Weisband & L. Atwater (Eds.), *Leadership at a distance*. Mahwah, NJ: LEA.
- [13] J. Burgoon, D. Buller, K. Floyd, R. Viprakasit. (1999). Does participation affect deception success? A test of the inter-activity effect. (Manuscript submitted for publication).
- [14] J. Burgoon, D. Buller, A. Ebesu, P. Rockwell. (1994). Interpersonal deception: V. Accuracy in deception detection. *Communication Monographs*, 61, 303-325.
- [15] W. Chen, T. Swift, D. Warren. (1995). Efficient top-down computation of queries under the well-founded semantics. *Journal of Logic Programming*, 24(3):161-201.
- [16] P. Cholewinski, W. Marek, M. Trzuscynski. (1996). Default reasoning system DeReS. In *International Conference on Principles of Knowledge Representation and Reasoning*, pp. 518-528. Morgan Kaufman.
- [17] K. Clark. (1978). Negation as failure. In Gallaire, H., & Minker, J., editors, *Logic and Data Bases* (pp. 293-322). Plenum Press, New York.
- [18] A. Colmerauer, H. Kanoui, R. Pasero, P. Russel. (1973). Un systemem de communication homme-machine en francais. Technical report, Groupe de Intelligence Artificielle Universitae de Aix-Marseille.
- [19] E. Erdem, V. Lifschitz, M. Wong. (2000). Wire routing and satisfiability planning. *Proceedings of CL-2000*, pp. 822-836.
- [20] M. Fitting. (1985). A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4):295-312.
- [21] M. Gelfond, N. Leone. (2002). Logic programming and knowledge representation - an A-prolog perspective. *Artificial intelligence* (to appear).
- [22] M. Gelfond, M. Balduccini. (2002). Diagnostic reasoning with A-Prolog. (accepted for publication in *Theory and practice of logic programming*).
- [23] M. Gelfond, T.C. Son. (1998). Reasoning with prioritized defaults. In J. Dix, L.M. Pereira, & T. Przymusinski (Eds.), *Lecture notes in artificial intelligence*, 1471, pp. 164-224, 1998.
- [24] M. Gelfond, H. Przymusinska. (1996). Towards a theory of elaboration tolerance: logic programming approach. *Journal on software and knowledge engineering*, 6, 89-112.
- [25] M. Gelfond. (1994). Logic programming and reasoning with incomplete information. *Annals of mathematics and artificial intelligence*, 12, 89-116.
- [26] M. Gelfond, V. Lifschitz. (1993). Representing actions and change by logic programs. *Journal of logic programming*, 17, 301-323.
- [27] M. Gelfond, H. Przymusinska. (1993). Reasoning in open domains. In L. Pereira & a. Nerode (Eds.), *Logic programming and nonmonotonic reasoning*, (pp. 397-413). Cambridge, MA: MIT Press.

- [28] M. Gelfond, V. Lifschitz, A. Rabinov. (1991). What are the limitations of the situation calculus? In S. Boyer (Ed.), *Automated reasoning, essays in honor of woody bledsoe* (pp. 167-181). Kluwer Academic Publishers.
- [29] M. Gelfond, V. Lifschitz. (1991). Classical negation in logic programs and disjunctive databases. *New generation computing*, 9, 365-385.
- [30] M. Gelfond, V. Lifschitz. (1988). The stable model semantics for logic programming. In *Proceedings of ICLP*, 88, 1070-1080.
- [31] P. Johnson, S. Grazioli, K. Jamal, R. Berryman. Detecting deception: Adversarial problem solving in a low base-rate world. *Cognitive Science*.
- [32] S. Jones, M. Wilikens, P. Morris, M. Masera. (2000, December). Trust requirements in e-business. *Communications of the ACM*, 43 .
- [33] C. Koch, N. Leone. (1999). Stable model checking made easy. In *Proceedings of IJCAI '99*.
- [34] M. Kollingbaum, T. Norman. (2002, July). Trust and reputation: Supervised interaction: creating a web of trust for contracting agents in electronic environments. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*.
- [35] R. Kowalski. (1979). *Logic for problem solving*. North-Holland.
- [36] R. Kowalski. (1974). Predicate logic as a programming language. *Information Processing* 74, (pp. 569-574).
- [37] K. Kunen. (1987). Negation in logic programming. *Journal of Logic Programming*, 4(4):289-308.
- [38] V. Lifschitz. (1996). Foundations of logic programming. In Brewka, G., editor, *Principles of Knowledge Representation*, pp. 69-128. CSLI Publications.
- [39] W. Marek, M. Truszczynski. (1999). Stable models and an alternative logic programming paradigm. In *The Logic Programming Paradigm: a 25-year perspective*, pp. 375-398, Springer-Verlag.
- [40] J. McCarthy, P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B. & Michie, D., editors, *Machine Intelligence*, 4:463-502. Edinburgh University Press, Edinburgh.
- [41] C. Michael, A. Ghosh. (2002, August). Simple, state-based approaches to program-based anomaly detection. *ACM Transactions on Information and System Security* , 5 .
- [42] R. Moore. (1985). Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75-94.
- [43] L. Mui, M. Mohtashemi, A. Halberstadt. (2002, July). Trust and reputation: Notions of reputation in multi-agents systems: a review. *Proceedings of the first international joint conference on autonomous agents and multi-agent systems*.
- [44] I. Niemela, P. Simons. (2000). Extending the Smodels system with cardinality and weight constraints. In Miner, J., editor, *Logic Based AI*, pp. 491-522. Kluwer.
- [45] I. Niemela, P. Simons. (1997). Smodels - an implementation of the stable model and well-founded semantics for normal logic programs. In *Proceedings of the 4<sup>th</sup> international conference on logic programming and non-monotonic reasoning*, pp. 420-429.
- [46] I. Niemela. (1998). Logic programming with stable model semantics as a constraint programming paradigm. In *Proc of the Workshop on Computational Aspects of Nonmonotonic Reasoning*, pp. 72-79, Trento, Italy.
- [47] J. Pujol, R. Sangüesa, J. Delgado. (2002, July). Group and organizational dynamics: Extracting reputation in multi-agent systems by means of social network topology. In

*Proceedings of the first international joint conference on autonomous agents and multiagent systems.*

- [48] R. Reiter. (1980). A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81-132.
- [49] J. Sabater, C. Sierra. (2002, July). Group and organizational dynamics: Reputation and social network analysis in multi-agent systems. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems.*
- [50] J. Sabater, C. Sierra. (2001, May). Regret: reputation in gregarious societies. In *Proceedings of the fifth international conference on autonomous agents.*
- [51] S. Sen, N. Sajja. (2002, July). Trust and reputation: Robustness of reputation-based trust: boolean case. In *Proceedings of the first international joint conference on autonomous agents and multiagent systems.*
- [52] A. Van Gelder, K. Ross, J. Schlipf. (1991). The well-founded semantics for general logic programs. *Journal of ACM*, 38(3):620-650.
- [53] M. Warren, W. Hutchinson. (2001, June). Network security and intrusion detection: Deception, a tool and curse for security management. In *Proceedings of the 16th international conference on Information security: Trusted information: the new decade challenge.*
- [54] C. White, J. Burgoon. (2001). Adaptation and communicative design: Patterns of interaction in truthful and deceptive conversations. *Human Communication Research*, 27, 9-37.