Book Title
Book Editors
IOS Press, 2003

Assumption Based Peg Unification for Crime Scenario Modelling [1]

Jeroen Keppens [a] Burkhard Schafer [b]
[a] *King's College London, Dept. of Computer Science, Strand, London WC2R 2LS, UK*
[b] *The University of Edinburgh, School of Law, Edinburgh EH8 9YL, UK*

Abstract. An important cause of miscarriages of justice is the failure of crime investigators and lawyers to consider important plausible explanation for the available evidence. Recent research has explored the development of decision support systems that (i) assist human crime investigators by proposing plausible crime scenarios explaining given evidence, and (ii) provide the means to analyse such scenarios. While such approaches can generate useful explanations, they are inevitably restricted by the limitations of formal abductive inference mechanisms. Building on work presented previously at this venue, this paper characterises an important class of scenarios, containing "alternative suspects" or "hidden objects", which cannot be synthesised robustly using conventional abductive inference mechanisms. The work is then extended further by proposing a novel inference mechanism that enables the generation of such scenarios.

Keywords. crime investigation, truth maintenance, abduction, peg unification

1. Introduction

In a previous Jurix paper [9], we have discussed largely informally the logic that governs hypothetical reasoning with evidence in a criminal law context. This paper builds on the results developed there, and addresses several points made by the then referees, in particular the desirability of a fully explicit formal account of that we had proposed. Hypothetical reasoning plays an important part in crime investigation and also in criminal trials. At the same time, difficulties with hypothetical reasoning strategies during crime investigations have been identified as a common source for miscarriages of justice [6].

While our focuses in the past has been primarily on the use of hypothetical reasoning during crime investigations, the emphasis in this paper will be on hypothesis generation in the ensuing court room setting. The reason for this shift in emphasis is that our proposal is based on the "abductive diagnosis" paradigm for physical systems diagnosis [1]. Physical systems differ from crime investigation in that most, if not all, component objects of a physical system are known. Typical examples of unknown components in physical systems include leaks in a hydraulic system or short-circuits in an electrical systems. Existing abductive diagnosers can model the behaviour of such unknown components quite easily because they influence the overall physical system at only one location in its topology. Conversely, unknown persons and objects constitute an important feature in crime scenarios and their existence can affect many aspects of the crime scenario: When a body is first found under suspicious circumstances,

[1] Both authors have been supported by EPSRC grant GR/S63267.

the possible causes for the death are considerably broader than the range of possibilities that caused the breakdown of a piece of machinery. Some of the explanations will be "surprising" in the sense that they were impossible to anticipate.

This paper shows how existing diagnostic systems can be amended to incorporate this type of reasoning that is typical for legal proceedings. This paper therefore aims to demonstrate that existing abductive reasoning techniques can not adequately deal with this aspect of reasoning about crime scenarios and to propose a basic extension of our ATMS based decision support system to tackle this restriction. The remainder of this paper is organised as follows: Section 2 presents an overview of the existing decision support system that this paper is based on, Section 3 characterises the class of crime scenarios that are difficult to synthesise with the present approach, Section 4 introduces a novel assumption based peg unification technique that addresses the latter issue, and Section 5 concludes the paper.

2. Background

Although there is no consensus within the wider community of crime investigators as to what constitutes an effective methodology for evidence discovery, crime investigation and forensic argumentation in court, forensic scientists, statisticians and philosophers increasingly advocate the adoption of an abductive reasoning paradigm [2,7,16]. Decision support systems (DSS) aimed at assisting human investigators in aspects of abductive reasoning are inevitably restricted by the limitations of formal inference mechanisms. To alleviate such deficiencies, the approach discussed in [10] combines abduction in the narrow sense (i.e. inverse modus ponens) with model based reasoning techniques such as assumption based truth maintenance. To render the paper sufficiently self-contained, this section will briefly discuss this background material.

2.1. Assumption based truth maintenance

An assumption based truth maintenance system (ATMS) can assist a problem solver by maintaining how each piece of inferred information depends on presumed information and facts, and how inconsistencies arise. This section summarises the functionality of an ATMS as it is employed in this work. For more details, the reader is referred to the original papers [3].

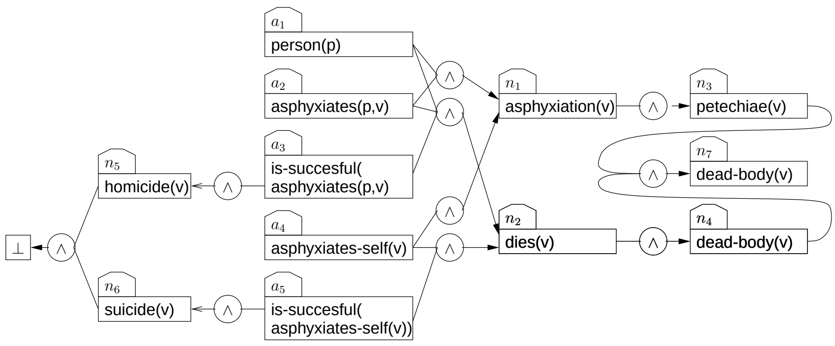

Figure 1. Graphical representation of a sample ATMS

In an ATMS, each piece of information of relevance to the problem solver is stored as a *node*. Some pieces of information are not known to be true and cannot be inferred easily from other pieces of information. In the ATMS, such information is represented by a special type of node, called *assumption*. Our model based DSS employs two types of assumptions: (i) conjectures, which cannot be deemed true or false a priori (e.g. "the victim asphyxiated himself"), and (ii) default assumptions, which should be presumed true unless their truth leads to inconsistencies (e.g. "the cause of death proposed by the medical examiner is correct") [15].

Inferences between pieces of information are maintained within the ATMS as inferences between the corresponding nodes. In its extended form, the ATMS can take inferences, called *justifications* of the form $n_i \wedge \ldots \wedge n_j \wedge \neg n_k \wedge \ldots \wedge \neg n_l \rightarrow n_m$, where $n_i, \ldots, n_j, n_k, \ldots, n_l, n_m$ are nodes (and assumptions) representing information that the problem solver is interested in. Figure 1 shows a graphical representation of an ATMS with seven regular nodes $n_1, \ldots, n_7$, five assumption nodes $a_1, \ldots, a_5$, and the following justifications: $a_1 \wedge a_2 \rightarrow n_1$; $a_4 \rightarrow n_1$; $a_1 \wedge a_2 \wedge a_3 \rightarrow n_2$; $a_4 \wedge a_5 \rightarrow n_2$; $n_1 \rightarrow n_3$; $n_2 \rightarrow n_4$; $a_3 \rightarrow n_5$; $a_5 \rightarrow n_6$; and $n_3 \wedge n_4 \rightarrow n_7$. This ATMS provides a very simple description of some of the situations and events that might explain the combined evidence of a dead body of a person v and petechiae on the eyes of v[1].

An ATMS can also take justifications, called *nogoods*, that have lead to an inconsistency. Nogoods are justifications of the form $n_i \wedge \ldots \wedge n_j \wedge \neg n_k \wedge \ldots \wedge \neg n_l \rightarrow \perp$. As such, they impose constraints upon the combinations of assumptions that constitute valid explanation for known observations. The latter nogood implies that at least one of the statements in $\{n_i, \ldots, n_j, \neg n_k, \ldots, \neg n_l\}$ must be false. In the sample ATMS of Figure 1, there is one nogood $n_5 \wedge n_6 \rightarrow \perp$, which indicates that the death of a victim can not be both homicide and suicide.

An *environment* in an ATMS is a set $\{a_1, \ldots, a_m\}$ of assumptions. Each environment describes the possible worlds in which all its assumptions are true. For each of its nodes, an ATMS maintains a description, called a label, denoting the environments that entail it. Because it would be inefficient to store all environments that entail a particular node, a label $L(n)$ of a node n is a set of environments, such that $L(n)$ is *minimal* ($L(n)$ does not contain any supersets of an environment that entails n), *complete* ($L(n)$ contains each environment that entails n or a subset thereof), *sound* ($L(n)$ contains no environment that does not entail n) and *consistent* ($L(n)$ does not contain an environment that entails the nogood node). In the sample ATMS of Figure 1, for instance: $L(n_7) = \{\{a_1, a_2, a_3\}, \{a_4, a_5\}\}$.

Given a set of nodes, assumption nodes, justifications and nogoods, an ATMS can be queried to provide useful information to an abductive problem solver. For example, an ATMS can determine whether a given environment is consistent (i.e. corresponds to a possible world), whether a given environment entails a given node, and which (consistent) environments are sufficient to explain a given node.

2.2. *A model based decision support system for reasoning about crime scenarios*

The model based decision support system presented in [10] synthesises an ATMS representing a *space of plausible scenarios* from a given set of available evidence, a given set of facts and a knowledge base of causal rules and constraints upon the consistency of scenarios. The approach follows roughly the following steps:

1. A new ATMS is initialised with one node for each piece of evidence and each fact. Each fact node is justified by the empty environment (in other words, it is deemed true in all possible worlds).

[1]For a substantially extended version of this example, the reader is referred to [10]

2. Through the application of inverse modus ponens of the causal rules in the knowledge base, the ATMS is filled with plausible explanations of the available evidence. In this step, the plausible scenarios that may have caused the evidence are built up and composed within the ATMS.
3. Through the application of modus ponens of the causal rules, the ATMS is filled with plausible consequences of the causes generated in Step 2. In this step, collectible evidence and the assumptions that such evidence depends upon is added to the ATMS.
4. Finally, the constraints of the knowledge base are instantiated in the ATMS in the form of nogoods.

Once this scenario space has been constructed, it can be analysed to provide decision support information by using standard ATMS operations to provide answers to a range of queries, including the following ones:

- Which scenarios can explain the evidence? Scenarios explaining available evidence can be identified by computing the complete and sound set of minimal and consistent environments in the ATMS that entail the available evidence, and then tracing the justifications from the assumptions in the environments and the facts in the ATMS to the given evidence. Our DSS provides the means to describe visually or textually such scenarios.
- Is a given hypothesis about the crime scenario underlying the evidence supported by the available evidence? Any hypothesis is supported by the evidence provided a consistent environment exists in the ATMS that entails both the hypothesis and the evidence.
- What pieces of evidence may be expected to be observed if a certain scenario/hypothesis were true? These are discovered by searching for evidence that logically follows from the/an environment entailing the given scenario/hypothesis, possibly extended with default assumptions.

3. Abductive inference mechanisms for abductive inference

While the DSS described in the previous section can generate a wide range of plausible scenario to explain a given set evidence, there are certain types of plausible scenarios it is not well suited to synthesise. This drawback is not due to any limitation of abductive reasoning, which is accepted as an appropriate methodology to infer information in reasoning about crime scenarios [16]. Instead, it is due to the limitation of our abductive reasoning algorithm.

The term "abduction" is used in philosophy and artificial intelligence to refer to relatively distinct concepts [17]. In philosophy, "abduction" is associated with the formulation of hypotheses or explanations[2]. In what follows, this concept is named *abductive inference*. In artificial intelligence (AI), "abduction" characterises a specific class of algorithms that implement inference mechanisms enabling knowledge based systems to compute hypotheses or explanations. Hence, the AI concept of "abduction" refers to *abductive inference mechanisms*. While abductive inference mechanisms aim to enable a machine to perform abductive inference, their repertoire of reasoning tends to be substantially narrower than that discussed in philosophy textbooks on abduction. This stems primarily from the fact that computer algorithms need to be substantially more precise than typical analyses of human reasoning.

An example may help illustrate these issues. Assume the body of a man is found in his home. The cause of death is identified as a single shot in the head and blood splatter evidence indicates the man was shot with a small calibre handgun at very short range. No hand gun has

[2]Some authors also include the evaluation of hypotheses, but this topic is beyond the scope of this paper.

been retrieved from the scene. Two witnesses claim to have information that may be relevant to the case. The first heard a male voice shouting threats followed by a single shot. The second saw a suspicious looking person running away from the home of the victim.

Abductive inference may yield several hypothetical causes that explain the pieces of evidence. The victim may have been shot by another person, he may have committed suicide or his death may have been an accident. The absence of the weapon that killed the victim at the crime scene suggests that it was either taken away by a person (such as the victim's killer in the homicide scenario) or that it fell in an awkward location at the crime scene (after a possible suicidal or accidental shooting). The male voice heard by the first witness may have been that of a, say, paranoid delusional victim, or another person, say the killer, threatening the victim. Finally, the suspicious person fleeing the scene may be the victim's killer or a person unrelated to the killing (e.g. the thief of the victim's gun).

The distinct plausible causes for the individual pieces of evidence, which have been identified through causal reasoning, do not yet constitute hypothetical scenarios. These can be identified by composing the plausible causes into coherent combinations of causes that explain all the available evidence. Two (of many) such scenarios in the ongoing example are:

- *Scenario 1*: The perpetrator threatened the victim at the victim's home, killed him, and ran away with the murder weapon, and
- *Scenario 2*: In a paranoid delusion, the victim shouted threats to kill himself and then shot himself in the head; after hearing this, a second person entered the victim's home, saw the dead body, took the weapon, and fled the scene.

Note that this abductive inference involves causal reasoning to identify the situations and events that may have generated the pieces of evidence, and a process that combines the situations and events into coherent hypothetical scenarios. Pierce referred to the latter process as "colligation" and deemed it an inherent feature of abductive (and inductive) inference [14]. Abductive inference mechanisms, however, tend to focus on the causal reasoning task and provide little means for colligation. The ATMS based technique discussed in Section 2 can infer a complete set of minimal and consistent combinations of the causes of the available set of evidence (i.e. solutions to Konolige's model of abduction [12]), but it is not well suited to discern the important possible links between the causes involving unidentified entities.

The latter issue can be explained more precisely by means of the ongoing example. Consider the following three causal rules, which may be part of a knowledge base that aims to aid in the synthesis of plausible scenarios, such as Scenario 1:

$$\begin{gathered}\text{person}(P) \wedge \text{victim}(V) \wedge \text{scene}(S) \wedge \text{at}(P,S) \wedge \text{near}(W,S) \wedge \\ \text{threatened}(P,V) \wedge \text{near}(W,P) \rightarrow \text{witness}(W, \text{threat}(\text{near}(S)))\end{gathered} \tag{1}$$

$$\begin{aligned}\text{person}(P) \wedge \text{scene}(L) \wedge \text{shot}(S,P,V,L) \wedge \text{gun}(S,G) \wedge \text{range}(S,R,) \rightarrow \\ \text{evidence}(\text{shot-at-range}(V,R)) \wedge \text{evidence}(\text{shot-with}(V,G))\end{aligned} \tag{2}$$

$$\text{person}(P) \wedge \text{scene}(S) \wedge \text{took}(P,G) \rightarrow \neg\text{evidence}(\text{recover}(G,S)) \tag{3}$$

Rules (1), (2) and (3) respectively provide explanations for the testimony of the first witness, the shot at short range with a handgun, and the absence of the weapon that killed the victim at the crime scene. In each of these rules, P refers to *an* unknown person. Therefore, P is considered to be existentially quantified.

While these rules can be employed by the abductive inference mechanism of Section 2 to produce useful explanations, the semantics of first order predicate logic can not distinguish between subtle but important variations of Scenario 1. The closest scenario that our abductive inference mechanism can produce to Scenario 1 is one where three different instances of P refer to the person threatening the victim, the person killing the victim and the person taking the gun from the scene. The very plausible case that the three persons are one and the same is not differentiated from its alternatives, even though this is a crucial distinction.

Situations such as the one in the example arise frequently in the crime investigation and legal reasoning domains. Because evidence usually relates to aspects or features of persons and objects, and the events and situations in which they occur, it is not always possible to uniquely identify these entities from the start of the investigation, or even during the subsequent court proceedings (as argued in the introduction). Nevertheless, the possible number of unidentified entities relevant to the crime must be considered and taken into account. The remainder of this paper proposes an approach to extend our abductive reasoning mechanism accordingly, thereby expanding the applicability of abductive inference mechanisms.

4. Assumption based peg unification

The task of identifying different references to the same entity is known as *coreference resolution* in computational linguistics. In the analysis of a discourse, it is important that references to the same entity are correctly associated with one another because each of the expressions that contains one of these references may add some information about the entity in question. For example, in the sentence "Every farmer who owns *a donkey*, beats *it*." "a donkey" and "it" refer to the same entity. The first half of the sentence conveys that the entities of interest are all donkeys owned by farmers. The second half of the sentence communicates that the entities of interest are beaten. Thus, the sentence as a whole imparts the knowledge that all donkeys owned by farmers are beaten.

A wide range of techniques has been devised to perform coreference resolution tasks, such as the one illustrated in the example. The vast majority of these techniques specialise in examining texts for discourse analysis or information extraction. An important property of the existing approaches is that they tend to consider only a single possible solution at any one time, while the present problem domain requires a method that can represent and reason with multiple possible worlds simultaneously.

4.1. Pegs

The objective of this work is to identify possible references to the same unknown or partially specified entities in the scenario space. In order to correctly distinguish such entities, the notion of *pegs* is adopted from the literature on coreference resolution [8,13]. Pegs refer to a specific entity whose exact identity remains unknown (or partially specified). In this paper, each peg is identified by an expression of the form $\_n$, where n is a non-negative natural number. At the start of the scenario space generation algorithm $n = 0$, and n is incremented by 1 after each generation of a new peg. As such, each new peg is identified uniquely.

New pegs may be introduced into the scenario space during the instantiation of causal rules of the form `if` $\{A_n\}$ `assuming` $\{A_s\}$ `then` $\{c\}$, where A_n is a set of antecedent predicates, A_s is a set of assumption predicates and c is a consequent predicate. Whenever a rule, whose antecedent or assumption predicates contain variables that do not occur in the consequent sentence, is applied during the inverse modus ponens phase of the scenario space generation algorithm (i.e. step 2), then those variables are instantiated by pegs. Consider, for instance, applying inverse modus ponens on rule `if` $\{\text{scene}(S)\}$ `assuming` $\{\text{person}(P), \text{took}(P, G)\}$ `then` $\{\neg\text{evidence}(\text{recover}(G, S))\}$ given the piece of evidence: $\neg\text{evidence}(\text{recover}(\text{handgun}, \text{home}(\text{victim}))\}$. The required substitution $\{G/\text{handgun}, S/\text{home}(\text{victim})\}$ does not provide an instance for P. Here, P refers to an unknown entity and it is therefore substituted by a peg, say, _0. Therefore, the assumptions $\text{person}(\_0)$ and $\text{took}(\_0, \text{handgun})$ are added to the scenario space.

Similarly, pegs may also be introduced during the modus ponens phase of the scenario generation algorithm (i.e. step 3). In this case pegs are introduced when a rule whose consequent predicates contain variables that do not occur in the antecedent or assumption sentences, is applied.

4.2. *Peg unification*

Because a peg refers to an unknown entity, it can be treated as a constant that uniquely identifies a that entity, or it can be unified with a ground term, including another peg or terms containing other pegs. In the latter case, the unification is possible if it is hypothesised that the entity represented by the peg and the entity represented by the term unified to the peg are the same one. This hypothesis must therefore be made explicit by means of an assumption whenever an inference is made that depends on the unification of a peg and a ground term. In the remainder of this paper, such assumptions are referred to as *peg unification assumptions*.

In this paper, each peg unification assumption takes the form $\text{bind}(\_n, t)$, where $\_n$ is a peg and t is a ground term (which may include a peg). A peg unification assumption $\text{bind}(\_n, t)$ is added to the scenario space for each pair of predicates that can be matched using a substitution that contains a mapping of the form $\_n/t$.

The binding relation implied by these assumptions is transitive. Therefore, peg unification can not only be assumed, but also be entailed by other peg unification assumptions. This knowledge is represented explicitly in the scenario space: for each pair of peg unification assumptions $\text{bind}(\_i, t_1(\ldots, \_j, \ldots))$ and $\text{bind}(\_j, t_2(\ldots, \_k, \ldots))$, the following new justification is added to the emerging scenario space:

$$\text{bind}(\_i, t_1(\ldots, \_j, \ldots)) \wedge \text{bind}(\_j, t_2(\ldots, \_k, \ldots)) \rightarrow \text{bind}(\_i, t_1(\ldots, t_2(\ldots, \_k, \ldots), \ldots))$$

4.3. *Scenario space generation*

Peg unification affects the way the scenario space generation algorithm operates, when applying causal rules of the form shown in 4.1 in steps 2 and 3 of the algorithm and constraints of the form `inconsistent {` I `}`, where I is a set of sentences that should be deemed nogood, in step 4 of the algorithm. In the extended approach, each application of a causal rule or constraint requires the following operations:

1. *Unify* the relevant sentences (i.e. the consequent of the causal rule during inverse modus ponens, the antecedents of the causal rule during modus ponens, or the inconsistent sentences of the constraint) with nodes in the emerging scenario space, and return the substitution σ required to achieve the unification.
2. *Record* each binding that unifies a peg with a term in the scenario space and a newly created set A_p. That is, for each binding $\_n/t \in \sigma$ or $t/\_n \in \sigma$, where $\_n$ is a peg and t is a term that the $\_n$ is unified with, the peg unification assumption $\text{bind}(\_n, t)$ is added to A_p.

3. *Instantiate* the remaining sentences (i.e. the antecedents and assumptions during inverse modus ponens or the assumptions and consequent during modus ponens) by applying the substitution σ and add those that do not already exist in the scenario space as new nodes.
4. *Generate* a justification of the form $\left[\wedge_{(a \in A_n \cup A_s)} \sigma a\right] \wedge \left[\wedge_{(u \in A_p)} u\right] \rightarrow \sigma c$ in case of an application of a causal rule, or a nogood of the form $\left[\wedge_{(a \in I)} \sigma a\right] \wedge \left[\wedge_{(u \in A_p)} u\right] \rightarrow \perp$ in case of an application of a constraint.

Consider, for instance, the following rule expressing the constraint that a person should be either male or female: $\text{person}(P) \wedge \text{gender}(P, \text{male}) \wedge \text{gender}(P, \text{female}) \rightarrow \perp$ and let the scenario space contain to references to persons, i.e. $\text{person}(\_0)$ and $\text{person}(\_1)$. Naturally, $\_0$ and $\_1$ may refer to the same person. Therefore, the antecedents of the gender constraint can match the ground terms $\text{person}(\_0)$, $\text{gender}(\_0, \text{male})$, and $\text{gender}(\_1, \text{female})$ assuming the peg unification assumption $\text{bind}(\_0, \_1)$ is made. Therefore, a valid instantiation of the constraint is $\text{person}(\_0) \wedge \text{gender}(\_0, \text{male}) \wedge \text{gender}(\_1, \text{female}) \wedge \text{bind}(\_0, \_1) \rightarrow \perp$.

The last example has show how additional predicates can describe certain features of the entities represented by the pegs and how such predicates may impose constraints on the plausibility of peg unification assumptions. It is important to point out that these features attributed to unknown entities, as with other information stored in an ATMS, can be modelled as dependent upon conjecture. As such, the approach allows for different pieces of evidence regarding similar entities to be modelled to refer to: (i) correctly interpreted evidence of multiple entities, (ii) conflicting evidence regarding the same entity, or (iii) evidence that bears no relation to any entity in existence.

4.4. *Scenario extraction*

In [10], we argue that a scenario driven decision support system (DSS) for crime investigation should explain the available evidence by means of the simplest scenario that entail this evidence and one of the main hypotheses of interest. For example, in the investigation of a suspicious death, our DSS provides the simplest scenarios that explain homicide, suicide, accidental death and natural causes. In [10], the simplest scenarios are defined as those entailed by the environments with the smallest number of conjectures. However, all other things being equal, scenarios with fewer entities are normally deemed to be less complex. Therefore, peg unification assumption simplify scenarios, and for that reason they treated as default assumptions.

While the ATMS label propagation algorithm provides an efficient means to determine the minimal (or smallest) environments (of conjectures) that entail a given set of nodes (representing pieces of evidence), it does not provide a facility to determine the maximum number of default assumptions. However, the basic candidate generation algorithm of the General Diagnostic Engine (GDE) [5] can be employed to extend an environment E of conjectures with a consistent set of peg unification assumptions. The candidate generation algorithm can be applied as follows:

1. Let P be the set of all pegs that occur in predicates that logically follow from E. Let E_p be the set of all peg unification assumptions that contain pegs in P and no other pegs. In other words E_p is the set of all peg unification assumptions the E can be extended with.
2. Let N be the set of nogood environments that are subsets of $E_p \cup E$. Formally, $N = \{E_i \mid E_i \subseteq E_p \cup E, E_i \in L(\perp)\}$. Each environment in N is a combination of peg unification assumptions that is inconsistent. Therefore, one peg unification as-

sumption from each environment in N needs to be removed from E_p in order to make it consistent.

3. If N is empty, then return the singleton $L_E = \{E \cup E_p\}$ and end the algorithm. Otherwise, proceed to step 4.
4. Let N' be the set of all combinations of peg unification assumptions from E_p that are inconsistent with E. In other words, N' is the set constructed by removing all conjectures in E from each of the nogood environments in N. Or formally $N' = \{E'_i \mid E'_i = E_i - E_p, E_i \in N\}$
5. Generate the set C of GDE candidates from the nogood environments in N' as follows: (i) Assemble a set C of all the cross product sets of the nogood environments in N. Each cross product set in is a set that contains at least one assumption from each of the environments in N. (ii) Remove each cross product set from C that is a superset of another cross product set in C. Each candidate set in C is a minimal set of peg unification assumptions from E_p that is not consistent with E.
6. Let $L_E = \{E' \mid E' = E \cup E_p - E_c, E_c \in C\}$. In other words L_E is set of all supersets of E that contain all the peg unification assumptions in E_p minus a candidate set from C.

The set L_E returned by the above procedure contains all consistent extensions of E with relevant peg unification assumptions. To illustrate this approach, consider the graphical representation of a partial scenario space in Figure 2.

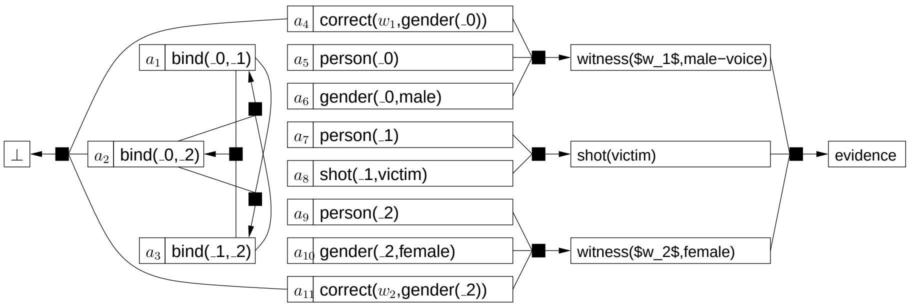

Figure 2. Sample scenario

In Figure 2, the node evidence, which represents all the available evidence, is supported by a single environment of conjectures $E = \{a_4, \ldots, a_{11}\}$. This environment can be extended with peg unification assumptions as follows: Step 1: The set of relevant pegs $P = \{\_0, \_1\}$ and hence, the set of relevant peg unification assumptions $E_p = \{\text{bind}(\_0, \_1), \text{bind}(\_1, \_2), \text{bind}(\_0, \_2)\}$; Step 2: The set of nogood environments $N = \{\{a_1, a_3, a_4, a_{11}\}, \{a_2, a_4, a_{11}\}\}$; Step 4: The combinations of peg unification assumptions inconsistent with E is $N' = \{\{a_1, a_3\}, \{a_2\}\}$; Step 5: The minimal candidates that can be generated from N' is $C = \{\{a_1, a_2\}, \{a_2, a_3\}\}$; Step 6: The set of valid extensions of E is $L_E = \{\{a_3\}, \{a_1\}\}$. Thus, the environments entailing the available evidence with minimal conjectures and maximal peg unification is: $\{\{a_3, a_4, \ldots, a_{11}\}, \{a_1, a_4, \ldots, a_{11}\}\}$.

5. Conclusions and future work

Building on the preliminary work presented in [9], this paper has characterised further the benefits of peg unification in abductive reasoning about crime scenarios and basic approach has been proposed. The novelty and research contribution of this approach is twofold. Firstly, the new representation formalism and inference mechanisms bring the benefits of peg unification from the computational linguistics and information extraction domains to the entirely new domains of reasoning about crime scenarios and evidence evaluation. As such, this work can enable abductive inference mechanisms in the latter domains to accomplish a broader range of abductive inference. Secondly, the peg unification approach is the first one to be integrated into an assumption based truth maintenance system (ATMS). While conventional peg unification mechanisms aim to find a single most likely set of coreferences, the use of the ATMS in peg unification enables the associated problem solver to reason about multiple plausible sets of coreferences that may be valid in different possible worlds.

As the approach presented herein is an initial proposal to incorporate peg unification into abductive synthesis of crime scenarios, a number of important issues remain to be addressed in future work. While the approach presented herein facilitates modelling certain types of abductive reasoning about crime scenarios, it does not address the broader problem of knowledge acquisition for this type of reasoning. Future work intends to tackle this issue by developing techniques aimed at extracting cause-and-effect knowledge from individual cases as they are encountered. Also, the important issue of time and space complexity of the approach proposed herein needs to be studied. Because ATMSs are constraint propagation mechanisms, they eliminate the need for search algorithms, such as backtracking and its variants. But that need not be the most efficient approach to identify consistent crime scenarios, and hence, the use of constraint satisfaction algorithms should be examined. Other future work will examine the integration of the peg unification approach into our Bayesian scenario space synthesis approach [11] and the enrichment of the existing knowledge representation formalism.

References

[1] L. Console and P. Torasso. A spectrum of logical definitions of model-based diagnosis. *Computational Intelligence*, 7(3):133–141, 1991.

[2] R. Cook, I. Evett, G. Jackson, P. Jones, and J. Lambert. A model for case assessment and interpretation. *Science and Justice*, 38:151–156, 1998.

[3] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[4] J. de Kleer. A comparison of ATMS and CSP techniques. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 290–296, 1989.

[5] J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32:97–130, 1987.

[6] D. Dixon. Police investigative procedures. In C. Walker and K. Starmer, editors, *Miscarriage of Justice. A Review of Justice in Error*, 1999.

[7] A. Jamieson. A rational approach to the principles and practice of crime scene investigation: I. principles. *Science & Justice*, 44(1):3–7, 2004.

[8] L. Karttunen. Discourse referents. *Syntax and Semantics*, 7:363–385, 1976.

[9] J. Keppens and B. Schafer. "murdered by persons unknown" - speculative reasoning in law and logic. In *Proceedings of the 17th Annual Conference on Legal Knowledge and Information Systems (JURIX)*, pages 41–51, 2004.

[10] J. Keppens and B. Schafer. Knowledge based crime scenario modelling. *Expert Systems With Applications*, 30(2), 2006.

[11] J. Keppens, Q. Shen, and B. Schafer. Probabilistic abductive computation of evidence collection strategies in crime investigation. In *Proceedings of the 10th Internation Conference on Artificial Intelligence and Law*, 2005.

[12] K. Konolige. Abduction versus closure in causal theories. *Artificial Intelligence*, 52:255–272, 1991.

[13] F. Landman. *Towards a theory of information. The status of partial objects in semantics*. Foris, Dordrecht, 1986.

[14] C. Peirce. *Collected Papers*. Harvard University Press, 1933. Edited by C. Hartshorne and P. Weiss.

[15] D. Poole. Explanation and prediction: An architecture for default and abductive reasoning. *Computational Intelligence*, 5(2):97–110, 1989.

[16] D. Schum. *The Evidential Foundations of Probabilistic Reasoning*. Northwestern University Press, 1994.

[17] P. Thagard and C. Shelley. Abductive reasoning: Logic, visual thinking, and coherence. In L. Dalla Chiara et al., editor, *Logic and scientific methods*, pages 413–427. Kluwer, 1997.