Journal of Applied Non-Classical Logics
Vol. 00, No. 00, Month 201X, 1–30

RESEARCH ARTICLE

Modeling Last-Act Attempted Crime in Criminal Law

(*Received 00 Month 201X; final version received 00 Month 201X*)

In the court of law, a person can be punished for attempting to commit a crime. An open issue in the study of Artificial Intelligence and Law is whether the law of attempts could be formally modelled. There are distinct legal rules for determining attempted crime whereas the last-act rule (also called proximity rule) represents the strictest approach. In this paper we provide a formal model of the last-act rule using structured argumentation.

Keywords: attempted crime; last-act (or proximity)-rule; structured argumentation

1. Introduction

Legal systems punish not only completed crimes, but also the attempts to commit them, at least for the most serious crimes.

In this paper, we aim to model the strictest standard for attempt, namely, the last-act test, which constitutes the core of all other tests. If the concerned case can be convicted by the last-act test then the conviction will be affirmed also according to other tests. Further if it is not possible to model the strictest test then it is impossible to model any other test.

The paper is organized as follows. In this section we introduce the law of criminal attempt, focusing on the last-act test, and we sketch our approach to modelling this test. In section 2 we briefly review the argumentation framework and MABA framework. We introduce our approach to model the dynamic of actions in section 3. We model the last-act, intention in the last-act test and attempt rules in section 4. As we cannot prove mathematically that our approach is correct, we evaluate it by applying it on several well-known court cases and also by formally showing that it is coherent system. In section 5 the structure of our system is elaborated, the definition of well-structured knowledge bases are proposed and its semantic are analyzed. We illustrate in section 6 the application of our proposal by applying it to the court cases. Relevant literatures specially actual causality are discussed in section 7. Finally we discuss and conclude.

1.1 *The law of attempt*

We may say that, in general terms, the attempt to commit a certain primary crime (not being itself an attempt), consists in an act that has been carried out purposefully but failed in completion of a primary crime[1] (Andenaes, 1965; LaFave, 2017; Williams,

[1]Criminal Attempts Act 1981 of UK law "*... a person does an act which is more than merely preparatory to the commission of the offence...*". Section 6 of Massachusetts General Laws states that "*whoever attempts to commit a crime by doing any act toward its commission,but fails in its perpetration,...,be punished as follows:...* ". Section 80 of Thai Criminal Code also states that "*whoever commences to commit an offence, but does not carry it through, or carries it through, but does not achieve its end...*". The Italian criminal code, at Article

1978). Thus, the attempt to commit an offence requires the agent to realise to a sufficient extent the material aspect of the offence, though not completing it, and that he has the purpose of realising the whole crime. As Fletcher says the difference between an attempt and the corresponding primary crime is that in the attempt "the harm, the death, the beating, the loss of property, the sexual penetration is absent"(Fletcher, 1998). This raises the issue of why an attempt should be punished. It has been argued that remedial action is needed against those people who have explicitly expressed their dangerousness, for preventive purposes. Also the police can legally intervene when they find a man attempting to commit any indictable offence (LaFave, 2017; Williams, 1978).

The purpose of the law of criminal attempts is crime prevention so that the police can legally intervene before the crime occurs. Many tests are introduced to distinguish the border line between the merely preparatory and attempt act in legal judgment. Besides the last-act test, there are many other more liberal tests which are applied in various courts. If a person can be convicted under the last-act test, he/she will also be convicted by other test. In this paper we provide a formalization of the last-act test in the law of attempts.

The legal punishment on attempts has a long history. Roman law punished attempts, and though for "ordinary" crimes an attempt got a lighter penalty than a success, for "atrocious" crimes attempts were apparently punished just as severely as completed crimes. Plato in the Laws presented the ideal legal regimes about attempted murder (Becker, 1974). However, in common law the case of Rex v. Scofield (Duff, 1996; Scofield, 1784) has been considered as the origin of the modern law of criminal attempts. The defendant intentionally set fire to burn his rented house by placing lighted candle with combustible material but the larger fire did not happen. The defendant was convicted for attempted arson.

1.2 *The last act rule*

Various legal theories exist concerning the extent up to which an offence must be realised for its partial realisation to constitute an attempt. The common law has traditionally adopted the so called proximity test, which requires that some act, beyond 'mere preparation', is accomplished, an act that should be 'proximate' enough, temporally or physically, to the completion of a crime. The most rigorous test for proximity is provided by so called *last-act test*, that requires that the criminal has done all that he could do to commit the crime, so that the completion of the conduct only depended on external factors (Fletcher, 1998; LaFave, 2017)[2]. Recently, UK law has apparently affirmed a broader notion of attempt, only requesting an act "which goes so far towards the commission of the offence attempted as to be more than an act of mere preparation". In US law an even broader notion is adopted, which only requires that the agent accomplishes "substantial step in a course of conduct planned to culminate in his commission of the crime" (US Model Penal Code).

Our analysis of the last-act rule will focus on result crimes, namely on those crimes that include non only a criminal conduct but also the realisation of a harmful result. Thus, we shall address the cases when the agent has completed action meant to produce the criminal result, but has failed to realise that result. For instance in State v. Mitchell (170 Mo. 633, 71 S.W. 175 1902) the criminal were condemned for attempted murder for

56 says that "*who perform idoneous acts, which are unequivocally directed to commit a delict, is responsible for attempt, if the action is not completed or the event does not take place.*".

[2] An extensive discussion on the last-act rule can be found in (Fletcher, 1998) where pros and cons of this rule are discussed at length.

shooting through a hole in the roof of a house to kill a person they mistakenly believed to be still there.

We shall also address those cases in which the agent has completed his conduct, but the realisation of the crime also requires some consequential action by others, that the criminal mistakenly expects will be accomplished as illustrated in the Eagleton case.

Example 1. *The baker in the parish of Great Yarmouth, Mr.Eagleton (Eagleton, 1855), signed a contract to supply bread to give to the poor. He got a ticket from a needy person for each loaf of bread. Then he reported the number of loaves supplied with collected tickets to an officer who would credit his account and pay him later. Before the first payment it was found that Mr.Eagleton had given bread loaves to the poor weighting much less than the contracted specification. Hence, he was convicted for attempting to obtain money by false pretences.*[3]

Concerning the subjective aspect of attempt, the intention to realise the criminal conduct, and its result (for result-crimes) is needed. Moreover, awareness of circumstances that are relevant the realisation of the crime is also needed(Duff, 1996). The court of Sanstrom provides a guidance for jury to draw intention from the defendant's acts: "*The law presumes that a person intends the ordinary consequences of his voluntary acts.*" (Sanstrom, 1979). So the commission of an act authorises the defeasible inference that not only the act, but also its normal effects were intended.

In this work we focus on the interpretation of the last-act rule by appellate courts and hence we do not deal with matter of facts and how qualitative facts could be drawn from quantitative evidences. There are clear legal guidelines for delineating the functions of district and appellate courts as stated in (The U.S. Court Role, 2016): "*At a trial in a U.S. District Court, witnesses give testimony and a judge or jury decides who is guilty or not guilty…The appellate courts do not retry cases or hear new evidence. They do not hear witnesses testify. There is no jury. Appellate courts review the procedures and the decisions in the trial court to make sure that the proceedings were fair and that the proper law was applied correctly.*". In short, the appellate courts deal only with the matter of law while the matter of facts has to be dealt with in trial courts.

1.3 *A model of the last-act attempt*

Two aspects need to be considered in modeling the last-act test:

- that exceptional circumstances which are unexpected to the defendant blocked the causal chain from the last-act to the completion of the crime, and
- the intention of the defendant to commit the criminal offence at the time of committing the last-act.

To model the first aspect, we structure our system into two modules where the first module contains facts, domain rules and legal rules of the case while the second module represents the hypothetical situation where unexpected circumstances, unknown to the defendant did not happen. In the Eagleton case, it shows that if the officer had not intervened, then the action of submitting credits, which is the last-act, would have caused the crime of obtaining money by false pretense.

To model the intention aspect we distinguish between two kinds of basic actions which are proactive actions that are executed proactively by the defendant and trigger actions

[3] It is interesting to note that had Eagleton not submitted the collected credits, he would not have been convicted in any court applying the last-act rule though he could still be convicted in an US court using the more liberal substantial-step rule provided that his intention to cheat could be proved.

that are executed by agents other than the defendant. Hence we can derive intention of the defendant when he/she deliberately executes proactive actions.

A key component of our model is a representation of the dynamics of action and the dynamic causal laws stating the causal effects of actions. A well-known approach for representing and studying the causal effects of actions integrating in a natural way like the situation calculus with time has been proposed and studied extensively in (Dung, 1993; Gelfond & Baral, 2000; Gelfond & Lifschitz, 1998, 1993). We adopt this language as it allows us to represent the legal rules in a natural way as in the work of (Dung & Sartor, 2011; Dung & Thang , 2008; Dung & Thang & Hung , 2010). However, other approaches like event calculus-based languages (Artikis & Sergot & Paliouras , 2015, 2014; Hadjisoteriou & Kakas, 2012; Kowalski & Sergot, 1986; Shanahan, 1999) could as well be used in modeling the law of attempts.

We apply a modular assumption-based argumentation (MABA) framework to model the modules of the defendant knowledge and the knowledge base of the case. Applying MABA we can model the hypothetical situation where had unexpected circumstances not happen, the last-act causally leads to the completion of the crime.

This paper could be viewed as a continuation of our approach in using argumentation to model laws like the private international law (Dung & Sartor, 2011) or contract laws (Dung & Thang , 2008; Dung & Thang & Hung , 2010). This suggests that argumentation is natural and appropriate platform for modeling legal reasoning.

2. Preliminaries

Our model of last-act test is based on modular assumption based argumentation frameworks, which rely on the theory of abstract argumentation.

2.1 *Abstract Argumentation*

An abstract argumentation framework (Dung, 1995) consists of a set of arguments AR and an attack relation att $\subseteq$ AR$\times$AR where (X,Y) $\in$ att means that the argument X attacks the argument Y.

A set S of arguments attacks an argument A if some arguments in S attacks A. S is said to defend A if S attacks any argument attacking A. A set of arguments S is admissible iff S is conflict-free and defends all arguments attacking its arguments. Several semantics are introduced. Preferred extensions are maximal (wrt set inclusion) admissible sets of arguments while complete extensions are admissible sets containing all arguments it defends. Grounded extension is the smallest (wrt set inclusion) complete extension. A stable extension is a conflict-free set of arguments attacking every arguments outside it.

Stable extensions are preferred extensions but not vice versa.

2.2 *Assumption-Based Argumentation Framework*

Given a language $\mathcal{L}$, an assumption-based argumentation(ABA) framework (Bondarenko & Dung & Kowalski & Toni, 1997; Dung & Kowalski & Toni , 2006; Gaertner & Toni , 2008) over $\mathcal{L}$ is represented by a triple $(\mathcal{R}, \mathcal{A}, \overline{})$ where $\mathcal{A} \subseteq \mathcal{L}$ is a set of assumptions, $\mathcal{R}$ is a set of rules of the form

$$\delta \leftarrow \delta_1, ..., \delta_n$$

where $\delta_i, \delta \in \mathcal{L}$ and $\delta \notin \mathcal{A}$, and $\overline{}$ is a (total) mapping from $\mathcal{A}$ into $2^{\mathcal{L}} \backslash \{\emptyset\}$ where for any $\alpha \in \mathcal{A}$, $\overline{\alpha}$ denotes the set of contraries of α.

A special kind of assumptions is represented by negation as failure literals of a form *not_l* whose contrary is l.

Definition 1.

- *An argument wrt ABA* $M = (\mathcal{R}, \mathcal{A}, \overline{})$ *is constructed by repeatedly applying the following steps finitely many times:*
  (i) *For any assumption* $\alpha \in \mathcal{A}$ *,* $[\alpha]$ *is an assumption-argument with conclusion* α*. The set of assumptions of* $[\alpha]$*, denoted by ass(*$[\alpha]$*) is* $\{\alpha\}$*.*
  (ii) *Let* $r \in \mathcal{R}$ *be a rule of the form* $\delta \leftarrow \delta_1, ..., \delta_n$*,* $n \geq 0$ *and* $A_1 \ldots A_n$ *are arguments with conclusion* $\delta_i, 1 \leq i \leq n$*, respectively.*
  Then $A = [\delta \leftarrow A_1, ..., A_n]$ *(also denoted by* $A = [r, A_1, ..., A_n]$*) is an argument with a conclusion* δ*, denoted by cnl(A), and the set of assumptions of A is ass(A)=*$ass(A_1) \cup \ldots \cup ass(A_n)$*.*

  The set of all arguments wrt M is denoted by AR_M*.*
- *We say that A attacks B, denoted by (A,B)*$\in att_M$ *iff the conclusion of A is a contrary of some assumptions of B, i.e. cnl(A)*$\in \overline{\alpha}$ *for some* $\alpha \in ass(B)$*.*
- *We define the argumentation framework wrt M by* $AF_M = (AR_M, att_M)$*.*

Abusing the notation slightly, we define the extensions of an ABA framework M as the extensions of AF_M.

We say that a sentence δ is skeptically(resp. credulously) derived from an ABA framework M, denoted by $M \vdash\!\sim_s \delta$(resp. $M \vdash\!\sim_{cr} \delta$) iff in each(resp. some) prefer extension, there is an argument A such that the conclusion of A is δ.

2.3 *Modular Assumption-Based Argumentation Framework*

A *modular assumption-based argumentation (MABA) framework* (Dung & Sartor, 2011) is a structure of distinct ABA modules having exactly one main module and several submodules. Each module is an assumption-based argumentation framework which allows rules of the form $\delta \leftarrow \delta_1, ..., \delta_n$; $n \geq 1$ where δ_i can be either a sentence or a module call of the form *call(l,M,t)* stating that $M \vdash\!\sim_t l$ with $t \in \{s, cr\}$.

Both types of semantics of module calls are useful for modeling private international law (Dung & Sartor, 2011) and contract law (Dung & Thang , 2008; Dung & Thang & Hung , 2010).

As for the purpose of modeling the last-act attempts only skeptical semantics are required, we often write $M \vdash\!\sim \alpha$ for $M \vdash\!\sim_s \alpha$. Furthermore, from now on we write $call(l, M)$ for $call(l, M, s)$.

In this paper we restrict ourself on stratified MABA frameworks where the modules are ranked such that all module calls of the form $call(l, M)$ in a module of rank k refer to module M of ranks lower than k.

The semantics of stratified MABA framework is given by inductively defining the semantics of the higher ranks modules based on the semantics of lower ranks modules. Suppose that the semantics of all module of rank $\leq$ n have been defined then the set of all arguments of a module M with rank n + 1 is defined as in definition 1 with an extra condition as follows:

- $[call(l, M')]$ of module M' of rank $\leq$ n is an argument (wrt module M) with the conclusion l and an empty set of assumptions iff $M' \vdash\!\sim_s l$.

Hence the semantics of module M is also fully defined. Repeatedly applying this step the semantics of the main module can be defined.

For practical purpose we extend the MABA framework by allowing to call module written in other programming language like JAVA, C, or whatsoever like many versions of Prolog (SWI-Prolog, 2019). Such module call has the form $call(M, In)$ or $not\_call(M, In)$ where M is a module from other languages, In is a list of input parameters and the return is true or false respectively.

3. Domain Representation

We assume a set $\mathcal{F}$ of fluents representing properties in the concerned domain and a set $\mathcal{A}$ of basic actions. Both $\mathcal{F}$ and $\mathcal{A}$ are finite and disjoint. For convenience, we use f, f_i to denote fluents, and a, a_i to denote basic actions. We also use F,G to denote finite sets of fluents and their negations.
For simplicity we often identify a singleton set {f} with its element f.

Executing a basic action can cause more than one fluents to hold. We represent the *dynamic causal laws* specifying causal effects of actions as defeasible rules of the form

$$\text{DCL}: \quad causes(a, f, F, T) \longleftarrow not\_ab_{DCL}(a, f, F, T)$$

$$where\ \neg f \in F$$

stating that if an action a is executed at time T where fluents in F hold then (barring some unforeseen intervention) it will cause fluent f to hold. The fluents in F are often refer to as preconditions of a and the fluent f is called the postcondition of a.

To see the reason for the requirement that $\neg f \in F$, consider a causal law stating that shooting a loaded gun causes death. This obviously means that the person being shot was alive before the shooting occurs. Therefore the causal law should be *causes(shoot, Dead, {Loaded, ¬Dead}, T).*

Remark. *Note that a rule with variables is considered as a shorthand of the set of all grounded instances of it. For example rule DCL refers to the set of all of its ground instances, where an instance of DCL is often referred to by DCL(a,f,F,T).*

Theorem 1. *Given are two dynamic causal laws*

$$causes(a, f, F, T) \leftarrow not\_ab_{DCL}(a, f, F, T)$$

$$causes(b, g, G, T) \leftarrow not\_ab_{DCL}(b, g, G, T)$$

such that $F \cup G$ is consistent. It holds that

(1) $g \notin F$ and $f \notin G$ hold, and
(2) {g,f} is consistent.

Proof.

(1) Suppose $g \in F$ then $F \cup G$ is not consistent because $\neg g \in G$. Similarly suppose $f \in G$ then $F \cup G$ is not consistent. Contradiction.
(2) Since $\neg f \in F$ and $g \notin F$, $g \neq \neg f$. Hence, $\{g, f\}$ is consistent. □

Remark. *Theorem 1 essentially states that if the preconditions of two actions are sat-*

isfied at the same time then both actions could be executed concurrently.

Convention Abusing the notation for simplicity, we often simply write $causes(shoot,\ Dead,\ Loaded,\ T)$ for $causes(shoot,\ Dead,\ \{Loaded, \neg Dead\},\ T)$ if there is no possibilities for misunderstanding.

In general, if there is no possibilities for misunderstanding we often write $causes(a,\ f,\ F,\ T)$ with $\neg f \notin F$ for $causes(a,\ f,\ F \cup \{\neg f\},\ T)$.

Remark. *The undercut of dynamic causal laws depends on their domain. For instance the dynamic causal law stating that shooting a person within 5 metres normally causes death is described by*

$$causes(shoot, Dead, \{Loaded, 5m\}, T) \leftarrow not\_ab_{DCL}(shoot, Dead, \{Loaded, 5m\}, T).$$

The rule will be undercut when the victim wears a bullet proof vest as represented by the following rule:

$$ab_{DCL}(shoot, Dead, \{Loaded, 5m\}, T) \leftarrow holds(Wearing\_bullet\_proof\_vest, T).$$

It is obvious that if executing an action a under precondition F causes the fluents in G and G' to hold then it also causes the fluents in $G \cup G'$ to hold.

CA : $$causes(a, G \cup G', F, T) \leftarrow causes(a, G, F, T), causes(a, G', F, T)$$

We use a predicate $holds(F, T)$ to denote that fluents in F hold at time T.

The meaning of the following rule should be obvious.

HS : $$holds(G \cup G', T) \leftarrow holds(G, T), holds(G', T), G \neq \emptyset, G' \neq \emptyset$$

The following rule states that the fluents in L hold (at some time point) if there is a time point T such that L hold at T.

H : $$holds(L) \leftarrow holds(L, T)$$

The rule below describes the effect of an action.

AE : $$holds(G, T+1) \leftarrow causes(a, G, F, T), happen(a, T), holds(F, T)$$

Fluents are normally inertial and hence do not change when an action is executed. There could be many exceptions to this rule like when the action causes the contrary of a fluent to hold or the action affects some fluent on which a fluent is related like when a person is shot dead then he can not sing. This property is specified by inertial rules below:

HD : $$holds(f, T+1) \leftarrow holds(f, T), not\_ab_{HD}(f, T)$$

UHD : $$ab_{HD}(g, T) \leftarrow holds(F, T), happen(a, T), Rel(g, a, F)$$

where $Rel(g, a, F)$ intuitively states that when a is executed under condition F then g is not inertial.

Predicate $Rel(g, a, F)$ is part of the domain representation where $Rel(f, a, F)$ is always included whenever $causes(a, \neg f, F, T)$ is included in the domain representation.

3.1 *Two Kinds of Actions*

We distinguish between two kinds of basic actions: proactive actions that are executed proactive and trigger actions that are executed when triggered.

Trigger actions are usually executed by agents other than the defendant while proactive actions could be executed by any agent. Trigger actions are denoted by $trigger(a, G)$ and proactive actions are denoted by $proactive(a)$ for any basic action a and precondition G. Predicate $executor(\alpha, a, T)$ denotes that agent α executes action a at time T.

An agent executes proactive actions only when she chooses to execute them or selects them among plausible available alternatives and she is fully aware of what she is doing.

In contrast, trigger actions are usually executed (by agents other than the defendant) whenever their preconditions hold, according to routines, commitments or norms. For instance, submitting a withdrawal slip to a bank teller satisfies the precondition of the trigger action $withdraw\_money$ and hence, the action $withdraw\_money$ will normally be executed within some period of time.

Trigger actions could be executed immediately after their preconditions hold or they will happen later after some period of time. For example, a client will get money within a few minutes after submitting a withdrawal slip to a bank teller. In another example, it could take days to get money from an insurer after a client has submitted a claim.

In other words, when a trigger action is triggered, i.e. when its precondition are satisfied, the action will happen soon (but not necessarily immediately). The meaning of how soon is "soon" depends on the domain-specific delay time t_{dl} and is represented in the following rules:

$$\text{TA}: \quad will\_happen(a, T + t_{dl}, T) \leftarrow trigger(a, G), holds(G, T), delay(a, t_{dl}), \\ not\_triggered(a, T - 1),\ T > 0$$

stating that when the preconditions G of a hold, the action a will normally happen at some point in the future between time T and $T + t_{dl}$ where $t_{dl} > 0$. $delay(a, t_{dl})$ states that t_{dl} is the limit of the delay time for executing action a after being triggered.

The assumption $not\_triggered(a, T - 1)$ in TA is to ensure that an action can only be triggered if it is not triggered yet, and once triggered, it remains triggered until it actually happens as represented by rule UTA below.

$$\text{UTA}: \quad triggered(a, T) \leftarrow will\_happen(a, T', T), trigger(a, G)$$

We use assumptions ϕ_{now} and ϕ_{next} to represent whether a triggered action could happen now or later. Both assumptions are a contrary of each other and hence, only one assumption will be selected at a time. If a triggered action is supposed to be executed now, then the assumption ϕ_{now} is selected. Otherwise the assumption ϕ_{next} is picked.

$$\text{HNO}: \quad happen(a,T) \leftarrow will\_happen(a,T',T),\ \phi_{now}(a,T',T)$$

$$\text{HNE}: \quad will\_happen(a,T',T+1) \leftarrow will\_happen(a,T',T),\ \phi_{next}(a,T',T)$$

where

$$\overline{\phi_{now}(a,T',T)} = \{\phi_{next}(a,T',T),\ \Phi_u(a,T),\ T > T'\}\ \text{, and}$$

$$\overline{\phi_{next}(a,T',T)} = \{\phi_{now}(a,T',T),\ \Phi_u(a,T),\ T \geq T'\}.$$

Rule HNE represents the situation when the triggered action a does not happen immediately while rule HNO represents the situation when the triggered action happens now.

The rule HNO and HNE guarantee that a triggered action will happen within the delay time t_{dl} specified in rule TA unless it is undercut by some external conditions represented by $\Phi_u(a,T)$. Specifically when the time limit is reached, then the assumption $\phi_{next}(a,T',T)$ will be rejected (as $T \geq T'$ is its contrary) and therefore only the assumption $\phi_{now}(a,T',T)$ can be selected. If the triggered action has not happened before the delay time limit, it will happen at the point of delay time limit.

$\Phi_u(a,T)$ is an undercut to the triggered action a. For instance, in the case of Eagleton the triggered action pay_money is undercut, i.e., it will not happen at all when the officer intervenes as captured by the following rule.

$$\text{UE}: \quad \Phi_u(pay\_money,T) \leftarrow holds(Officer\_intervention,T)$$

3.2 *Domain Knowledge base*

A domain rule base for modeling the last-act-attempt consists of rules as follows:

Definition 2. *A domain rule base RKB = (CAL, DOA, UB, DS) consists of*

(1) a set CAL of dynamic causal laws of the relevant actions of the domain, and
(2) the set DOA of rules presented above[4] *describing the dynamics of the actions and the rules classifying types of actions like rules stating whether an action is trigger or proactive, and*
(3) a set UB of possible undercut rules of the defeasible rules in CAL or HNE, HNO of the form

$$ab_{DCL}(a,f,F,T) \leftarrow holds(G,T)$$

$$\Phi_u(a,T) \leftarrow holds(G,T)$$

(4) and a set of DS containing
(a) The relation $Rel(g,a,F)$ specifying the fluents that are not inertial when executing a under condition F; and
(b) rules specifying other domain knowledge of the concerned case such that

[4]To be precise, they are the rules CA, HS, H, AE, HD, UHD, TA,UTA, HNE, and HNO.

- *all literals appearing in the same rules have the same temporal parameter; and*
- *for each literal L appearing in a rule in DS, the following conditions hold:*
  - *L is of the form holds(f,T) where f is a fluent;*
  - *L is of the form ab(...) or not_ab(...) and the predicate ab(...) does not appear in any rule in CAL, DOA, UB and L appears only in the heads of rules in DS if L is positive, otherwise only in the bodies of the rules in DS.*

For example the defendant mistakenly believed that a decoy that looks like a grazing deer is a deer.[5] *This can be captured by the following rule.*

$$holds(Deer(O),T) \leftarrow holds(Look\_like\_deer(O),T),\ not\_ab(O,T)$$

stating that if an object looks like a deer then it is a deer.

Definition 3. *An evidence base BE is defined to be a set of facts of the form holds(L,T) or happen(a,T) together with executor(D,a,T) where D is an executor, a is an action, L is a fluent literal and T is a time point.*

4. Modeling the Last-Act-Attempts

There are several approaches in distinguishing between mere preparation and attempted acts of the crime of attempts. Among all approaches the last-act test is the strictest one which simply considers whether the defendant has engaged in the last proactive action which normally can lead to the completion of the intended offence.

The last proximate act must be immediately connected with the substantive offence[6]. The proximity test requires the accused to engage in the last proximate act as explained in (LaFave, 2017):

> "The defendant has engaged in the last proximate act, that is, that he have done everything which he believes necessary to bring about the intended result."

The last proximate act is the act that lead to intended consequence naturally. For instance Mr.Eagleton had already submitted his claim for his credits. He would get his money without the intervention by the officer, which was unexpected circumstances for Mr.Eagleton. The court found that he had engaged the last proximate act which immediately connected to the substantive crime.

More formally we can say that when an agent has accomplished a deliberate action that would trigger a set of further actions and events without any involvement of the agent then the agent is considered to finish the last-act.

The commission of an attempt rather than the primary crime, depends on the fact that exceptional circumstances, unknown to the agent, have blocked the ***causal chain*** *from the last-act to completion.*

For instance, if the officer had not discovered that Mr.Eagleton had breached the contract by supplying underweight bread loaves and submitting credits for his payment, Mr.Eagleton would obtain money for his false pretense.

[5]See (Wilkinson, 1998)

[6]"The mere intention to commit a misdemeanour is not criminal. Some acts is required, and we do not think that all acts towards committing a misdemeanour are indictable. Acts remotely leading towards the commission of the offence are not to be considered as attempts to commit it, but acts immediately connected with it are...". (Allen, 2011; LaFave, 2017; Eagleton, 1855)

A key task in modeling last-act attempted crime is to model the causality between the defendant's last-act and the attempted offence. If the crime would not have completed without the defendant's last-act, then a causal connection is present. In other words, if the crime would have happened independently of the defendant's last-act, then there is no causal connection between them (Colvin, 1989).

Therefore to prove a causal connection, our system is structured to model two hypothetical situations

- if unexpected circumstances have not happened, then the crime would be completed, and
- if the defendant had not executed his last-act then the crime would not have occurred.

As we are modeling the applications of the law of attempts by the appellate courts, it is known to the courts what the defendant believed when he/she carried out his/her last-act as well as what the facts of the case are[7].

Hence, we structure a domain knowledge base system of a case, refered as the case knowledge base, into two modules. Each module contains the domain rule base RKB as defined in definition 2 but has different sets of facts and evidences.

(1) M_d =(RKB, BE_d) represents the knowledge base of the defendant at the time he/she committing the last-act where BE_d contains facts and evidences in which the defendant believed he/she was in up to the point of his last-act, and hence $happen(a, T) \in BE_d$ for the last-act a.

(2) Module M_c =$(RKB \cup LKB, BE_d \cup BE_u)$ is the entire knowledge base of the case where all facts and evidence are included, B_u contains facts that is unknown to the defendant at the time he was executing his last-act, and LKB is a set of rules representing the law of last-act attempted crime.

We describe the rules in LKB below.

We say the execution by the defendant of a basic action a at time T **causally leads to** the holding of a set L of fluent literals iff following conditions hold:

(i) $happen(a, T) \in \ BE_d$

(ii) $M_d \mathrel{|\!\sim} holds(L)$

(iii) $M_d \backslash \{happen(a, T)\} \mathrel{|\!\not\sim} holds(L)$

where $M_d \backslash \{happen(a, T)\} = (RKB, BE_d \backslash \{happen(a, T)\})$

Rule CLT below captures what we have just discussed:

$$\text{CLT}: \quad clead(a, L, T) \longleftarrow happen(a, T), call(holds(L), M_d),$$
$$call(not\_holds(L), M_d \backslash \{happen(a, T)\})$$

A proactive action a is considered as the **last-act leading to a criminal offence** iff executing a casually leads to L as represented in the following rule.

[7]The appellate courts deal only with matter of laws while matter of facts are dealt by lower court. (The U.S. Court Role, 2016).

CLA :
$$last\_act(a, L, T) \leftarrow proactive(a),\ happen(a, T),\ clead(a, L, T),\ offence(L)$$

where the predicate $offence(L)$ indicates that a criminal offence is committed if all the fluents in L hold. The definition of offence is domain specific and should be given. For example, in the Eagleton case, the offence consists of two fluents denoting distributing of underweight bread and obtaining money for it.

Attempts in criminal law requires the proof of an intent to commit an offence at the time the crime occurs. Intention is in a person's mind and it cannot be proven objectively. Thus the courts provide rules for deriving intention from what the defendant has already done as stated in the case of Niziolek (Niziolek, 1980) that a person is presumed to intend the natural and probable consequences of his own act (an elaborate discussion of this point is given in quote below[8]).

Similarly the court of Sanstrom provides a guidance for jury to draw intention from the defendant's acts: "*The law presumes that a person intends the ordinary consequences of his voluntary acts.*"(Sanstrom, 1979).

For instance, a person who has boarded an international flight has an intent to leave the country. Similarly as Mr.Eagleton had submitted the credits obtained from distributing underweight bread loaves he had demonstrated a clear intent to obtain the payment by false pretenses[9].

A person who has engaged in executing an act voluntarily intends the foreseen consequences of the act. Without unexpected interruption the intended result will happen if the person completes the act.

If executing a proactive action a by the agent D at time T causally leads to the holding of a set of fluent literals L, we conclude that the agent has the **intention to execute a at the time T to accomplish L** as formalized in the following rule.

$$\text{CIT}: \quad intent(D, a, L, T) \leftarrow proactive(a),\ happen(a, T),\ executor(D, a, T),\ clead(a, L, T),\ not\_ab_{CIT}(a, L, T)$$

The rule CIT could be undercut when the defendant is mentally disturbed and hence has no capability to foresee the consequence of his action. For example in the case of Stephenson (Stephenson, 1979) where the defendant ignited a straw stack to keep himself warm and caused fire and damage. There were evidences showing that the defendant had schizophrenia which made him incapable to appreciate the consequence of his action. Hence he was acquitted of the arson charge. This can be captured by the following rule.

$$ab_{CIT}(Stephenson, light\_fire, Burn\_straw, T) \leftarrow holds(Have\_schizophrenia(Stephenson), T)$$

[8]"*Intent is a state of mind. The intention of a person ... is to be ascertained by his acts and the inference is to be drawn from what is externally visible. Intent ordinarily cannot be proven directly because there is no way of reaching into and examining the operations of the human mind, but you may determine the defendant's intent from any statement or act done or act omitted and all the other circumstances which indicate his state of mind, provided you first find that any or all of such circumstances occurred. A person is presumed to intend the natural and probable consequences of his own acts.*"(Niziolek, 1980)

[9]"...by returning the tickets to the relieving officer he intended to represent that he had delivered the loaves mentioned in them of the weights stated."(Eagleton, 1855)

which undercuts the instance of rule CIT:

$$\begin{aligned}&intent(Stephenson, light\_fire, Burn\_straw, T) \longleftarrow\\ &\quad proactive(light\_fire),\ happen(light\_fire),\ executor(Stephenson, light\_fire, T)\\ &\quad clead(light\_fire, Burn\_straw, T),\ not\_ab_{CIT}(light\_fire, Burn\_straw, T)\end{aligned}$$

We are now almost ready to define when an agent should be considered as having attempted a crime wrt the last-act rule.

An agent D is considered to have attempted a criminal offence by executing a proactive action at time T iff his execution of his act would have causally led to the criminal offence with his intent to accomplish the consequence of his action but fails to reach to the completion of the crime. This can be capture by the following rule.

CAT :

$$attempt(D, a, L, T) \leftarrow last\_act(a, L, T),\ intent(D, a, L, T),\ not\_holds(L)$$

To summarize, the component LKB representing the law of last-act attempt consists of the following rules:

$$\begin{aligned}\text{CLT}: \quad & clead(a, L, T) \longleftarrow happen(a, T),\ call(holds(L), M_d),\\ & \qquad call(not\_holds(L), M_d \backslash \{happen(a, T)\})\\ \text{CLA}: \quad & last\_act(a, L, T) \longleftarrow proactive(a),\ happen(a, T),\\ & \qquad clead(a, L, T),\ offence(L)\\ \text{CIT}: \quad & intent(D, a, L, T) \longleftarrow proactive(a),\ happen(a, T),\ executor(D, a, T),\\ & \qquad clead(a, L, T),\ not\_ab_{CIT}(a, L, T)\\ \text{CAT}: \quad & attempt(D, a, L, T) \longleftarrow\\ & \qquad last\_act(a, L, T),\ intent(D, a, L, T),\ not\_holds(L)\end{aligned}$$

5. Well-Structured Knowledge Bases

The semantics underlying our approach to model the last-act-attempt is based on preferred extensions.

In general if an argumentation framework has a prefer extension that is not a stable extension then there is some anomaly in it. An example is the argumentation framework with exactly one self-attacking argument that has an unique preferred extension that is the empty set and not stable.

An argumentation framework is *coherent* if each preferred extension is stable.

Further an ABA framework is said to be coherent iff its prefer extensions are stable.

We believe that in practice knowledge bases for convincing legal cases should be coherent. It turns out that under general and reasonable condition on the domain knowledge represented by the DS component, the knowledge bases for the last-act-attempt cases are coherent.

We first introduce a notion of dependency graph adopted with a slight modification from similar notion in (Dung, 1995).

The *dependency graph* of an ABA framework M $= (\mathcal{R}, \mathcal{A}, \overline{})$ is a directed graph where nodes are sentences in $\mathcal{L}$, edges could be either negative or positive, and

- there exists a positive edge labelled by a rule r $\in \mathcal{R}$ from a node α to a node δ iff α is a head of r and δ is an element of the body of r, and
- there is a negative edge from a node α to a node δ iff $\delta \in \overline{\alpha}$ [10].

A loop is a sequence $N_1, \ldots N_m$ such that $N_1 = N_m$ and $\forall\, 1 \leq i, j < m$, $N_i \neq N_j$ for $i \neq j$.

The following lemma holds obviously.

Lemma 1. *Let M be an ABA framework and A be an argument wrt M with δ being the conclusion of A and α being an assumption of A. Then there is a path consisting of only positive edges from δ to α in the dependency graph of M.*

Lemma 2. *Let $A_1 \ldots A_n$ be a loop in AF_M[11]. Then there is a loop containing n negative edges in the dependency graph of M.*

Proof. Let δ_i= cnl(A_i) and $\alpha_i \in ass(A_i)$ such that $\delta_{i+1} \in \overline{\alpha_i}$. From Lemma 1, there is a positive path pa_i (i.e. pa_i contains only positive edges) from δ_i to α_i. Since $\delta_{i+1} \in \overline{\alpha_i}$, there is a negative edge from α_i to δ_{i+1}. Therefore we can construct a path, pa in the dependency graph of M as follows.

$$\begin{tikzcd} pa \equiv & pa_1 \arrow[r, "-"] & pa_2 \arrow[d, "-"] \\ & pa_{n-1} \arrow[u, "-"] & \cdots \arrow[l, "-"] \end{tikzcd}$$

where each pa_i is a positive path. It is clear that pa is a path in the dependency graph of M with exactly n negative edges. $\square$

We introduce next the notion of well-structured modules.

Definition 4. *(***Well-Structured Knowledge Bases***)*
M_c is well-structured iff the component DS is acyclic.

From the definition 4, it is obvious that M_c is not well-structured if DS contains a rule like $holds(Dead, T) \leftarrow holds(Dead, T)$.

Since $M_d \subset M_c$, if M_c is well-structured, it is clear that M_d is also well-structured.

Lemma 3.

(1) Let $\alpha \xrightarrow{r} \beta$ be a link in the dependency graph of M_c and α has the form P(..., T). Then β should have a form Q(..., T') such that $T \geq T'$.

[10]Our notion of a dependency graph is slightly different from an atom dependency graph for logic programming in the work of Dung (Dung, 1995) where a negative edge is drawn from an atom a to an atom b if there is a clause with a as its head and not_b appearing in its body while in our definition there will be a positive edge from a to not_b and a negative edge from not_b to b.
[11]I.e. $\forall\, 1 \leq i < n$, A_{i+1} attacks A_i, $A_n = A_1$ and $A_i \neq A_j$ for $i \neq j$.

(2) *Let pa be a path in the dependency graph of M_c. Suppose α be the first sentence in pa in the form of P(..., T). Then all other sentences of this path will be the form of Q(...,T′) such that $T \geq T'$.*

Proof.
It is obvious that (2) is followed from (1).

To prove (1) we do a case analysis of r.

i) Let r belongs to the component CAL. Hence $\alpha = causes(.,.,.,T)$. Then $\beta = not\_ab_{DCL}(.\,,\,.\,,\,.\,,T)$. It is clear that α and β have the same time parameter.
ii) Let r belongs to the component UB. Hence α is either $ab_{DCL}(.\,,.\,,.\,,T)$ or $\Phi_u(.,T)$. Then $\beta = holds(.,T)$. It is obvious that both α and β have the same unit of time.
iii) Let r belongs to the component DOA. There are several cases as follows.
   - Let r be CA. Hence $\alpha = causes(.,.,.,T)$. Then $\beta = causes(.,.,.,T)$. It is clear that α and β have the same time parameter.
   - Let r be HS. Hence $\alpha = holds(.,T)$. Then $\beta = holds(.,T)$. It holds that α and β have the same time parameter.
   - Let r be AE. Hence $\alpha = holds(.,T+1)$. Then β can be either $causes(.,.,.,T), happen(.,T)$, or $holds(.,T)$. It is clear that the time parameter of α is greater than those of β.
   - Let r be HD. Hence $\alpha = holds(.,T+1)$. Then β can be either $holds(.,T)$ or an assumption $not\_ab_{HD}(.\,,T)$. It is obvious that α has the time parameter greater than those of β.
   - Let r be UHD. Hence $\alpha = ab_{HD}(.\,,T)$. Then β can be either $happen(.,T)$, or $holds(.,T)$. It is clear that α and β have the same time parameter.
   - Let r be TA. Hence $\alpha = will\_happen(.,.,T)$. Then β can be either $holds(.,T)$, or $not\_triggered(.,T-1)$. It is clear that α has a time parameter greater or equal to β.
   - Let r be UTA. Hence $\alpha = triggered(.,T)$. Then β can be $will\_happen(.,.,T)$. Obviously α and β have the same time parameter.
   - Let r be HNO. Hence $\alpha = happen(.,T)$. Then β is either $will\_happen(.,.,T)$ or $\phi_{now}(.\,,\,.\,,T)$. It is clear that α and β have the same time parameter.
   - Let r be HNE. Hence $\alpha = will\_happen(.,.,T+1)$. Then β is either $will\_happen(.,.,T)$ or $\phi_{next}(.\,,\,.\,,T)$. It is clear that the time parameter of α is greater than those of β.
iv) Let r belongs to LKB. There are several cases as follows.
   - Let r be CLT. Hence $\alpha = clead(.,.,T)$. Then $\beta = happen(.,T)$. It is clear that α and β have the same time parameter.
   - Let r be CLA. Hence $\alpha = last\_act(.,.,T)$. Then β is either $happen(.,T)$ or $clead(.,.,T)$. It is obvious that α and β have the same time parameter.
   - Let r be CIT. Hence $\alpha = intent(.,.,.,T)$. Then β is either $happen(.,T), clead(.,.,T)$ or $not\_ab_{CIT}(.\,,\,.\,,\,T)$. It is obvious that α and β have the same time parameter.
   - Let r be CAT. Hence $\alpha = attempt(.,.,.,T)$. Then β is either $last\_act(.,.,T)$ or $intent(.,.,.,T)$. It is clear that α and β have the same time parameter.
v) $r \in DS$. Obvious from the definition of DS.

□

Lemma 4. *Suppose M_c is well-structured. All loops in the dependency graph of M_c have* *exactly two negative edges.*

Proof.

i) We first show that if a loop of M_c contains a negative edge then this edge is of the form $\alpha \xrightarrow{-} \delta$ s.t. $\delta \in \overline{\alpha}$ and α is of the form $\phi_{now}(a,T',T)$, *or* $\phi_{next}(a,T',T)$.

Suppose the contrary.

(a) Let *pa* be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where $\alpha = not\_ab_{DCL}(a,f,F,T)$. Hence pa has a structure as illustrated in figure 1.
- From Lemma 3 all nodes in *pa* have the same time parameter.
- DCL is the only rule that has $not\_ab_{DCL}(a,f,F,T)$ appears in its body. Furthermore rules having $ab_{DCL}(a,f,F,T)$ in its head are only rules in the component UB whose bodies are in the form of $holds(.,T)$. Therefore pa has the form as represented in figure 2.
- The possible rules to apply for r_2 are AE, HS, HD or the rules in DS. However if r_2 is AE or HD then the time point will be reduced by one. Contradiction.
- Hence r_2 could only be HS or from DS. Continuing this line of reasoning we can conclude that *pa* has the form as illustrated in figure 3.
- Then AE should be applied as r_1. However the time parameter of its head is $T+1$. Hence AE cannot be applied to *pa*. Contradiction.

Therefore we conclude that $not\_ab_{DCL}(a,f,F,T)$ cannot appear in pa.

(b) Let *pa* be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where $\alpha = not\_ab_{HD}(f,T)$. Then *pa* has a structure as illustrated in figure 4.
- From Lemma 3 all nodes in *pa* have the same time parameter.
- There is only rule HD where $not\_ab_{HD}(f,T)$ appears in its body. Furthermore the only rule having $ab_{HD}(f,T)$ in its head is UHD. Therefore pa has the form as represented in figure 5.

  Let pa_1 be the path $\alpha \xrightarrow{r_2} pa_0 \xrightarrow{r_1} \delta$ in pa

  where α is either $happen(.,T)$, $Rel(.,.,.)$ or $holds(.,T)$ and $\delta = holds(.,T+1)$.

  This path is a contradiction to Lemma 3 (second assertion). Therefore we can conclude that the assumption $not\_ab_{HD}(f,T)$ cannot appear in *pa*.

(c) Let *pa* be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where $\alpha = not\_triggered(a,T-1)$. Hence the only rule having $triggered(a,T)$ in its head is UTA. Therefore pa will be represented as figure 6 and 7.

Let pa_1 be the path $\alpha \xrightarrow{r_2} pa_0 \xrightarrow{r_1} \delta$ in pa

where α is either $will\_happen(.,.,T-1)$ or $trigger(.,.)$ and $\delta = will\_happen(.,.,T)$.

This path is a contradiction to Lemma 3 (second assertion). Therefore we can conclude that the assumption $not\_triggered(a,T-1)$ cannot appear in *pa*.

(d) Let *pa* be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where α is in a form $not\_ab_{CIT}(a,L,T)$. Therefore there is an edge in the loop from a node β labelled by intent(., ., .,T) to α. Thus there is an edge in the loop from a node γ labelled by $attempt(.,.,.,T)$ to β. As literal of the form $attempt(.,.,.,T)$

do not appear in the body of any rule, there is no edge in the loop to γ. Contradiction.

(e) Let pa be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where α is in a form $not\_holds(L)$. Since α appears only in rule CAT and the head of CAT does not appear in the body of any other rules. Therefore there will be no edge from any other nodes to the head of CAT. Contradiction.

(f) Let pa contain only negative edges from the rules in DS. Therefore all rules appear in pa are either from DS or the rule HS. As DS is acyclic, rule HS appears at least one time in pa. Let $pa = \alpha_0, \ldots, \alpha_n, \alpha_0$ and without loss of generality, $\alpha_0 \xrightarrow{\text{HS}} \alpha_1$ be an edge in pa. Therefore $\alpha_0 = holds(G_0, T)$ where G_0 is not empty. From the condition that if a hold-literal appears in rules in DS, they are of the form $hold(f, T)$ where f is a single fluent. Therefore we can conclude that $\alpha_n \xrightarrow{\text{HS}} \alpha_0$ is an edge in pa. Hence α_n is also of the form $holds(G_n)$ such that G_n is a strict super set of G_0. Continuing this line of reasoning, we could conlude that α_i, $i = 0, n, n-1, ..., 0$ are labelled by $holds(G_i, T)$ such that $G_0 \supset G_n \supset G_{n-1} \supset G_1 \supset G_0$. Contradiction.

ii) Let pa be a loop containing a negative edge of the form $\alpha \xrightarrow{-} \delta$ where α is in a form $\phi_{now}(a, T', T)$ or $\phi_{next}(a, T', T)$.

- Let $\alpha = \phi_{now}(a, T', T)$. We show below that δ must be $\phi_{next}(a, T', T)$.
  - If $\delta = T > T'$, there will be no edge out from δ to any other node and hence, pa cannot be a loop. Hence this case cannot happen.
  - Suppose $\delta = \Phi_u(a, T)$. Therefore pa will have a structure as illustrated in figure 8.

    Let pa_1 be the path $\quad \alpha \xrightarrow{r_2} pa_0 \xrightarrow{r_1} \delta \quad$ in pa

    where $\alpha = holds(., T)$ and $\delta = happen(., T)$.

    Rules having $holds(., T)$ in its head are AE, HD, HS or from DS. However AE and HD cannot be applied since they are contradiction to Lemma 3 (second assertion).

    Hence r_2 can be only rule HS or from DS. Continuing this line of reasoning, pa has the form as shown in figure 9.

    There is no rule having $holds(., T)$ in its head and $happen(., T)$ in its body with the same time parameter. Hence this case cannot happen.
  - Therefore δ is $\phi_{next}(a, T', T)$.
- Let $\alpha = \phi_{next}(a, T', T)$. Applying similar reasoning as in the previous case, we can prove that $\delta = \phi_{now}(a, T', T)$.
- Hence we can conclude that pa is of the form $\phi_{next}(a, T', T), \phi_{now}(a, T', T), \phi_{next}(a, T', T)$ consisting of exactly two negative edges.

Therefore we have shown that loops in the dependency graph of M_c contains exactly two negative edges. □

Theorem 2. *Let M_c be well-structured. Then the argumentation frameworks AF_{M_c}, AF_{M_d} are coherent.*

Proof. It follows from Lemma 4 that only loops with two negative edges occurs in the dependency graphs of M_c, M_d. From Lemma 2, it follows that there exist only loops containing two arguments in AF_{M_c}, AF_{M_d}. Dunne & Bench-Capon has proved that an argumentation framework with only loops of even length are coherent (Dunne & Bench-

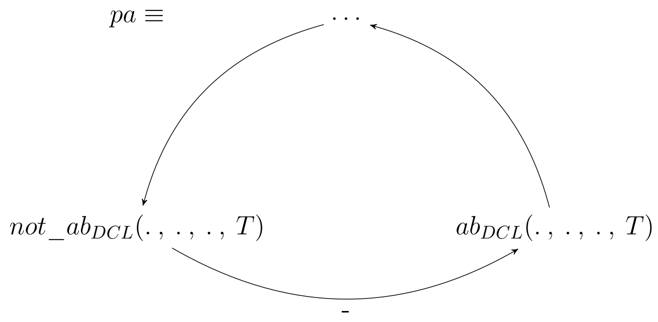

Figure 1.

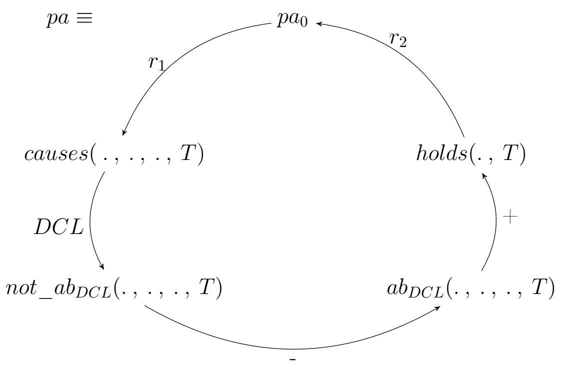

Figure 2.

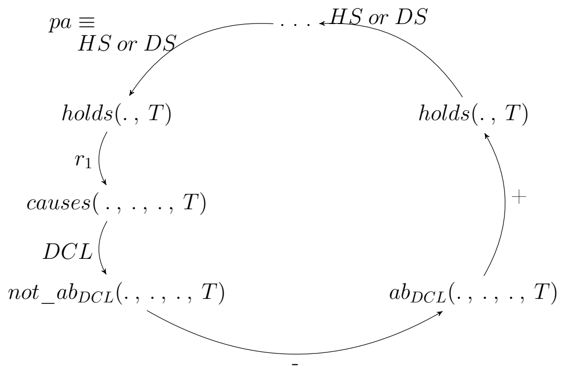

Figure 3.

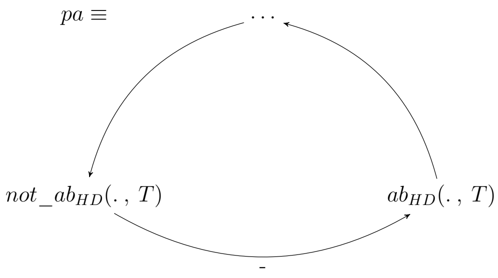

Figure 4.

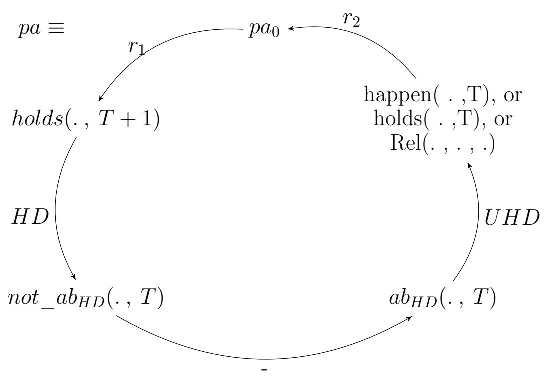

Figure 5.

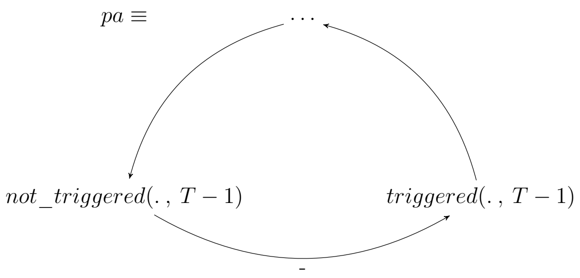

Figure 6.

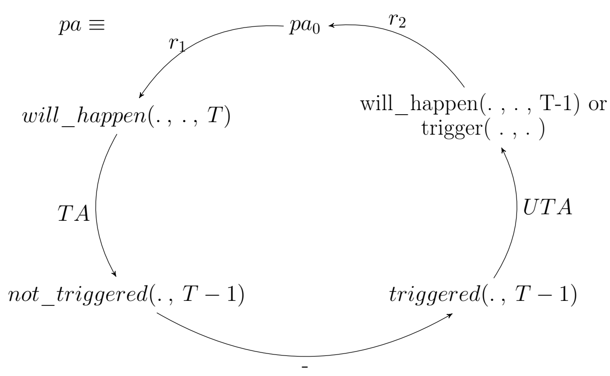

Figure 7.

Capon , 2002). Therefore AF_{M_c}, AF_{M_d} are coherent. □

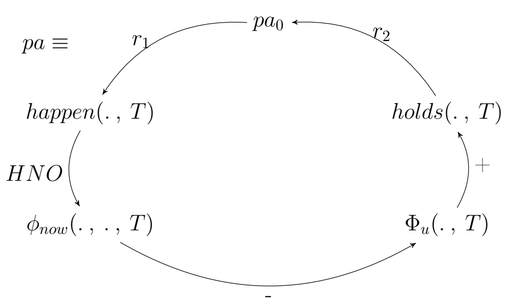

Figure 8.

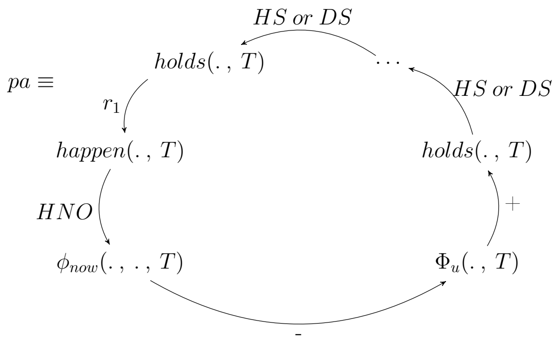

Figure 9.

6. Illustration

We give a detailed representation of the Eagleton case.

Example 2. *R v Eagleton 1855 (Continuation of Example 1)*

We start with a description of the domain of the case.
Relevant Fluents are:

Distributed_ underweight_ bread, Collected_ tickets,
Submitted_ credits, Obtained_ money

which are respectively abbreviated by Db , Ct , Sc , Om .

There are two actions: *submit_ credits* (or *sc* for short) is a proactive action and *pay_ money* (or *pm* for short) is a trigger action in which will be executed whenever the credits are submitted.

The domain rule base of Eagleton case can be defined as RKB=(CAL,DOA,UB,DS) where

- CAL contains the causal rules specifying the effect of actions :

$$DCL_{sc}: \quad causes(sc, Sc, Ct, T) \leftarrow not\_ab_{DCL_{sc}}(sc, Sc, Ct, T)$$

  stating that executing the action *submit_ credits* when the fluent *Collected_ tickets* holds normally causes the fluent *Submitted_ credits* to hold.

$$DCL_{pm}: \quad causes(pm, Om, Sc, T) \leftarrow not\_ab_{DCL_{pm}}(pm, Om, Sc, T)$$

  stating that executing the action *pay_ money* when the fluent *Submitted_ credits* holds normally causes the fluent *Obtained_ money* to hold.

- DOA contains rules specifying the type of basic actions :

$$proactive(sc) \longleftarrow$$

$$trigger(pm, Sc) \longleftarrow$$

  together with a rule defining that distributing underweight bread (Db) and obtaining money (Om) constitutes a criminal offence :

$$offence(\{Db, Om\}) \longleftarrow$$

We assume that the delay time for pm is 1, i.e., $delay(pm, 1) \longleftarrow$

- UB contains the rule :

$$UE: \ \Phi_u(pay\_money, T_0 + 1) \leftarrow holds(Officer\_intervention, T_0 + 1)$$

- DS is empty.

The domain knowledge base of the Eagleton case is structured into 2 modules M_d, M_c where M_d represents the defendant's knowledge base and M_c presents the entire knowledge base of the case.

M_d consists of RKB together with the evidence base BE_d containing the following facts:

- *happen(sc, T_0)*
  stating that the action *submit_ credits* happened at time T_0.
- *holds(Db,T_0), holds(Ct,T_0)*
  stating that the fluents *Distributed_ underweight_ bread*, and *Collected_ tickets* hold at time T_0.
- *executor(Eagleton, Sc,T_0)*
  stating that Eagleton had executed the fluents *Submitted_ credits* at time T_0.

Since the component DS is empty, both M_d and M_c are well-structured and hence their argumentation frameworks are coherent as discussed in Theorem 2. Therefore their prefer extensions are stable.

Applying rule AE with the dynamic causal rule DCL_{sc} together with the fact that the defendant had collected tickets and submitted credits at time T_0 gives us an argument stating that the credits are submitted at time $T_0 + 1$ (figure 10).

$$\begin{array}{ccc}
 & holds(Sc, T_0 + 1) & \\
 & \uparrow AE & \\
happen(sc, T_0) & causes(sc, Sc, Ct, T_0) & holds(Ct, T_0) \\
| & \uparrow DCL_{sc} & | \\
\square & not_ab_{DCL_{sc}}(sc, Sc, Ct, T_0) & \square
\end{array}$$

Figure 10.

The action *pay_ money* (pm) is triggered at time T_0+1 and will happen between time $T_0 + 1$ to $T_0 + 2$ as the delay time limit is 1. The arguments P_0, P_1(figure 11) represent situations when the action *pay_ money* happens at time $T_0 + 1$ or $T_0 + 2$ respectively.

Since $\phi_{now}(pm, T_0+2, T_0+1)$ is a contrary to $\phi_{next}(pm, T_0+2, T_0+1)$ and vice versa, A_1 attacks P_0 at A_0 while A_0 also attacks P_1 at A_1.

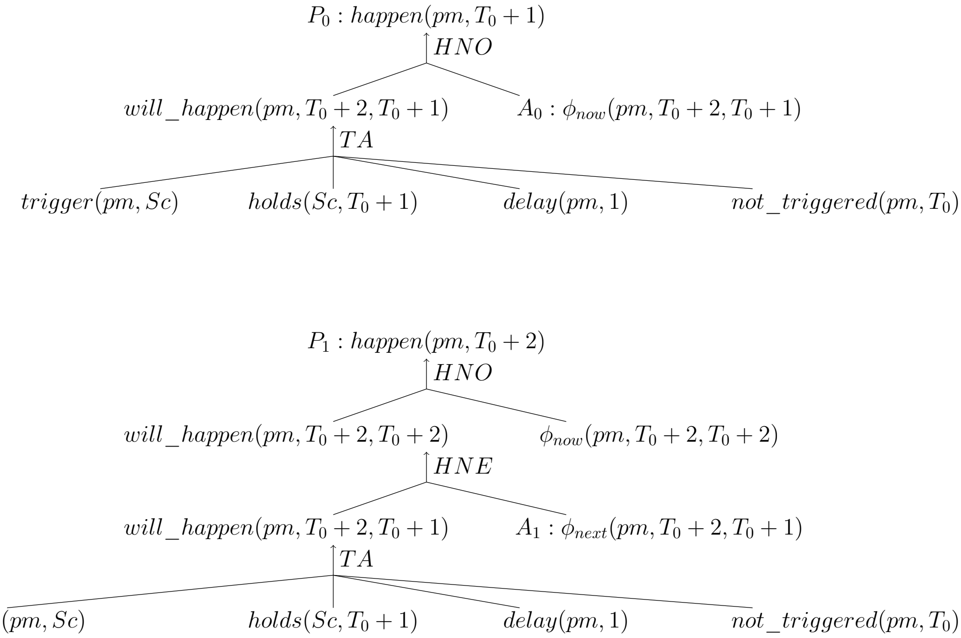

Figure 11.

There are two stable extensions for M_d, one (referred to as E_0) contains P_0 and the other (referred to as E_1) contains P_1.

The defendant had distributed underweight bread at time T_0 which causes the fluent *holds(Db,T_0)* to hold. Due to its inertia the fluent still holds as long as there exists no execution of any other action that causes its contraries. Hence applying rule HD we get argument I_2, I_3 representing *holds(Db,T_0+2)* and *holds(Db,T_0+3)* as in figure 12.

It is not difficult to see that the stable extension E_0 and E_1 also contains respectively argument Q_0 and Q_1 as illustrated in figure 13 and 14.
Hence we can conclude $M_d \mathrel{|\!\sim} holds(\{Db, Om\})$.

Therefore $call(holds(\{Db, Om\}), M_d)$ holds in M_c.

If submitting credits does not happen, we cannot establish an argument with a conclusion stating $holds(Sc, T_0 + 1)$. Hence we cannot derive $holds(\{Db, Om\})$ in $M_d \backslash \{happen(sc, T_0)\}$.

Thus $call(not\_holds(\{Db, Om\}), M_d \backslash \{happen(sc, T_0)\})$ holds in M_c.

Hence applying rule CLT we can conclude $clead(sc, \{Db, Om\}, T_0)$.

M_c consists of $RKB \cup LKB$ and the evidence base $BE_d \cup BE_u$ where BE_u contains the unexpected fact:

- $holds(Officer\_intervention, T_0 + 1)$
  stating that the fluent Officer_intervention hold at time $T_0 + 1$.[12]

[12]I.e.,for simplicity we assume that the officer intervened immediately after the defendant had submitted credits.

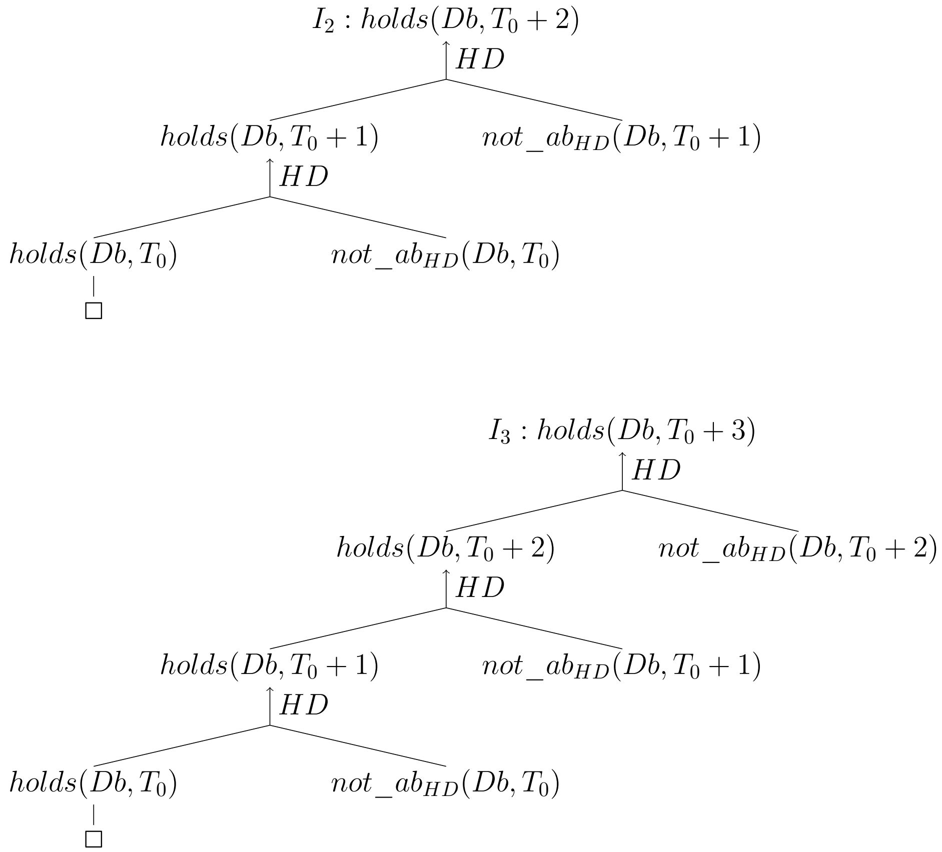

Figure 12.

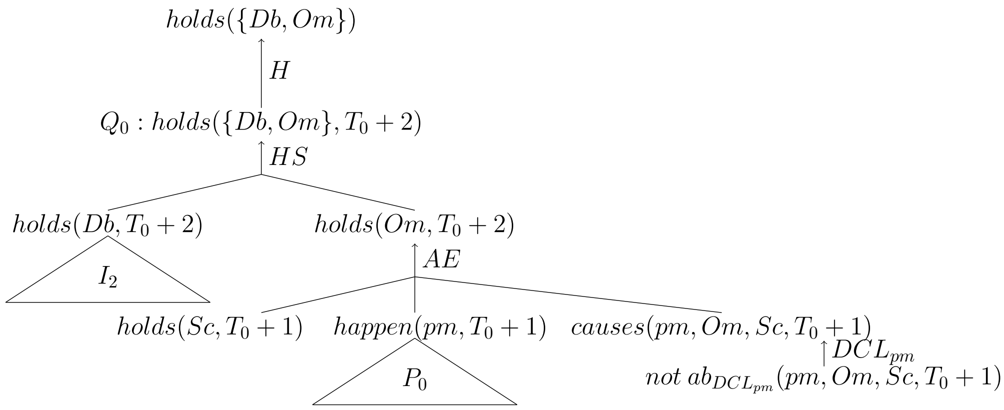

Figure 13.

Applying rule CLA we can conclude $last\_act(Eagleton, sc.pm, \{Db, Om\}, T_0)$.

Applying rule CIT we can conclude $intent(Eagleton, sc, \{Db, Om\}, T_0)$ as there is no argument undercutting the assumption of this rule.

In figure 15 we give argument stating that the officer intervention undercut the execution of paying money at time $T_0 + 1$. Hence both P_0 and P_1 are undercut. Therefore there exists only one stable extension wrt M_c where neither P_0 nor P_1 is accepted. Obviously there is no acceptable argument supporting $holds(\{Db, Om\})$.

Therefore we can apply rule CAT and we can conclude

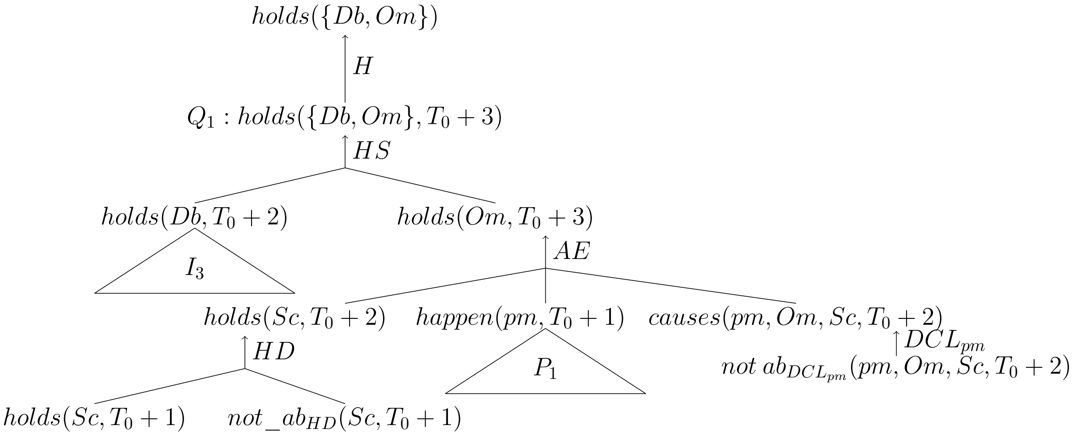

Figure 14.

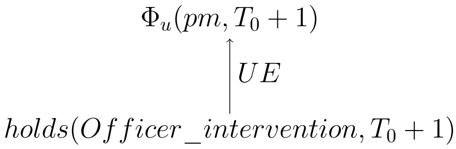

Figure 15.

attempt(Eagleton, sc.pm, { Db, Om}, T_0)

stating that Eagleton attempted to commit the offence of obtaining money by distributing underweight bread.

Let us apply our model to another important case where the notion of the last-act was applied. In

Rex v. Scofield (Duff, 1996; Scofield, 1784) the defendant intentionally placed lighted candle with combustible material to burn his rented house but the larger fire did not happen. The defendant was convicted for attempted arson.

The formal model of the case of R v. Scofield could be modelled as follows.

$M_d = (RKB, BE_d)$, $M_c = (RKB \cup LKB, BE_u)$ with RKB = (CAL,DOA,UB,DS) where

(i) CAL contains only one rule :

$$causes(light\_candle, Burn\_house, Combustible, T) \longleftarrow$$
$$not\_ab_{DCL}(light\_candle, Burn\_house, Combustible, T)$$

where Combustible is a fluent representing the fact that combustible material had been put in the house.

(ii) DOA contains rules in RKB[13] together with the following rule specifying the type of the light_candle action:

$$proactive(light\_candle) \longleftarrow$$

together with a rule defining that burning a house is a criminal offence :

[13]To be precise, they are the rules CA, HS, H, AE, HD, UHD, TA,UTA, HNE, and HNO.

$offence(Burn\_house) \longleftarrow$

(iii) UB consists of one undercut rule :

$ab_{DCL}(light\_candle, Burn\_house, Combustible, T) \longleftarrow holds(P, T)$[14]

(iv) DS is empty.

(v) BE_d contains the following facts:

$happen(light\_candle, t_0)$, $executor(Scofield, light\_candle, t_0)$ and $holds(Combustible, t_0)$.[15]

(vi) BE_u contains one fact which is $holds(P, t_0)$.

It is clear that the defendant has already done the last-act and his last-act causally leads to a crime of arson (i.e., $M_d \mid\!\sim holds(Burn\_house)$ and $M_d \backslash \{light\_candle\} \not\mid\!\sim holds(Burn\_house)$).

7. Related Research

Different strands of research in are relevant to our work.

7.1 *Argument-based models of rules and cases*

A number of studies have addressed legal reasoning through formal argumentation, within the AI and law research (for a review, see (Prakken & Sartor, 2015)). Norm-based legal reasoning has been modelled by viewing legal norms as defeasible inference rules (Prakken, 1997), or as inputs to defeasible inference schemes (Gordon & Walton, 2006): the arguments based on such rules or schemes are defeasible, being subject counterarguments (Sartor, 2018). The latter approach, namely, viewing legal rules as inputs to inference pattern was anticipated by Hage's reason-based logic (Hage, 1997).

Precedent-based legal reasoning has been modelled through argumentation, where rationes decidendi of precedents can be applied, directly or analogically to new cases or challenged through distinctions (see (Horty, 2011; Prakken & Sartor, 1998)). Also factual reasoning has been approached through argumentation, considering that evidential inferences support defeasible conclusions, since such inferences are based on argument schemes that can be challenged by arguments to the contrary (Bex & Prakken & Reed & Walton , 2003).

Our model relates to such approaches, being also based on defeasible argumentation, but has a different focus, since rather then modelling legal rules or cases, we address those inferences pertaining to the normal course of human actions, once a plan has been set in movement, as needed to capture the relationship between an attempt and the corresponding completed crime, according to the last-act approach.

[14]We could not find any material specifying why the house was not burned. Therefore P represents the special circumstances in this case that prevented the house to be burned down.
[15]We assume that the action happened at a constant time t_0.

7.2 *The event calculus*

In a way, the knowledge bases in our approach could be viewed as a kind of assumption-based modular argumentation system for reasoning about events. Naturally, well-known systems for event calculus (Artikis & Sergot & Paliouras , 2015, 2014; Hadjisoteriou & Kakas, 2012; Kowalski & Sergot, 1986; Shanahan, 1999) could also be applied to represent our knowledge bases as long as they could be extended to allow modularity. However, we believe that a semantic based on argumentation better captures legal reasoning than other approaches. Further the concept of triggered events in (Shanahan, 1999) where once triggered the actions will happen immediately, could be viewed as a special case of our concept of trigger actions that allow delay time. On a more technical level, the axioms underlining the event calculus systems like (Shanahan, 1999) are distinct from ours. For example, we allow *causes-predicate* of the form $causes(a, f, F, T)$ stating that executing action a under the conditions F causes f. To capture this, in event calculus, we would need a clause like $Initiates(a, f, T)\, if\, holds(F, T)$. The *clipped-predicate* is closely related to the abnormal predicate ab_{HD} though again they are different technically. A formal study of the correspondence between the two approaches would be interesting but it is beyond the scope of this paper as the focus of the paper is on modelling the last-act-attempt.

7.3 *Evidential reasoning*

A number of contributions in AI& law have addresses the issues of evidence, using different frameworks: probabilistic, story-based, and argument-based approaches has been proposed as well as attempts to integrate the three approaches ((Bex , 2011; Verheij et al., 2016)).

Our approach may be linked both to stories and to argumentation, since it provides for scenarios of assumed context, in which causal arguments build a story explaining production of criminal results.

There are extensive amount of research in the literature on reasoning about mental states of agents such as beliefs, desires and intentions. This line of research is especially about the relationship between an agent's intention and his future actions and plans (Baral & Gelfond, 2005; Blount & Gelfond, 2012; Cohen & Levesque, 1990; Lorini & Herzig, 2008; Rao & Georgeff, 1991; Wooldridge, 2002). However in attempted crime we are only concerned about intention of past actions of the defendant as we have studied in this paper.

7.4 *Actual Causality*

In our model proposed in sections 3-4 we consider cases where the criminal offence had not been materialized. In principle, it is imaginable that a defendant has done the last act to commit a crime and the crime somehow also happened but not because of the defendant's act (though we do not find any real court case of last-act attempt similar to this scenario). In other words, the action of the defendant has been preempted. In this case, the law of last-act attempt still applies and the defendant should be convicted of attempted crime. As the rule CAT can not be applied in this case, the question is how the rule CAT could be generalized to deal with cases like this ?

Let us look at this problem through a well-known story adapted from (Halpern, 2016).

Assassins A and B were out to get victim V. Assassin A shot V with a gun and B shot with a bow. V died after being shot by A. B would have killed V had V not died from A's shot. Later B was arrested. How should B be tried? Should he be convicted of murder or attempted murder?

To handle such cases, we would need a test to check whether B's shot is an actual cause of V's death. The most well-known test for actual causality is the but-for-test. It is not difficult to see that the but-for-test can not determine whether A's shot or B's shot is the cause of V's dead in our story. Hence more general tests are needed. There are many proposals of such tests. Halpern and Pearl (Halpern, 2016, 2015; Halpern & Pearl, 2005; Pearl, 2000) have proposed three distinct tests for actual causality referred to by Halpern (Halpern, 2016) as the original, updated and modified tests, based on the structural equation model of events. Though all three tests deliver the correct result for the example above (i.e A's shot is the actual cause of V's death and B's shot is not), they produce distinct results in many cases (Halpern, 2016). There are cases (Halpern, 2016) where some of the tests provide rather unnatural conclusion.

More recently Bochman has advanced a test based on the NESS concept (Bochman , 2016). While it provides intuitive conclusion in many cases, there are also rather unnatural conclusions. A recent work of Liepina&Sartor&Wyner (Liepina & Sartor & Wyner , 2019) suggests an informal argument-based approach to causality in law that in some way could be viewed as the NESS approach, though again no algorithmic test for causality in law has been proposed.

As there are many proposals for actual causality tests, a judge in a court case would have to pick (or invent) one test she/he considers the most appropriate for the case. To capture this situation, we introduce a new rule CATN where the test for actual causality is left as a parameter:

$$\text{CATN}: \ \mathit{attempt}(D,a,L,T) \leftarrow \mathit{last\_act}(a,L,T),\ \mathit{intent}(D,a,L,T),\ \mathit{holds}(L)$$
$$\mathit{call}(NAC, a, M_d \cup BE_u, L)$$

where $\mathit{call}(AC, a, M_d \cup BE_u, L)$ holds iff a is an actual cause of L wrt event structure represented by $M_d \cup BE_u$. Here AC represents a module specifying an actual causality test.[16]

As it remains an open and challenging problem whether and how the tests studied in the literature (Bochman , 2016; Halpern, 2016, 2015; Halpern & Pearl, 2005; Pearl, 2000) could and should be applied in legal contexts and how such tests should be adapted to knowledge representation frameworks other than the structural equation model (Liepina & Sartor & Wyner , 2019), we leave the problem of finding an appropriate test for actual causality in legal contexts for future works.

8. Conclusion

The main contribution of this work is to model the last-act attempt law. The key idea is to use modular assumption-based argumentation to model two different knowledge bases representing defendant's knowledge when he carried out his last act and the real situation of the case. We evaluate our model by applying it on some real court cases and by showing that the model is coherent under a general condition of well-structuredness.

The last-act attempt is only the strictest test among other tests of the law of criminal attempt. It would be interesting to see how the structure we proposed in this paper will be customized to capture other liberal tests. We plan to look at this problem as a future work.

[16] AC could be written in any language, say Java or Python ect.

T

References

Alechina, N., & Halpern, J., & Logan, B. (2017). Causality, Responsibility and Blame in Team Plans. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, (pp. 1091–1099). International Foundation for Autonomous Agents and Multiagent Systems.

Allen, M. (2011). *Textbook on Criminal Law* (11st ed.). OUP Oxford.

Andenaes, J. (1965). *General Part of the Criminal Law of Norway.* Fred B Rothman & Co.

Artikis, A., & Sergot, M., & Paliouras, G. (2015). An event calculus for event recognition. In *IEEE Transactions on Knowledge and Data Engineering*,(pp. 895–908).

Artikis, A., & Sergot, M., & Paliouras, G. (2014). Reactive reasoning with the event calculus. In *Proceedings of the 21st International Workshop on Reactive Concepts in Knowledge Representation (ReactKnow 2014)*, (pp. 9–15). Prague, Czech Republic, August 18-22.

Baral, C., & Gelfond, M. (2005). Reasoning About Intended Actions. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, (pp. 689–694). AAAI Press.

Bex F.J. (2011). Arguments, Stories and Criminal Evidence: A Formal Hybrid Theory, Springer, Dordrecht.

Bex, F., & Prakken, H., & Reed, C., & Walton, D. N. (2003). Towards a Formal Account of Reasoning about Evidence: Argumentation Schemes and Generalisations. *Artificial Intelligence and Law*, *11*, 125–65.

Becker, C. L. (1974). Criminal Attempt and the Theory of the Law of Crimes. *Philosophy & Public Affairs*, *3*, 262–294.

Blount, J., & Gelfond, M. (2012). Reasoning about the intentions of agents. In *Logic Programs, Norms and Action, volume 7360 of Lecture Notes in Computer Science*, (pp. 147–171). Springer Berlin / Heidelberg.

Bochman, A. (2018). Actual Causality in a Logical Setting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*,(pp. 1730–1736).

Bondarenko, A., & Dung, P. M., & Kowalski, R., & Toni, F. (1997). An Abstract, Argumentation-Theoretic Approach to Default Reasoning. *Artificial Intelligence*, *90*, 63–101.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, *42*, 213–261.

Colvin, E.(1989). Causation in Criminal Law. *Bond Law Review*, *1(2)*.

Commonwealth v. Niziolek. (1980). 380 Mass. 513 , 528. Retrieved June 2016, from http://masscases.com/cases/sjc/380/380mass513.html

Duff, R.A. (1996). *Criminal Attempts.* Clarendon Press Oxford.

Dunne, P. E., & Bench-Capon, T.J.M. (2002). Coherence in Finite Argument Systems. *Artificial Intelligence*, *141*, 187–203

Dung, P. M. (1995a). An Argumentation Theoretic Foundation for Logic Programming. *Journal of Logic Programming*, *22*, 3151–177.

Dung, P. M. (1995b). On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, *77*, 321–357.

Dung, P. M. (1993). Representing Actions in Logic Programming and Its Applications in Database Updates. In *Proceedings of the Tenth International Conference on Logic Programming*, (pp.222–238). The MIT Press.

Dung, P. M., & Kowalski, R., & Toni, F. (2006). Dialectic proof procedure for assumption based admissible argumentation. *Artificial Intelligence and Law, 170(2-3)*, 114–159.

Dung, P. M., & Sartor, G. (2011). The modular logic of private international law. *Artificial Intelligence and Law, 19(2-3)*, 233–261.

Dung, P. M., & Thang, P. M. (2008). Modular Argumentation For Modelling Legal Doctrines in Common Law of Contract. In *JURIX08, Frontiers in Artificial Intelligence and Applications*, (pp.108–117). IOS Press.

Dung, P. M., & Thang, P. M., & Hung N. D. (2010). Modular Argumentation for Modelling Legal Doctrines of Performance Relief. *Argument & Computational Journal, 1*, 47–69.

Fletcher, George P. (1998). *Basic Concepts of Criminal Law.* Oxford University Press.

Gaertner, D., & Toni, F. (2008). Hybrid Argumentation and Its Properties. In *Proceedings of the 2008 Conference on Computational Models of Argument: Proceedings of COMMA 2008*, (pp.183–195). IOS Press, Amsterdam.

Gelfond, M.,& Baral, C. (2000). Reasoning Agents In Dynamic Domains. In *Logic-Based Artificial Intelligence*, (pp.257–279). Springer US.

Gelfond, M.,& Lifschitz, V. (1998). Action Languages. *Electronic Transactions on AI, 3*.

Gelfond, M.,& Lifschitz, V. (1993). Representing action and change by logic programs. *The Journal of Logic Programming, 17*,301 – 321.

Gordon, T. F., & Walton, D. N. (2006). The Carneades Argumentation Framework - Using Presumptions and Exceptions to Model Critical Questions. In *Proceedings of the First International Conference on Computational Models of Argument (COMMA-06).*

Hadjisoteriou, E., & Kakas, A. (2012). Argumentation and the Event Calculus. *Logic Programs, Norms and Action*, 103–122.

Hage, J. C. (1997). *Reasoning with Rules: An Essay on Legal Reasoning and Its Underlying Logic.* Kluwer.

Hall, N. (Hall). Structural Equations and Causation. *Philosophical Studies, 132(1)*, 109–136.

Halpern, Y. J. (2016). *Actual Causality.* Cambridge, MA: MIT Press.

Halpern, Y. J. (2015). A modification of the Halpern-Pearl Definition of Causality. In *International Conference on Principles and Practice of Multi-Agent Systems*, (pp.20–39). Springer International Publishing.

Halpern, Y. J., & Pearl, J. (2005). Causes and Explanations: A Structural-Model Approach Part I: Causes. *The British Journal for the Philosophy of Science, 56*, 843–887.

Hitchcock, C. (2009). Structural Equations and Causation: Six Counterexamples. *Philosophical Studies, 144(3)*, 391–401.

Horty, J. F. (2011). Rules and reasons in the theory of precedent. *Legal theory, 10*, 1–33.

Kolkorn, M. (2016). *Representing and Reasoning with Custom Law in Logic Programming.* Unpublished Master Thesis, Asian Institute of Technology, TH.

Kowalski, R.,& Sergot, M.(1986). A logic-based calculus of events. *New Generation Computing, 4*,67–95.

LaFave, Wayne R. (2017). *Criminal Law* (6th ed.). St. Paul, Minn:West Academic Publishing.

Liepina, R.,& Sartor G.,& Wyner A. (2019). Arguing about causes in law: a semi-formal framework for causal arguments. *Artificial Intelligence and Law.*

Lorini, E.,& Herzig, A.(2008). A logic of intention and attempt. *Syn- these, 163*, 45–77.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge UP,2000.

Pooksook, J.,& Dung, P. M.,& Satoh, K. (2016). Modeling Attempted Crime in Criminal Law. In *Proceedings of the 10th International Workshop on Juris-Informatics*

(JURISIN 2016), (pp. 59–72). Yokohama, Japan.
Prakken, H. (1997). *Logical Tools for Modelling Legal Argument: A Study of Defeasible Reasoning in Law.* Kluwer.
Prakken, H., Sartor, G. (1998). Modelling Reasoning with Precedents in a Formal Dialogue Game. *Artificial Intelligence and Law*, *6*, 231–87.
Prakken, H., Sartor, G. (2015). Law and logic: A review from an argumentation perspective. *Artificial Intelligence*, *227*, 214–45.
R v. Eagleton. (1855). 6 Cox C.C. 559. Retrieved August 2016, from http://www.casebriefs.com/blog/law/criminal-law/criminal-law-keyed-to-lafave/attempts/regina-v-eagleton/ and http://swarb.co.uk/lisc/Crime18491899.php
R v. White. (1910). 2 KB 124, 4 Cr App R 257. Retrieved January 2019, from https://swarb.co.uk/rex-v-white-1910/ and https://www.lawteacher.net/cases/r-v-white.php
R v. Stephenson. (1979). EWCA Crim 1. Retrieved September 2017, from http://www.bailii.org/ew/cases/EWCA/Crim/1979/1.html
Rao, A. S.,& Georgeff, M. P. (1991). Modelling rational agents within a BDI-architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann Publishers.
Rex v. Scofield. (1784). Cald 397, 403. Retrieved July 2018, from http://www.okcca.net/online/oujis/oujisrvr.jsp?oc=OUJI-CR%202-14
Sanstrom v. Montana. (1979). 442 U.S. 510. Retrieved June 2016, from https://scholar.google.co.th/scholar_case?case=5244531001775040757
Sartor, G. (2018). Defeasibility in Law. In *Handbook of Legal Reasoning and Argumentation*, Springer, 315–64.
Shanahan, M. (1999). The event calculus explained. *Artificial Intelligence Today,*, (pp.409–430). Springer-Verlag.
State v. Wilkinson. (1998). 724 so.2d 614. Retrieved April 2017, from https://caselaw.findlaw.com/fl-district-court-of-appeal/1265311.html
SWI-Prolog Foreign Language Interface. Retrieved April 2019, from http://www.swi-prolog.org/pldoc/man?section=foreign
The U.S. Court Role. Retrieved October 2016, from http://www.uscourts.gov/about-federal-courts/court-role-and-structure
Verheij, B., Bex, F., Timmer, S. T., Vlek, C. S., Meyer, J-J. C., Renooij, S., & Prakken, H. (2016). Arguments, scenarios and probabilities: connections between three normative frameworks for evidential reasoning. *Law, Probability & Risk*, *15*, 30–70.
Williams, G. L. (1978). *Textbook of criminal law*(pp.542–554). London, UK:Stevens.
Wooldridge, M. (2002). *An Introduction to Multiagent Systems.* John Wiley & Sons.