The Law of Inertia and the Frame Problem in Attempto Controlled English

Norbert E. Fuchs
Department of Computational Linguistics
University of Zurich
fuchs@ifi.uzh.ch
http://attempto.ifi.uzh.ch

Abstract

Daily experience teaches us that a situation remains unchanged unless somebody or something changes it. Leibniz called this experience the law of inertia. Early attempts to formalise the law of inertia failed because they offered no easy way to describe that after a partial change of a situation the unaffected rest remains unchanged. This so-called frame problem was efficiently solved by later approaches, specifically by the event calculus and the default logic. Focusing on default logic, I will show that it can express the law of inertia not only in first-order logic, but also quite naturally in Attempto Controlled English. Furthermore, I will use the Attempto reasoner RACE to efficiently reason with the law of inertia.

1 The Common Experience of Inertia

When you return to your office in the morning you expect to find the items on your desk in exactly the same order as you left them in the evening before – unless somebody or something moved them. This common experience was expressed by the philosopher Leibniz as the law of inertia: "Everything is presumed to remain in the state in which it is." (Leibniz, 1679). At about the same time Newton published his three laws of motion, the first of which expresses the specific case of the law of inertia for moving physical bodies (Newton, 1687).

I will show that Leibniz' law of inertia can be formalised in Attempto Controlled English (ACE)[1] and that this formalisation allows us to reason with the law. In section 2, I describe early attempts to formalise common sense, specifically the law of inertia, the encountered frame problem, and a variant of the Yale Shooting Problem. Section 3 presents two solutions of the frame problem, the event calculus and the default logic. In section 4, I express the default logic in ACE. Section 5 describes reasoning with the law of inertia using the Attempto reasoner RACE[2]. Section 6 shows that incorporating the law of inertia into RACE facilitates the reasoning. Section 7 revisits the Yale Shooting Problem in ACE/RACE. Section 8 summarises the paper and briefly addresses the fact that the law of inertia and, specifically, its formalisations are asymmetric with respect to time.

2 Formalising Common Sense and the Frame Problem

Beginning in the 1960s researchers began to formalise common sense, predominantly in first-order logic. When trying the express the law of inertia they encountered the so-called frame problem, that is how to effectively and efficiently describe the unaffected part of a situation after a partial change.

Initial attempts to solve the frame problem failed, mostly because only the changed parameters were taken into account and the unchanged parameters ignored.

The shortcomings were strikingly demonstrated by the impossibility to adequately solve the so-called "Yale Shooting Problem" (Hanks and McDermott, 1987). In this paper I replace the original problem by a less violent, but "problem-identical" one.

[1] http://attempto.ifi.uzh.ch/

[2] http://attempto.ifi.uzh.ch/race/

Initially, a wine glass is empty and a wine bottle is not open. Opening the bottle, waiting a moment to read the label, and then pouring the wine should fill the glass. If this situation is formalised in first-order logic by only taking the changed parameters into account and ignoring the unchanged ones, it cannot be uniquely proved that the wine glass is finally full. In one logical solution the wine glass is actually full; in another logical solution the wine bottle is again not open and the wine glass remains empty.

Here is a simple formalisation[3] of my version of the Yale Shooting Problem using four time points 0, 1, 2, 3 and the two fluents – conditions that can change their truth value over time – *empty* and *open* expressed by the following first-order formulas:

$$empty(0) \quad (1)$$

$$\neg\, open(0) \quad (2)$$

$$true \rightarrow open(1) \quad (3)$$

$$open(2) \rightarrow \neg\, empty(3) \quad (4)$$

As it turns out, two evaluations of the fluents are consistent with the formulas, the first one

{empty(0), ¬ open (0)}, {empty(1), open (1)}, {empty(2), open (2)}, {¬ empty(3), open (3)}

describing the intended behaviour that the glass is not empty at time 3, the second one

{empty(0), ¬ open (0)}, {empty(1), open (1)}, {empty(2),¬ open (2)}, {empty(3),¬ open (3)}

describing the non-intended behaviour that the bottle strangely is not open again at time 2 and the glass is empty at time 3.

The problem is that the formulas only describe the changes and that they do not specify that fluents unaffected by the actions remain unchanged. Several so-called frame axioms – e.g. *empty(1) = empty(0)* – would be needed to restrict the solutions to the one with the intended behaviour.

3 Solutions of the Frame Problem

In the end, several correct solutions for the frame problem and the Yale Shooting Problem were developed. Among the solutions that express Leibniz' law of inertia directly are the Event Calculus (Kowalski and Sergot, 1986) and the Default Logic (Reiter, 1980).

The Event Calculus uses the following conceptualisation: linear time, fluents F that hold or do not hold at time points, and events E that happen at time points and initiate or terminate fluents. There is one domain-independent axiom of inertia

holdsAt(F, T) ← happens(E1, T1) ∧ initiates(E1, F) ∧ T1< T ∧ ¬ ∃ E2, T2 [happens(E2, T2) ∧ terminates(E2, F) ∧ T1< T2 ∧ T2< T (5)

with the meaning "The fluent F holds at a time T if an event E1 happens at a time T1 before T and E1 initiates F and there is no event E2 and there is no time T2 between T1 and T so that E2 happens at T2 and E2 terminates F."

To describe a concrete situation, a set of domain-dependent axioms would be needed to specify the actual fluents and the actual events that initiate and terminate the fluents.

The Default Logic – which will be used in this paper – relies on non-monotonic logic with assumptions and exceptions expressed as default inference rules of the form

$$\frac{prerequisite : justification}{conclusion} \quad (6)$$

meaning "If the prerequisite is true and the justification is consistent with the known facts then the conclusion can be drawn."

As a concrete example here is the default inference rule for the law of inertia

$$\frac{r(X,S) : r(X,\ do(A,S))}{r(X,\ do(A,S))} \quad (7)$$

paraphrased as "If r(X) is true in a situation S and it can be assumed that r(X) remains true after the action A is applied to S then r(X) remains true."

Being inspired by (Erdem et al., 2016), I reformulated (Equation 7) as an implication using logical negation (¬) and negation as failure (*not*).

r(X,S) ∧ not (¬ r(X, do(A,S))) → r(X, do(A,S)) (8)

with the paraphrase "If r(X) is true in a situation S and it is not provable that r(X) is false after the action A is applied to S then r(X) remains true."

[3] adapted from https://en.wikipedia.org/wiki/Yale_shooting_problem

Replacing the situation parameter by time points we get

$r(X,T1) \wedge T2>T1 \wedge not\ (\neg\ r(X,T2)) \rightarrow r(X, T2))$ (9)

that is "If r(X) is true at a time T1 and there is a later time T2 and it is not provable that r(X) is false at T2 then r(X) is true at T2."

4 Default Logic in Attempto Controlled English

Default logic's basic axiom for inertia (Equation 9) can be paraphrased in Attempto Controlled English (ACE) as

If something X holds at a time T1 and there is a time T2 that is after T1 and it is not provable that X does not hold at T2 then X holds at T2. (10)

Note that I introduced a practically identical axiom when briefly discussing the frame problem in (Fuchs, 2016).

As a concrete example let's model the situation of a sleeping person that can be expressed in English as "If a person falls asleep and the person does not wake then the person continues to sleep.", ignoring the duration and nature of human sleep to solely focus on the law of inertia. Using verbs of the sleep example (*fall asleep*, *wake*, *sleep*) directly instead of the verb *hold* we can customise (Equation 10) for the sleep example

If a person falls asleep at a time T1 and a time T2 is after T1 and it is not provable that the person wakes at T2 then the person sleeps at T2. (11)

or simply using only the verb *sleep*

If a person sleeps at a time T1 and a time T2 is after T1 and it is not provable that the person does not sleep at T2 then the person sleeps at T2. (12)

5 Reasoning with in Attempto Controlled English

Having default logic's basic axiom for inertia expressed in ACE we may want to reason with it. There are several reasoners for ACE of which I will use the Attempto reasoner RACE (Fuchs, 2012; Fuchs, 2016). RACE can

- determine the (in-) consistency of an ACE text
- deduce one ACE text from another one
- answer an ACE query from an ACE text

RACE is supported by about 100 auxiliary axioms – expressed in Prolog, using ACE's internal logical notation – that provide domain-independent knowledge like the relations between plural and singular nouns, the relations between numbers, the substitutions for query words, and much else. In spite of their large number auxiliary axioms have little impact on RACE's performance since they are only called individually and only when needed.

RACE is implemented in Prolog, has a web-service and a web-interface whose output window will be used in the following.

RACE has a restriction relevant for the problem at hand: For technical reasons RACE does not accept logical negation within the scope of negation as failure (*it is not provable that ... not ...*). Thus, we will have to replace negated verbs within negation as failure by verbs that express the intended negation, e.g. *does not sleep* → *wakes*.

Reasoning Examples: Continuous Sleep and Interrupted Sleep. In the following RACE will prove that a person whose sleep is not interrupted will continue to sleep, while a person whose sleep is interrupted does no longer sleep.

First the case of continuous sleep. To the axiom of the law of inertia of a sleeping person (Equation 11) two axioms are added to describe a concrete situation: one axiom to introduce a person sleeping at an initial time and a second axiom to introduce a later time. The theorem checks whether the person will sleep at the later time.

If a person sleeps at a time T1 and a time T2 is after T1 and it is not provable that the person wakes at the time T2 then the person sleeps at the time T2.

A person sleeps at an initial time. A later time is after the initial time.
⊢
A person sleeps at a later time.

Submitting this reasoning example to RACE's web-interface we get the expected result (Figure 1) that RACE proves that the person sleeps at a later time. To present the result, I use a screenshot of the output window of RACE's web-interface. This window contains the axioms, the theorem, and the minimal subset of the axioms needed to prove the theorem. The entry "parameters" is used for testing.

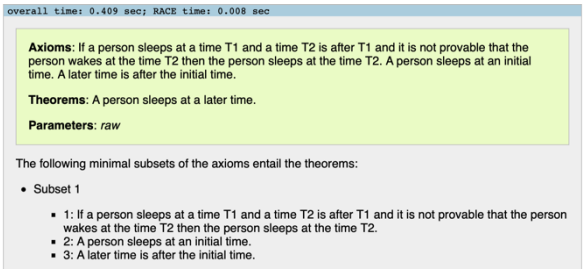

Figure 1: Continuous Sleep

Following is the case of interrupted sleep. To the axiom of the law of inertia of a sleeping person (Equation 11) four axioms are added: one axiom relates waking to not sleeping, one axiom introduces a person sleeping at an initial time, one axiom introduces a later time, and one axiom states that the person wakes at the later time. The theorem checks whether the person will sleep at the later time.

If a person sleeps at a time T1 and a time T2 is after T1 and it is not provable that the person wakes at the time T2 then the person sleeps at the time T2.

If a person wakes at a time T then the person does not sleep at the time T.

A person sleeps at an initial time. A later time is after the initial time. The person wakes at the later time.

⊢

A person sleeps at a later time.

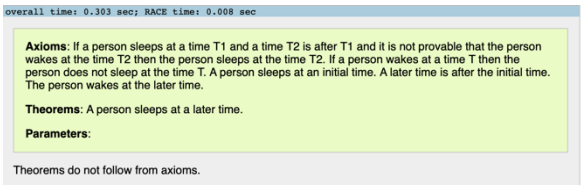

Figure 2: Interrupted Sleep

Submitting this reasoning example to RACE we get the expected result (Figure 2) that RACE cannot prove that the person sleeps at the later time.

6 Incorporating the Law of Inertia into RACE

For a complex situation involving many fluents we would have to formulate for each fluent a separate ACE axiom of inertia thus blowing up the axiomatisation of the situation. Going back to ACE's general axiom of inertia (Equation 10) offers no solution since for each fluent of the concrete situation we would have to introduce bridging axioms – as in the event calculus – thus again creating a blow-up. Instead, I decided to incorporate an abstract version of (Equation 10) as three auxiliary Prolog axioms into RACE: one axiom for intransitive verbs, one axiom for transitive verbs, and one axiom for the copula plus adjective. This threefold division is necessary since these verbs have different internal representations. There is no axiom for ditransitive verbs because they do not seem to cause the frame problem. By abstracting away from the details of the fluents – concretely replacing nouns, verbs etc. by variables – these three auxiliary axioms can cover any fluent. One could say that RACE now "knows" the law of inertia in the same way as it "knows" the relation between plural and singular nouns.

There is an unexpected bonus: the new auxiliary axioms can deal with logical negation within the scope of negation as failure (*it is not provable that ... not ...*), eliminating the need to replace negated verbs by other verbs.

Let us now return to the previous examples.

Reasoning Examples: Continuous Sleep and Interrupted Sleep Revisited. In the following RACE will again prove that a person whose sleep is not interrupted will continue to sleep, while a person whose sleep is interrupted does no longer sleep.

Note that the verb *wake* of the initial example can now be replaced by *not sleep*.

Further note that in both cases the ACE axiom expressing inertia (Equation 12), is no longer necessary for the proof. However, it is left crossed out as a reminder to the reader.

Following is the case of the continuous sleep.

~~*If a person sleeps at a time T1 and a time T2 is after T1 and it is not provable that the person does not sleep at the time T2 then the person sleeps at the time T2.*~~

A person sleeps at an initial time. A later time is after the initial time.

⊢

A person sleeps at a later time.

We get the expected result (Figure 3) that – using the (hidden) auxiliary axiom "Frame Axiom 1: Persistence of intransitive verb." in addition to the (visible) axioms describing the concrete situation – RACE proves that the person sleeps at a later time.

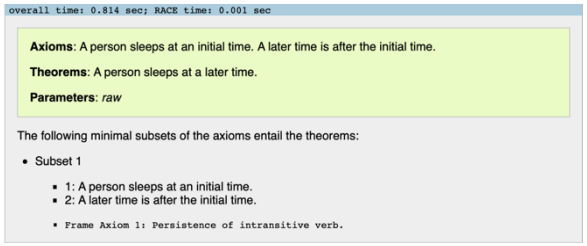

Figure 3: Continuous Sleep Revisited

Now the case of interrupted sleep. Note that also the axiom relating waking to not sleeping, that I used previously, is no longer needed.

~~*If a person sleeps at a time T1 and a time T2 is after T1 and it is not provable that the person does not sleep at the time T2 then the person sleeps at the time T2.*~~

A person sleeps at an initial time. A later time is after the initial time. The person does not sleep at the later time.

⊢

A person sleeps at a later time.

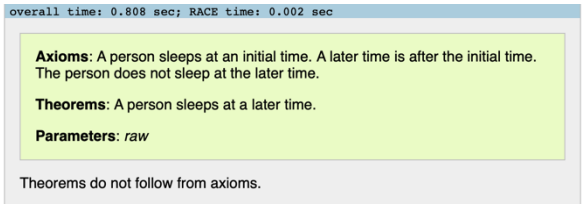

Figure 4: Interrupted Sleep Revisited

We get the not at all surprising result (Figure 4) that RACE cannot prove that the person sleeps at a later time.

7 The Yale Shooting Problem Revisited

RACE can now correctly and efficiently solve my version of the Yale Shooting Problem. Here again the version in English

A wine glass is initially empty and a wine bottle is initially not open. Opening the bottle, waiting a moment and then pouring the wine should fill the glass.

and here the ACE version

A glass is empty at a time T0 and a bottle is not open at T0. A time T1 is after T0 and the bottle is open at T1. An intermediate time T2 is after T1. A final time T3 is after T2. If a glass is empty at T2 and a bottle is open at T2 then the glass is not empty at the final time T3.

⊢

There is a final time. A glass is not empty at the final time.

In addition to the three frame axioms introduced before, a fourth frame axiom is needed to relate the state of some fluents to the precondition of an implicative axiom, concretely the fluents "*empty glass*" and "*open bottle*" at the time T2 to the precondition of the implicative axiom "*If a glass is empty …*".

Submitted to RACE we get:

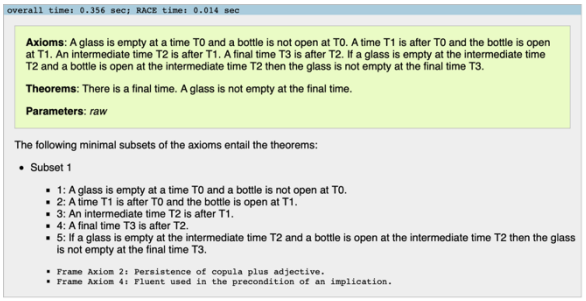

Figure 5: Yale Shooting Problem Revisited

The Yale Shooting Problem presented in section 2 had two solutions, one expected, the other one unexpected. RACE – using its built-in inertia axioms "Frame Axiom 2: Persistence of copula plus adjective." and "Frame Axiom 4: Fluent used in the precondition of an implication." – generates only the expected one.

8 Conclusions

I arrive at the following conclusions

- default logic effectively formalises the omnipresent law of inertia,
- default logic – originally expressed in first-order logic – can be naturally formulated in Attempto Controlled English (ACE),
- the Attempto reasoner RACE can reason with the law of inertia,
- reasoning with the law of inertia can be simplified and generalised by expressing the basic axioms not in ACE but as auxiliary Prolog axioms, quasi incorporating the law of inertia into RACE,
- RACE can correctly solve the Yale Shooting Problem.

Leibniz' law of inertia connects the present implicitly to the future, it does not make any assump-

tions about a possible past. This is explicitly reflected in the formalisations presented. To reason backwards in time, one would need other approaches, for instance abduction, a simple form of which is found in (Fuchs, 2016).

Acknowledgments

I enjoyed intensive discussions with Bob Kowalski and Rolf Schwitter which motivated me to investigate the formalisation of the law of inertia in Attempto Controlled English. Many thanks go to the first reviewer who not only carefully put this paper into the context of my previous publications, but also made valuable suggestions for clarifications, extensions and future research. I would also like to thank the Department of Computational Linguistics, University of Zurich, for its hospitality.

References

E. Erdem, M. Gelfond and N. Leone. 2016. *Applications of Answer Set Programming*. ACM. (www.depts.ttu.edu/cs/research/documents/5.pdf)

N. E. Fuchs. 2012. *First-Order Reasoning for Attempto Controlled English.* In *Proc. of the Second International Workshop on Controlled Natural Language (CNL 2010).* Marettimo, Italy.

N. E. Fuchs. 2016. *Reasoning in Attempto Controlled English: Non-monotonicity.* In *Proc. of the Fifth International Workshop on Controlled Natural Language (CNL 2016).* Aberdeen, UK.

S. Hanks and D. McDermott. 1987. *Nonmonotonic logic and temporal projection*. Artificial Intelligence, 33(3):379–412.

R. Kowalski and M. Sergot. 1986. *A logic-based calculus of events*. New Generation Computing, 4 (1): 67–95 (1986).

G. Leibniz. 1679. *An Introduction to a Secret* Encyclopaedia.

I. Newton. 1687. *Philosophiæ Naturalis Principia Mathematica*.

R. Reiter. 1980. *A logic for default reasoning*. Artificial Intelligence, 13:81-132 (1980).