Simplicity in solving the Frame Problem

Victor Jauregui and **Maurice Pagnucco** and **Norman Foo**[1]

Abstract. This paper presents an approach for reasoning about action and change which appeals to the principle of Occam's razor—roughly stating that the simplest explanations are the best ones—as the underlying theme of commonsense reasoning. We conduct a preliminary investigation of how appealing to simplicity allows us to address some of the challenges faced when reasoning about the effects of actions. In particular, we apply Occam's razor to the transformations between worlds specified by an action. To formalise these notions we use Kolmogorov complexity to identify the intended (simplest) transformation given an action description. We give some preliminary support for our claims by showing that the formalism captures our intuitions on some simple examples.

1 Introduction

Imagine you are standing before a high-rise building with n ($n \gg 0$) windows, each equally accessible and individually numbered. Assume, further, that each window is presently shut, and we specify an action which opens the k-th window (for some $k \leq n$). Our intuition suggests that window k should, in the state ensuing (known as the *successor state*), open, with the other windows remaining shut. Such reasoning is *non-monotonic* (as opposed to the monotonic inferences obtained with first-order logic); our action specifies only the intended *direct effects* of the action (the opening of the k-th window) yet, even though we have not specified, for instance, that our action leaves the j-th ($j \neq k$) window shut, we infer it as a matter of common sense. Specifying only the direct effects is done by design. We do not wish to specify that the action *opens window k and leaves all the other windows shut*, unless there was a particular reason why the other windows should open as well. The "leave all the other windows shut" part is to be (non-monotonically) inferred.

In general, we work with incomplete information. Here, apart from the fact that we have only described the windows and have left the rest of the world unmodelled, our action specification did not provide complete information about the intended successor state, but required that we make certain reasonable (non-monotonic) assumptions. Of necessity we work only with incomplete action descriptions, otherwise the specification cost would be so overwhelming it would render the problem of specifying an action impossible.

Our assumptions—in this case, that none of the other windows are affected by the action which opens window k—are reasonable in the following sense: given an initial state of the world, and the intractability of having an action fully specify its successor state, our intention, when we provide an action description, is that certain underlying (physical) mechanisms which bring about the action's intended effects are entailed. Moreover, in the absence of further information, we should only conjecture those mechanisms essential to bring about the specified effects. The latter is articulated as follows by Li and Vitanyi [3]:

> We are to admit no more causes of natural things (as we are told by Newton) than such as are both true and sufficient to explain their appearances.

This, essentially, is an appeal to the principle of Occam's razor—we should accept the simplest hypotheses that explain the observed phenomena. In the previous example, the opening of window $j \neq k$ would imply the existence of a superfluous cause in satisfying the action of opening window k. As such, we reject this effect.

This example highlights what is known as the *frame problem*—the problem of inferring what remains unaffected when we give an incomplete description of an action.

A solution to the frame problem, it is claimed in the literature, is to provide a non-monotonic formalisation of what Shanahan [8] calls the *commonsense law of inertia*, which states, among other things, that:

> Normally, given any action (or event type) and any fluent, the action doesn't affect the fluent. [p. 18]

Shanahan claims:

> As a scientific claim, the commonsense law of inertia wouldn't stand up to much scrutiny. But it's much better thought of, not as a statement about the world, but either as a useful representational device or as a strategy for dealing with incomplete information. [p. 18]

However, if we specify an action which now specifies that *all windows but the k-th are opened,*[2] then the first statement above no longer holds. By contrast, now typically every fluent (i.e., every window) is affected by the action. It appears more rational to conclude that, for this action, everything changes uniformly. In other words, our default rule, or commonsense inference, should be that of regular change with, as the exceptions, those fluents that behave inertially.

This is the sense in which we associate *simplicity* and *commonsense*, and it is for this reason that we look to generalise the commonsense law of inertia. In this instance, the simplest interpretation for an action that opens all the windows except, say, the k-th, is one under which regular change is assumed. In this context, it is more plausible that the k-th window also opens, *unless we specify explicitly that it remains shut*.

From this perspective, Occam's razor espouses the rationalisation of simplicity in our interpretations of action specifications. It generalises the commonsense law of inertia to the case when our scenario descriptions involve regular change.

[1] School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia. email: {vicj,morri,norman}@cse.unsw.edu.au

[2] By which we mean, we require that all but the k-th window be open, though we have not specified the state of the k-th window.

The goal of this paper is to furnish a foundation from which to guide our investigations of commonsense reasoning about action and change. Essentially, we make a commitment to a principled motivation of the problem of commonsense reasoning about action—in particular, we appeal to Occam's razor when reasoning with incomplete information. We do this in an attempt to overcome the *ad hoc* nature with which techniques for such reasoning have been applied to solve the problem, and which, in many instances, has led to more questions being raised than answers to existing questions being provided.

In Section 2 we provide the necessary background we use to formalise the intuitions captured in the example above and introduce the measure of simplicity we use to capture Occam's razor. In Section 3 we develop our formalism to identify how we appeal to the simplicity of transformations to obtain the desired successors states following an action in a given world. Then, in Section 4, we show that we can formally capture the intuitions we have indicated and discuss some of the issues that surface with our proposal. The final section concludes with a brief outline of the main concerns of this paper.

2 Background

In order to develop some of the basic ideas we now introduce a simple, traditional formalism used in the reasoning about action literature known as the *situation calculus* (see McCarthy and Hayes [6]).

2.1 The Situation Calculus

Essentially, the situation calculus is a first-order language in which we can describe world scenarios, actions and their effects. We develop it with the windows example we used above (for a recent and more extensive introduction see Shanahan [8], or for an alternative treatment see Reiter [7]).

Suppose we have again $n \gg 0$ windows before us, each initially open. In the situation calculus this scenario might correspond to the situation term S_0 with each window described by a *fluent*—essentially a property of the system in a given situation. Here the k-th window is described by the fluent w_k. To describe the state of our system we would then have fluents:

$$w_1, w_2, \ldots, w_k, \ldots, w_n.$$

To denote that window k is open we have a predicate H which stands for 'Holds'. So if window k is open in the initial situation we write: $H(w_k, S_0)$.

Hence, the initial state/situation would be described by the formulae:

$$\neg H(w_1, S_0), \neg H(w_2, S_0), \ldots, \neg H(w_n, S_0).$$

That is, all the windows are initially shut (i.e., not open).

Now, suppose we are given an action specification a, which specifies that, after performing the action, window k is open. In the situation calculus this would correspond to an action with direct effects specified by the *effect axiom*:

$$H(w_k, Result(a, s))$$

where s is a situation variable—the universal closure of unbound variables is taken to hold.

The worlds we *a priori* consider as possible candidates to be successor states are those worlds in which the direct effects of the action hold. That is, those in which the fluent w_k holds.[3]

In this case, this would correspond to the set of worlds w which make $H(w_k, Result(a, S_0))$ true. The intended successor is the one which, in addition, makes $\neg H(w_j, Result(a, S_0))$ remain true for all $j \neq k$. In this case this corresponds to the most inertial transformation consistent with the specified change.

The commonsense law of inertia, or principle of minimal change, furnishes a justification for this argument. In order to capture this intuition formally, what is typically done is the addition of the *abnormality predicate*, $Ab(f, a, s)$, denoting that fluent f is abnormal with respect to action a in situation s. The intuition is that, as the commonsense law of inertia tells us that typically actions don't affect a fluent, a fluent is abnormal if its value changes when an action is performed. This is captured by the *frame axiom*:

$$\neg Ab(a, f, s) \rightarrow [H(f, Result(a, s)) \longleftrightarrow H(f, s)].$$

In order to obtain the desired successor(s), all that remains is to ensure there are no more changes than are necessary. In particular, this corresponds to *minimising the extension of the predicate Ab in a manner consistent with the action specification.*

This, essentially, is what the technique known as circumscription, developed originally by McCarthy [5], captures. Other formalisms, like Reiter's successor state axioms [7], and Clarke's predicate completion in logic-programming [1], allow us to capture similar intuitions.

The ideas expressed here resonate with the principle of Occam's razor. We can view the commonsense law of inertia (the principle of minimal change) as expressing a preference for simplicity in our interpretation of change. In this sense, the commonsense law of inertia is, fundamentally, an attempt to apply Occam's razor to commonsense reasoning about action by associating complexity with change. Appeals to non-monotonic formalisms, like circumscription, used to implement it can be viewed as a rational attempt to provide the simplest consistent explanation of an effect and its causes.

In particular, in the limiting case, strict inertia corresponds to the simplest possible transformation between states—the identity transformation. The *simplicity of transformations* will recur as major theme explored in the rest of this paper.

2.2 Measures of simplicity

The principle of minimal change can be viewed as one measure of simplicity that has been proposed to solve the frame problem. In it we prescribe the simplicity of the transformations to be the instances of changes of fluents, ordered by inclusion. A transformation between states is simpler than another if the second has the same (instances of) changes as the first in addition to some others.

A number of authors have suggested problems with the approach of minimising change: initially, Hanks and McDermott [2] and subsequently, among others, McCain and Turner [4] and Thielscher [9], highlighting the acausal nature of the policy of minimal change. It would seem that the principle of minimal change serves as a good heuristic for capturing this rational notion of simplicity which we seek, to model Occam's razor, but perhaps lacks the sophistication to achieve the desired effects. To improve on this we have to look to more elaborate measures of simplicity.

One well established measure of simplicity, or its dual—complexity—is that of *Kolmogorov complexity* (see, for example, Li and Vitanyi [3]). Kolmogorov complexity is a measure of the complexity of an object (by which we mean a string) determined by the shortest program which generates it. The intuition is that if we supply a computer (indeed, any universal computing machine) with

[3] That the intended successor be among these worlds we may wish to relax—consider a transient effect—but for the present paper we assume this condition to hold.

a program that generates a string, we have effectively, in whatever programming language we have used, provided a computational description of that object/string. The simplest programs are the shortest ones, and it is their length which is a measure of the complexity of the string. Occam's razor is borne out directly in such a framework and for these reasons we will pursue this technique in the analysis to follow. Formally, the Kolmogorov complexity of a string x is defined as follows (this is an adaptation taken from [3]):

Definition 1 *Let x and p be binary strings. The* ***Kolmogorov complexity*** *of x, denoted $K(x)$, is given by:*

$$K(x) = \min\{|p| \ : \ U(p) = x\}.$$

where U is a universal reference machine and $U(p) = x$ denotes that the result of running U on input (program) p is x.

For our purposes the following alternative, which provides the complexity relative to another object/string, will prove useful (see [3]):

Definition 2 *Let x, y and p be binary strings. The* ***conditional Kolmogorov complexity*** *of x given y, denoted $K(x|y)$, is given by:*

$$K(x|y) = \min\{|p| \ : \ U(\langle p, y\rangle) = x\}.$$

where $\langle p, y\rangle$ denotes that the pair of inputs p and y are supplied.[4]

(Conditional) Kolmogorov complexity will be used to identify the complexity/simplicity of the transformations we associate with an action; the intended transformations being the simplest.

2.3 On the computational paradigm

Traditionally, as we saw when we introduced the situation calculus, the field of reasoning about action has been formalised at the logical level; mostly in terms of propositional, first-order logic and, when we delve into non-monotonic formalisms, like circumscription, second-order logic. Our developments in the computational domain are not, however, unfounded. The initial investigation in reasoning about action and, in particular, the frame problem, were carried out by McCarthy and Hayes [6]. In this paper they use a metaphysical model of the world consisting of "a system of interacting discrete automata" [p. 469].

We continue in this vein, but generalise our automata to Turing machines (TM's). That Turing machines provide us with a natural formalism to capture the intuitions we wish to develop is unclear, however, given that the measure of simplicity we use is developed in this context, it provides us with a good starting point.

3 Formal theory of change

We will now endeavour to develop the formalism in parallel with the recurring window example we have developed so far in order to motivate the intuition behind our formal theory.

If we look at the windows example we developed earlier, and formalised in the situation calculus, we find we can naturally encode the initial situation S_0 by the string:

$$w = \underbrace{00\ldots0}_{n}.$$

Here, bit k in our string gives the truth value of the k-th fluent. As such, we can identify a world/situation with the input string on the tape of a TM.

Consider the situation calculus specification of the direct effects: $H(w_k, Result(a, s))$. In model-theoretic terms, this formula represents the set of all possible worlds/models which make it true. In terms of Turing machines and strings, the latter corresponds to the set R_a of all world strings with the k-th bit set to 1. That is:

$$R_a \ \equiv \ \{x1y \ : \ |x| = k-1, |x1y| = n\}.$$

In general, this set will be supplied based on a specification of the action, as we had when we wrote $H(w_k, Result(a, s))$.

The challenge now is identifying the set of worlds which we consider as the intuitive successors given an initial world and a specification, as we provided above, of the direct effects of an action.

To do this we consider the transformations that take our initial state w to some possible candidate successor $v \in R_a$. The claim is that the intended interpretation(s) for the action a are the simplest transformations from w to some $v \in R_a$. We depict such a transformation schematically as:

$$w \xrightarrow{a} v$$

The simplest transformations are taken to be the shortest programs. That is, the transformations with smallest Kolmogorov complexity. Formally:

Definition 3 *The successors of an action a, performed in a world w, denoted $S_{a,w}$, are those worlds v in R_a with minimal conditional Kolmogorov complexity given w. That is:*

$$S_{a,w} \ \equiv \ \min_{v}\{K(v|w) \ : \ v \in R_a\}.$$

The intuition above is that we consider the simplest transformations taking w into R_a as the intended interpretations of a and the worlds that these yield are the desired successor states.

In our example, intuitively, the simplest program that transforms the world:

$$w = \underbrace{00\ldots0}_{n}$$

so as to produce a 1 at the k-th bit is the program that simply moves the Turing machine head along to the k-th bit position and writes a 1. This yields the state:

$$v = \underbrace{\overbrace{0\ldots0}^{k-1}10\ldots0}_{n}.$$

In order to capture this intuition we have to develop our formalism a bit further. The universal reference machine U we use, for our definition of complexity (see Definition 2), is a 3-tape TM as follows. The first tape will be designated the *program* tape and will store the program that performs the transformation we associate with action a to produce state v.

The second tape will be referred to as the *world* tape and on it is supplied, as input, the description of the initial world/situation w. This tape, after it has undergone the transformation dictated by the program tape, will hold the output state v.

Finally, there is a third tape which is a *work*, or *data*, tape, on which is supplied additional data particular to the situation/action. The intention is that our program p, on the program tape, will transform the world tape using the data on the data tape, to produce the

[4] In practice the pair of inputs $\langle p, y\rangle$ will correspond to inputs on separate tapes of a multi-tape machine U.

successor v. To show the role of the data tape, let us return to our example.

Our action specification is *shut window* k. Now, the value k is particular to our action and is an artifact due to our numbering of the windows. Given that we intend no discrimination between window k and any other window, we wish to factor out the number k from the action specification. Otherwise, the numbering of the windows can potentially affect which program is shortest. Indeed, if k is encoded in our program, transformations of the lowest numbered windows will seem favourable, as we incur a greater overhead in specifying windows labelled with larger numbers.

The motivation for separate program and data tapes is to maintain the program/data distinction. In Li and Vitanyi [3] a similar distinction is made, where *two-part codes* are considered. The program encodes what is known as the *model*, which in our case corresponds to the nature of the transformation (in our case, the setting of a bit to 1), and the second constitutes the particular data (specifying that the bit to set is the k-th).

4 Results and Discussion

We now proceed to show that our formalism captures some of the intuitions outlined in the motivations provided earlier. We do not provide rigorous proofs. In many cases such proofs are not obtainable—the Kolmogorov complexity function is not computable (see Li and Vitanyi [3]). Moreover, the nature of commonsense reasoning means we can only appeal to 'correctness' with respect to intuition. As intuitions are largely subjective it is impossible to claim that what we do is correct in any absolute sense of the word. What we can show is that our formalism adheres to our own intuitions which, we hope, coincide with those that are widely accepted.

Let us begin with our windows example. We want to show that the intended transformation is:

$$\underbrace{0\ldots000\ldots0}_{n} \xrightarrow{a} \underbrace{\overbrace{0\ldots0}^{k-1}10\ldots0}_{n}$$

The simplest (shortest) program that yields this transformation is simply the one with $(k-1)$ 1's on the data tape (indicating the number of shifts to be performed to identify the k-th bit) and which moves both heads to the right while there is a 1 on the data tape; writing a 1 on the world tape when a blank is encountered on the data tape. We encode this as:

$$\begin{array}{l}(\mathtt{q}_0,\genfrac{}{}{0pt}{}{\times}{\mathtt{1}},\genfrac{}{}{0pt}{}{\mathtt{R}}{\mathtt{R}},\mathtt{q}_0)\\(\mathtt{q}_0,\genfrac{}{}{0pt}{}{\times}{\mathtt{\_}},\genfrac{}{}{0pt}{}{\mathtt{1}}{\times},\mathtt{q}_{\mathtt{H}}).\end{array}$$

These tuples are machine instructions for the 3-tape TM described earlier. In particular, a tuple: $(\mathtt{q},\genfrac{}{}{0pt}{}{\mathtt{s}}{\mathtt{t}},\genfrac{}{}{0pt}{}{\mathtt{a}}{\mathtt{b}},\mathtt{q}')$ is an instruction which says if our TM is in state $\mathtt{q}$ with the world-tape head reading symbol $\mathtt{s}$ and the data-tape head reading symbol $\mathtt{t}$, then perform action $\mathtt{a}$ on the world tape and action $\mathtt{b}$ on the data tape, and goto state $\mathtt{q}'$.

Moreover, reading the symbol × is shorthand indicating 'any' symbol and performing action × indicates that we do nothing (the null-op). Otherwise, the symbol _ indicates a blank and L and R indicate moving the head left and right, respectively. Finally, the state $\mathtt{q}_{\mathtt{H}}$ comprises a *halting* state.

The only conceivable alternative simplest program that writes a 1 on the k-th tape square is that one which simply writes k ones—corresponding to the transformation:

$$\underbrace{00\cdots00}_{k}0\ldots0 \xrightarrow{a} \underbrace{11\cdots11}_{k}0\ldots0$$

This, however, is slightly more complex (longer), comprising the program:

$$\begin{array}{l}(\mathtt{q}_0,\genfrac{}{}{0pt}{}{\times}{\mathtt{1}},\genfrac{}{}{0pt}{}{\mathtt{1}}{\mathtt{0}},\mathtt{q}_0)\\(\mathtt{q}_0,\genfrac{}{}{0pt}{}{\times}{\mathtt{0}},\genfrac{}{}{0pt}{}{\mathtt{R}}{\mathtt{R}},\mathtt{q}_0)\\(\mathtt{q}_0,\genfrac{}{}{0pt}{}{\times}{\mathtt{\_}},\genfrac{}{}{0pt}{}{\mathtt{1}}{\times},\mathtt{q}_{\mathtt{H}})\end{array}$$

Even though we have not supplied the proof, intuitively it seems clear that there can be no shorter program which when supplied with data k (actually $k-1$) can produce a transformation into $R_a = \{x1y \;:\; |x| = k-1, |x1y| = n\}$. So it bears out that the intended program, for this simple example, is indeed the simplest.

Clearly an appeal to minimal change suffices to produce the desired successor for this example, so we have shown that our framework can capture the desired results when the intuitions behind minimal change prevail. Our next example shows that our formalism is more general than minimal change by showing that in cases when appealing to minimal change fails our intuitions (in the context of temporal projection) our theory still captures them.

Consider the Hanks–McDermott problem, or the Yale Shooting Problem, (see [2]). The scenario consists of a turkey and a gun which is used to shoot the turkey. We identify two fluents: $Alive$ and $Loaded$ to indicate that the turkey is alive and the gun is loaded, respectively. There are also three actions $Load$, $Wait$ and $Shoot$, with the obvious meanings. Suppose our initial situation S_0 has the turkey alive and the gun unloaded. These actions are specified according to the following effect axioms:

$$\begin{array}{c}H(Loaded, Result(Load, s))\\H(Loaded, s) \rightarrow \neg H(Alive, Result(Shoot, s))\end{array}$$

Note that, as the wait action is intended to do nothing, its effect axiom is omitted.

Consider performing the sequence of actions, $Load$ then $Wait$ followed by $Shoot$. Intuitively we expect the following model, which we have depicted pictorially:

$$\overset{A,\overline{L}}{\bullet}\xrightarrow[\Delta L]{Lo}\overset{A,L}{\bullet}\xrightarrow{Wa}\overset{A,L}{\bullet}\xrightarrow[\Delta\overline{A}]{Sh}\overset{\overline{A},L}{\bullet}$$

where the Δ's below the arrows indicate the occurrence of an abnormality with the respective fluent. Unfortunately, the following anomalous model is also admitted when we minimise abnormalities/changes (to see this observe that there are as many Δ's in the anomalous model as in the intended one, however, they occur at different times with different fluents):

$$\overset{A,\overline{L}}{\bullet}\xrightarrow[\Delta L]{Lo}\overset{A,L}{\bullet}\xrightarrow[\Delta\overline{L}]{Wa}\overset{A,\overline{L}}{\bullet}\xrightarrow{Sh}\overset{A,\overline{L}}{\bullet}$$

This second, anomalous model is clearly counter-intuitive. There is no justification for the gun becoming unloaded during the wait action. In our framework we can show that the anomalous model is rejected. In particular, the $Wait$ action, having been specified as not doing anything, receives as its intended interpretation the simplest program which does nothing—the empty program. Carrying over the arguments from the windows example, the load action receives the program that sets the bit associated with the $Loaded$ fluent (and only this bit) and the $Shoot$ action gets the program which checks if the $Alive$ bit is set and resets it if it is.

The program that performs the composite sequence of actions consisting of $Load$, $Wait$ and $Shoot$ actions is then the composition of these programs. This composite program, associated with the composite action, clearly yields only the intended model above and not the anomalous model. In particular, we cannot trade a change during the $Wait$ action with a change during the $Shoot$ action, as takes place in the anomalous model under minimisation of change.

These two examples suggest that, at least to a non-trivial extent, appealing to the simplicity of a transformation, in the context of specifying an action, provides us with an underlying theme which we can apply to common-sense reasoning about action. However, there are a number of considerations which suggest the formalism, as it currently stands, is not quite satisfactory.

Firstly, our choice of universal 3-tape Turing machine, though not completely arbitrary, provides relatively weak justifications for the computational measure of simplicity provided coinciding with our commonsense notion. What is perhaps more desirable is a better motivated world-simulator machine which gives us a more realistic simulation of world transformations.

To further stress this point, some simple transformations, such as setting the first k bits to 1 comprise a highly inefficient process to set the k-th bit. We take efficiency here in the thermodynamic sense where a process is inefficient to the degree to which it is irreversible—in this sense there is cost associated with an increase in entropy. One feels that a physical notion of efficiency should be factored in to mirror the descriptional notion of efficiency supplied by Kolmogorov complexity. A good choice of a simulator machine would facilitate in identifying good measures of physical efficiency.

Moreover, just as the second law of thermodynamics supplies a direction of time and causation to macroscopic mechanics, so we imagine that information theoretic arguments could supply arguments for commonsense causal reasoning about action. Causal arguments (or the lack thereof) were one of the main concerns regarding the use of minimal change in commonsense reasoning. The hope is that, by addressing such arguments in the framework above, we can supply a formal characterisation of causal reasoning. This is one of the central long-term goals of this research.

There are a number of other criticisms we can identify with the present formalism. We will identify two of the more immediate. The first concerns the use of a data tape along with a program tape. The intention of using a data tape was to factor out particular information, like the fact that we had numbered our window k, and remove such particulars from consideration of the complexity of the transformation. The problem is distinguishing what is particular and what we can legitimately attribute to the 'nature' of the transformation. The pitfall here is that we can move more and more of the program describing the transformation into the data tape, turning the program tape into a trivial program with essentially all the information regarding the transformation encoded as data on the data tape. We need to supply criteria for formalising the program/data distinction so as to avoid this happening.

Another potential concern is that, as we have formalised the problem, we work with transformations on *literal* representations of the world; by which we mean strings encoding the values of fluents. It may be more natural to work with descriptions of these literal representations rather than the literal encodings themselves. For example, rather than specifying that the initial world is the string:

$$w = \underbrace{0 \ldots 0}_{n}$$

denoting that all the windows are open, we might work on a description of this string. This would correspond to the shortest program which generates a string of n 0's. Perhaps it could be argued that as humans we tend to work with the most natural representations rather than literal ones—where here we might identify the most natural representations with the shortest descriptions of the literal world. That the same arguments and results hold given this view is unclear.

5 Conclusion

The purpose of this paper was to motivate a unifying philosophy towards the problem of reasoning about action. In particular, we looked towards the principle of Occam's razor—roughly stating that the simplest explanations for a theory are the most rational—as the underlying foundation guiding our interpretation of commonsense when reasoning about action.

Our investigations are still preliminary at this stage but we feel that the philosophy and motivations are sound even if the formalism proves, subsequently, to require modification or elaboration. This is likely to be the case but we feel the work here provides a solid starting point for such investigations.

We supplied some evidence, through simple examples, that the formalism we developed captures our intuitions. While this is clearly not evidence that our approach here is correct in any strong sense of the word, it suggests that we are at least on the right track towards capturing something interesting.

As we outlined in the discussion following our introduction to the formalism, there are many issues which we have neglected to address. Indeed, the ultimate goal of the research is to provide a general framework under which commonsense notions, such as causality, can be analysed. The present work has not delved into these issues, though some attempt has been made to show that the relevant concepts feature adequately in our theory. Such work remains for future research.

REFERENCES

[1] K. Clark, 'Negation as failure', *Logic and Databases*, 293–322, (1978).

[2] S. Hanks and D. McDermott, 'Nonmonotonic logic and temporal projection', *Artificial Intelligence*, **33**, 379–412, (1987).

[3] M. Li and P. Vitnayi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, New York, 2nd edn., 1997.

[4] N. McCain and H. Turner, 'A causal theory of ramifications and qualifications', in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, ed., Chris Mellish, pp. 1978–1984, San Francisco, (1995). Morgan Kaufmann.

[5] J. McCarthy, 'Circumscription —a form of nonmonotonic reasoning', *Artificial Intelligence*, **13**, 27–39, (1980).

[6] J. McCarthy and P. Hayes, 'Some philosophical problems from the standpoint of artificial intelligence', in *Machine Intelligence 4*, eds., B. Meltzer and D. Michie, 463–502, Edinburgh University Press, (1969).

[7] R. Reiter, *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*, MIT Press, 2001.

[8] M. Shanahan, *Solving the frame problem*, MIT Press, Cambridge, Mass., 1997.

[9] M. Thielscher, 'Ramification and causality', *Artificial Intelligence*, **89**(1-2), 317–364, (1997).