Balls and String: Simulations and Theories*

Dr. G. White
Department of Computer Science
Queen Mary and Westfield College
University of London
London E2 9QD
email graham@dcs.qmw.ac.uk

December 10, 1997

Contents

1 Theories and Simulations

There is, in the philosophy of mind, a debate about the way in which people reason about other minds. The distinction usually contrasts the traditional view, according to which naive cognisers, when they reason about their and

*During the composition of this report, the author was paid by Project Dynamo, supported by the United Kingdom Engineering and Physical Sciences Research Council under grant number GR/K 19266. The views expressed in this paper are the author's own, and the principal investigators of the project – John Bell and Wilfrid Hodges – bear no responsibility for them.

others' minds, use "*nomological* ... generalisations in providing the links between each mental-state concept and suitably chosen inputs, outputs, and other mental states" [7, p. 369],[1] with a view, advocated by Goldman and others, according to which "one can predict another person's choices or mental states by first imagining himself in the other person's situation and then determining what he himself would do or how he would feel" [7, p. 371]. The first account of how people reason is generally know as the "theory theory", and the second is known as the "simulation theory".

1.1 Response to the Psychological Theory

Discussion of these issues has centred on experimental – and particularly developmental – evidence; [9, 5] give evidence in favour, [15] evidence against, and there are arguments about the interpretation of that evidence throughout the literature. on whether the simulation theory will give rise to predictions that are "cognitively penetrable", i.e. "sensitive to what we know and don't know about the laws that govern the system" [21, p. 67] – the argument here is that our behaviour is cognitively penetrable, that the theory theory can model this but the simulation theory cannot. Or whether simulation is itself dependent on theory in order to isolate, for example, the similarities and differences between simulator and simulated.

Now clearly this is not directly applicable to the frame problem – AI investigators do not imagine that they are blocks or bullets, and even if they did it would probably be hard to get computers to do so. However, Goldman elsewhere gives a more abstract account of the simulation theory, which *will* be useful to us.

> There are three fundamental elements in the simulation model. First, the simulator must construct a pretend or feigned mental state that mirrors the state of the simulated agent as closely as possible. This construction will be based on the simulator's prior information about the agent's condition (however good or bad that information is). Second, the simulator must feed these pretend states into a cognitive process or mechanism that operates on these inputs to generate further mental states. It is assumed that the process operates on the feigned states in the same manner that it operates on actual states of the same sort. A practical reasoning process, for example, generates decisions from surrogate belief and desire inputs in the same way that it generates decisions from genuine beliefs and desires. Third, instead of acting on the generated output (e.g., the decision), the simulator takes it "offline" and uses it as a basis for attributing the output state to the agent. [6, pp. 89f]

[1] See also [8, pp. 168f]: "It is the position of many philosophers that common sense terms such as 'believes' are *theoretical* terms, the meanings of which are fixed in the same way as theoretical terms in general: by the set of laws and generalisations in which they figure. This view is widely (but not universally) assumed in functionalist accounts of folk psychology. (It is the offspring of the dispositional theories that were popular in the days of philosophical behaviourism). Presumably, mastery of the concept of belief would then be a matter of internalising a sufficiently large number of laws or generalisations in which the term 'belief' (and related verb forms) occurs."

This account, then, *is* worth exploring in the context of the frame problem. A mental setup for reasoning about blocks on tables, bullets in guns, and the like, could similarly have three elements. Firstly, one would have to have a mental state that simulated the state of the blocks; secondly, one would have to have a mental process that operates on these inputs to generate other mental states, corresponding to the state of the blocks after the necessary changes had taken place; and, thirdly, one would have some sort of inspection mechanism that read off predictions from the ensuing mental states.

Now I do not think that any of this is hideously implausible – in fact, since I have quite a vivid visual imagination, I think that it is overwhelmingly likely that mental states of this sort are produced and manipulated in at least some cases of reasoning about change. And if (for whatever reason) one does not want to rely on the evidence of introspection, then it would seem that some sort of talk about spin glasses and the like could be produced to make the possibility of this more academically respectable.

1.2 The PDP Approach

In fact, the theorists of connectionism – [16] – have come up with such an approach. They start with what they call "schemata" as emergent features of PDP networks.

> Schemata are not "things". There is no representational object which is a schema. Rather, schemata emerge at the moment they are needed from the interaction of large numbers of much simpler elements all working in concert with one another. Schemata are not explicit entities, but rather are implicit in our knowledge and are created by the very environment that they are trying to interpret – as it is interpreting them. [16, p. 20]

In this context, neural nets can be thought of as constraint satisfaction networks:

> the maxima in the goodness-of-fit space ... are those states that maximize the particular set of constraints acting at the moment. [16, p 21]

And, in such a context, one can indeed talk about simulation: [16, p 40].

2 Simulation as a Basis for Reasoning

Suppose, then, that we have a simulation-based account of human reasoning. How shall we use it to solve instances of the frame problem? An obvious answer is this: that we find some mathematical object which will correspond to the situation in question, and we find some transformations acting on this mathematical object which will match the evolution of the situation. We can, then, repeat Goldman's procedure on p. 2 above, using our mathematical objects instead of the mental states: that is, we first construct our mathematical object in correspondence with the initial state, then we apply transformations to it, thus simulating its evolution, and finally we examine it to see what state it has got into. Assuming we can find such objects, this will be a plausible basis for reasoning about change.

What mathematical objects and transformations shall we use? Following the work of Thielscher [22] and others, We will use associative, commutative rewrite systems with units (ACI-rewrite systems, as they are usually known [12]). We can define them as follows:

Definition 1. An ACI-rewrite system on an alphabet $\mathcal{A}$ is defined as follows. The *words* on $\mathcal{A}$ are given by expressions of the form I, A for $A \in \mathcal{A}$, and $W \bullet W'$ for words W and W'. Words are considered the same under the following equivalence relation:

$$\begin{array}{rcl} W \bullet I & \simeq & W \\ W \bullet W' & \simeq & W' \bullet W \\ W \bullet (W' \bullet W'') & \simeq & (W \bullet W') \bullet W''. \end{array}$$

We are also given a set of primitive rewrites, of the form

$$W \quad \rightsquigarrow \quad W'$$

Rewrites are composites of primitive rewrites and ACI equivalences, of the form

$$\begin{array}{rcl} W_1 \simeq X_1 \bullet W_1' \bullet Y_1 & \rightsquigarrow & X_1 \bullet W_2 \bullet Y_1 \\ \simeq X_2 \bullet W_2' \bullet Y_2 & \rightsquigarrow & X_2 \bullet W_3 \bullet Y_2 \\ \simeq \cdots & & \\ & \vdots & \\ \cdots & \rightsquigarrow & X_{n-1} \bullet W_n \bullet Y_{n-1} \\ \simeq W_n' & & \end{array}$$

where $W_1' \rightsquigarrow W_2, \ldots W_{n-1}' \rightsquigarrow W_n$ are primitive rewrites.

Since we will not need to distinguish general rewrites from primitive rewrites, we use the same notation $(a \rightsquigarrow b)$ for both.

We will use ACI words (or, rather, a subset of them) to model states, and ACI rewrites to model state transformations.

2.1 Motivation

Why do we use these mathematical objects and not others? Basically for two reasons:

1. the representation works – that is, it is expressively adequate and pleasingly direct;
2. such rewrite systems are computationally extremely tractable; and
3. they are mathematically well-behaved.

Although these reasons would be quite sufficient in common mathematical practice, they seem, in an AI context, to be not quite the thing. Why? There are, I believe, two reasons.

2.1.1 *A Priori* Considerations

Now there is a tradition in AI of coming up at this point with justifications which are fancier, and certainly more *a priori*. Those who incline towards a theory-based approach – and who thus tend towards the logic-based account of AI – typically use justifications such as the following:

> The claim is simply that the underlying meaning of any representational scheme is given by logical model theory. The closer the actual representational formalism is to the language of logic, of course, the easier it is to uncover this meaning. In practice though, for many reasons, the actual formalism used in a program may not be close to logic at all. [18, p. 12]

This sort of claim is frequently made, but is a little tendentious. Philosophically, one would like to know what this "underlying meaning" is, and especially what "underlying" means here: what sort of priority is involved? Why is "uncovering" this meaning a desirable task? It is also not clear what mathematical claims are being made here: whether one is saying that logic, and first-order model theory, constitute a sort of "universal mathematics" which will take precedence here over any other sort of mathematics. But this sort of claim seems to be trivially false: however one might be able in theory to reduce mathematics to set theory and logic, mathematics thus represented would be absolutely impractical. Mathematical thought surely depends on some sort of reference to mathematical entities. Even if it could be done, what would be gained by translating, say, the Atiyah-Singer index theorem into the language of sets and quantifiers? In what way would it be a closer approach to the "underlying meaning" of the representations involved?

2.1.2 Dichotomies

There is another opportunity for misunderstanding, which is the supposed dichotomy between connectionist and logical methods. There is an assumption that

> ... the traditional approach to AI [is] the *symbolic paradigm*, intending to emphasise that in this approach, cognitive descriptions are built of entities that are *symbols* both in the semantic sense of referring to external objects and in the syntactic sense of being acted on by 'symbol manipulation'. [20, p. 306]

Conversely, the connectionist approach denies this: connectionist networks manipulate elements which are not symbols. But in the symbolic approach,

> [f]ormal cognitive models have taken their structure from the syntax of formal languages, and their content from the semantics of natural language. [20, p. 307]

Thus, it has been assumed that, in a symbolic approach, the symbols are necessarily structured like a formal language. But simulation approaches – such as our rewrite systems – are symbols, they correspond to elements of reality, but they are not structured like formal languages. Similarly, if one models, say, the flow of a liquid by means of a vector field, one has set up a correspondence between vectors of the field and elements of reality: but the vectors are not structured like elements of a formal language, and they cannot be combined using logical connectives.

Thus, although some aspects of the PDP advocacy of simulation, quoted in Section 1.2 above, are appealing, it must be emphasised that one does not need any commitment to a subsymbolic approach to AI in order to use simulation; one can remain a symbolist and use it. Indeed, one could wonder whether much of the appeal of connectionism – despite the stated ideology – lies in the fact that many of their systems are simulations rather than theories, and that it is really the simulation/theory dichotomy that is important here.

3 Balls and String

Consider the following examples.

Example 1. There are two balls (b_1 and b_2) linked together with a string s of length 1. There are three locations: l_0 is on the surface of a table, l_1 is one unit above it, and l_2 is two units above l_0. At time 0 the two balls are at l_0 and the string is slack; b_2 is then raised so that, at time1, it is at l_2 and, in consequence, b_1 is hanging below it at l_1.

The second example is much the same, but more complex:

Example 2. There are n balls, $b_1 \ldots b_n$, and there are m strings $s_1 \ldots s_m$ each linking two of the balls. String i has length d_i. Initially the balls are all together on a table, with the strings slack. At time 0 a lift action occurs, which raises one of the balls so that at time 1, it is d units above the table. The problem is to predict the configuration of the balls and strings.

Note that the final configuration of the strings satisfies the following property:

Shortest Path Property If string s, of length l, links balls b and b', then string s is taut iff

- it lies on the shortest path between b and b' in the weighted graph defined by the strings, and
- one of the balls is at a height greater than l above the table.

We now have to consider the formalisation of our example, in our two styles: simulation and theory.

3.1 Theory

We will consider a typical approach to representing this problem logically, namely McCarthy's circumscription [13]. This is only an example, however; exactly similar problems arise with other approaches, such as, for example, Shoham's "causal theories" [19].

We should first ward off a possible misunderstanding about what a solution would be. In formulating such a problem, we can generally axiomatise the direct effects of actions, and of the direct effects of changes in causing other changes: these are what Shanahan calls *effect axioms* [18, Chapter 1], and they individually give sufficient conditions for fluents to be changed. But we cannot assume immediately that we have any axioms which tell us which fluents are *not* affected by a given axiom: these are called *frame axioms*, and having a procedure for providing them is tantamount to solving the frame problem. Neither, of course, are we allowed to assume axioms which are trivially equivalent to frame axioms: for example, we are not allowed to give, for a particular fluent, an axiom which gives *necessary* and sufficient conditions for that fluent to be changed, since negation of those conditions would immediately give the frame axiom. Thus, we are not allowed to assume axioms of the form

> A ball B at height H is either on the table ($H = 0$), or held at height H, or hanging on a taut string S from a ball $B2$ at height $H2 > H$.

because this amounts to giving necessary and sufficient conditions for the position of the ball to be affected by something.

Rather, we start with the features of the situation which are obviously captured in effect axioms – features such as would be produced by a low-level analysis such as that described in [4] – and we apply the normal procedure for deriving frame axioms from effect axioms, that is, circumscription. Lifschitz [11, pp. 328–334] summarizes the procedure.

- First one selects appropriate fluents for the situation. In our case, they will be those of the form $\mathsf{at}(b, l)$, where b is a block and l is a location, and two fluents describing the condition of the string: $\mathsf{taut}(s)$ and $\mathsf{slack}(s)$.

- Secondly, "we need to describe first when a combination of values of the frame fluents is 'consistent', that is, attained by some situation" [11, p. 329]. In our case, the constraints are given by the length of the string: to avoid the verbosity of explicit computations with lengths and distances, we simply give the following constraints:

$$\begin{array}{rcl}
\mathsf{at}(b_2, l_0) & \leftrightarrow & (\mathsf{slack}(s) \wedge \mathsf{at}(b_1, l_0)) \\
 & & \vee(\mathsf{taut}(s) \wedge \mathsf{at}(b_1, l_1)) \\
\mathsf{at}(b_2, l_1) & \leftrightarrow & (\mathsf{taut}(s) \wedge \mathsf{at}(b_1, l_0)) \\
 & & \vee(\mathsf{slack}(s) \wedge \mathsf{at}(b_1, l_1)) \\
 & & \vee(\mathsf{taut}(s) \wedge \mathsf{at}(b_1, l_2)) \\
\mathsf{at}(b_2, l_2) & \leftrightarrow & (\mathsf{taut}(s) \wedge \mathsf{at}(b_1, l_1)) \\
 & & \vee(\mathsf{slack}(s) \wedge \mathsf{at}(b_1, l_2)) \\
 & & \vee(\mathsf{taut}(s) \wedge \mathsf{at}(b_1, l_3)) \\
\mathsf{at}(b, l) \wedge \mathsf{at}(b, l') & \rightarrow & l = l' \\
\mathsf{slack}(s) & \leftrightarrow & \neg\mathsf{taut}(s)
\end{array}$$

  (Note that we have added another location, l_3, for the sake of symmetry.)

- Next we introduce actions: in our case there is only one, namely the lift action, whose postconditions are $\mathsf{at}(b_2, l_2)$.

- There is also the general machinery of the situation calculus: there is a predicate $\mathsf{holds}(f, s)$, where f is a fluent and s is a situation, and the abnormality predicate $\mathsf{noninertial}(f, a, s)$, where f is a fluent, a is an action, and s is a situation. Relating all of this, there is the generic law of inertia

$$\neg\mathsf{noninertial}(f, a, s) \quad \to \quad [\mathsf{holds}(f, \mathsf{result}(a, s)) \leftrightarrow \mathsf{holds}(f, s)]$$

- Finally we compute the resulting state by circumscription: we circumscribe the extent of the $\mathsf{noninertial}$ predicate while holding fixed the post-conditions of the actions and obeying the constraints.

The result of this procedure is the following: we minimise the extent of our predicate $\mathsf{noninertial}$, which means that we minimise the number of fluents which change. Now if the constraints are satisfied, the two fluents corresponding to the position of the balls must change anyway: the other fluent is the condition of the string. So circumscribing change means we minimise changes in the condition of the string, i.e. we keep it slack; given the constraints, this means that both balls end up at l_2, which is clearly incorrect.

One might think that a more complex circumscription policy might aid matters: one might think, for example, that, since the position of the balls determines the condition of the string, one should circumscribe abnormalities in the position of the balls before one circumscribes abnormalities in the condition of the string. However, this is not so: come what may, the position fluents are abnormal, and the only fluents about which there is any choice of normality or abnormality are those giving the condition of the string, so that whenever we circumscribe that, we end up with the same wrong answer.

3.1.1 The More Complex Example

Of course, one may be able to come up with a theory that correctly formalises these examples, but the shortest path property shows that there is a certain irreducible complexity to such treatments. Suppose that one had a theory, and that, in this theory, one could derive the positions of the balls – subject, of course, to certain conditions. This theory would, one think, contain entailments such as

$$\Gamma, \mathsf{at}(b, p_i) \wedge \mathsf{connects}(s, b, b') \wedge \mathsf{length}(s, d) \Supset \mathsf{at}(b', p_{i-d})$$

for some nonmonotonic entailment relation $\Supset$ and some theory Γ. Then this machinery will implicitly have to conclude that s *is* the shortest path from b to b'; that is, it will have to be able to show that there are no theorems of the form $\mathsf{path}(d', b, b')$ in some appropriate theory. This gives an irreducibly global character to such reasoning.

3.2 Rewrites

We will now give a rewrite system which simulates the behaviour of balls and string. There are three sorts of rules, firstly the rules which say how the string can change the positions of balls:

$$\mathsf{at}(b, l_i) \bullet \mathsf{at}(b', l_j) \quad \rightsquigarrow \quad \mathsf{at}(b, l_i) \bullet \mathsf{at}(b', l_{i-1}) \quad j < i$$

and secondly the rules which say how the position of balls affects the condition of the string:

$$\begin{array}{rcll} \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_j) \bullet \mathsf{slack}(s) & \rightsquigarrow & \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_j) \bullet \mathsf{taut}(s) & i \neq j \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_i) \bullet \mathsf{taut}(s) & \rightsquigarrow & \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_i) \bullet \mathsf{slack}(s) & \end{array}$$

and, finally, the rewrite which says what the action does:

$$\mathsf{time}(T) \bullet \mathsf{occurs}(\mathsf{lift})(T) \bullet \mathsf{at}(b_2,l_0) \quad \rightsquigarrow \quad \mathsf{time}(T+1) \bullet \mathsf{at}(b_2,l_2)$$

Now start with a situation in which the action occurs, and apply the rewrites:

$$\begin{array}{rl} \mathsf{time}(0) \bullet \mathsf{occurs}(\mathsf{lift})(0) \bullet \mathsf{at}(b_1,l_0) \bullet \mathsf{at}(b_2,l_0) \bullet \mathsf{slack}(s) & \rightsquigarrow \\ \mathsf{time}(1) \bullet \mathsf{at}(b_1,l_0) \bullet \mathsf{at}(b_2,l_2) \bullet \mathsf{slack}(s) & \rightsquigarrow \\ \mathsf{time}(1) \bullet \mathsf{at}(b_1,l_1) \bullet \mathsf{at}(b_2,l_2) \bullet \mathsf{slack}(s) & \rightsquigarrow \\ \mathsf{time}(1) \bullet \mathsf{at}(b_1,l_1) \bullet \mathsf{at}(b_2,l_2) \bullet \mathsf{taut}(s) & \end{array}$$

Note that at the final line, which corresponds to the final state of the system, we can apply no more rewrites.

3.2.1 The More Complex Example

This is, likewise, easy to handle. We simply generalise the rewrite rules a little, in order to accommodate strings of varying lengths; we also give the lift action a pair of arguments, so that we can talk about raising a specific ball to a specific location:

$$\begin{array}{rlr} \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_j) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) & \rightsquigarrow & (1) \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_{i-d}) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) & & j < i-d \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_j) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) \bullet \mathsf{slack}(s) & \rightsquigarrow & (2) \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_j) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) \bullet \mathsf{taut}(s) & & |i-j| = d \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_i) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) \bullet \mathsf{taut}(s) & \rightsquigarrow & (3) \\ \mathsf{at}(b,l_i) \bullet \mathsf{at}(b',l_i) \bullet \mathsf{connects}(s,b,b') \bullet \mathsf{length}(s,d) \bullet \mathsf{slack}(s) & & |i-j| < d \\ \mathsf{time}(T) \bullet \mathsf{occurs}(\mathsf{lift}(b,l))(T) \bullet \mathsf{at}(b,l_0) & \rightsquigarrow & (4) \\ \mathsf{time}(T+1) \bullet \mathsf{at}(b,l) & & \end{array}$$

For a correctness result, we need a couple of standard properties of rewrite systems (see [10, pp. 4f]):

Definition 2. Two terms are *convertible* if they are equivalent under the equivalence relation generated by $\rightsquigarrow$.

Definition 3. A term a of a reduction system is in *normal form* if no rewrite $a \rightsquigarrow b$ applies to it.

Definition 4. A rewrite system is *strongly normalising* if every sequence of rewrites eventually terminates.

Definition 5. A rewrite system has the *normal form property* if, whenever a and b are convertible and a is in normal form, then $b \rightsquigarrow a$.

Now we have the following.

Definition 6. A *single-valued* term of the rewrite system is one in which, for each ball b, there is exactly one factor $\mathsf{at}(b, l)$ in the term, and for each string s there is either a factor $\mathsf{slack}(s)$ or a factor $\mathsf{taut}(s)$.

Note that the rewrites we have given preserve single-valuedness.

Proposition 1. *The rewrite systems defined above are strongly normalising on single-valued terms.*

We will prove this by a long sequence of tiny lemmas. Note that the rewrite system corresponding to Example 1 is a special case of that corresponding to Example 2, so we only need to prove the proposition for the latter.

First we have

Definition 7. For balls b_i and b_j, let $\sigma_{i,j}$ be the length of the shortest path from b_i to b_j.

Lemma 1. *If ball b_i is raised to height d, then the final position of ball b_j is $l_{\phi(j)}$, where $\phi(j) = l_{d-\sigma_{i,j}}$ if $d > \sigma_{i,j}$, and $\phi(j) = l_0$ otherwise.*

Proof. This is an obvious consequence of the Shortest Path Property. □

Lemma 2. *If the final positions of balls b and b' are l_j and l_k, and if the two balls are connected by a string s of length e, then string s is taut if $|j - k| = e$, and slack otherwise.*

Proof. Even more obvious. □

Definition 8. If we have a single-valued term of the rewrite system, and if it assigns to ball b_j the position $l_{\psi(j)}$, then its *badness*, $\beta = \Sigma_j(\phi(j) - \psi(j))$.

Lemma 3. *The badness of a single-valued term strictly decreases at each application of rule* (1)*, and is left unchanged by the rules* (2) *and* (3).

Proof. The rules (2) and (3) do not change the positions of the balls: the rule (1) does, and always moves balls upwards. □

Lemma 4. *For each single-valued term, the badness of terms accessible by rewrites from it is bounded below.*

Proof. Such a term assigns initial positions to the balls: let l be the highest of these positions. It is easy to check inductively that the rewrites cannot move a ball above l. The badness of a term gets smaller the higher the balls get, and so the badness of the terms in question is bounded below by the badness of the term in which all balls are at l. □

Lemma 5. *For single-valued terms, there can be no infinite sequence of applications of the rules* (2) *and* (3).

Proof. If we only apply the rules (2) and (3), the positions of the balls do not change; for fixed positions of the balls, we can, for each string, only apply one at most of (2) and (3), and there is a finite number of strings. □

Now we can prove Proposition 1.

Proof. Consider a sequence of rewrites. Note first that we can only apply the rule (4) once, because after that the occurs factor disappears. Now consider the badness of terms: we get a sequence $\beta_1, \beta_2, \ldots$ which may have finite plateaux (where we apply rules (2) and (3)) but otherwise strictly decreases and is bounded below. The sequence of rewrites must thus terminate. $\square$

Now consider the normal forms that we can reach from our initial state.

Lemma 6. *In any sequence of rewrites of the initial term, the balls are never above their final positions.*

Proof. An easy induction. $\square$

Lemma 7. *If a term t can be reached from the initial term in a finite sequence of rewrites, and if the badness of t is non-zero, then we can always reduce the badness by some rewrite.*

Proof. If the badness of t is non-zero, then some ball must not be in its final position. Choose such a ball which is highest; apply all of the rewrites (1) corresponding to strings leading from the ball. At least one of them must pull the ball upward, thus reducing the badness. $\square$

The above gives us

Proposition 2. *Up to ACI-equivalence, there is only one term in normal form attainable from the initial term; it corresponds to the final configuration of the system.*

Proof. If a term is accessible from the initial term, no ball is above its final position; if it is in normal form, its badness is zero. This implies that all of the balls are in their correct positions. If the conditions of the strings are wrong, we can always use appropriate rewrites to correct them. $\square$

The propositions, then, give us a correctness result. Whatever sequence of rewrites we apply, we will get to the same term in the end, and that term corresponds to the final configuration of the system.

4 Further Developments

One might say that these results are fine as far as they go, but that the traditional AI approach, by virtue of being conducted in a logical calculus, is inherently more expressive and flexible. That is, there are problems such as abduction, and, more generally, reasoning with incomplete information, which can be handled by a logical calculus but cannot be done with rewrite systems.

However, this argument begs the question. It assumes that, because rewrite systems are not *overtly* embedded in a logical calculus, that they *cannot* be embedded in such a calculus. And, indeed, these rewrite systems can be very directly embedded in a logical calculus, namely linear logic. I have shown myself that linear logic can successfuly deal with the unadorned frame problem [24, 23], and that linear logic, augmented with Girard's fixpoint theorem, can directly encode GOLOG [25]; this itself would guarantee a treatment of the rewrite rules that we discuss here. Furthermore, McDowell *et al.* [14] have shown how linear logic, together with Girard's fixpoint theorem, can be used to encode transition systems in a very direct and appealing way. All of this work is done by the sort of proof-theoretic methods which will justify logic programming with the logic in question: that is, one will have a very good treatment of abduction and imperfect information.

5 Assessment

We have, then, presented two approaches to reasoning about problems with balls and string; these correspond to what philosophers call the theory theory and the simulation theory. What our examination has shown us is that the theory theory seems to have appreciable problems in dealing with quite simple physical problems, and that simulation can deal with those problems much better.

One has to be a little circumspect about both elements of this. One probably *could* find some deductive formulation of the problem that would be successful: however, it would most likely involve a radical change in the way that the problem was originally stated. For example, John Bell[2] has remarked that the simple example can be solved, using model preference theories, if one divides the action into two temporal segments, in the first of which the string becomes taut while the second ball remains on the table, and in the second of which the second ball is raised from the table. This may well be true, although, in practice, one still has unwanted models to deal with. However, it seems to sacrifice the naturalness of the original example; common sense reasoning seems to be perfectly capable of working out the final configuration of the balls without worrying tremendously much about the precise intermediate configurations (and certainly without doing the sort of intricate segmentation of the temporal evolution which would be necessary for more complex examples). Similarly, this approach would contradict a common intuition about ramification – discussed in [17] – namely that ramification ought to deal with a *series* of effects of an action, rather than looking at them one at a time.

Similarly, one could deal with the examples by changing the stock of fluents, from those of the form $\mathsf{at}(b, l)$ to something like $\mathsf{above}(b, l)$, which would say that a ball is at or above a certain location. One would have to circumscribe the $\mathsf{above}(\cdot, \cdot)$ predicates before the inertial circumscription. Although this might work, it would greatly increase the number of fluents and decrease their naturality: it would be very difficult to argue that this is how people reasoned about these problems.

This seems, then, to be a worrying example for what has been described as the "common sense law of inertia" [18, Chapter 1], namely the idea that one can do a good deal of reasoning about change simply by regarding change as exceptional and persistence as normal. One might be able to do that, but at a cost: the cost is the loss of naturalness and the feeling that one is dealing with some sort of elaborate coding rather than a genuine representation.

The simulation-based theory, by contrast, has two great advantages, advantages which seem also to have been noticed in earlier work on the subject [2, 3]. The first is its naturalness: it deals with the problem as stated, using a very simple set of fluents. The rules, too, have a very simple interpretation: rule (1) says that strings pull balls upwards, which is, after all, what strings do, and

[2] Personal communication.

rules (2) and (3) say how strings become taut or slack when the balls at their ends move around.

The second advantage is this: the rules are entirely local. One can apply a rewrite – say using an instance of (1) – and one only need worry about the two balls and the single string involved. Of course, the new position of the lower ball might be wrong; there may be a shorter path between those balls than the string that we have just pulled along. However, if we have put in a false position, we can correct it by applying further rewrites, and strong normalisation tells us that everything will work out right in the end. By contrast, what made the classical logical approach less tractable and less robust is that it is not local in this sense; before one infers the position of a ball from the length of a string, one has to know that there is no shorter path. Because classical reasoning of this sort cannot erase, deductions have to get it right the first time.

On the other hand, it would be a mistake to be too naive about the relation between the rewrite system and reality. The evolution of the rewrite system may not correspond, in any simple-minded way, to the evolution of the physical system; the rewrite system may have intermediate states which are not physically realisable (because, for example, they put balls further apart than the strings would allow). However, in many ways this is true to the way in which people reason: Nancy Cartwright writes of scientific reasoning (although I believe this account holds of much human reasoning in general):

> It is important that the models we construct allow us to draw the right conclusions about the behaviour of phenomena and their causes. But it is not essential that the models accurately describe everything that actually happens; and in general it will not be possible for them to do so ... This does not mean that the right lessons cannot be drawn. Adjustments are made where literal correctness does not matter very much in order to get the correct effects where we want them; and very often ... one distortion is put right by another. [1, p. 140]

References

[1] Nancy Cartwright. Fitting facts to equations. In *How the Laws of Physics Li*, pages 128–142. Clarendon, Oxford, 1983. Originally published in Richard Grandy and Richard Warner (eds), *Philosophical Grounds of Rationality: Intentions, Categories and Ends*, Oxford University Press.

[2] Brian V. Funt. Problem-solving with diagrammatic representations. *Artificial Intelligence*, 3:201–230, 1980.

[3] Francesco Gardin and Bernard Melzer. Analogical representations of naive physics. *Artificial Intelligence*, 38:139–159, 1980.

[4] Andrew Gelsey. Automated reasoning about machine geometry and kinematics. In Daniel S. Weld and Johan de Kleer, editors, *Readings in Qualitative Reasoning about Physical Systems*, The Morgan Kaufmann Series in Representation and Reasoning, pages 580–591. Morgan Kaufmann, San Mateo, CA, 1990. Originally Appeared in 3rd IEEE Conference of AI Applications, Orlando, Florida, 1987, pp. 182–187.

[5] Alvin I. Goldman. In defence of the simulation theory. *Mind and Language*, 7:104–119, 1992.

[6] Alvin I. Goldman. *Philosophical Applications of Cognitive Science*. Westview, Boulder CO, 1993.

[7] Alvin I. Goldman. The psychology of folk psychology. In Alvin I. Goldman, editor, *Readings in Philosophy and Cognitive Science*, pages 347–380. MIT, Cambridge MA, 1993. Originally published in *Behavioural and Brain Sciences* 16 (1993), 15–28.

[8] Robert M. Gordon. Folk psychology as simulation. *Mind and Language*, 1:58–171, 1986.

[9] Daniel Kahneman and Amos Tversky. The simulation heuristic. In Daniel Kahneman, Paul Slovic, and Amos Tversky, editors, *Judgement Under Uncertainty: Heuristics and Biases*. Cambridge University Press, 1982.

[10] J. W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume II, pages 1–116. Oxford University Press, 1992.

[11] Vladimir Lifschitz. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 297–352. Clarendon, 1994. Volume Coordinator D. Nute.

[12] Ernst W. Mayr and Albert R. Meyer. The complexity of the word problem for commutative semigroups and polynomial ideals. *Advances in Mathematics*, 46:305–329, 1982.

[13] John McCarthy. Epistemological problems in artificial intelligence. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 24–30. Morgan-Kaufmann, Los Altos, 1977. 1985.

[14] Raymond McDowell, Dale Miller, and Catuscia Palamidessi. Encoding transition systems in sequent calculus: Preliminary report. *Electronic Notes in Theoretical Computer Science*, 3, 1996. Available from `http://www.elsevier.nl/locate/entcs/volume3.html`.

[15] Josef Perner and Deborrah Howes. ‘He thinks he knows’: And more developmental evidence against the simulation (role taking) theory. *Mind and Language*, 7:72–86, 1992.

[16] D.E. Rumelhart, P. Smolensky, J.L. McClelland, and G.E. Hinton. Schemata and sequential thought processes in PDP models. In James L. McClelland, David E. Rumelhart, and the PDP Research Group, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2, pages 7–57. MIT Press, 1986.

[17] Erik Sandewall. Transition cascade semantics and first assessment results for ramification: Preliminary report. Invited paper at a workshop at FAPR ’96. Accepted to appear in *Spatial and Temporal Reasoning*, edited by Oliviero Stock, to be published in 1997 by Kluwer. Available at `ftp://ftp.ida.liu.se/pub/publications/techrep/1996/r-96-19.ps`.

[18] Murray Shanahan. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia.* MIT Press, 1997.

[19] Yoav Shoham. *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence.* MIT Press, Cambridge, 1988.

[20] Paul Smolensky. Connectionism and the foundations of AI. In Derek Partridge and Yorick Wilks, editors, *The Foundations of Artificial Intelligence: A Sourcebook*, pages 306–326. Cambridge University Press, Cambridge, 1990. A shortened version of an article in *Behavoural and Brain Sciences* 11 (1988).

[21] Stephen Stich and Shaun Nichols. Folk psychology: Simulation or tacit theory? *Mind and Language*, 7:35–71, 1992.

[22] Michael Thielscher. Computing ramifications by postprocessing. In *International Joint Conference on Artificial Intelligence*, pages 1994–2000, 1995.

[23] Graham White. A linear meta-interpreter for the situation calculus. Submitted to the *Journal of Logic and Computation.*

[24] Graham White. The design of a situation-based Lygon metainterpreter: I. simple changes and persistence. Technical Report 729, Department of Computer Science, Queen Mary and Westfield College, London, October 1996.

[25] Graham White. GOLOG and linear logic. Technical report, Department of Computer Science, Queen Mary and Westfield College, London, December 1997.