Chapter 11
An Approach to Human-level Commonsense Reasoning

Michael L. Anderson, Walid Gomaa, John Grant & Don Perlis

Abstract
Commonsense reasoning has proven exceedingly difficult both to model and to implement in artificial reasoning systems. This paper discusses some of the features of human reasoning that may account for this difficulty, surveys a number of reasoning systems and formalisms, and offers an outline of active logic, a non-classical paraconsistent logic that may be of some use in implementing commonsense reasoning.

Introduction

Humans reason – of that there is no doubt. But what sort(s) of reasoning do we do? Clearly there are some among us who do mathematical reasoning, and do it well. And, it has been argued that all reasoning is an attempt to reach the ideal model of mathematics, i.e., to arrive at true conclusions (from given assumptions).

Perhaps for this reason, efforts to examine human reasoning have tended to be in formal logical dress, mimicking the rigor of mathematics, e.g., Aristotle, Leibniz, Boole, and Frege.[1] But there is evidence – recounted below – that human reasoning is not always aimed specifically at true conclusions.

The field of artificial intelligence also followed in this math-based mode, at least initially, despite many doubters. One such doubter, Marvin Minsky, pointed out

Michael L. Anderson[1,4], Walid Gomaa[5,6], John Grant[3,2,1], Don Perlis[1,2]
(1) Institute for Advanced Computer Studies, University of Maryland.
(2) Department of Computer Science, University of Maryland.
(3) Department of Mathematics, Towson University.
(4) Department of Psychology, Franklin & Marshall College.
(5) Department of Computer and Systems Engineering, Alexandria University.
(6) Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology.

[1] Note that we do *not* make the converse claim – that formal logic tends to aim at modelling reasoning processes (i.e., psychologism).

an embarrassingly obvious difficulty: much of human reasoning tends to be nonmonotonic. What we conclude depends not only on what we believe but also on what we do not believe. That is, we draw conclusions on the basis of not having certain beliefs.

Minsky's famous example is essentially this: Told Tweety is a bird, we may reasonably come to believe that Tweety can fly. But if we had been told Tweety is a bird and furthermore is a penguin, we would not have drawn that conclusion. That is, the original conclusion (Tweety can fly) was performed by an inferential process that can be halted by the presence of additional information. This completely flies (pardon the pun) in the face of mathematical reasoning, where only ironclad guaranteed true conclusions are of interest. Minsky's example highlights the fact that in everyday life very often we are content (indeed may have no other choice except) to seek a very highly plausible conclusion. The real world has far too many parameters for us to be able to have strict data on them all, so we end up reasoning as if we had beliefs such as "most birds fly", and so on. And this relaxing of the truth-demand into a plausibility demand opens the door to retractions in the face of further evidence.

Well, this did not stop the logic-based AI-ers from using logic. It simply encouraged them to find better logics, so-called non-monotonic logics, where an enlarged set of assumptions might lead to a different set of conclusions that happens to be missing one of the original ones. A number of proposals for such logics soon surfaced, and had high degrees of success, most notably those of McCarthy and of Reiter.[2]

Nevertheless, smart artificial systems did not spring up. It seems that being nonmonotonic is not enough. In fact, it was pointed out a number of times that these new logics tended to be designed for the purpose of specifying the kinds of conclusions a smart reasoner ought to come to, but were not in general useful to system designers. The logics did not lend themselves to specifying ways for a system to actually arrive at these conclusions. The biggest roadblock was that the logics in question – like those before them – provided a characterisation of the set of all theorems that would follow from given axioms (or beliefs). This set typically is infinite, and the logics give little or no indication as to how and in what order these theorems are to be proven. Thus a system designer is stuck with the task of building an inference *engine* that produces, little by little, the theorems specified by one of the formal logics.

Yet even if one solves these problems, another surfaces. Very many of the theorems are of no use whatsoever to a given system's activity. Logics in general tend to have promiscuous rules of inference, concluding sentence after sentence without regard for their usefulness. This problem of *relevance* had long been recognised. But possibly an even worse aspect is that time (a *lot* of time) is being used up, both on these irrelevant results and on the enormous (even infinite) set of all theorems. Somehow a real-world reasoner must exercise some control over its reasoning so that time is not wasted without regard to real-world exigencies.

[2] While details are complicated, most such approaches aim at the inference of special additional "normal" or "typical" formulas – such as Flies(x) from Bird(x) – when *not* ruled out by axioms.

This is a major difficulty because time is highly significant in almost all endeavours, and because formal inference crunches on and on forever, oblivious to time. Clearly, on-board logic[3] must be able to take into account the fact that time passes as reasoning is going on, and what is important at one moment might not be so at another. In short, a reasoner ought to realise that "now" changes out from under it, it never stands still. So reasoning about time is slippery. As an example, the following makes perfect sense and is essential for effective on-board logic, yet is absurd from the point of view of spec logic:

$$\text{From } Now(t) \text{ infer } \neg Now(t).$$

That is, as soon as the time is known to be t, it no longer is t. Given a small unit or grain of time (e.g., a second, or a millisecond), the gist of the above can be approximated by this rule:

$$\frac{t : Now(t)}{t+1 : Now(t+1)}$$

The above "clock" rule is the essential feature of so-called active logics, a species of on-board logic. While it might not seem particularly revolutionary, it has major consequences. Three of the most important are as follows:

1. Reasoning can keep up with deadlines. Given a noon lunch appointment, one reasons at 11:30am that at 11:45 one should start walking to the restaurant. Then at 11:44 one reasons that it is time to stop reading the newspaper and put on one's coat. And by the time one's coat is on, one reasons that it is time to walk (because by the time that reasoning has been done it will be close enough to 11:45). Trivial enough, but impossible to do with spec logics. But the clock rule makes deadline sensitive reasoning possible.[4]

Here is a much-simplified example of the above form of reasoning in active logic, involving a deadline. We have annotated each time-step in the reasoning with the actual time on the left; via the clock rule, the logic has effective access to this information as well, assuming it is started off with the correct time. In each step below we have placed the agent's relevant beliefs at that time, with any new ones listed first; among these is always the current time, $Now(t)$. And the last step shown has the newly inferred belief "Walk" as well. Note that beliefs of the form Now(t) are not inherited to the next step (see above clock rule) but that in general other beliefs

[3] Let us call a logic that is used by a real-world reasoning agent (human or otherwise) as it goes about its business an "on-board" logic (as opposed to a specification – "spec" – logic that characterises limiting behaviours such as the set of all sentences that (eventually) can be proven). Thus we are using the term "logic" quite broadly, to include any systematic method for drawing conclusions from premises.

[4] So-called tense logics and temporal logics express propositions about past, present, and future, but the present is not represented as evolving: *Now* does not change as theorems are proved, in contrast with the above Clock Rule. In other words, tense logics are also spec logics, rather than on-board logics.

– such as that one should start walking at 11:45 – tend to be retained (precisely which beliefs are to be retained is a subtle issue; in particular cases there are useful heuristics but no single general principle). Note that, in general, a belief at one time is carried forward (remains a belief) at later times – for instance

$Now(11:45) \rightarrow Walk$

remains a belief indefinitely, whereas some special beliefs, such as knowledge of the current time, are dropped at the next step and replaced by a new belief (in this case due to the above clock rule, because time is always changing; but something similar can occur whenever there is reason to no longer hold a belief). Here is the example:

[11:30]: Now(11:30), Now(11:45) → Walk
[11:30:01]: Now(11:30:01), Now(11:45) → Walk
...
[11:44:59]: Now(11:44:59), Now(11:45) → Walk
[11:45]: Now(11:45), Now(11:45) → Walk
[11:45:01]: Now(11:45:01), Walk, Now(11:45) → Walk

At time 11:45 above, *modus ponens* goes to work on the then-current beliefs, and by 11:45:01 has inferred Walk. One simplifying assumption here is that it takes one "step" of time to apply an inference rule. Note that the belief that at 11:45 the Walk action should begin is still there among the beliefs, even though it is not likely to be useful; this can be "pruned" by a cleanup rule that drops conditionals of the form

$Now(t) \rightarrow X$

after time t has passed; after all, $Now(t)$ will never be true again after that time so the conditional will always remain true but never useful in concluding anything except at time t.

2. Inconsistency is a disaster for spec logics. They simply accept all sentences as theorems, making them useless. Paraconsistent logics adopt various means to avoid this "explosion" of consequences. But what is really needed is a paraconsistent logic with the ability not only to side-step a contradiction, but to notice it and consider what to do about it, possibly altering its status as a belief. After all, it might be an important clue to something amiss. Again, time comes to the rescue, providing a temporal "stratification" of theorems according to when they are proven, so that the time at which one sentence is proven (believed) allows inferences at the next time-step to comment on the previous result, such as that it is in contradiction with other beliefs and should be abandoned:

$$\frac{t : P, \neg P}{t+1 : Contra(P,t)}$$

However, active logic does not discover all inconsistencies; that is in general not computable in finite time. It simply scans the current knowledge base for an occurrence of a wff and its negation. If deeper inconsistencies remain, so be it: just as a human may unknowingly entertain contradictory beliefs, so with active logic. Only when a contradiction is noticed – such as in the form of a direct contradiction between a formula and its negation, is an agent (human or otherwise) in a position to do anything about it.

Also, once P and $\neg P$ are noticed and removed from the KB, there is no general method for adjudicating between them. In general, the reasoning agent may have to be content with uncertainty. In particular cases, there are heuristics that may be useful, such as deciding in favour of P if the evidence that produced it is more compelling than that for $\neg P$. That of course requires additional machinery.

3. Inconsistency is just one example of a situation needing some sort of change (e.g., distrust various sentences). But more generally, any manner of change might be called for in a given situation. Even a change in language may be needed, if for example that is a plausible way to resolve an inconsistency. For instance, given the beliefs “John is reading,” and “John is wagging his tail," one might consider that the word John is being used to name two different entities. This might then prompt the introduction of two new names, John^1 and John^2. But to do this, the reasoning must be able to have breathing room, time to make such changes before the inference engine rushes ahead to all the infinitely many theorems that would arise from the two earlier beliefs that together are implausible: that a dog is reading.

So where the does this leave us? Are we closer to commonsense reasoning? We think so. One feature that we have identified, as a key to commonsense reasoning, is the ability to notice – and respond usefully to – anomalies. And it turns out that anomalies can easily be cast in the form of mismatches between expectations and observations, i.e., a contradiction – or close enough so that the Contra and Distrust rules can go into action. An evolving-time logic such as active logic provides just this capability.

Human paraconsistency

Humans are very good at dealing with – reasoning and acting in the face of – uncertainty, change and even contradictions. In contrast, AI systems, especially those implemented with logic-based reasoning mechanisms, are notoriously bad at coping with these pervasive features of real environments. One widely-accepted conclusion from these observations has been that humans do not use logic-based mechanisms to implement their core reasoning abilities. And, indeed, there is a great deal of empirical evidence that seems to point in this direction. Humans often fail to achieve the ideal of valid logical deduction, and in many contexts we seem to utilise representational formats more suited to non- or extra-logical manipulations.

For instance, a large body of research has established that people are less likely to judge instances of *modus tollens* to be valid than instances of *modus ponens*.[5] Moreover, people are subject to some characteristic logical fallacies, such as the converse error (example 1) and the inverse error (example 2):

[5] For an overview of the various findings reported in this paragraph, see Evans (1982).

Example 11.1.
If the horses went to the watering hole, we would see their tracks.
We see their tracks.
∴ The horses went to the watering hole.

Example 11.2.
If the horses went to the watering hole, we would see their tracks.
The horses did not go to the watering hole.
∴ We will not see their tracks.

Interestingly, despite trouble with *modus tollens* in general, people have little trouble with that logical form in the following sort of case:

Example 11.3.
If the horses went to the watering hole, we would see their tracks.
We do not see their tracks.
∴ The horses did not go to the watering hole.

This pattern of results has suggested to many that what looks like logical reasoning is actually *causal* reasoning. Rather than building formal logical models from these sentences and judging the validity of the argument, we are in fact building causal models of the situations depicted, and judging the likelihood of the outcome. And, indeed, by those standards, arguments 11.1 and 11.2 represent fairly plausible inferences.

Similarly, results from the Wason card selection task in Johnson-Laird & Wason (1970) apparently point to the use of inference mechanisms that are not logic-based. In this task, participants are shown four cards, e.g. $[A, K, 2, 7]$, given a rule of the form "If a card has a vowel on one side, it has an even number on the other" $(p \rightarrow q)$, and asked to choose the cards they need to turn over to check the validity of the rule. The majority of participants choose A and 2 (i.e. p and q), even though the logically correct choice is A and 7 (p and $\neg q$). To cite just two examples of how this evidence has been interpreted, Oaksford & Chater (1994) take it to indicate that decision making is instead driven by considerations of information yield (according to their analysis, turning over A and 2 yields more information about the rule than does turning over any other two cards), while Cosmides (1989) – after noticing that participants make the logically correct choices when the abstract rule is replaced with one governing social conduct, e.g., "If you drink beer you must be over 21" – argues for the existence of mechanisms specialised for reasoning about social exchanges.

Such results – and there are many more like them – are of course deeply interesting, and assimilating them will be crucial to articulating a complete model of the mechanisms supporting human reasoning. And while we do not wish to question the existence and importance of the many different non-logical mechanisms that have been proposed to account for the vast amounts of available data on human reasoning and decision-making – including causal and other mental models (Gentner & Stevens (1983); Johnson-Laird (1983)), Bayesian inference (Oaksford & Chater

(2007)), social exchange modules (Cosmides (1989)), expected utility curves (Kahneman & Tversky (1979)), frequency sensitivity (Gigerenzer (1994)), and expected information gain (Oaksford & Chater (1994)) – we would like to suggest that there is nevertheless room for continued empirical attention to human *logical* reasoning, for at least the following reasons.

First, and most obvious, from the fact that humans possess some inferential mechanisms that are not logic-based, it does not follow that we do not have and use some native logic-based reasoning abilities. It might be noted in support of this thought that people's vulnerability to fallacies like those presented in arguments 11.1 and 11.2 largely disappears when the propositions involved are *not* causally related as they are in the examples. This suggests that causal-model-based mechanisms may be *interfering* with logic-based ones in circumstances in which both potentially apply.

Second, from the fact that logic-based AI is brittle while humans are not, it does not follow that human flexibility is necessarily or entirely the result of non-logical capacities. It may be that human logic takes a special form, or has certain features, or interacts with non-logical capacities in particular ways, and these attributes of human logic have simply not been captured in prevailing logic-based AI systems.

Third, even if it is proven that humans have no natural, native, logic-based inference mechanisms, the fact that humans can nevertheless reason logically would mean that our non- and extra- logical capacities can be harnessed to this end. Thus, investigating human logical reasoning, particularly in the face of contradiction and change, may help us understand what is special about our implementation of logic such that it supports the observed flexibility of human reasoning.

Fourth and finally, given the significant advantages of logic-based implementations in AI – including the fact that rule-based systems are relatively easy for humans to understand, and therefore to trust, and that changing their behaviour is as simple and quick as changing the rules that govern it (something that is not the case in systems that require extensive (re-)training) – it behooves us to consider how logic-based systems can be made more robust in the face of various perturbations. Human perturbation-tolerance can be a source of ideas and inspiration in this task.

Unfortunately, perhaps because human flexibility has been largely taken as an indication of non- or extra-logical mechanisms at work, there has been relatively little empirical work on human performance in the face of contradictory or changing information in specifically logical contexts. There has nevertheless been *some* work along these lines, enough to draw some preliminary conclusions that can be used to guide the development of more robust logic-based systems. We will first review the results, and then discuss what we take the implications to be.

In one interesting set of experiments Dean Sharpe & Guy Lacroix (1999) asked adults and children how they resolve assertions of the form $p\&\neg p$, such as the response "yes and no" to the question "Was the movie good?" In this work, 24 adults and 48 children (ranging in ages from 3 to 8) were told a story about two characters having dinner. At the end of the meal, one asks the other, "Did you like your supper?", to which the other character replies "Yes and no. I liked my supper and I didn't like it." Participants were asked to explain what the second character meant by the response.

The vast majority of participants (around 70%, including some children as young as 4), dealt with the contradiction by reinterpreting the statement *p* (I liked my supper) to take advantage of the internal structure of the object "supper". That is, they took the character to be asserting that he liked one part of the supper, but didn't like a different part. In addition, two other strategies were employed. Two adults and nine children reinterpreted the meaning of *p* by drawing attention to the applicability of the predicate "like". These participants said things like: the supper was average, so he neither liked it nor didn't like it. In addition, four of the adults, but only one of the children simply denied *p*, explaining that he didn't like the supper, but was trying to be polite. There were no other resolution strategies employed. The authors summarise their main findings by noting that "adults and even preschoolers posses interpretive structures – particularly object structure – that are non-classical in the sense that they can be used to resolve apparent contradictions." (Sharpe & Lacroix (1999) p.489)

A different set of experiments revealed some similar tendencies. Renee Elio (1997; 1998) asked what strategies people use to resolve logical contradictions of the form $\{p, p \rightarrow q, \neg q\}$. Participants were given premises like:

Example 11.4.

A If the ignition key is turned the car will start.
B The ignition key was turned.

They were then told:

C The car did not start

and asked: assuming that C is true, which statement A or B do you think it is more plausible to disbelieve? What revision would you make to that statement to make it consistent with the other premises?

Overall, participants were more inclined (around 60% of the time) to doubt $p \rightarrow q$ than they were to doubt *p*,[6] and when they did so they usually (around 63% of the time) made the statement consistent by re-interpreting the meaning of *p*, typically by adding conditions. Thus, participants might revise the statement to read "If the ignition is turned and the battery is not dead, then . . ." Most of the remaining revisions (around 30%) involved reinterpretations of *q*, with the effect of turning the rule into a default, e.g. "If the ignition key is turned the car will *usually* start."

This last finding is related to an interesting discovery by Ruth Byrne (1989), that reasoners seem to tacitly treat *many* rules as defaults, and thus can be made to suppress valid inferences under certain conditions. In her studies she found that while participants were happy to accept as valid inferences like:

[6] Although this preference was reversed when the initial statement was a definition such as: if a mineral is a diamond then it is made of compressed carbon.

Example 11.5.
If she has an essay to write then she will study late in the library.
She has an essay to write.
∴ She will study late in the library.

they will suppress the logically valid inference if certain additional premises are added, as in the following.

Example 11.6.
If she has an essay to write then she will study late in the library.
If the library stays open then she will study late in the library.
She has an essay to write.
∴ She will study late in the library.

In the case of argument 11.6, participants' chance of accepting the conclusion that she will study late in the library drops from 96% to 38%.

So, what do these interesting findings mean?

1. Humans maintain control over their inferences, and don't necessarily come to all logically valid conclusions.
2. This control is *content based*, in that they do not manage inference by ceasing to apply valid rules to all applicable forms, but instead selectively block application of valid rules to *certain* formulas. As Byrne concludes: "The moral of these experiments is that in order to explain how people reason, we need to explain how the premises of the same apparent logical form can be interpreted in quite different ways."
3. Reinterpretation of the meanings of premises is the most commonly used strategy for dealing with contradictory formulas. People maintain consistency of beliefs by changing their meanings in appropriate ways.
4. People use only a few strategies to address inconsistencies; these strategies nevertheless suffice for the purposes of everyday reasoning.

Can these features be captured in a formal system? We think so, and active logic is intended as one proposal for how that might be done. For instance, feature 1 is captured by active logic's stepwise character – an active logic reasons in time and, through the use of rules like $contra()$, permits "inspection" of its beliefs at each step. This allows an active logic to decide whether to continue to trust certain beliefs, or cease using them in further inference. In conjunction with this, active logic allows sentences to be "superscripted", as in the earlier example of the two Johns. This is a formal device implementing features 2 and 3, above. Its effect is to give an active logic the freedom to resolve contradictions by giving sentences different interpretations. Exactly how all of this is effected by active logic is described in detail in the section on active logic below, and in Anderson, Gomaa, Grant & Perlis (2008). Before getting to that, however, we turn to a brief survey of some of the many other approaches to implementing AI reasoning systems. This will allow us to better highlight the unique, and we think valuable, features of active logic.

Formal models of human reasoning

From the theoretical perspective any AI *reasoning system* typically consists of two main components: (1) a logical formalism for knowledge representation and (2) an inference engine to conclude new knowledge from existing knowledge. Based on the logical formalism and the theoretical and philosophical motivations behind the reasoning system, the inference mechanism can either be deductive, inductive, non-monotonic, default, defeasible, etc. An important issue in the implementation of the inference engine is the use of heuristics for typically the complexity of an algorithmic approach is prohibitively high. In the following subsections we survey some reasoning systems that take different approaches towards knowledge representation and inferencing.

General intelligence in human beings can be analyzed in terms of levels of description (see Newell (1990)). Each level corresponds to a particular degree of abstraction or, more concretely, to a particular timescale of intelligent tasks. Every increase in the order of magnitude on the timescale would instantiate a new higher level of abstraction. Levels can be grouped into three bands (see Rosenbloom, Laird, Newell & McCarl (1991)): (1) the neural band which corresponds to levels that do not exceed the order of few milliseconds; this band is the focus of the connectionist community, (2) the cognitive band which corresponds to levels starting with few milliseconds and up to levels with few seconds; this band is the focus of the cognitive science community, and (3) the rational band which corresponds to complex goal-oriented planning and actions which take at least the order of seconds; this band is the focus of the logicist and expert systems communities.

Soar

Soar (see Laird, Newell & Rosenbloom (1987); Rosenbloom, Laird, Newell & McCarl (1991)) is an implementation of a theoretical-based approach to general intelligence that focuses on the cognitive band. The relationship of Soar to other bands are investigated in Newell (1990); Rosenbloom (1989); Rosenbloom, Newell & Laird (1990). Soar assumes no distinction between human intelligence and machine intelligence, hence it has been extensively used both for developing artificial intelligence applications and cognitive models.

The architecture of Soar can be described by four levels of abstraction. First it uses an *associative parallel* memory to store long-term knowledge, and to identify and retrieve knowledge relevant to the current problem solving context. This knowledge is stored as a set of productions of the form $P\colon \mathit{condition} \rightarrow \mathit{action}$, where the correct action is performed when its preconditions hold. Memory access consists of the parallel execution of these productions. The result of this access is the retrieval of information into a short-term *working memory* that stores contextual information in the form of interrelated objects with attribute-value pairs. For example, an object representing a blue Ford car owned by Heather might look like

$$[Id = te12, type = car, model = Ford, color = blue, owner = Heather]$$

The second level of abstraction in Soar's architecture is the decision making mechanism which proceeds in two elaborate-decide cycles. During elaboration memory is accessed repeatedly and the corresponding relevant productions are executed in parallel. Then one or more of the retrieved actions is performed based upon preference knowledge about what actions are acceptable and/or desirable.

Above the decision making comes the determination of *goals*. Goals are set out whenever the decision procedure has reached a situation (called impasse) where alternatives do not exist any more or there are alternatives, but not enough discriminating information to choose among them (Rosenbloom, Laird, Newell & McCarl (1991)). Along with the determination of a new goal, a new problem context is generated which allows the continuation of decision making. If in the new context another impasse is encountered, then a new sub-goal and context are generated and the whole process recurs.

The final layer of abstraction is learning. When Soar resolves an impasse it summarises and generalises all the reasoning that led to its resolution. This adds new knowledge to its long-term memory that will prevent the occurrence of such an impasse in similar future situations. Soar's learning mechanism can be used to learn new conceptual knowledge, learn new procedures, and correct its knowledge from the feedback obtained from its interactions with the surrounding environment.

Cyc

Cyc is a reasoning system that focuses on the construction of a vast knowledge base (KB) of trivial and commonsense knowledge (see Lenat, Guha, Pittman, Pratt & Shepherd (1990); Lenat & Guha (1990)). The rationale behind Cyc is as follows. The research and design of AI reasoning systems have largely been concentrating on the development of a logical formalism for knowledge representation and an efficient inference engine based on that formalism. However, little attention has been given to the construction of a real, or at least an approximation to a real, KB that grounds the whole enterprise in reality (the raw material over which the reasoning engine operates). This KB would encode commonsense knowledge about the world that we take for granted concerning things such as time, space, agenthood, life, death, etc.

The early systems lacked the kind and amount of knowledge that would make them effective. With modest-sized KBs (10^2 to 10^3 domain-specific assertions or rules), such systems sometimes showed very impressive performance in *narrow* task domains but notable problems remained. For example, consider an expert system that contains the following rules from Lenat, Guha, Pittman, Pratt & Shepherd (1990):

if frog(x), then amphibian(x)
if amphibian(x), then lays_eggs_in_water(x)
if lays_eggs_in_water(x), then lives_near_lots_of(x,water)
if lives_near_lots_of(x,water), then ¬ lives_in_desert(x)

Given the assertion that Freda is a frog, the expert system can conclude various facts about Frida such as Frida is amphibian, lays eggs in water, lives near lots of water, etc. However, it can not answer simple commonsense questions, that would otherwise seem trivial to humans, such as: Does Freda lay eggs i.e., instead of asking about laying eggs in water? Is Freda sometimes in water? Is Freda a living being?, etc. Hence, such expert systems with complex detailed knowledge were very rigid, non-robust, and could easily fail when encountering a situation or question that is slightly different from the intended narrow domain.

Cyc is an attempt to overcome this brittleness. Its philosophy is to build a vast KB (size at least the order of millions of facts) containing general commonsense facts, domain-specific facts, general heuristics, specific heuristics, and heuristics for analogizing.

The construction of Cyc is, by its very nature, incremental. This includes the representation language, the inference engine, and of course the KB itself.

OSCAR

As opposed to Soar which is intended to simulate the cognitive band, OSCAR is constructed to simulate the *rational band* (Pollock (1992)). It is an architecture for rational agents based upon an evolving philosophical theory of rational cognition (Pollock (1999)). The general architecture is described in Pollock (1995). OSCAR's overall behaviour can be briefly described by the following cycle: (1) OSCAR has beliefs representing the surrounding environment, (2) it evaluates the current situation according to these beliefs, then (3) it engages in an activity to change the world to its liking and to update its belief system. The most distinguishing feature of OSCAR is that most of its rational cognition is performed by *epistemic cognition*, cognition about what to believe, as opposed to *practical cognition* which is cognition about what to do.

OSCAR is essentially a *defeasible* reasoner. Additionally, by providing it with the axiom schemas of first-order logic it becomes a *complete theorem prover* for that logic (that is OSCAR is able to deduce every valid first-order formula). Defeasible reasoning leads to conclusions that are not necessarily deductively valid. The truth of the premises along with a rationally compelling argument provide good support of the conclusion, even though it is still possible for the premises to be true and the conclusion false. Such premises are called *prima facie* reasons. Conclusions supported defeasibly might have to be withdrawn later in the face of new additional information (Pollock (1999)). For instance, if something looks red to me, that gives me a prima facie reason for thinking that it is red. But if someone I trust insists that it is not red then that gives a rebutting defeater. This kind of defeater attacks the conclusion. Another kind of defeater would attack the relationship between the premises and the conclusion. For example, learning that there was red light illumination should weaken my belief that the object is red. The interested reader may consult Pollock (1987, 1989, 1991a,b) for further details.

SNePS

SNePS, the Semantic Network Processing System (Shapiro (1979); Shapiro & Rapaport (1987, 1992); Shapiro (1993)), is a *logic-based* approach to natural language understanding and commonsense reasoning. Its ultimate goal is to acquire new knowledge through natural language interaction either with human agents or through media such as books, journals, radios, TVs, etc. SNePS should generally be able to represent everything expressible in natural language and should be able to reason in the presence of incomplete, circular, or inconsistent information.

Reasoning in SNePS is done through a formalism called SNePS logic SNePSLOG which is an enhanced version of first-order logic that is adapted to the natural language context (Shapiro (2000)). For example, one of the features of SNePSLOG is the implementation of a new logical connective $andor(i, j)$, which can be used to express the fact that an object satisfies some properties among several alternatives. This is not easily expressible in first-order logic because it is neither *inclusive or* nor *exclusive or*. The general formal syntax of $andor(i, j)$ is:

$$andor(i, j)\{P_1, \ldots, P_n\}$$

is true if and only if at least i and at most j of the first-order properties $P_1, \ldots, P_n$ are true. Another improvement to first-order logic is the addition of the connective *thresh* which has the following syntactical form:

$$thresh(i, j)\{P_1, \ldots, P_n\}$$

and is true if and only if fewer than i or more then j of $P_1, \ldots, P_n$ hold. This connective could be used to capture equivalences among first-order properties. More connectives, quantifiers and other logical features are included in SNePSLOG (see Shapiro (2000)).

SNePS memory is a *semantic network* modeled as a directed graph. Nodes in this graph represent concepts, individuals, general and specific rules, and propositions. The neighbours of any node in the semantic network can determine more complex structural properties of that node. For example, composite rules, propositions, and concepts can be formed by following a path of several nodes along the edges. Figure 11.1 shows an example of a SNePS semantic network. The nodes 'Max', 'David', and 'John' represent individuals, the node 'Male' represents a property, and the nodes ‘Has_gender’, ‘Is_parent_of’, ‘Is_child_of’, and ‘Equiv’ represent binary relations.

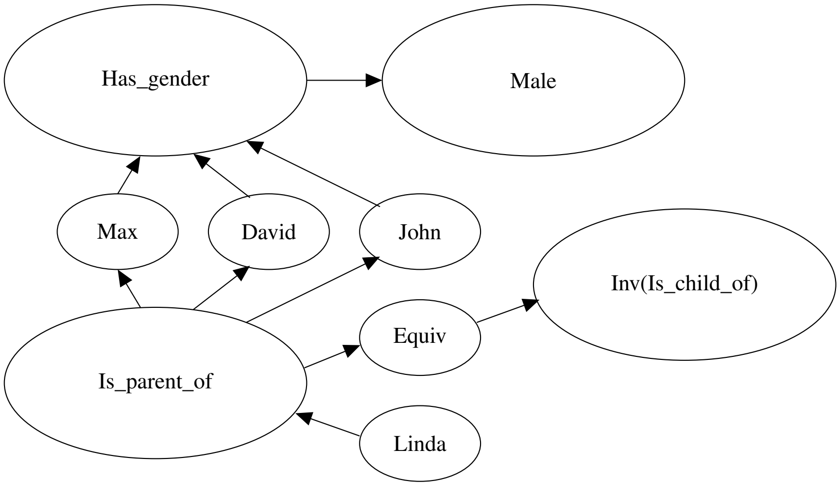

Fig. 11.1 An example of a SNePS network (adapted from Shapiro, Woodmansee & Kreuger (1968))

ACT-R

The ACT-R architecture is a simulation environment that supports the creation of cognitive models capable of predicting and explaining human behaviour (Anderson, Bothell, Byrne, Douglass, Lebiere & Qin (2004); Lebiere & Anderson (1993)). The architecture is constrained by the theory of *rational analysis* which is an empirical program that aims at explaining the functions and purposes of cognitive processes (Anderson (1990, 1991); Oaksford & Chater (1999)). According to rational analysis it is important to step back from the investigation of human methods and mechanisms to ask about the environment within which these mechanisms are applied (Gray, Sims, Fu & Schoelles (2006)). In the context of ACT-R, each component of the cognitive system is optimised with respect to environmental demands, given computational limitations (Taatgen, Lebiere & Anderson (2006)). According to this pragmatic approach *truth* is not a fundamental notion in ACT-R, though it is a derivative one: useful demand-based knowledge (either sensed directly from the surrounding environment or extracted from the current beliefs given the contextual environment) is usually true (weaker than defeasible reasoning described above in OSCAR); however, true knowledge is not necessarily useful (deducing Fermat’s Last Theorem or solving the Continuum Hypothesis are not useful in everyday activities). This is in contrast to purely logical-based systems built upon (presumed) true premises that are acted upon by sound reasoning rules irrespective of *usefulness*, which is not a logical notion. As will be seen below, this notion of usefulness/utility upon which ACT-R is based is manifested in the design of its memory.

ACT-R has two kinds of memory: declarative memory for facts and procedural memory for rules. Declarative memory is defined by items called *chunks*. Chunks have different levels of *activation* which reflect both their general access pattern and their relevance to the current context. Chunks that are frequently accessed receive a high activation. This activation decays stochastically over time if the chunk is not used. Procedural memory is defined by a set of *production rules*. Similar to the use of activation in declarative memory, each production rule has an associated *utility* value that determines its usefulness in reaching the desired goal. Selection of productions is based on the values of this attribute which are updated stochastically through the use of learning mechanisms.

Active Logic

In contrast with most of the systems outlined above, active logic was explicitly designed to capture some of the non-classical aspects of human commonsense reasoning, including time-awareness, control of inference, paraconsistency and non-monotonicity, including the ability to re-interpret the meanings of formulas. We have provided a detailed semantics (for a propositional version of active logic) in Anderson, Gomaa, Grant & Perlis (2008), but we offer some of the highlights here.

Formulas in active logic are expressed in a sorted first-order language $\mathscr{L}$ with two parts $\mathscr{L}_w$, a propositional language in which are expressed facts about the world, and $\mathscr{L}_a$, a first-order language used to express facts about the agent, including the agent's beliefs, for instance that the agent's time is now t, that the agent believes P, or that the agent discovered a contradiction in its beliefs at a given time.

$\mathscr{L}_w$ is a propositional language consisting of the following symbols:

- a set S of sentence symbols (propositional or sentential variables) $S = \{S_i^j : i, j \in N\}$ (N is the set of natural numbers).
- the propositional connectives $\neg$ and $\rightarrow$
- left and right parentheses (and)

$Sn_{\mathscr{L}_w}$ is the set of sentences of $\mathscr{L}_w$ formed in the usual way. These represent the propositional beliefs of the agent about the world. For instance S_1^0 might mean "John is happy". For later use we assume there is a fixed lexicographic ordering for the sentences in $Sn_{\mathscr{L}_w}$.

$\mathscr{L}_a$, contains the unary predicate symbol Now, used to express the agent's time, the binary predicate symbol $Contra$, used to indicate the existence of a direct contradiction in its beliefs at a given time, and the binary predicate symbol Bel, which expresses the fact that the agent had a particular belief at a given time. $\mathscr{L}_a$ contains only the connective $\neg$; hence statements such as $Bel(\theta,t) \to Bel(\theta,t+1)$ are not in the language.

All inferences in active logic depend on the knowledge base (KB) of the agent. The agent's knowledge base at time t, KB_t, is a finite set of sentences from $\mathscr{L}$, that is, $KB_t \subseteq Sn_{\mathscr{L}}$. In the case of KB_0 we allow only formulas of $Sn_{\mathscr{L}_w}$ whose superscripts are all 0.

For $\mathscr{L}_w$, we use a fairly standard notion of interpretation $h : Sn_{\mathscr{L}_w} \to \{T,F\}$ over the sentences in $\mathscr{L}_w$ that extends an $\mathscr{L}_w$-truth assignment h as follows:
$h(\neg\varphi) = T \iff h(\varphi) = F$
$h(\varphi \to \psi) = F \iff (h(\varphi) = T \text{ and } h(\psi) = F)$

We also stipulate a standard definition of consistency for $\mathscr{L}_w$: a set of $\mathscr{L}_w$ sentences is *consistent* iff there is some interpretation h in which all the sentences are true. Notationally we write the usual $h \models \Sigma$, to mean that all the sentences of Σ are assigned T by h.

The interpretation for $\mathscr{L}_a$ is somewhat more unusual. The symbol for the interpretation is H^{Σ}_{t+1}; it is an interpretation at time $t+1$ based on Σ, where Σ is to be understood formally as any set of sentences from $\mathscr{L}$. For current purposes, the most important aspects of the interpretation are as follows:

- The predicate symbol *Now* has the following semantics: $H^{\Sigma}_{t+1} \models Now(s) \iff s = t+1$ and $Now(t) \in \Sigma$; otherwise $H^{\Sigma}_{t+1} \models \neg Now(s)$.
- The predicate symbol *Contra* has the following semantics: $H^{\Sigma}_{t+1} \models Contra(\sigma,s) \iff$ either $s < t$ and $Contra(\sigma,s) \in \Sigma$ or $s = t$ and $\exists \sigma, \neg\sigma \in \Sigma$; otherwise $H^{\Sigma}_{t+1} \models \neg Contra(\sigma,s)$.
- The predicate symbol *Bel* has the following semantics: $H^{\Sigma}_{t+1} \models Bel(\theta,s) \iff$ either $s < t$ and $Bel(\theta,s) \in \Sigma$ or $s = t$ and $\theta \in \Sigma$; otherwise $H^{\Sigma}_{t+1} \models \neg Bel(\theta,s)$

For this version of active logic, we assume that the sentences in $\mathscr{L}_a$ are consistent, but allow for the possibility of inconsistency in the set of $\mathscr{L}_w$ sentences. We use the term Γ to refer to the potentially *inconsistent* set of $\mathscr{L}_w$ sentences in Σ: $\Gamma = \Sigma \cap Sn_{\mathscr{L}_w}$.

In order to model the sentences in Γ, active logic uses an "apperception function". The notion of an apperception function is intended to help capture, at least roughly, how the world might seem to an agent with a given inconsistent belief set Γ. For a real agent, only some logical consequences are believed at any given time, since it cannot manage to infer all the potentially infinitely many consequences in a finite time, let alone in the present moment. Moreover, even if the agent has contradictory beliefs, the agent still has a view of the world, and there will be limits on what the agent will and won't infer. This is in sharp distinction to the classical notion of a model, where (i) inconsistent beliefs are ruled out of bounds, since then there are no models, and (ii) all logical consequences of the KB are true in all models.

The idea is simple: suppose S^0_i, $S^0_i \to S^0_j$ and $\neg S^0_j$ are all in Γ, we imagine that the agent might not realise, at first, that the two instances of S_i are in fact instances of the same sentence symbol. That is, it might seem to the agent that the world is one in which, say, S^1_i is true, and so is $S^2_i \to S^0_j$.

The apperception functions we define can make changes only to Γ. An apperception function does not change $\Sigma - \Gamma$. We use the same notation ap when the apperception function is applied to an occurrence of a sentence symbol, a sentence, or a set of sentences. We start by defining a function that changes the superscripts of sentence symbols to 0. This is used to recover the original direct contradictions that were modified by the assignment of superscripts.

Definition 1 *For any sentence $\phi \in Sn_{\mathscr{L}_w}$, let $z(\phi)$ be the sentence ϕ with all superscripts reset to* 0. *If $\Sigma \subseteq Sn_{\mathscr{L}_w}$, then $z(\Sigma) = \{z(\phi) | \phi \in \Sigma\}$.*

Definition 2 *An apperception (awareness) ap is a function $ap\colon \Sigma \to \Sigma'$ where Σ and Σ' are sets of $\mathscr{L}$-sentences. An ap is represented as a finite sequence of non-negative integers: $\langle n_1, \ldots, n_p \rangle$. The effect of ap on Σ is as follows:*

1. *Let Σ be a set of $\mathscr{L}$-sentences and let $\Gamma = \Sigma \cap \mathscr{L}_w$. Using the lexicographic order given earlier, let the k^{th} sentence symbol in Γ be S_i^j. The effect of the $ap = \langle n_1, \ldots, n_p \rangle$ is to change S_i^j to $S_i^{n_k}$ if $1 \leq k \leq p$, otherwise S_i^j is unchanged.*
2. *$ap(\Sigma) = (\Sigma - \Gamma) \cup ap(\Gamma)$. (ap does not change $\Sigma - \Gamma$).*

Example 11.7. Let $\Sigma = \{Now(5), Bel(S_2^0, 4), \neg S_2^1, S_2^1, S_1^0 \to S_5^4\}$. In this case $\Gamma = \{\neg S_2^1, S_2^1, S_1^0 \to S_5^4\}$. Writing the elements lexicographically yields $ord(\Gamma) = \{S_2^1, \neg S_2^1, S_1^0 \to S_5^4\}$. Consider $ap = \langle 1, 3, 2, 16, 7 \rangle$. Then
$ap(\Sigma) = \{Now(5), Bel(S_2^0, 4), S_2^1, \neg S_2^3, S_1^2 \to S_5^{16}\}$.

The purpose of the apperception functions is to get rid of inconsistencies in Σ. Hence we are interested only in apperception functions that output consistent sets. The set of apperception functions that do this depends on Σ.

Definition 3 *Let AP denote the class of all apperception functions. $AP^{\Sigma} = \{ap \in AP | ap(\Sigma)$ is consistent $\}$.*

It turns out that AP^{Σ} is never empty (Anderson, Gomaa, Grant & Perlis (2008)).

At this point we are ready to define the notion of *active consequence* at time t – the active logic equivalent of logical consequence. Here again, the full technical details are given in Anderson, Gomaa, Grant & Perlis (2008), but we outline some of the more important elements here. We start by defining the concept of *1-step active consequence* as a relationship between sets of sentences Σ and Θ of $\mathscr{L}$, where $\Sigma \subseteq KB_t$ and Θ is a potential subset of KB_{t+1}. When we define this notion we want to make sure that Θ contains only sentences required by Σ and the definition of H_{t+1}^{Σ}. This is the reason for the next definition.

Definition 4 *Given Σ and $ap \in AP^{\Sigma}$, define*
$dcs(\Gamma) = \{\phi \in \Gamma | \exists \psi \in \Gamma$ such that $z(\phi) = \neg z(\psi)$ or $\neg z(\phi) = z(\psi)\}$.
$ap^z(\Gamma) = ap(\Gamma) - dcs(\Gamma)$.

The meaning of Definition 4 is that we are removing direct contradictions from $ap(\Gamma)$ while ignoring the superscripts.

Definition 5 *Let $\Sigma, \Theta \subseteq Sn_{\mathscr{L}}$. Then Θ is said to be a* 1-step active consequence *of Σ at time t, written $\Sigma \models_1 \Theta$ if and only if $\exists ap \in AP^{\Sigma}$ such that*

i. *if $\sigma \in \Theta \cap Sn_{\mathscr{L}_w}$ then $ap^z(\Gamma) \models \sigma$ (σ is a classical logical consequence of $ap^z(\Gamma)$), and*
ii. *if $\sigma \in \Theta \cap Sn_{\mathscr{L}_a}$ then $H_{t+1}^{(\Sigma-\Gamma)\cup z(\Gamma)} \models \sigma$.*

Definition 6

i. *Let $\Sigma, \Theta \subseteq Sn_{\mathscr{L}}$. Then Θ is said to be an n-*step active consequence *of Σ at time t, written $\Sigma \models_n \Theta$, if and only if*

$$\exists \Delta \subseteq Sn_{\mathscr{L}} \colon \Sigma \models_{n-1} \Delta \text{ and } \Delta \models_1 \Theta. \tag{11.1}$$

ii. *We say that Θ is an* active consequence *of Σ, written $\Sigma \models_a \Theta$, if and only if $\Sigma \models_n \Theta$ for some positive integer n.*

Next we give some examples to illustrate the concept of active consequence.

Example 11.8.

i. Let $\Sigma = \{Now(t), S_1^0, S_1^0 \to S_4^0, S_{12}^0\}$ and $\Theta = \{Now(t+1), S_4^0, S_{12}^0\}$. Let $ap \in AP^{\Sigma}$ be the identity function. It is easy to see that $\{S_4^0, S_{12}^0\}$ are logical consequences of $\{S_1^0, S_1^0 \to S_4^0, S_{12}^0\}$. Also by definition $H_{t+1}^{\Sigma} \models Now(t+1)$. Hence $\Sigma \models_1 \Theta$.
ii. Let $\Sigma = \{S_1^0, S_2^0, S_2^0 \to \neg S_1^0\}$ and $\Theta = \{Contra(S_1^0, t+1)\}$. We will see that $\Sigma \models_2 \Theta$. Let $\Delta = \{S_1^1, \neg S_1^2\}$. Then $\Sigma \models_1 \Delta$, through the apperception function $ap(\Sigma) = \{S_1^1, S_2^2, S_2^2 \to \neg S_1^2\}$. Then $\Delta \models_1 \Theta$ by the second part of the definition, regardless of the apperception function applied in this step.

Note that in Example 11.8.ii, it is not the case that $\Sigma \models_1 \{Contra(S_1^0, t)\}$ even though the conditions for the later appearance of the relevant direct contradiction were already in place at time t. This underlines the fact that in active logic it can take time for consequences to appear in the *KB*. Apperception functions give active logic agents control over which inferences to make, and which to suppress. They allow the agent to have inconsistent beliefs while still having a consistent world model. Moreover, this allows us to see how an agent with inconsistent beliefs could avoid vacuously concluding *any* proposition, and also reason in a directed way, by applying inference rules only to an appropriately apperceived subset of its beliefs.

For instance, consider the following active logic inference:

Definition 7 *If $\varphi, \neg\varphi \in KB_t$, where $\varphi \in Sn_{\mathscr{L}_w}$, then the* direct contradiction inference rule *is defined as follows:*

$$\frac{t : \varphi, \neg\varphi}{t+1 : Contra(\varphi, t)}$$

This inference is sound based on the definition and interpretation of $Contra$. And because of this, along with apperception functions, the following inference is *unsound*:

Definition 8 *Let* $\Sigma \subseteq Sn_{\mathscr{L}_w}$ *be inconsistent. Let* $\psi \in Sn_{\mathscr{L}_w}$*. We define the* explosive rule *with respect to the language* $\mathscr{L}_w$ *as follows.*

$$\frac{t : \Sigma; Inconsistent(\Sigma)}{t+1 : \psi}$$

The explosive inference rule is unsound. For consider the case where ψ is $\neg(S_1^0 \to S_1^0)$. No apperception function ap that turns Σ into a consistent set can logically derive ψ. Hence $ap(\Sigma) \not\models_1 \psi$.

This shows that active logic is paraconsistent. We hope that this approach to paraconsistency can shed some light on focused, step-wise, resource-bounded reasoning more generally. More details on the semantics for active logic, and many more examples of its use, can be found in Anderson, Gomaa, Grant & Perlis (2008).

Comparison with reasoning systems and formalisms

Active logic possesses several interesting properties. It has a temporal component so that inference occurs in time: for a set of formulas Γ at time t deduce formula ϕ at time $t+1$. Active logic is paraconsistent as both ϕ and $\neg\phi$ may hold at some time t. Active logic is also non-monotonic because a formula ϕ that holds at time t does not necessarily hold at time $t+1$; this happens in particular when ϕ and $\neg\phi$ are replaced by the $Contra$ formula.

We are not aware of any other logic system that possesses such a temporal component as well as paraconsistency and non-monotonicity. SOAR, Cyc and ACT-R do not appear to incorporate any of these features, and while OSCAR is non-monotonic, it is neither time-tracking nor paraconsistent. The closest of the above systems to having the distinctive features of active logic is SNePS, but there are some important differences between the two approaches. For instance although SNePS incorporates a time-tracking feature, in a SNePS-based agent *NOW* is a meta-logical variable, rather than a logical term fully integrated into the SNePS semantics. The variable *NOW* is implemented so that it does, indeed, change over time, but this change is the result of actions triggering an external time-variable update. In active logic, in contrast, reasoning itself *implies* the passage of time. Perhaps in part because of this difference, SNePS is a monotonic logic, whereas active logic is non-monotonic, leveraging the facts that beliefs are held at times, and beliefs can be held about beliefs, to easily represent such things as "I used to believe *P*, but now I believe $\neg$*P*" using the *Bel* operator. SNePS is also able to represent beliefs about beliefs, but there is no indication that this ability is leveraged by SNePS to guide belief updates. Rather, all beliefs are about states holding over time, so that belief change is effected by allowing beliefs to expire, rather than by formally retracting them.

This is a strategy similar to that employed by the situation calculus (which does not itself incorporate a changing *Now* term) (McCarthy & Hayes (1969)). Finally, although SNePS is a paraconsistent logic, in SNePS contradictions imply nothing at all, whereas in active logic contradictions imply *Contra*, a meta-level operator that can trigger further reasoning.

Nevertheless, although there are few examples of implemented systems with the features of active logic, we know that a substantial amount of work has been done on non-monotonic paraconsistent logics. While these logics are not really comparable to active logics, we provide here information on some such systems.

An early influential paraconsistent non-monotonic logical system was presented in Priest (1989). The logic **LP** has 3 truth values: True, False, and Both. The connectives and entailment in **LP** are defined as in classical logic, but on account of the third truth value, **LP** is paraconsistent. **LP** is then extended to $\mathbf{LP_m}$ with consistency as a default assumption and a notion of default consequence relation $\models_m$ is defined using minimal models. $\mathbf{LP_m}$ is a non-monotonic paraconsistent system.

Another such system is a combination of LEI (Logic of Epistemic Inconsistency) and IDL (Inconsistent Default Logic), called IDL&LEI. We refer to Martins, Pequeno & Pequeno (2002) for details about it including a multiple world semantics. Formulas in LEI are divided into 2 groups: the irrevocable formulas and the plausible formulas; the latter are distinguished by a question mark, as in α?. No contradictions are allowed involving any irrevocable formula; contradictions are allowed only for plausible formulas. LEI is paraconsistent. non-monotonicity is obtained by adding default rules using IDL. The IDL&LEI system has both an elegant syntax and a multiple world semantics.

Finally we mention the work in Arieli & Avron (1998) where a non-monotonic paraconsistent logic uses Belnap's 4-valued logic with a notion of logical consequence based on minimal preferential models. The approach here is primarily semantical. (Actually, it turns out that a 4-valued semantics is available also for IDL.) The recent paper by Arieli (2007) uses quantified Boolean formulas in the context of multiple-valued logics to represent several non-monotonic paraconsistent logics. This paper also contains many references to recent related work.

Conclusions

As shown by many psychological experiments, the logic used by human is substantially different from classical logic, and for just this reason may be more useful to commonsense reasoning. Hence logic-based AI systems should be attuned to, and where possible implement, these non-classical features. We have described several AI reasoning systems, as well as active logic, a logic designed to capture features such as time-awareness, control of inference, paraconsistency, and non-monotonicity, that we think are important to human commonsense reasoning.

Acknowledgements We would like to thank the referees for many helpful comments and suggestions.

References

Anderson, J. (1990). *The Adaptive Character of Thought*. Lawrence Erlbaum.

Anderson, J. (1991). Is Human Cognition Adaptive? *Behavioral and Brain Sciences* 14(3): 471–517.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of Mind. *Psychological Review* 111(4): 1036–1060.

Anderson, M.L., Gomaa, W., Grant, J., & Perlis, D. (2008). Active logic semantics for a single agent in a static world. *Artificial Intelligence* 172: 1045–1063.

Arieli, O. (2007). Paraconsistent Reasoning and Preferential Entailments by Signed Quantified Boolean Formulae. *ACM Transactions on Computational Logic* 8(3): Article 18.

Arieli, O. & Avron, A. (1998). The Value of Four Values. *Artificial Intelligence* 102(1): 97–141.

Byrne, R. (1989). Suppressing valid inferences with conditionals. *Cognition* 31: 61–83.

Cosmides, L. (1989). The logic of social exchange: has natural selection shaped how humans reason? Studies with the Wason selection task. *Cognition* 31: 187–276.

Elio, R. (1997). What to believe when inferences are contradicted: The impact of knowledge type and inference rule. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pp.211–216.

Elio, R. (1998). How to disbelieve p $\rightarrow$ q: Resolving contradictions. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society*, pp.315–320.

Evans, J.S.B. (1982). *The psychology of deductive reasoning*. London: Routledge Keegan Paul.

Gentner, D. & Stevens, A. (1983). *Mental models.* Lawrence Erlbaum.

Gigerenzer, G. (1994). Why the distinction between single-event probabilities and frequencies is important for psychology (and vice versa). In G. Write & P. Ayton (eds.) *Subjective Probability*, pp.129–161. New York: John Wiley and Sons.

Gray, W., Sims, C., Fu, W.T. & Schoelles, M. (2006). The Soft Constraints Hypothesis: A Rational Analysis Approach to Resource Allocation for Interactive Behavior. *Psychological Review* 113(3): 461–482.

Johnson-Laird, P. (1983). *Mental models.* Cambridge: Cambridge University Press.

Johnson-Laird, P., Wason, P. (1970). A theoretical insight into a reasoning task. *Cognitive Psychology* 1: 134–148.

Kahneman, D. & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica* 47(2): 263–291.

Laird, J., Newell, A. & Rosenbloom, P. (1987). Soar: An Architecture for General Intelligence. *Artificial Intelligence* 33(1): 1–64.

Lebiere, C. & Anderson, J.R. (1993). A Connectionist Implementation of the ACT-R Production System. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, pp.635–640.

Lenat, D. & Guha, R.V. (1990). Cyc: A Midterm Report. *AI Magazine* 11(3): 32–59.

Lenat, D., Guha, R.V., Pittman, K., Pratt, D. & Shepherd, M. (1990). Cyc: Toward Programs with Commonsense. *Communications of the ACM* 33(8): 30–49.

Martins, A.T., Pequeno, M. & Pequeno, T. (2002). A Multiple Worlds Semantics to a Paraconsistent Nonmonotonic Logic. In W.A. Carnielli, M.E. Coniglio & I.M.L. D'Ottavio (eds.) *Paraconsistency, the Logical Way to the Inconsistent*, pp.187–212. Marcel Dekker.

McCarthy, J. & Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer, D. Michie (eds.) *Machine Intelligence*, pp.463–502. Edinburgh: Edinburgh University Press.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge: Harvard University Press.

Oaksford, M. & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review* 101: 608–631.

Oaksford, M. & Chater, N. (eds.) (1999). *Rational Models of Cognition.* Oxford: Oxford University Press.

Oaksford, M. & Chater, N. (2007). *Bayesian rationality: The probabilistic approach to human reasoning.* Oxford: Oxford University Press.

Pollock, J. (1987). Defeasible Reasoning. *Cognitive Science* 11: 481–518.

Pollock, J. (1989). *How to Build a Person*. Cambridge: Bradford/MIT Press.

Pollock, J. (1991). A Theory of Defeasible Reasoning. *International Journal of Intelligent Systems* 6: 33–54.

Pollock, J. (1991). Self-Defeating Arguments. *Minds and Machines* 1: 367–392.

Pollock, J. (1995). *Cognitive Carpentry.* Cambridge: Bradford/MIT Press.

Pollock, J. (1999). Rational Cognition in OSCAR. In N. Jennings, Y. Lesperance (eds.) *Proceedings of ATAL-99*. Springer Verlag.

Pollock, J. (1992). How to Reason Defeasibly. *Artificial Intelligence* 57: 1–42.

Priest, G. (1989). Reasoning about Truth. *Artificial Intelligence* 39(2): 231–244.

Rosenbloom, P.S. (1989). A Symbolic Goal-Oriented Perspective on Connectionism and Soar. In R. Pfeifer, Z. Schreter, F. Fogelman-Soulie & L. Steels (eds.),

Rosenbloom, P., Laird, J., Newell, A. & McCarl, R. (1991). A Preliminary Analysis of the Soar Architecture as a Basis for General Intelligence. *Artificial Intelligence* 47: 289–325. *Connectionism in Perspective*. Amsterdam: Elsevier.

Rosenbloom, P.S., Newell, A. & Laird, J.E. (1990). Towards the Knowledge Level in Soar: The Role of the Architecture in the Use of Knowledge. In K. VanLehn (ed.), *Architectures for Intelligence*. Lawrence Erlbaum.

Shapiro, S.C. (1979). The SNePS Semantic Network Processing System. In N. Findler (ed.) *Associative Networks: The Representation and Use of Knowledge by Computers*, pp.179–203. New York: Academic Press.

Shapiro, S.C. (1993). Belief Spaces as Sets of Propositions. *Journal of Experimental and Theoretical Artificial Intelligence* 5: 225–235.

Shapiro, S.C. (2000). SNePS: A Logic for Natural Language Understanding and Commonsense Reasoning. In Ł. Iwańska & S.C. Shapiro (eds.) *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, pp.175–195. AAAI Press/MIT Press.

Shapiro, S.C. & Rapaport, W. (1992). The SNePS Family. *Computers and Mathematics with Applications* 23: 243–275.

Shapiro, S.C. & Rapaport, W. (1987). SNePS considered as a fully intensional propositional semantic network. In N. Cercone, G. McCalla (eds.) *The Knowledge Frontier*, pp.263–315. New York: Springer-Verlag.

Shapiro, S.C., Woodmansee, G.H. & Kreuger, M.W. (1968). A Semantic Associational Memory Net that Learns and Answers Questions (SAMENLAQ). Technical Report, Computer Science Department, University of Wisconsin, Madison, WI.

Sharpe, D. & Lacroix, G. (1999). Reasoning About Apparent Contradictions: Resolution Strategies and Positive-Negative Asymmetries. *Journal of Child Language* 26(2): 477–490.

Taatgen, N.A., Lebiere, C. & Anderson, J.R. (2006). Modeling Paradigms in ACT-R. In R. Sun (ed.) *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, pp.29–52. Cambridge: Cambridge University Press.