WHEN THINGS GO WRONG

A major theme in AI from the start has been so-called commonsense reasoning. Another (overlapping) theme is AI planning: automated production of plans (sequences of actions) that one might expect (based the system's KBN – both know-how and know-that) to lead to a given goal.

But plans are not infallible, as Robert Burns pointed out in 1786:

> But, Mousie, thou art no thy lane [you aren't alone]
> In proving foresight may be vain:
> The best laid schemes o' mice an' men
> Gang aft a-gley [often go awry]
> An' lea' us nought but grief an' pain,
> For promised joy.

Well, hopefully grief and pain are not universal consequences of plans (schemes) that go awry. Sometimes, yes, alas. But while it probably is the case that the vast majority of plans – when put into action – do not pan out just as expected (and that may sound even gloomier than what Burns said) still the vast majority of those are so easily fixed that we barely notice we are fixing them at all.

We are experts at dealing with things gone wrong, it seems.

This observation is one way to think about commonsense reasoning, at least as that phrase is used in artificial intelligence.

Commonsense Reasoning

As already noted earlier, in 1958-9, John McCarthy published a paper with the title Programs with common sense. In it he discusses – among other things – the idea of an advice-taking program, one that can be taught, especially when it is having trouble achieving a goal. It was an ambitious idea back then, and still is today. But some progress has been made.

Note that the advice-taker is a lesser goal than that of building a program so smart that it can figure everything out itself. The latter has also been discussed, and at times held up as a worthy goal of AI – with the caveat that it might mean we as a species make ourselves irrelevant, if we do manage to build machines smarter than we are (and there are differing views as to whether that is a good thing or bad; we may consider this issue toward the end of the semester). But no one has even come close to doing that, nor has there been a viable research program that shows any serious progress along those super-smart lines.

However, McCarthy's idea was that being an advice-taker, an agent with the ability to be taught – is a step along the way to human-level intelligence, and that this, together with other abilities, might lead to fully human-level mechanical intelligence. And that is a research program that is moving along noticeably (albeit slowly).

Why do things go wrong? That is, why do things not behave as we expect? Why cannot we have perfectly true understanding about things – at least about simple everyday things? Well, for some things we can – mostly for things that we define to be as they are: formal rules (e.g., graduation requirements), certain games (chess, checkers, etc), arithmetic (or math in general). But when the topic is the world out there, apart from out "toy" versions of it, our understanding of it tends to be only vague approximations based on limited experience. Thus birds fly. Usually. Unless they are penguins (or sick, or dead, or babies, or mutants, or…); in fact, flying is not part of the biological definition of bird (as given in Wikipedia):

Birds (class Aves) are feathered, winged, bipedal, endothermic (warm-blooded), egg-laying, vertebrate animals.

Things get worse: What is it to be feathered? Is having one feather enough? And again, what of baby birds, sick birds whose feathers have fallen off; and what about male birds don't lay eggs; etc? It seems that the natural world (i.e., apart from the human-defined world of formal concepts as in games and rules and math) is rather messy, and the "natural laws" we commonly refer to are indeed mere approximations. So we cannot fully "qualify" all the conditions under which creatures can fly, or have feathers, or even are birds at all; the counterexamples are endless; this is called the qualification problem.

Indeed, many AI problems are special cases of the general problem of the world not following precise formal rules. Yet we cannot give up on putting things into classes; we have no other way to handle data.

What about physics, then? Doesn't that subject purport to capture precise – exact – rules of how things really are in the world? Perhaps so; but those rules are far beyond that range of everyday things that an agent can use to get around in the world. Imagine trying to decide how to get to San Francisco, by using fundamental principles from particle physics! The equations themselves simply would be unstatable – let alone solvable – for large-scale objects such as cities, airports, and so on. And the way that physicists show that their equations are realistic in large-scale cases is by using statistics (i.e., approximations) based on those equations, such as the ideal gas law.

So we are back where we started: approximations that are usually pretty accurate but not always. Thus the real world is in general simply (far) too complicated for there to be concise general principles that are strictly true in all cases.

Ok, then, how about this: devise ways that an agent can use approximations, since they often work pretty well. And AIers have been hard at work doing just that in the commonsense reasoning area.

We will now take a brief look at some of this work.

The birth of nonmonotonic reasoning. In 1974, Marvin Minsky pointed out that when told of a bird, we reasonably imagine a creature that can fly. This initial knowledge be represented in a KB as follows:

Bird(x) -> Flies(x) [a general piece of background knowledge about birds]
Bird(tweety) [a fact about a particular bird]

Then based on these beliefs (or knowledge) we infer

Flies(tweety) [which then enters our KB as a new belief]

Clearly this is a very useful ability: to take a general principle (birds fly) and apply it to a specific case. In fact it is much like the generalized modus ponens of FOL.

But Minsky then pointed out that if we had instead started with an additional piece of information – if our KB is enlarged – then we might in fact block (or take back) that very conclusion. Suppose we had also been told that Tweety is a penguin; the initial KB then probably looks like this:

Bird(x) -> Flies(x) [a piece of general background knowledge about birds]
Bird(tweety) [a fact about a particular bird]
Penguin(tweety) [another particular fact]
Penguin(x) -> -Flies(x) [another piece of general background knowledge]
Penguin(x) -> Bird(x) [more general background]

A human would not infer Flies(tweety) from this (because we know penguins are exceptions to the general rule that birds fly). This means that the first principle (birds fly) is not to be interpreted as universally quantified – not true of all birds – but rather as a kind of rule of thumb. We tend to use a rule of thumb as if it were true, unless we know it does not apply to the item in question. Put in contrapositive form: if we do not know the rule does not apply to an item, we go ahead to use the rule on that item.

And in fact a great deal of human reasoning is like that; we use rules of thumb to guide very many of our inferences and decisions. Nor is this laziness; as noted above, for much of the real world this is the only possibility: there are very few general precise strictly true rules about the everyday world.

As a result, new discoveries were needed, in ways to capture some kind of formal reasoning that allowed for the expression and use of rules of thumb, or default rules as they are often called. And soon (by 1980) a number of methodologies had been found and were being intensively studied; among them were Circumscription (McCarthy), Default Logic (Reiter), and Nonmonotonic Logic (McDermott and Doyle).
While they differ in many respects, what they all have in common is some means for determining that an item is not known to be an exception to the rule in question. This is what allows them to conclude Tweety can fly – if we don't know Tweety is a penguin – and yet on the other hand blocks that same conclusion if we do know Tweety is a penguin.

Any type of reasoning for which the inclusion of additional axioms can prevent the inference of a conclusion that would have occurred without that new axiom, is said to be nonmonotonic. See www.cs.nott.ac.uk/~bsl/G52HPA/articles/Reiter:87a.pdf for a wonderful survey of the basics.

Numerous observations were soon made, concerning curious features of nonmonotonic reasoning (NMR). Below are some famous examples that had significant impact on the further development of NMR:

1. Flip-flopping nested defaults (Fahlman)
A mollusc typically is a shell-bearer.
A cephalopod is a mollusc except it typically is not a shell-bearer.
A nautilus is a cephalopod except it typically is a shell-bearer.
A naked nautilus is a nautilus except it typically is not a shell-bearer.

2. Nixon diamond (Reiter)
Nixon was a Quaker and a Republican. Quakers tend to be pacificists; Republicans tend not to be pacifists. What should a reasoning method conclude about Nixon?

3. Yale Shooting Problem (McDermott)
At time t=0 Fred is alive, and a gun is loaded and aimed at him.
At t=1 nothing is known to happen.
At t=2 the trigger is pulled.
The question then is the state of the world at t=3: Is Fred dead or alive at that time? The intuitive answer – and the answer we'd like a formal reasoning system to produce – is that he is dead. But what is involved in such reasoning? Let's consider the KB. Presumably it has in it an axiom such as If the gun is loaded and the trigger is pulled at a time t, then Fred will be dead at time t+1. So, is the gun loaded at time t=2 (when the trigger is pulled)? One presumes so; there is an implicit default rule: If a gun is loaded and the trigger is not pulled at a time t, then it is still

loaded at time t+1. However, there are also default rules such as: People who are alive at time t will still be alive at time t+1.

But now we have a problem: the alive rule suggests Fred will be alive at time t=1, t=2, t=3, and so on, whereas the loaded-gun rule suggests that gun will be loaded at t=1 and t=2 (when the trigger is pulled; hence not at t=3). Clearly some principle needs to pick out for us that loaded and alive are both true at t=1 and t=2 (and then by the death-by-shooting axiom alive becomes false at t=3), and not the alternative of alive staying true at t=1,2,3,4,… and loaded having become false at t=2 (due to some unknown activity at t=1). When researchers introduced a suitably broad notion of causality into the reasoning, this was solved; but it took a few years of intense study to reveal this.

4. Stuffy Room (Ginsberg)

A room has two ventilation ducts built into the floor. At time t=0 duct_1 is blocked by a flower-pot. Then at t=1 a TV set moved onto duct_2. What happens? Presumably both ducts are now blocked, so the room becomes stuffy (no fresh air can get in). But at t=0 various things were true (and noted in a presumed KB), including not-Stuffy, Blocked(duct_1), and not-Blocked(duct_2), as well as the general axiom Blocked(duct_1) & Blocked(duct_2) → Stuffy. At t=1, when not-Blocked(duct_2) becomes false (and removed from the KB, since we are told of this change), why should not-Stuffy also be taken out of the KB, rather than Blocked(duct_1)? One of the two must go, since if both ducts are blocked we know Stuffy will hold. But after all, something could have dislodged the flower-pot from duct_1. While this has some similarities with the Yale shooting problem, it also resembles the Nixon diamond: there are competing defaults at work in all three problems. (One proposed solution involves the notion of so-called nearest possible worlds.)

<u>Knowledge representation and reasoning (KRR).</u> We saw above that at times it is necessary to expand a formal language to include new notions, such as causality; and questions then arise as to the best way to do that – what representation to use for the new notion in order to facilitate reasoning with and about it: a constant, a predicate, maybe something altogether different? This happens so frequently that KRR itself is a recognized subarea of AI. However, it is closely allied with commonsense reasoning, and so I am addressing it here. Let us look at one example:

Suppose Mary wants John to telephone Sue. She says to him, "Call Sue." But why? This silly-sounding question becomes not silly at all when we consider the case when Mary is an artificial agent that we are designing. What does Mary's KB have to have, in order for her to decide that saying "Call Sue" to John is a sensible action to take? The following seems like a good possibility for Mary's KB (including an inferred belief):

1. If I ask John to do something, he will do it.
2. I want him to call Sue.
3. So if I ask him to call Sue, he will call Sue. [this is basically modus ponens]

But this will not do. Belief 1 is far too bold. John won't do something if he does not understand what it is, or if he does not know how, or lacks needed resources. For instance, if Mary wants to have dinner in Paris, saying "Take me to Paris for dinner tonight" is not likely to work if they are currently in the US. So 1 should be amended to have preconditions to the effect that John understands the meaning, knows how, and has the resources. But these can get rather tricky:

To know the meaning of "Call Sue" John has to know not only English but also who Sue is; that is, he must have in his KB something like Person(ID, Sue) where ID contains enough info for him to contact her. And that means Mary's KB should have something like Knows(John, Person(ID,Sue)). But we are now leaving standard FOL, since we have a predicate symbol occurring as argument to another predicate symbol. Things get worse: John might know Sue, but not know her by that name. Maybe he knows her as "Ms_Smith", whereas (say) Mary knows her by both names. Then for Mary, "Sue = Ms_Smith" is true, so for her Person(ID,Sue) and Person(ID,Ms_Smith) are equivalent; but not for John.

Worse still: even if John knows Sue by that name, that does not mean he can call her. To be able to call her, he needs to know her number. But what kind of knowledge is that? He knows lots of numbers, all sorts of numbers. What he needs is to know that a particular number X is her number: Number(X,Sue), say. And even now we are not done: What sort of thing is this X? Not an ordinary number, even though we might write it as 301-344-3085. It is a string or list of digits.

So, yes, all of this can be put into a language and then the appropriate wffs might in fact be in Mary's KB in adequate form for her (realistically, this time) to infer that it in fact is a good plan to ask John to call Sue. But we had to go to a lot of trouble to see the details, and make a lot of choices along the way (which we did not fully spell out) about how to represent John's beliefs, names, numbers, etc.

What all this adds up to is that a big part of KRR is picking an appropriate ontology for the task at hand. An ontology is a specification of the things that will be represented in an agent language. Things like: names, numbers, people (not the same as names!), physical objects (liquids too?), words, ideas, colors, beliefs of others, moments in time, time in general, actions, relations, categories, etc. The more one includes, the more complicated the language gets, and the more complicated it can be to choose appropriate axioms about those things, the more likely an inconsistency might occur, and the harder to see how to make intuitively appropriate inferences.