Multi-Agent Systems

For the most part, this course has proceeded under a very strong assumption: that our agent is the only agent in the world. We briefly challenged this assumption in two places: in the context of reactive agents, we looked at what might happen if we had a society comprising a large number of reactive agents, all reacting to their world and to each other; and, in the last of our lectures on AI planning, we noted the techniques used by nonclassical planners to deal with the possibility that the world might change in ways that were not predicted by the agent's actions.

Systems in which there are multiple agents are called *multi-agent systems*, and it's time we looked at these more fully. (The investigation of multi-agent systems sometimes also goes under the name of *distributed AI*.)

In these notes, I will use the phrase "our agent" to refer to the agent which we are building and from whose perspective we are seeing things.

If there are multiple agents in the world, two things follow:

Our agent needs to model other agents. The actions of others may hinder, help or have no effect on the plans of our agent. So, when our agent is building its plans, it must anticipate the actions of other agents. It can only do this if it has, in its own knowledge base, representations of these other agents.

Our agent needs to communicate with other agents. One way other agents can cooperate with our agent is to provide useful information or to comply with requests from our agent to execute certain actions. One way other agents can compete with our agent is to provide misinformation or ignore requests from our agent.

So the rest of this lecture is split into two.

1 Models of Other Agents

Up to now, our agent's knowledge base has contained representations of the initial state of the world, the available operators, the goal and background knowledge (especially commonsense knowledge). But now it must also contain representations of other agents. We need to represent what we believe other agents believe about the world and what we believe their goals to be.

We'll need quite an expressive language for this, so we will use logical rather than iconic representations. While FOPL can be used, using an extension to FOPL is more common. The logic we use is called an *epistemic logic* and it is one example of a *modal logic*. (Wffs in this logic can be translated into corresponding wffs of FOPL, but the translation is far from concise.)

The idea is to introduce into logic a new operator Bel, which is supposed to capture the notion of an agent believing something. It's a binary operator: the first argument is a term denoting an agent; the second argument is a wff that the agent believes. Here's the new syntax rule:

- For any term T and wff W, then $\mathrm{Bel}(T, W)$ is a wff.

Notice that Bel is not a predicate symbol because predicate symbols only allow *terms* as their arguments. But here the Bel operator has a *wff* as one of its arguments. (There are alternatives to this approach. In some of the alternatives, Bel is a binary predicate whose second argument is a string: the string believed by the agent.)

Although it is very bad practice to introduce new syntax without precisely stating its semantics, we're going to do just that. Our previous approach to doing semantics, using just a universe of discourse and an interpretation function, is no longer adequate. The semantics of the Bel operator are quite complicated. (They are usually done using what's called a *possible worlds semantics*.) We'll muddle along with just our intuitive idea of what this operator means.

Here's an indication of the range of new wffs we can now write. (Let *ann*, *ben* and *col* be constant symbols denoting agents, where *ann* is 'our agent' and *ben* and *col* are other agents. Imagine these wffs are in *ann*'s knowledge base.)

Exercise. *Write down paraphrases of these wffs.*

- $\mathrm{Bel}(\mathit{ben}, \mathit{on}(a, b))$
- $\neg\,\mathrm{Bel}(\mathit{ben}, \mathit{on}(a, b))$
- $\mathrm{Bel}(\mathit{ben}, \neg \mathit{on}(a, b))$
- $\mathrm{Bel}(\mathit{ben}, \mathit{on}(a, b)) \vee \mathrm{Bel}(\mathit{ben}, \mathit{on}(a, c))$
- $\mathrm{Bel}(\mathit{ben}, \mathit{on}(a, b) \vee \mathit{on}(a, c))$
- $\mathrm{Bel}(\mathit{ben}, \mathit{on}(a, b)) \vee \mathrm{Bel}(\mathit{ben}, \neg \mathit{on}(a, b))$
- $\mathrm{Bel}(\mathit{ben}, \mathrm{Bel}(\mathit{col}, \mathit{on}(a, b)))$
- $\mathrm{Bel}(\mathit{ben}, \mathrm{Bel}(\mathit{ann}, \mathit{on}(a, b)))$
- $\mathrm{Bel}(\mathit{ann}, \mathit{on}(a, b))$

We now have to come up with more inference rules. Here are some possibilities. (They're meant to be suggestive of rules we might use. I'm not giving a rigorous treatment of this material.)

The first inference rule is that our agent knows how to do Modus Ponens ($\Rightarrow$-ELIM):

$$\frac{\mathrm{Bel}(T, W_1), \mathrm{Bel}(T, W_1 \Rightarrow W_2)}{\mathrm{Bel}(T, W_2)}$$

Then there is positive introspection: if the agent believes something, then it believes that it believes it:

$$\frac{\mathrm{Bel}(T, W)}{\mathrm{Bel}(T, \mathrm{Bel}(T, W))}$$

You might also want negative introspection (although some researchers argue against this one): if you don't believe something then you believe that you don't believe it:

$$\frac{\neg\,\mathrm{Bel}(T, W)}{\mathrm{Bel}(T, \neg\,\mathrm{Bel}(T, W))}$$

Our agent can now start to represent other agents and reason about those other agents.

In fact, we probably want to extend our logic even further. We may wish to introduce another operator: Want, where, e.g., $\mathrm{Want}(\mathit{ben}, \mathit{on}(a, b))$ signifies that $\mathit{on}(a, b)$ is one of Ben's goals. Some people (but not all) go further and introduce Intend, which is an operator with which we can write wffs that signify that an agent has committed to a goal or an action. (For example, in the literature you might come across the phrase 'BDI agents'. Here BDI stands for Belief-Desire-Intention. This is one particular proposed way of specifying and building agents.)

(Here are a couple of advanced points, which you can ignore.

- The trouble with our inference rules for belief is that they make our agent *logically omniscient*. The first inference rule, for example, means that our agent believes all the wffs that can be derived from its beliefs. If our agent believes this of all other agents in its world (including humans, for whom this is clearly not the case), then our agent will be reaching some wrong conclusions about these other agents.
- Suppose Lois Lane believes that Superman can fly. Suppose we believe that Clark Kent is Superman. Would it be true to say that Lois Lane believes that Clark Kent can fly? In one sense, yes: she does believe that the person who we are referring to using the name Clark Kent is a person who can fly. But in another sense, no: if you asked her whether Clark Kent can fly, she'd say no. Approaches to the modelling of belief try to handle this issue in a variety of ways. They talk about *referential transparency* and *referential opacity*. There's much debate and research in this area.)

2 Communication with Other Agents

Communication between *reactive* agents, as we saw, was typically implicit, unintended and indirect (through the effects of the agents' actions in the environment). But at this point in the course, we are interested in communication between *intelligent* agents, and this communication will typically be planned: it will be explicit, intentional and direct.

In this section, I'm going to use terminology that is very much concerned with 'talking'. The agent doing the communicating will be called the *speaker*; the one receiving the communication will be called the *hearer*; acts of communication will be called *speech acts*; and I will talk about the speaker producing *utterances*. Although the terminology is all very speech-oriented, in AI we use this terminology to refer to any form of communication (writing, hand signals, smoke signals, digital, etc.), not just speech.

Before we go any further, let's ask ourselves the following:

Question. *Why do agents communicate? What advantages does it bring?*

Speaking is an action, just like any other action, e.g. moving a block. These actions, like other actions, can be planned for. In saying that speaking is an action, I'm not really talking about the act of producing language (moving your tongue, etc.). I'm talking about the acts we can perform *using* language. These acts include such things as

- making statements,
- asking questions,
- issuing requests or commands,
- apologising,
- making promises,

and so on. These actions are called *speech acts*.

Consequently, an utterance has to convey two things, which I shall give the following names:

The force: what act is being performed (making a statement, asking a question, etc.)

The content: what the act refers to (e.g. the statement being made, the question being asked, etc.)

And we can expect speech acts to have *effects* on the hearer (e.g. if we make a statement, the effect is that the hearer's knowledge base is updated; if we ask a question, the effects is that the hearer utters some reply to our question, etc.). It's because we want these effects that we communicate in the first place. Of course, we cannot guarantee that we will get the effects we expect: communication often fails.

Given all that we've said about planned speech acts, it follows that we can write operators that can be used by a planning algorithm (such as POP, or maybe something tailored to this task) to decide on the speech acts we will later execute.

Here are example operators:

The action of informing the hearer of W
Op(ACTION: inform(*speaker*, *hearer*, W)
PRECOND: $\mathrm{Bel}(speaker, W) \wedge \neg \mathrm{Bel}(hearer, W)$
EFFECT: Delete: $\neg \mathrm{Bel}(hearer, W)$
Add: $\mathrm{Bel}(hearer, W)$)

The action of requesting the hearer to make W true
Op(ACTION: request(*speaker*, *hearer*, W)
PRECOND: $\mathrm{Want}(speaker, W) \wedge \neg \mathrm{Want}(hearer, W)$
EFFECT: Delete: $\neg \mathrm{Want}(hearer, W)$
Add: $\mathrm{Want}(hearer, W)$)

Planning using these operators requires the full sophistication of the planning algorithms we looked at. It's likely that we need to use hierarchical planners: at the highest levels, we would plan the force and content; at lower levels, we would plan the syntax and vocabulary to use; at the lowest level, we would plan the physical actions necessary to make the utterance (e.g. tongue movements).

We would also need to use nonclassical planners, because we would have to deal with unexpected effects. For example, the other agent might not answer a question we ask or might not comply with a command or request. Conditional planning and interleaved planning and execution (execution monitoring & replanning, continuous planning) are both necessary.

When it comes to the actual communication, there are two possibilities.

- Our agent could directly transmit the force (as a suitable symbol) and the content (as a wff of our logic) to the other agent. This, of course, requires that the two agents share the language (the logic) and the vocabulary, i.e. that they use the same constant symbols, function symbols and predicate symbols to mean the same things.

  This is a conceivable approach for communication between agents that have all been built to the same specification. However, even in this scenario, it makes very strong assumptions. For example, if one agent learns new vocabulary during its interaction with the world, then it cannot so easily communicate any wffs that use this vocabulary. If it did, the other agent would either not know the vocabulary or might even have some other meaning for the same symbol.
- The alternative is to translate the force and the content into some external communication language. Other agents can then translate from this language back into their own internal language (which might use a different syntax and a different vocabulary from the speaker's internal language).

Let's look at this second option in more detail.

The shared, external communication language might be

- another formal language, e.g. another logic; or
- a natural language, such as English, Gaelic or Navaho.

Communication using natural languages is well-suited to communication with human agents; communication with formal languages is well-suited to communication between programs. Let's look at formal languages first.

A number of formal languages for agent communication have been proposed. We'll mention two that are gaining in currency.

KQML (Knowledge Query and Manipulation Language) is a formal language designed to allow programs to communicate attitudes about information. In other words, it is mostly about communicating what I have called the 'force' of a speech act. (Unfortunately, if you ever read anything about KQML, you'll have to be careful because they use the word 'force' with a different meaning from here.) The 'forces' they have specified include: `tell` for making statements, `deny` for retracting statements, `ask-if` to ask certain questions, `reply` to reply to certain questions, `sorry` to signal an inability to answer a question, and `achieve` to issue a command/request.

KQML has nothing to say about how the 'content' of a speech act will be encoded. Some other language is used for this. But, in KQML, agents can also encode the identity of the sender, the identity of the receiver, the identity of the language used for the 'content', the identity of the vocabulary ('ontology') used for the content, and a few other things.

A complementary language is KIF (Knowledge Interchange Format). This is a variant of FOPL, but it has a syntax that is suitable for today's keyboards. It has been proposed to allow interchange of knowledge between disparate computers. It is suitable for encoding the 'content' part of a KQML message. (KQML allows you to use any language you wish; KIF is a common choice.)

The alternative to using formal languages such as KQML and KIF is to use natural languages. What are the differences between formal languages and natural languages?

The first difference is that natural languages are the results of evolution or are God-given. Formal languages, on the other hand, are human-designed artifacts. Swahili, for example, is the way it is today because, depending on your point of view, God gave it to us that way, or a long process of evolution made it that way, whereas the programming language Algol was defined by a committee.

A second important distinction is that natural languages are general purpose. It is possible, using a natural language, to express thoughts on almost any subject, from whether you like sugar in your tea, to philosophically abstruse topics such as the fallacy of Cartesian dualism, to talking about language itself (as we are doing here). Non-natural languages are always designed with a much narrower purpose in mind: while you can use Pascal, for example, to write programs to compute any computable function, you cannot use it to convey your feelings about Géricault's "The Raft of the Medusa".

A final distinction, that perhaps follows from the other two, is that natural languages possess properties that non-natural languages typically do not. The best example is that we can express ourselves in natural languages using terms that are *ambiguous*. An ambiguous sentence is one that has *multiple* meanings, as in (1), which is a renowned audacious First World War British newspaper headline:

(1) *"British push bottles up German rear."*

Our choice here of a humorous example makes the ambiguity obvious. From this it would be easy to conclude that ambiguity rarely occurs and is obvious when it does. This is not so. Most natural language sentences are ambiguous. But, in most text the reader does not notice more than one meaning. Somehow the alternative meanings are not spotted (or, if they are spotted, they are filtered out at a sub-conscious level of thought). This is remarkable when you realise that, in the presence of multiple sources of ambiguity in a sentence, the number of readings for the sentence grows multiplicatively rather than additively. Each of the words in sentence (2), for example,

(2) *"Flares save plane crash party."*

has (at least) two meanings: *"flares"* can be beacons or trousers, to *"save"* can mean to rescue or to conserve, a *"plane"* can be an aircraft or a wood-smoothing tool, a *"crash"* can be a collision or a loud noise, and a *"party"* can be a group of people or a social event. The meanings of this sentence therefore include the beacon-induced rescue of the group of people involved in an aircraft collision, and the conservation by voluminous trousers of social gatherings in which loud noises are made using carpenter's tools. Were we to enumerate all the combinations of the meanings of the words, we would find that this five word sentence has at least thirty-two (i.e. $2 \times 2 \times 2 \times 2 \times 2$), not ten (i.e. $2 + 2 + 2 + 2 + 2$), possible meanings.

Question. *What properties (other than ambiguity) distinguish natural languages and formal languages?*

What we're going to do in the next few lectures is look at communication from the perspective of the hearer. In particular, we're going to assume that the hearer agent receives an utterance that is expressed in a natural language (in this case, English) and we will see how it might translate from the English into its own internal language (logic). This is a considerable problem: the hearer may have to recover the force and content from a vague, ambiguous, ungrammatical, highly-contextualised English utterance.