LECTURE 1: The Role of Logic in AI

Intelligent systems need large amounts of knowledge, and the ability to use that knowledge for inference and purposeful action. Logic is an essential tool in the REPRESENTATION and use of much of this knowledge.

The need for knowledge

The crucial dependence of intelligent behavior on having large amounts of knowledge is by now a truism in AI; it was not obvious at first. But no general problem solving strategy, search strategy or learning strategy by itself has taken us very far toward intelligent behavior. Even very limited systems for language understanding, or systems for narrow tasks like chess, transportation planning, medical diagnosis, credit advising, etc., require large amounts of knowledge.

Some examples illustrating this point:

1. Story understanding; Some commonsense inferences from simple "stories" (the first was already mentioned in class):
   - Alicia drove back home from her holiday visit to the US; (what borders did she cross? Where might she live? how do we infer this?)
   - When Randy learned that he had terminal metastatic cancer, he decided against further chemotherapy and to enter a hospice program for palliative care. (Explain his decision.)

   These examples are from a paper, whose accompanying slides are at http://www.cs.rochester.edu/~schubert/papers/cognitum15-slides.pdf

Chung Hee Hwang and I did a detailed computational study of another example: from Little Red Riding Hood, just after the point where the wolf has knocked on the door of grandmother's cottage, pretending to be Little Red Riding Hood:

   The worthy grandmother was in bed, not being very well, and cried out to him, "Pull out the peg and the latch will fall."

   The wolf drew out the peg and the door flew open. Then he sprang upon the poor old lady and ate her up in less than no time, for he had been more than three days without food.

In a detailed analysis of the last passage (see Hwang and Schubert in Minds and Machines 3(4), Nov.93, or in Iwanska & Shapiro’s Natural Language Processing and Knowledge Representation, 2000, or http://www.cs.rochester.edu/~schubert/papers/el-meets-lrrh.pdf), the following facts were found to be necessary for understanding, among many others. Essentially what "understanding" means here is to figure out how the described events and states "hang together" in a causally coherent way, including agents' goals, intentions, beliefs, & behavioral dispositions:

- Causation is transitive
(Roughly) whatever a part of an event causes, the event as a whole causes as well
- If the conjunction of two facts holds because of a third fact, then each of the conjuncts holds because of the third fact
- If something flies open then it opens very quickly
- If something is "very P" then it is P
- If a person does a certain kind of action, and believes that this kind of action is a possible way to bring about a certain kind of event that is harmless to him, then he may INTEND that his action bring about that kind of event.
- If someone wants the door of a room or house to open while he is outside of it, that's probably because he wants to enter it.
- If one enters a room, one will then be inside the room.
- Individuals who are in the same room are near each other.
- Bad individuals are apt to do something bad when the opportunity arises.
- Eating requires food
- Creatures are very hungry when they have not eaten for more than a day
- Two successive event sentences in a narrative usually indicate that the two events happened in succession
- etc., etc.

2. **Dealing with a threat in nature**, e.g., when a bear shows up at your campfire breakfast on a canoe trip (an example based on personal experience). This can involve knowledge about bear behavior (with regard to food preferences, aggression, flight, etc.), operating a camera for filming the adventure and knowing what to do if the battery happens to die just then, knowing what can be used in self-defense (axe? paddle? noise?), everything involved in breaking camp quickly, loading up canoes and fleeing as soon as the bear goes off with the box of powdered milk, etc. Also recall the harrowing tale I mentioned where survival depended on human knowledge & ingenuity, as related in Truman Everts' Lost in the Yellowstone (see http://www.yellowstonepark.com/2007/01/truman-everts/).

3. **Medical diagnosis**: have a look at some medical text, such as Krupp et al., Current Medical Diagnosis and Treatment!

The same goes for any other technical field or branch of mathematics.

People know at least millions of particular and general facts, such as

- the names, personal characteristics (gender, approximate age, general appearance, interests, skills, occupations, personal history, etc) of many family members, friends, associates, and public figures
- general facts about people (appearance, body-parts, that they eat and breathe and sleep, how they come into being, mature, age, and die, how they think and feel, how they interact socially, or in education, in business and commerce, etc)
- where they live, what their daily routine is, what their belongings are, their particular characteristics and where they are located
- general facts about classes of natural objects, artifacts and substances such as what dogs look like, that they bark, are generally friendly domestic animals, have fur, are mammals; that pieces of writing paper are flat and rectangular and of a certain size, and can be written, typed or drawn on, folded, crumpled, burnt, etc.; that parking lots typically have large paved level surfaces, and markings indicating permissible locations of cars, etc
- particular and general facts about history, movies, sports, politics, music, crime, math, science, literature, etc.
- What sorts of situations and events are apt to occur in the world -- dogs barking, traffic moving on roads, people working at jobs, people eating, sleeping, opening doors, chatting with family, friends, co-workers, watching TV, etc., the patterns of day/night and seasons, clouds moving overhead, the sun and other light sources casting shadows, etc., etc. -- we appear to know tens of millions of such often-encountered situations and patterns of events.

- particular and general facts about words and their meanings

People can state these facts in ordinary language. And they can use them for reasoning and action. I use the term "reasoning" very generally, to include such things as

- deducing logical consequences of given facts;
- figuring out a plan to achieve a goal;
- engaging in familiar patterns of behavior & interaction (chatting, dining,....)
- inferring causal connections between what we see or are told;
- inferring explanations for given events or situations;
- predicting what will happen in a given situation
- recognizing certain types of situations and events as (un)familiar;
- figuring out what a speaker meant or intended;

If we are going to endow machines with human-like intelligence, they will have to acquire an amount and variety of knowledge comparable to that of people, in a form that will enable them to talk about and use this knowledge like people do.

While the goal of endowing machines with commonsense knowledge and reasoning abilities is the most central in AI, we should also note the need for representing formal mathematical and technical knowledge and reasoning. For example, it is a long-standing dream to build an automated mathematician or mathematician's assistant that would greatly boost progress in mathematics, helping with very complex proofs and perhaps even formulating mathematical concepts that are natural and relevant in a particular subdomain. In fact, significant results have been achieved along such lines. Similarly we would like reasoners that can design or verify complex circuits or programs, or help with biochemical synthesis or analysis, or with the complex mathematics and reasoning involved in modern physics (e.g., the physics of elementary particles, or of plasma instability in attempts at controlled fusion, etc.).

But even mathematical and technical work involves heavy use of acquired knowledge and much intuitive commonsense reasoning, so we probably shouldn't draw any sharp lines: mathematical and scientific knowledge and reasoning are refinements of commonsense knowledge and reasoning, and probably continuous with them.

The road to knowledge and commonsense reasoning

The knowledge acquisition bottleneck was already alluded to in "lecture 0"; it refers to the difficulty of doing the above. How do we get this vast amount of knowledge into computers, and how do we get them to use it effectively? Some possible answers:
• **Knowledge engineering:** Find a general way to **represent** knowledge; have "knowledge engineers" code up the knowledge (e.g., Doug Lenat, CACM 38(11)); or acquire the knowledge **through language** (being told, reading), perception/physical exploration, and through concept formation and generalization learning. A special case of coding up knowledge is what the Big Tech companies are doing: They are code factories employing tens of thousands of coders to engineer "skills" that make their products, such as Alexa or Siri, marketable. Their goal is not a well-founded basis for general understanding and problem solving, and expansion of skills depends on further investment of human expertise & labor.

• **Rule-based systems:** Build mechanisms for making inferences, basically by pattern-matching: we use rules that say if you have a set of facts matching such and such patterns (i.e., having a certain form), then you can draw such-and-such a conclusion (having a certain form based on the input facts). E.g., two such rules might be (one is quite generic and "sound", the other not so much):

  Given: [fact1 or fact2], not(fact1)  
  Conclude: fact2

  Given: bird(x)  
  Conjecture: flies(x)

• **Neobehaviorism:** Mount a big "behavioral engineering" effort; instead of worrying about representation and inference, focus on what an intelligent agent DOES, what its skills and modes of behavior are; build small subsystems to capture these skills and modes of behavior (using inspiration from natural systems, as well as from math, control theory, engineering, introspection, etc.); learn how to put them together so that the right behaviors are "in charge" at the right time, and they cooperate; layer them into ever larger, more ambitious systems, ...

• **Universal ML:** Build a very general learning machine, based as closely as possible on neural mechanisms and brain organization, with a built-in tendency to act in a way that optimizes "rewards" (positive feedback); embody it & place it in a stimulating, realistic learning environment; it should learn facts and inference methods implicitly, just like people. E.g., see the Blue Brain project (now focused on the mouse brain)
  http://bluebrain.epfl.ch/cms/lang/en/pid/56882,
  or Chris Eliasmith's approach to building a brain (focused on neuromorphic engineering),
  http://arts.uwaterloo.ca/~celiasmi/
  The DNN aspects of Google's DeepMind,
  are also viewed as mimicking some aspects of human brain function (pattern learning, and reinforcement learning).

• **Simulated evolution/ genetic programming:** Build virtual creatures capable of rapid self-reproduction, and genetic change, letting many variants of them thrive and reproduce in some virtual environment that contains hazards and rewards... Intelligence and learning, including learning of facts, will come automatically as the artificial beings evolve...

My belief is that we should and will eventually do ALL of these things. I absolutely do not think we can circumvent the knowledge representation and inference approach, and build a human-like agent just by engineering skills and behaviors, or building the ultimate learning machine (e.g., deep neural nets), or simulating evolution.
Here’s a longstanding “law” of technological progress which probably still holds, despite DNNs:

*If you can't provide a convincing, detailed argument why a device X that you are proposing to build will do Y (e.g., understand language; do commonsense reasoning; learn to talk or plan; etc.), it won't.*

The point is this: We can provide convincing, detailed arguments (and of course, many sample systems) showing how we can use explicit knowledge representations and explicit inference methods to enable a computer to talk and reason about various problem domains -- not yet at a human level, but nontrivially.

There is as yet nothing to compete with that -- e.g., no account of how behavioral engineering, or use of GPT (generative pretrained transformer) models, would lead to a system that can converse meaningfully (not just mimic meaningful discourse it has been trained on), or figure out how to stack up some blocks (or more generally solve problems or plan). And they won't become competitive in those areas till we CAN give such an account, in convincing detail.

Of course, there are also kinds of "how-to" knowledge that probably are best represented in some non-symbolic information processing mechanism:
- how the tune for "Twinkle, twinkle" goes
- how to ride a bike, catch a ball, use chopsticks, tie shoelaces
- how to interpret images falling on our retinas
- how to speak grammatically
- how to learn

**Levels of description, declarativism, proceduralism**

I think in discussions of these issues there is often a confusion about *levels of description* of a knowledge-based agent (see R & N, p152-2):

- the *knowledge level*, or *epistemological level*: what an agent knows, and knows how to do
- the *logical level*: the actual form (syntax) and semantics of the representations used
- the *implementation level*: the code and data structures

For instance, when we talk about neural nets in terms of the layers of neurons, their interconnections, activation functions, and backpropagation, we are describing a (possible) agent largely at the implementation level. We will also have to learn to describe it at the logical and epistemological level before we can successfully engineer (coherent) talking or reasoning systems using neural nets. Neural nets aren't an *alternative* to representation (nor are "subsumption architectures"); they are a way of implementing logical-level knowledge (in the broadest senses of “implement” and "logical").

Finally, a word about the distinction between *declarative* and *procedural* knowledge. Declarative knowledge consists of explicit, interpretable representations of facts, such as

(forall x) dog(x) => can-bark(x),
dog(Odie)

B & L emphasize that declarative representations are (i) transparent, (ii) modifiable, and (iii) potentially usable for many purposes. By contrast, procedural knowledge (e.g., how to multiply two numbers, invert a matrix, ... or catch a ball, ride a bicycle, or pronounce a word) is typically
special-purpose, and computer code that does such tasks is hard to understand and modify, or use for a different purpose.

This is true, but I think the key distinction is not so much the declarative/procedural one as the distinction between representations that contain *externally meaningless symbols* and those that don't. For instance, consider the following two procedural representations of a routine that makes a robot say "hello" when a person comes near:

1. declare (G,string,"hello")
2. declare (F,boolean,nil)
3. declare (F',boolean,nil)
4. turn-on(proximity-sensor)
5. Repeat-at-intervals(1, sec):
   a. let F' := read(proximity-sensor)
   b. IF F' and (not F) THEN write-stream(voice-output,G)
   c. let F := F'.

1. Turn-on(proximity-sensor)
2. Repeat-at-intervals(1, sec):
   IF on(proximity-sensor, now) AND
   off(proximity-sensor,ago(1,sec))
   THEN utter("hello").

When we consider knowledge relevant to a particular domain, *externally meaningless symbols* are symbols that don't denote anything in that domain -- typically, they are parameters and operators concerned with *internal* book-keeping and data manipulation.

In the first version, the following symbols are externally meaningless: declare, G, F, F', string, boolean, nil, let, :=, read, write-stream, voice-output.

They denote nothing *in the domain*, and have only internal significance.

In other words, clarity, modifiability, and re-usability require that we ABSTRACT AWAY from the level of internal data manipulation, specifying actions strictly at the level of domain entities, properties, and operators.

NOTE: Cooking recipes and furniture assembly instructions can be just as understandable as statements of fact or stories -- yet they are procedural!

A comment on B & L's "Snow is white" example: Their first prolog program directly encodes that the correct response to the query "Color of snow?" is to print "Snow is white", while the second program bases its response on the FACT that snow is white, explicitly represented (along with other color knowledge) as a prolog clause. This nicely illustrates that basing "what to do" on factual knowledge (rather than simply behavioral rules) provides flexibility and generality. But observe that both programs still embody behavioral rules, and apart from the "!" operator ("don't do any goal-chaining on the remaining literals of the clause) are quite comprehensible at the domain level -- i.e., how to answer certain color questions. In that sense, the examples don't impugn procedural representations at all.
CSC 444: Comments on Brachman & Levesque's introduction

Their emphasis is on defining knowledge representation and reasoning, and justifying an approach to AI based on KR&R.

They define knowledge in terms of propositions that are known, believed, expected, hoped-for, etc., by an agent.

According to Brian Smith's "KR hypothesis", mechanized intelligence will require internal symbolic representations of propositions that are interpretable by the designer (and other observers), and that are the basis for the intelligent agent's operation.

Reasoning is the derivation of new propositions from ones explicitly represented. They give a nice example:

Suppose you are told that
- John is allergic to penicillin; and
- Anyone allergic to penicillin is also allergic to ampicillin.

We would want to conclude that John is allergic to ampicillin. Another example they give is more verbally oriented:

Suppose you are told that
- John loves Mary (and these are persons); and
- Mary is coming to John's party.

Then we would agree that "Someone that John loves is coming to his party".

One point they make is that it is hard to see how facts like those mentioned in the examples (or ones like "Nearly half of Peru's population lives in the Andes") could be committed to memory by an agent *without* some form of propositional representation. Keep in mind that these facts could be used in combination with a variety of other facts to make inferences; and there seems to be no procedural representation that could make use of those facts in all situations.

The point could be further dramatized by imagining learning from a book. For example, suppose you read about the geography, history, and culture of Peru. After doing the reading (if you make some mental effort in the process!) you'll know quite a bit of what you've read, and could answer questions, plan a vacation that suits your tastes, make conjectures about the future of Peru, etc. Surely the simplest assumption to make about how this is possible is that we represent the memorized information in some symbolic internal language, perhaps not far removed from language. (Since we use language to communicate knowledge, it would be surprising if the internal encoding of knowledge were vastly different from language at all levels of structural and functional abstraction.)

They relate the kinds of inferences mentioned above to *logical entailment* (and I think it's a good idea to introduce this concept so early in the book). They explain entailment in terms of imagining how the world would have to be if the given propositions are true. For example, in the case of the inference concerning penicillin and ampicillin, if we imagine a world in which John is allergic to penicillin, and everyone who is allergic to penicillin is also allergic to ampicillin, then in such a world John *must* be allergic to ampicillin.
They justify the use of logic in terms of entailment, pointing out that one way of looking at logic is as a theory of entailment relations that hold between symbolically represented propositions.

They also already introduce the notions of *soundness* and *completeness*: a logical system is sound if the conclusions it allows you to derive are always entailments of the premises; and it is complete if it allows you to derive *all* entailments of any given premises.

A caveat:

At a couple of points, B & L imply that it is appropriate to think of the *beliefs* (or knowledge) of an agent, assuming that it represents propositions symbolically, as all the entailments of its explicitly represented propositions.

But this is a notion of "belief" (or knowledge) that is badly out of joint with intuitive notions of belief. Do you know the rules of chess, or Go? If so, do you also believe that the first player can, in principle, force a win? Or that the second player can force a win? Or that neither is the case? I doubt it very much -- even though one of those conclusions follows from the rules!

In general, what you can *infer* is not the same as what you *know* (or *believe*); and it is entirely possible that a proposition that is entailed by what we know is unknown to us -- yet becomes known when the entailment is pointed out. Example: though you may know that the assassins of Lincoln and JFK were John Wilkes Booth and Harvey Lee Oswald, you probably didn't know (until now!) that their names have same number of letters -- even though this is an easy consequence of your prior knowledge!