COMMENT ⁻ VALID 00003 PAGES C REC PAGE DESCRIPTION C00001 00001 C00002 00002 C00003 00003
CS326
C00033 ENDMK C⁻;

CS326
Winter 1987

Formalizing Commonsense Knowledge and Reasoning: An Introduction

1. We are studying the art of *formalizing commonsense knowledge* in logical languages.

The process of formalizing a body of knowledge begins with designing a language in which the facts involved can be represented. By translating the facts into that language we get a set of sentences which can be called a *knowledge base.*

The next step is to make sure that *the process of commonsense reasoning* translates into *deduction:* whenever common sense tells us that a sentence A follows from the facts included in the database, we want A to be a *logical consequence* of those facts. Usually we find out that this is not the case unless some "self-evident" facts, implicit assumptions about the objects of reasoning, are added to the knowledge base. It may be non-trivial to decide whether such additional axioms are, in fact, needed; this is similar to independence problems in logic. Also, formalizing the implicit assumptions is at times a difficult task. It may require major changes in the design of the language.

2. As an example, consider the blocks world with 3 blocks: A, B, C. The blocks A and B are on the table, C is on A. Which blocks are clear? We want to express the given facts in a logical language and then derive the answer (B and C are clear, A is not) by deduction.

The given facts can be formalized in the language with variables for blocks, which has 3 object constants A, B, C and 2 predicate constants: the binary predicate *on* and the unary predicate *ontable*. Then the information about the configuration of blocks can be expressed by the formulas:

$$ontable\ A, \quad ontable\ B, \quad on(C, A). \tag{1}$$

A block is clear if it has no blocks on top of it. There are two ways to formalize this definition. We can add the unary predicate *clear* to the language and the formula

$$clear\ x \equiv \neg\exists y\ on(y, x)$$

to the knowledge base. Alternatively, we can define *clear* as a metamathematical abbreviation: for any term t, $clear\ t$ stands for the formula $\neg\exists x\ on(x, t)$, where x is a variable which does not occur in t. Let's treat *clear* as an abbreviation. We want to show that the formulas $\neg clear\ A$, $clear\ B$ and $clear\ C$ follow from axioms (1).

It is easy to prove the first of these formulas: it is equivalent to $\exists x\ on(x, A)$ and thus follows from the axiom $on(C, A)$. But the formulas *clear* B and *clear* C can't be derived from (1). More axioms are needed.

3. One essential assumption which is not made explicit in (1) is that the blocks world contains *no blocks other than* A*,* B *and* C. If there are blocks different from those then some of them may be located on top of B or C, so that B and C may not be clear. This assumption can be expressed by the formula

$$x = A \lor x = B \lor x = C. \tag{2}$$

Axioms of this kind are called *domain closure* axioms.

Remark. It may be argued that (2) isn't the right way to interpret the given description of the blocks world. We have never actually said that there are no blocks other A, B and C. We can imagine that the description is followed by additional information: "There is also a tower of 10 blocks $D_1, \ldots, D_{10}$ located on the table". Perhaps the reasoning here is "reasoning by default"; since no other blocks are mentioned, we conclude that there are none (or at least that no other blocks are essential for finding the right answer). This form of reasoning is called *domain circumscription*. It was described in the paper "Epistemological Problems of Artificial Intelligence" by McCarthy (1977). Formalizing domain circumscription is an important and relatively difficult problem. We will discuss it later in the course.

4. Axioms (1), (2) are still not sufficient for proving that B and C are clear. How do we know that a block can't be located on two blocks at once? If this is possible then we can imagine that C rests partially on A and partially on B, so that B is not clear. Let's assume then that

$$on(x, y) \land on(x, z) \supset y = z. \tag{3}$$

Similarly, we have to assume that a block cannot be located on the table and on another block at the same time:

$$on(x, y) \supset \neg ontable\ x. \tag{4}$$

Without this assumption, it is possible that A rests partially on the table and partially on B, so that B is not clear again.

5. Even with these additional axioms, it is impossible to prove that B and C are clear. How do we know that the objects denoted by A, B and C are actually different from each other? Perhaps A and B are two names of the same block, and axiom (3) doesn't prevent C from being on B. Let us add 3 more facts to the knowledge base:

$$A \neq B, \quad A \neq C, \quad B \neq C. \tag{5}$$

Assumptions of this kind are called *unique names* axioms.

Remark. As with the domain closure conditions, it is possible to treat the uniqueness of names in a more subtle way: since we are not told that A and B denote the same block, we conclude by default that this is not the case. This is discussed in "Applications of circumscription to formalizing commonsense knowledge" by McCarthy (1986), and we'll return to this issue later.

6. With axioms (1)—(5), we can prove that B is clear. Assume that it is not. Then there is a block x which is on B. By (2), x is one of the blocks A, B, C. It can be neither A nor B, because, by (1), A and B are on the table, and then (4) implies that they can't be on B. It follows that $x = C$. Thus C is on B. Since, by (1), C is on A, we can conclude, using (3), that $A = B$, contrary to (5). This contradiction shows that B is clear. In a similar way, we can prove $clear\ C$. (Do it!)

Remark. For those familiar with logic programming we can make the following observation. Formulas (1) can be viewed as the rules of a Prolog program defining the predicates *on* and *ontable* (except for the fact that, in Prolog, constants must begin with lower-case letters, the syntactic detail which we'll ignore for the moment). Then we can run Prolog to find all blocks x satisfying the condition $on(x, B)$. The answer will be empty, which means that B is clear. In a similar way, Prolog will tell us that C is clear also. This result may seem puzzling: we have seen that axioms (1) are insufficient for determining whether B and C are clear, but Prolog somehow manages to arrive at the answer without using any additional facts. The reason is that Prolog treats a program as the "complete definition" of the predicates involved; because formulas (1) don't provide information on any blocks located on B or C, Prolog concludes that there are none. Reasoning of this kind is called *closed-world reasoning*. Later in the course we'll discuss closed-world reasoning and the semantics of logic programming in more detail.

7. Knowledge bases like the one constructed above are axiomatic theories, and we can study their properties using concepts of mathematical logic. For instance, we can ask whether such a theory is complete, whether its axioms are independent, what its models look like, etc.

Consider, for instance, some properties of the theory T^b with axioms (1)—(5). First, we want to construct a model of this theory and show in this way that the theory is consistent. (It would have been strange, of course, if formalizing an informally described "toy world" could lead to an inconsistent theory. But in more complex cases the intuition we are trying to formalize may be somewhat vague, and then it may be worthwhile to check that the axioms don't lead to an inconsistency or other undesirable results.)

As the universe of the model M we can select any set of cardinality 3; let's call its elements A^*, B^* and C^*. They will represent the object constants A, B and C. Symbolically:

$$|M| = \{A^*, B^*, C^*\}, \quad M[\![A]\!] = A^*, \quad M[\![B]\!] = B^*, \quad M[\![C]\!] = C^*.$$

(By $M[\![s]\!]$ we denote the interpretation of a symbol s in M). The interpretations of *on* and *ontable* are defined as follows:

$$M[\![on]\!] = \{\langle C^*, A^*\rangle\}, \quad M[\![ontable]\!] = \{A^*, B^*\}.$$

It is easy to check that all axioms (1)—(5) are true in M, so that M is indeed a model of T^b. (How should the definition of M be modified if *clear* is treated as a primitive rather than an abbreviation?)

Problem 1. Prove that every model of T^b is isomorphic to M (i.e., for every model M' of T^b there exists a 1—1 correspondence between the universe of M and the universe of M' which maps the intepretation of any symbol s in M into its interpretation in M').

If all models of a theory are isomorphic then the theory is called *categorical.* Thus we can say that T^b is categorical. In the foundations of mathematics we hardly ever deal with categorical first-order theories: it is known that such theories can't have infinite models. In applications of logic to AI the situation is different, because in AI we are often interested in finite systems.

Every categorical theory is complete, i.e., every sentence is either provable or refutable in it. (Why?) Thus axioms (1)—(5) are sufficiently strong not only to decide which blocks are clear but to answer any questions stated in the language of T^b.

A theory *admits quantifier elimination* if, in that theory, every formula is equivalent to a formula without quantifiers.

Problem 2. Show that T^b admits quantifier elimination. Which axioms of T^b did you use in the proof?

If a theory admits quantifier elimination, and every atomic sentence ("ground atom") can be either proved or refuted in that theory, then the theory is complete. (Why?) This method of proving completeness is called the *quantifier elimination method.*

Problem 3. Prove the completeness of T^b using quantifier elimination.

8. In Sections 2—5 we used informal arguments to justify the inclusion of axioms (2)—(5). Each of these arguments can be presented as a proof of the fact that the corresponding axiom is indeed necessary. For instance, let us show that it is impossible to prove *clear* B without axiom (4). Let the interpretation of *on* in M be changed as follows:

$$M[\![on]\!] = \{\langle A^*, B^*\rangle, \langle C^*, A^*\rangle\}.$$

The new structure satisfies all axioms but (4), and the formula *clear* B is false in it.

Problem 4. Show that *clear* B cannot be proved without axioms (5).

Problem 5. Show that axiom (2) doesn't follow from the other axioms.

9. The investigation of "toy worlds", such as the blocks world discussed above, is useful only as long as it leads to ideas applicable to larger and more complex systems. In connection with that, let us return to axioms (5). of the uniqueness of names axioms in a world with N objects is proportional to N^2. This may be objectionable if the knowledge base is to be made a part of a reasoning program. Several approaches to this problem are proposed in McCarthy's 1986 paper. Here is one of them.

Let U_N ($N \geq 0$) be the theory with N object constants $C_1, \ldots, C_N$ (and no function or predicate constants other than equality), whose axioms are

$$C_i \neq C_j \qquad (1 \leq i < j \leq N).$$

Each of these axioms is independent. (Prove it!) But it is possible to replace the axioms of this theory by $N+1$ short axioms if we extend its language. Let U'_N be the theory whose language contains, in addition to $C_1, \ldots, C_N$, the binary predicate constant $<$, and whose axioms are

$$C_1 < C_2, \ldots, C_{N-1} < C_N,$$

$$x < y \wedge y < z \supset x < z, \quad \neg x < x.$$

It is easy to see that each axiom of U_N is a theorem of U'_N.

The predicate $<$ plays an auxiliary role. The axioms for it are selected just for the purpose of making the axioms of U_N provable. We don't expect them to imply any additional facts in the language of U_N. More precisely, any theorem of U'_N which doesn't contain $<$ is provable in U_N. An extension T_2 of a theory T_1 is called *conservative* if every theorem of T_2 which belongs to the language of T_1 is provable in T_1. Thus we can say that U'_N is conservative over U_N.

To prove this claim, we observe that every model M of U_N can be extended to a model of U'_N by interpreting $<$ as follows:

$$M[\![<]\!] = \{\langle M[\![C_i]\!], M[\![C_j]\!]\rangle \;:\; i < j\}.$$

Hence every formula without $<$ which is true in all models of U'_N is true in all models of U_N.

Problem 6. U_N is extended by the unary predicate *small* and the axioms

$$\exists x\, small\ x, \quad \exists x \neg small\ x.$$

Prove that this extension is conservative if and only if $N > 1$.

Problem 7. Is U_{N+1} conservative over U_N?

10. Axiom (1) describes one particular configuration of blocks, whereas the other axioms describe general properties of the blocks world. We can visualize 13 different configurations of blocks: 6 in which all blocks form a single tower, 6 in which two blocks form a tower and the third is on the table, and 1 in which all blocks are on the table. Axioms (2)—(5) do not give a complete description of all common properties of these configurations. They do not tell us, for instance, that A and B can't be both located on top of C. They do not tell us that a block can't be located on its own top, that A can't be on B when B is on A, and so forth.

Intuitively, it is impossible to have both $on(A,B)$ and $on(B,A)$ because $on(x,y)$ implies that x is "above" y, and, in any given configuration, the blocks are partially ordered by the relation "above". Axioms (2)—(5) don't grasp that idea.

It is not that easy to fix this problem. In a given configuration, x is above y if there is a sequence of blocks $x_1,\dots,x_n$ $(n>1)$ such that

$$x = x_1,\ on(x_1,x_2),\ \dots,\ on(x_{n-1},x_n),\ x_n = y.$$

Without using the fact that there are only 3 blocks in this blocks world, we can't restrict n and express this definition in the language of T^b.

Research Problem. Add the binary predicate *above* to the language of T^b as a new primitive and write axioms in the extended language which express the common properties of possible configurations of blocks. Check that your formalization is adequate by proving that the axiom set has exactly 13 non-isomorphic models, each corresponding to one particular configuration. Investigate the independence of the axioms.

11. The same body of facts can be formalized in many different ways, using different formal languages. Consider, for instance, the formalization of the problem from Section 2 in the language which contains, in addition to variables for blocks, also variables for *locations*.

We will use the symbols b, b_1, ... and l, l_1, ... as variables of these two sorts. The definition of a many-sorted language should specify, for each object constant, its sort; for each predicate constant, the sort of each of its arguments; for each function constant, the sort of each of its arguments and the sort of its value. The new formalization of the blocks world has the block object constants A, B, C, the location object constant $Table$, and two unary function constants, top and $location$, both applied to blocks and returning locations. There are no predicate constants, so the only atomic formulas are equalities. (The syntax requires that both sides of an equality be of the same sort.) Notice that, even though there are function symbols in this language, it has only a finite number of ground terms, and consequently a finite number of ground atoms. (How many?)

The axioms of the new theory T^{bl} are:

$$A \neq B, \quad A \neq C, \quad B \neq C,$$

$$top\ b \neq Table,$$

$$top\ b_1 = top\ b_2 \supset b_1 = b_2,$$

$$b = A \lor b = B \lor b = C,$$

$$location\ A = Table, \quad location\ B = Table, \quad location\ C = top\ A.$$

Problem 8. Define $clear\ t$ as $\neg\exists b(location\ b = t)$ (where t is a location term) and prove

$$\neg clear\ top\ A, \quad clear\ top\ B, \quad clear\ top\ C$$

in T^{bl}.

Problem 9. Construct a model of T^{bl}. (To define its universe, you have to specify two disjoint sets, the domain $|M|^b$ of blocks and the domain $|M|^l$ of locations.)

12. The language of T^{bl} is essentially more expressive than the language of T^b. For instance, in the language of T^{bl} we can express the assertion that there are locations other than the table and the tops of blocks.

Problem 10. Show that this assertion can be neither proved nor refuted in T^{bl}.

Problem 11. Are there formulas without location variables which can be neither proved nor refuted in T^{bl}?

We can describe the transition from T^b to T^{bl} by saying that we introduced locations in our *ontology* (things that exist; things that may be the values of variables), or that we *reified* locations. In this course we'll encounter still richer formalizations of the blocks world, in which other things are reified, such as states (the configurations of blocks) and events (for instance, moving a block).

Whatever can be expressed in the language of T^b, can be also expressed in the language of T^{bl}, and whatever can be proved in T^b can be proved in T^{bl} also, after an appropriate reformulation. We can make this assertion precise if we define a translation α which assigns to each formula F in the language of T^b the formula αF in the language of T^{bl} which, intuitively, has the same meaning as F. The translation is defined as follows. First we substitute distinct variables for blocks for the object variables in F. Second, we replace each part of the form $on(t_1, t_2)$ by $location\ t_1 = top\ t_2$, and each part of the form $ontable\ t$ by $location\ t = Table$. The resulting formula is αF.

The image of each axiom of T^l is provable in T^{bl}. (Check this for axioms (3) and (4).) It follows that the images of all theorems of T^l are provable in T^{bl} also.

Problem 12. Can it happen that αF is provable in T^{bl} even though F is not provable in T^b?