Jespersen, O. (1917). Negation in English and other languages. *Historisk-filologiske Meddelelser* 1.5.
Khanmagomedov, B. G.-K. (1967). Tabasaranskij jazyk. *Yazki naradov SSSR*. Tom I–V. Moscow: Nakua.
Klima, E. (1964). Negation in English. In Fodor and Katz, (eds) *The Structure of Language: Readings in the Philosophy of Language*. Englewood Cliffs: Prentice Hall.
Kuipers, A. (1967). *The Squamish Language. Grammar, Texts, Dictionary*. The Hague: Mouton.
Langdon, M. (1970). A grammar of Diegueno. The Mesa Grande dialect. *University of California Publications, Linguistics* 66. Berkeley and Los Angeles: University of California Press.
Lehmann, W. (1974). *Proto-Indo-European Syntax*. Austin and London: University of Texas Press.
Li, C. (ed.) (1975). *Word Order and Word Order Change*. Austin and London: University of Texas Press.
Steele, S. (1975). On some factors that affect and effect word order. In C. Li (ed.) *Word Order and Word Order Change*. Austin and London: University of Texas Press.
Tesnière, L. (1959). *Eléments de Syntaxe Structurale*. Paris: Klincksieck.
Voegelin, C. F. and Voegelin, F. M. (1977). *Classification and Index of the World's Languages*. New York: Elsevier.
Welmers, Wm. E. (1973). *African Language Structures*. Berkeley and Los Angeles: Univ. of California Press.
Wessén, E. (1965). Svensk språkhistoria. III. Grundlinjer till en historisk syntax. Stockholm: Almqvist and Wiksell.

The organisation of purposeful dialogues*

RICHARD POWER

Abstract

The first part of this paper describes a computer program which models a conversation between two robots. The robots co-operate to achieve a simple practical goal in a world of a few objects, and their co-operation is genuinely facilitated by a conversation in which every remark is ultimately related to the goal they are trying to achieve. They conduct several types of conversational exchange (e.g. agreeing plans, obtaining information), each exchange being controlled by a 'conversational procedure', which is a list of instructions to be performed by two agents, a 'first speaker' and a 'second speaker'. These procedures are called by the procedures which control the underlying planning.

The second part of the paper discusses some deficiencies of the program. In particular, it is argued that the robots lack an explicit representation of how utterances achieve their purposes, and with the aid of some recent philosophy of language a possible representation is suggested.

Part I. A Computer Program Which Conducts Simple Conversations

1. *Introduction*

1.1. *Why the Program was Written.* Linguistic acts belong to the repertory of methods by which we achieve our goals. If I want to know the time I can either look at a clock or ask a passer-by; if I want the light on I can either press the switch myself or ask someone who is closer. A speaker is not just putting a meaning into words, he is also trying thereby to do something to achieve a purpose. And a hearer has not understood a remark unless he has perceived its purpose as well as its literal meaning.

There has been a tendency for the discipline concerned with language to ignore the purposes for which language is actually used and to regard

Linguistics 17 (1979), 107–152.

it solely as a means of representing facts. This narrow view has been corrected in philosophy by Wittgenstein (1953) and Austin (1962), and Austin's notion of 'Speech Act' has infiltrated linguistics. But until very recently (e.g. Cohen, 1977) there has been little work in Artificial Intelligence on the purposeful use of language. This is surprising because there has been a good deal of work on language understanding and goal-directed behaviour; but the two have not got together. Winograd's SHRDLU (1971), for example, has a planning system (a variant of Hewitt's PLANNER [unpublished]) coupled with an impressive language-understanding system, but the planning system cannot achieve goals by producing utterances, and the language-understanding system cannot perceive the goals of the human interrogator. This is of course a perfectly legitimate restriction of subject-matter; my point is that Artificial Intelligence as a whole should not neglect these problems.

The inability of programs like SHRDLU to use language purposefully is brought out by the fact that they could not converse with near copies of themselves. When SHRDLU is interrogated by a human operator, the operator always takes the initiative, and the processes which give the conversation its overall structure are concealed inside his head. One way of making these processes explicit, therefore, is to create a system in which the conversants are two sections of program rather than a program and an operator.

This paper describes such a system (1974). Two sections of nearly identical program called John and Mary inhabit a world consisting of a few objects. They are able to achieve simple practical goals in this world, either by working separately or by co-operating. When they co-operate they use conversation in order to agree plans, exchange information, compare beliefs, and assess the results of their actions. The conversation is conducted in a small subset of English, and is printed out for the benefit of observers.

When the program is compiled, John and Mary are identical sections of code. Before running a conversation, the operator has to initialise a number of special variables which determine the robots' beliefs, goals, and abilities. This is how John and Mary are individuated. The operator also initialises a variable representing the state of the world. He then calls a function which I shall call the 'chairman', which operates a simple time-sharing arrangement. It calls the John and Mary sections alternately, control normally passing every time an utterance is made, and returns control to the operator when the robots have nothing further to say.

Practical co-operation is naturally only one of the functions of conversation. This type of conversation was chosen because its purpose

is definite: a remark is relevant if and only if it is related to the process of achieving the goal. The subtler uses of language (humour, sarcasm, rhetoric, etc.) are disregarded, not because they are unimportant, but because they are more difficult and should be studied later.

A difficulty in writing a program of this kind is that to understand the point of a remark one must normally begin by understanding its literal meaning. Assuming that the programmer has nothing to say about the latter, he must either: (a) run the conversation in the program's internal semantic notation, (b) translate between this notation and English by unprincipled short cuts. In the program, the second option was followed so that the conversation would be intelligible to human observers. The program actually analyses an English sentence by picking out key words and trying to convert them into an expression of the kind it is expecting.

I want to comment briefly on Artificial Intelligence as a method in theoretical psychology. There are two main reasons why computer models are used. First, they give quick accurate feedback on the coherence, completeness, and detailed consequences of a theory. Theories of cognitive processes are necessarily complicated, and are likely to be too complicated for the unaided human intelligence to cope with. Second, computer models are an appropriate medium of expression for a theorist who believes that the mind is a computational system. Even theorists who don't write programs have found computational concepts useful (see Neisser, 1967).

We are not yet able to construct formally precise theories of advanced cognitive processes such as language understanding. It is important to be clear about this. The problem is not that we do not know which theory is correct, but rather that we cannot construct any theory at all which explains the basic facts. Under these circumstances, all that can be expected from a formal model is that it should be a step towards the construction of an adequate theory. The present model will have served its purpose if it (a) highlights some problems in the organisation of dialogue which the reader may not have explicitly noticed, (b) explores a clear set of ideas for solving these problems (the most important idea in this case being the 'conversational procedure'), and (c) exposes the limitations of these ideas and therefore helps someone to construct a better theory.

1.2. *Some Problems of Organising a Dialogue*. Conversation comes easily to us because it is a highly practised skill for which the fundamental processes are automatic. In this section I shall examine a fragment of dialogue and spell out some of the processes which lie behind it. I shall then list some problems suggested by this analysis.

Imagine that John and Mary are discussing how to spend the evening:

(1) a. John: Let's go to the cinema.
b. Mary: What's on?
c. John: A cartoon.
d. Mary: All right.

The background to this exchange might be as follows. John and Mary have decided to go out together that evening, and before (1a) each is mulling over possible entertainments. John is considering films, while Mary would like a drink and is wondering which pub to suggest. When John says (1a) Mary notices his suggestion and interrupts her mental pub tour in order to attend to it. She knows that John has made a suggestion about how to spend the evening, that she must evaluate it and respond, and that a positive response commits them both to the intention of going. Meanwhile John is aware of all this too, and may be making contingency plans as he awaits her reply.

Mary's evaluation depends on what kind of film is showing. She considers several ways of finding out: phoning the cinema, looking in the local paper, asking John: and chooses the latter since it is quickest. She says (1b). John notices (1b) and perceives that it is not a reply to (1a), that Mary doesn't know what kind of film is showing, and that she wants to know in order to evaluate his suggestion. He says (1c). Mary perceives this as a reply to (1b) and resumes her evaluation, which turns out positive. Had it been negative she would have said so, and then resumed her tour of pubs until she had a concrete suggestion. She says (1d). After (1d) John and Mary both know that it is their common intention to see the film, and their thoughts move on to matters arising: time, place, means of transport, where to eat beforehand, and so on. Mary's plans about pubs, and the alternatives thought up in case she declined (1a), are all abandoned.

I shall now list some of the problems that this story illustrates. For concreteness I shall use the example of a question like (1b), but similar problems arise with other types of utterance. Suppose S asks H a question Q in order to elicit some information K.

1. How does S know that he needs to know K?
2. How does S know that asking Q in the presence of H might lead to his knowing K?
3. Why does S expect H to reply?
4. How does S know what to do with H's reply?
5 How does S know what range of replies is appropriate?

6. How does S remember that he has asked Q, so that he doesn't say it twice?
7 How does H know that he is meant to reply?
8. How does H know how to work out a reply?
9. How is it that Q can direct H's mind to any of a variety of responses: simply answering, refusing to answer, saying he doesn't know, denying a presupposition of the question, disputing the sincerity of the question, asking for clarification, and so on?
10. When S asks Q, how is H able to break off his train of thought and switch to an unconnected topic?
11. When H has replied, how are S and H able to take up their previous trains of thought?
12. If S's question requires special thought, how is H able to turn his mind to solving the necessary problems, and then remember to reply?
13. If H was about to tell K to S before S asked Q, how does H know that there is no need to tell K to S after he has answered Q? More generally, how do some interruptions cause us to revise our earlier intentions, to return to a different train of thought?
14. If H says he doesn't know the answer to Q, how does S avoid asking Q again when he returns to his previous thoughts, or when the need to know K arises later?
15. If H replies to Q with a counter-question, or in some other unexpected way, how does S know that H is not yet answering him?
16. And if S answers H's question, and then H answers his, how does S know that H is answering Q, and not confuse it with the intervening dialogue? In general, how is it that to some extent we can conduct nested conversations?

1.3. *Outline of How These Problems are Solved in the Program.* The program is based on a view of conversational structure which resembles that of Schegloff and Sacks (1973). Schegloff and Sacks note that conversation consists largely of what they call 'adjacency pairs'. An adjacency pair is a pair of adjacent utterances made by different speakers, the second being a response to the first. Examples: greeting–greeting; question–answer; offer–acceptance/refusal. Some types of utterance (e.g. questions, offers) are always the first parts of a pair; some (e.g. answers, refusals) always second parts; and some (e.g. greetings) can be either. There are restrictions on the range of second-part utterances which are acceptable replies to a given first-part utterance. The second part usually has two functions: (a) to show that the first part has been understood; (b) to give the specifically required response, for example by choosing one of the range of alternatives defined by the first part. In a few cases the second function is absent: for instance, in reply to a statement the remark 'I see' may just indicate comprehension.

Although the concept 'adjacency pair' is useful, the phrase itself seems inappropriate. The important feature of the Schegloff and Sacks pairs is not their adjacency, but the fact that the second remark is a response to the first. Thus (1a) and (1d) clearly constitute a pair in the intended sense in spite of the two intervening utterances. Nevertheless I shall continue to use the phrase 'adjacency pair' to avoid creating an extra term.

There are strong grounds for considering the adjacency pair to be a more important unit of conversation than the utterance. For example, the natural way to split Diagram 2 into functional parts is to make the main break at the full stop after *Wembley*, although S2's remark is thereby divided.

(2) S1: Where does that bus go?
S2: Wembley. Shall we catch it?
S1: Okay.

We find this natural because the point of a question or an answer is to inform the questioner, and it is only through the whole adjacency pair — or to be exact, through the adjacency pair in conjunction with its associated mental processes — that this purpose is achieved.

The program implements these ideas in the following way. For each type of adjacency pair that they use, the robots have a list of instructions which lays down how each utterance in the pair should be produced and interpreted. This list of instructions is called a *conversational procedure*. Conversational procedures have the special feature that some instructions are assigned to one robot and some to the other. The two roles are called *first speaker* and *second speaker*. Diagram 3 gives an English translation of the conversational procedure which controls question-and-answer exchanges. The first line means that the procedure is named ASK and that it has three input variables: S1 (first speaker), S2 (second speaker), and Q (the state of affairs that S1 wants to ask about). ASK has no output variable; its job is to update S1's world model.

(3) CONVERSATIONAL PROCEDURE ASK (S1, S2, Q)

1. S1 composes a sentence U which expresses Q as a question, and utters it.
2. S2 reads U and obtains a value for Q. He records that S1 cannot see the object mentioned in Q, and then inspects his world model to see if Q is true. If he finds no information there he says I DON'T KNOW, otherwise YES or NO as appropriate.

3. S1 reads S2's reply. If it is YES or NO he updates his world model appropriately. If it is I DON'T KNOW he records that S2 cannot see the object mentioned in Q.

When John asks Mary a question, each of them runs a copy of ASK, John taking the role S1 and Mary the role S2. The next problem is: how are adjacency pairs initiated? Or in terms of the program: how are appropriate conversational procedures called in the minds of both robots?

The answer depends on whether the robot is first speaker or second speaker. S1 initiates a conversational procedure in order to achieve some purpose connected with his goals; therefore in S1's mind the conversational procedure is called by his planning procedures, the procedures which try to achieve his main goal. S2, on the other hand, starts a conversational procedure on receiving the opening remark, or some preparatory remark, from S1. S2's copy of the conversational procedure is not called by his planning procedures.

There is thus a distinction in the program between two kinds of procedure: those which control short sections of conversation, and those which control the underlying planning process. Conversational procedures are run jointly, planning procedures privately. The program is written so that it is possible for the robots to interrupt the currently running procedure and resume it later, because in a conversation one must be prepared to break off a train of thought and respond to the partner.

To conclude this section I shall say briefly how the 16 problems listed in Section 1.2 are solved. To save space I shall not repeat the problems; the reader must therefore refer back.

1. S's planning procedures have searched his world model for K but found no information. (K will probably concern whether a goal has been achieved.)
2. S's planning procedures call the conversational procedure ASK if (a) S doesn't know K (b) S has no record stating that H doesn't know K.

3-9. All this knowledge is contained in the definition of ASK, or is computed while ASK runs.

10. H calls ASK on top of whatever procedure he was previously executing.
11. When ASK terminates, control returns to whatever procedure lies beneath it in each robot's control stack.
12. H can call another procedure on top of ASK, control returning to ASK when this procedure terminates.
13. It is possible for the procedure on top of the control stack to modify the control stack itself, e.g. by aborting the procedure beneath it.

14. S has a model of what H can see (and therefore of what H knows about) which is updated during a run of ASK. This model is consulted whenever S decides whether to call ASK.
15. Conversational procedures check that utterances made by the other speaker belong to the limited set of remarks which can be made at that point.
16. One conversational procedure can call another.

2. *What the Program Does*

2.1. *The Setting*. The setting is a simple definite world of four objects, three actions, and three laws of nature, in which several goals and plans are possible. The objects are John, Mary, a door, and a bolt. The state of the world is defined by the positions of these objects. The robots can be IN or OUT, depending on which side of the door they are; the door can be OPEN or SHUT; the bolt UP or DOWN. The bolt is on the inside and can fix the door in either the OPEN or SHUT positions. There is room either side of the door for both robots to be IN, or both OUT.

For each type of object there is a corresponding action: MOVE for robots, PUSH for doors, and SLIDE for bolts. An action is an ATTEMPT to change the position of an object, and is said to have occurred whether or not this position actually changes. Unfortunately the words MOVE and SLIDE wrongly imply that the world changes, but I could find no concise alternatives. I shall therefore always use capital letters for actions and for object positions, so that words like MOVE and IN are not confused with their normal senses.

The consequences of the three actions are defined by the following laws of nature:

(a) If a robot MOVES, he changes position provided that the door is OPEN.

(b) If a robot PUSHES the door, the door changes position provided that the bolt is UP.

(c) If a robot SLIDES the bolt, the bolt changes position provided that the robot is IN.

Note that all actions are their own inverses. If John MOVES when the door is OPEN, then if he was previously OUT he becomes IN, while if he was previously IN he becomes OUT. If a robot who is OUT SLIDES the bolt, we imagine him trying to manipulate a non-existent object.

The objects, actions, laws of nature, and minds of the robots are all represented by data structures or procedures. The minds of the robots are derived from a single section of code which is compiled twice, each

variable being prefixed by J the first time and M the second. Several special J and M variables are then initialised differently to individuate the robots. To take a simple case, JMYNAME and MYOURNAME both receive the value JOHN, JYOURNAME and MMYNAME the value MARY. The operator can also give the robots different beliefs, goals, and capacities. For example he might make John unable to see the door, or Mary unable to SLIDE the bolt. Though artificial, this increases the scope of the conversation, for instance by motivating such questions as 'Is the door open?' and also brings about an objective need to co-operate based on the robots' different capabilities of perception and action.

Apart from the robots' minds, there are three small sections: (a) the functions which print out what is happening; (b) the chairman, the function which arranges the timesharing between John and Mary; and (c) the data structures which represent the state of the world. The latter section contains the current object positions, held in a variable WOBJECTS, and the last utterance made by a robot, WMESSAGE. If the last robot with an opportunity to speak said nothing, WMESSAGE is set to NIL.

Some interactions between sections of the program are given schematically below. JVAR, MVAR, and WVAR represent variables in John's mind, Mary's mind, and the world, and the symbol '→' means 'brings about some change in'.

JVAR → WMESSAGE	John speaks
WMESSAGE → MVAR	Mary hears what he said
JVAR → WOBJECTS	John changes the world by an action
WOBJECTS → JVAR	John perceives the world
JVAR1 → JVAR2	John thinks
JVAR → MVAR	Telepathy, assumed impossible
WVAR1 → WVAR2	A change in the world not due to the robots; this never happens.

The program is written in POP-2, in the version described by Burstall, Popplestone and Collins (1972). It prints out the conversation at the rate of an utterance every two or three seconds.

2.2. *Example of Output*. This section gives a complete transcription of a conversation between John and Mary. To understand the transcript it is important to know the initial settings of (a) the world, (b) the robots' capabilities, (c) the robots' beliefs, and (d) the robots' goals. These were as follows:

Positions of objects: JOHN OUT, MARY IN, DOOR SHUT, BOLT UP.

Perception: John can see all four objects; Mary is blind and cannot see any of them.

Action: John can MOVE and SLIDE, but not PUSH; Mary can do all three.

Beliefs: John believes that:

If a robot MOVES, nothing happens.

If a robot PUSHES the door, the door changes position.

If a robot SLIDES the bolt, nothing happens.

Mary believes that:

If a robot MOVES, he changes position provided the door is OPEN.

If a robot PUSHES the door, the door changes position.

If a robot SLIDES the bolt, nothing happens.

Main goals: John's goal is JOHN IN; so is Mary's. (To simplify matters, the bolt plays no part in the example.)

What follows is an exact copy of the program's output, except for the sections in small type, which are annotations, and the comments in round brackets, which specify the conversational procedure instructions which produced each utterance, and may help the reader later. When referring to the transcript later I will call the utterances U1, U2, U3, etc.

1. JOHN: MARY. (call-by-name)
2. MARY: YES. (acknowledgement)
3. JOHN: I WANT TO SUGGEST A GOAL. (ANNOUNCE 1)
4. MARY: GO AHEAD. (ANNOUNCE 2)
5. JOHN: WILL YOU HELP ME GET IN? (AGREEGOAL 1)
6. MARY: BY ALL MEANS. (AGREEGOAL 2)

The robots have now agreed to co-operate to achieve John's goal. Mary has learned John's goal and John has learned that she is willing to help. Had Mary been given a goal different from John's, she would have refused to help him.

7. JOHN: SHALL WE MAKE A PLAN? (ANNOUNCE 1)
8. MARY: JOHN. (call by name)
9. JOHN: YES. (acknowledgement)
10. MARY: MAY I ASK YOU SOMETHING? (ANNOUNCE 1)
11. JOHN: GO AHEAD. (ANNOUNCE 2)
12. MARY: ARE YOU IN? (ASK 1)
13. JOHN: NO. (ASK 2)

Mary interrupted because she did not know whether the goal was already achieved (she is blind); the robots have very limited powers of inference. Before selecting a plan, a robot always checks that the goal is not already achieved and that a plan is therefore needed. The interruption at 8 causes both robots to forget 7; John was trying to call a conversational procedure at 7, but Mary over-ruled him, and it is she who now starts the conversation to choose a plan.

14. MARY: SHALL WE MAKE A PLAN? (ANNOUNCE 1)
15. JOHN: OKAY. (ANNOUNCE 2)
16. MARY: JOHN. (call-by-name)
17. JOHN: YES. (acknowledgement)
18. MARY: MAY I ASK YOU SOMETHING. (ANNOUNCE 1)
19. JOHN: GO AHEAD. (ANNOUNCE 2)
20. MARY: CAN YOU MOVE? (ASK 1)
21. JOHN: YES. (ASK 2)
22. MARY: JOHN. (call-by-name)
23. JOHN: YES (acknowledgement)
24. MARY: MAY I ASK YOU SOMETHING? (ANNOUNCE 1)
25. JOHN: GO AHEAD. (ANNOUNCE 2)
26. MARY: IS THE DOOR OPEN? (ASK 1)
27. JOHN: NO. (ASK 2)
28. MARY: I SUGGEST THAT WE GET THE DOOR OPEN AND THEN YOU MOVE. (AGREEPLAN 1)

After John agreed to help choose a plan at 15, Mary tried to devise one. To do this she needed to know whether the door was open and whether John could move. Since she is blind she had to ask about the door. At the point we have reached, Mary has learned the following: that John is out and the door is shut, that John's goal is to get in, that John can see the door and himself, and that John can move. John has learned that Mary cannot see him or the door, and that she is willing to help him. Note that Mary's questions are nested in a conversation to make a plan.

29. JOHN: MARY. (call-by-name)
30. MARY: YES. (acknowledgement)
31. JOHN: I WANT TO EXPLAIN SOMETHING. (ANNOUNCE 1)
32. MARY: GO AHEAD. (ANNOUNCE 2)
33. JOHN: IF ONE MOVES, NOTHING HAPPENS. (DISCUSS 1)
34. MARY: I DISAGREE. IF ONE MOVES WHEN THE DOOR IS OPEN, ONE CHANGES POSITION. (DISCUSS 2, 3)
35. JOHN: I SEE. (DISCUSS 4)
36. MARY: I SUGGEST THAT WE GET THE DOOR OPEN AND THEN YOU MOVE. (AGREEPLAN 1)

37. JOHN: ALL RIGHT. (AGREEPLAN 2)

At 29 John has decided that Mary's plan will not work, and he interrupts to explain why, giving the rule which he used in judging the plan. Mary counters by giving the rule on which she based the plan. This exchange allows the robots' beliefs to be brought into line if they differ. A proper discussion being beyond them, they simply compare beliefs, the more complicated being judged the winner. If the two beliefs are equally complicated, no change occurs except that the robots add to their knowledge of each other. In this case Mary's belief is accepted because it contains an extra proviso. She repeats her plan at 36 because the program is written so that if a plan is disputed, the suggestion is cancelled, and must either be changed or reasserted. Having been converted to Mary's belief, John this time accepts her plan.

38. MARY: SHALL WE MAKE A PLAN? (ANNOUNCE 1)
39. JOHN: OKAY. (ANNOUNCE 2)
40. MARY: I SUGGEST THAT I PUSH THE DOOR. (AGREEPLAN 1)
41. JOHN: ALL RIGHT. (AGREEPLAN 2)

Mary now pushes the door, and the state of the world changes to JOHN OUT, MARY IN, DOOR OPEN, BOLT UP. Since they were already in agreement as to the consequences of pushing doors there was no discussion of the plan.

42. MARY: I WANT TO TELL YOU SOMETHING. (ANNOUNCE 1)
43. JOHN: GO AHEAD. (ANNOUNCE 2)
44. MARY: I HAVE PUSHED THE DOOR. (TELL 1)
45. JOHN: I SEE. (TELL 2)
46. MARY: LET'S ASSESS THE RESULT OF MY ACTION. (ANNOUNCE 1)
47. JOHN: OKAY. (ANNOUNCE 2)
48. MARY: NOTHING HAS HAPPENED. (ASSESS 1)

Since Mary is blind she does not observe a change in the position of the door and assumes it is still where it was. (This is rather foolish of her but it does provide a motive for some further exchanges.) As with a plan, if an assessment is challenged it must be changed or reasserted. Thus 54/2 is a revision of 48.

49. JOHN: MARY. (call-by-name)
50. MARY: YES. (acknowledgement)
51. JOHN: I WANT TO TELL YOU SOMETHING. (ANNOUNCE 1)
52. MARY: GO AHEAD (ANNOUNCE 2)

53. JOHN: THE DOOR IS OPEN. (TELL 1)
54. MARY: I SEE. THE DOOR HAS CHANCED POSITION (TELL 2, ASSESS 1)
55. JOHN: YES. (ASSESS 2)
56. MARY: THE DOOR IS NOW OPEN. (ASSESS 3)
57. JOHN: RIGHT. (ASSESS 4)

There are two parts to assessing an action: saying which object moved, if any, and saying whether the action achieved its goal. The first part is done by 48 and revised by 54/2, the second is done by 56. After saying RIGHT at 57 John moved, and the state of the world has now changed to JOHN IN, MARY IN, DOOR OPEN, BOLT UP.

57. JOHN: I WANT TO TELL YOU SOMETHING. (ANNOUNCE 1)
58. MARY: GO AHEAD. (ANNOUNCE 2)
59. JOHN: I HAVE MOVED. (TELL 1)
60. MARY: I SEE. (TELL 2)
61. JOHN: LET'S ASSESS THE RESULT OF MY ACTION. (ANNOUNCE 1)
62. MARY: OKAY. (ANNOUNCE 2)
63. JOHN: I HAVE CHANGED POSITION. (ASSESS 1)
64. MARY: JOHN. (call-by-name)
65. JOHN: YES. (acknowledgement)
66. MARY: I WANT TO TELL YOU SOMETHING. (ANNOUNCE 1)
67. JOHN: GO AHEAD. (ANNOUNCE 2)
68. MARY: YOU ARE OUT. (TELL 1)
69. JOHN: I DISAGREE. I HAVE CHANGED POSITION. (TELL 2, ASSESS 1)
70. MARY: YES. (ASSESS 2)
71. JOHN: I AM NOW IN. (ASSESS 3)
72. MARY: RIGHT. (ASSESS 4)

Being unable to update her world model by perception, Mary challenges (64–68) John's claim that he has changed position (63) but she is belatedly persuaded (69–72). Having decided that the action has achieved its aim, the robots now go into a halt state since the main goal is achieved and they have no motive to do or say anything further.

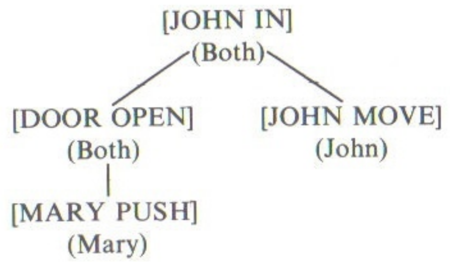
3. *How the Program Works*

The remainder of Part I describes how the program works. To do this properly it will be necessary to give details of several procedures, but I shall use English for this purpose rather than the formalism in which the program was actually written.

The program contains several kinds of procedures. The most abstract of these is the *planning tree*. This is constructed and executed by *planning procedures*. The planning procedures sometimes call *conversational procedures* in order to perform subtasks which require dialogue. Finally, there are *executive procedures*, which interpret planning procedures and conversational procedures.

3.1. *Planning*. The conversation transcribed in section 2.2 is based on a single planning tree. Diagram 4 reproduces this tree.

(4)



Each node of the tree is a goal (in square brackets) with a label indicating who is responsible for achieving the goal (round brackets). The links between nodes represent goal–subgoal relationships: for instance, [MARY PUSH] is a subgoal of [DOOR OPEN]. The node at the top of the tree is the main goal; the terminal nodes are all acts which the robots can perform directly. Each non-terminal node has a goal which describes a state of affairs, represented by a symbol referring to an object (e.g. DOOR) followed by a symbol referring to a possible position of the DOOR) followed by a symbol referring to a possible position of the object (e.g. OPEN). The acts at the terminal nodes are each represented by a symbol referring to a robot followed by a symbol referring to one of the three possible actions (MOVE, PUSH, or SLIDE).

The robots are capable of planning separately as well as co-operatively; they only co-operate when one of them has a goal which he fails to achieve on his own. The same planning procedures are used for co-operative planning as for separate planning. The only difference is that during co-operative planning some goals on the planning tree are

labelled as joint responsibilities; the planning procedures are at several points sensitive to this labelling. The usual result is that they employ conversational procedures at these points in order to perform subtasks which would otherwise have been performed by planning procedures. For example, when it is time to select a plan to achieve a goal labelled 'Both', the plan is chosen by a conversational procedure AGREEPLAN which ensures that both robots accept the plan and add it to their copies of the planning tree.

To simplify the exposition I shall now describe the planning process as it would be conducted by one robot working on his own. The complications which arise in co-operative planning are introduced afterwards.

At any stage in the life of a planning tree one of the goals will be the *current goal*, the one the system is currently working on. If achieved or abandoned goals are removed from the tree, the current goal will always be the bottom leftmost one; on Diagram 4 it is [MARY PUSH]. Assuming that the tree has got started (i.e. that it already contains the main goal at least), the planning procedure for developing it further can be stated in terms of operations on the current goal. This procedure is named ACHIEVEGOAL; it is called when the main goal has already been selected, and its job is to achieve it. ACHIEVEGOAL is defined in Diagram 5. The first line of the definition means that ACHIEVEGOAL takes an input variable M (the main goal) and delivers an output R (a report on whether M has been achieved). The variables M, R, G, P are all local variables of the procedure.

(5) PLANNING PROCEDURE ACHIEVEGOAL(M) → R
1. Make M the main goal of the planning tree.
2. Identify the current goal. Call it G.
3. If G is an act which can be performed directly, perform it; then remove G from the tree and return to instruction 2.
4. If on the other hand G refers to a state of affairs, test whether the state of affairs has been achieved.
5. If so, remove G from the tree; and if G is the main goal M, exit and report success.
6. If not, select a plan to achieve G. Call it P.
7. If unable to find a satisfactory plan, remove the plan of which G is a part (i.e. G and its sisters) from the tree, marking this plan as unsatisfactory; and if G is the main goal M, exit and report failure.
8. Attach G to the tree below G, and return to instruction 2.

Apart from straightforward operations on the planning tree (e.g. removing goals, affixing plans) there are three subtasks in this procedure:

performing a direct act, testing whether a state of affairs has been achieved, and selecting a plan. The first of these is straightforward but the others need further comment.

A robot tests whether a state of affairs has been achieved by consulting his model of the world. This model lists the positions of the four objects, the special symbol UNKNOWN being used to represent ignorance. Initially every object position is marked UNKNOWN, but the robots look at the world occasionally and record the positions of those objects they can see. In the setting of the transcript Mary is blind, but she is still able to update her world model by asking John questions. The robots are able to ask and answer questions even when they are planning separately.

To select a plan to achieve a state of affairs, a robot consults his theory of how the world works; that is, his beliefs concerning the consequences of the three actions. These beliefs are preset by the operator, but they can alter during a run of the program either (a) through conversation, or (b) because experience proves them false. Alteration (b) is needed to prevent unsuccessful plans being repeated, and is accomplished by a trick which I prefer to pass over. An example of alteration (a) is the sequence U33–U35 in the transcript. The robots compare beliefs, and the more complex is judged the winner. This part of the program is blatantly unprincipled and was included solely in order to vary the types of conversational exchange. The only feature which can be taken seriously is that the robots, like people, respond to a disagreement over the value of a plan by discussing the beliefs on which the plan was based.

Each robot has three beliefs about the consequences of actions These take the following form: act X will shift object Y provided condition Z holds. For instance, ROBOT MOVE will shift ROBOT provided condition DOOR OPEN holds. Either Y or Z can be null. If Y is null the implication is that X achieves no result under any circumstances. If Z is null the implication is that X always shifts Y. A belief in which Y or Z is null is regarded as less complex than one in which Y and Z have definite values, and in a confrontation the former will yield to the latter. This is because in the robots' world, a belief in which Y or Z is null is always false. The operator endows a robot with false beliefs if he wants to provoke either the failure of a plan or a disagreement between the robots.

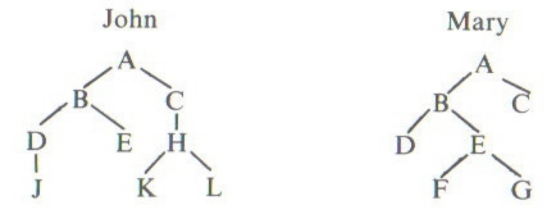
The procedure which selects a plan to achieve the current goal is given in Diagram 6. G is the goal for which a plan is sought, and P is either a plan or a failure report.

(6) PLANNING PROCEDURE FINDPLAN(G) → P

1. Identify the type of object in G. Call it T. (T will be ROBOT, DOOR, or BOLT.)
2. Find a belief in which T is the Y entry (i.e. the object to be shifted). Call it B. If there isn't one, exit and report failure.
3. Identify the action in part X of B, and check that you can perform it. If not, exit and report failure.
4. Wherever the symbol ROBOT occurs in B, substitute your own name.
5. Test whether condition Z of B is achieved. (If Z is null it is regarded as achieved whatever the situation.)
6. If Z is achieved, P should consist of act X only; if not, P should consist of state of affairs Z followed by act X.

3.2. *Co-operative Planning*. When two parties co-operate on a task, they construct and execute a single planning tree. Each party must have a copy of the tree in his mind, and although these copies need not correspond exactly, they must be consistent with one another. Diagram 7 shows two different but consistent trees. Goals A and B are joint responsibilities, but C and D are John's responsibility and E is Mary's. John performs actions J, K, L, and Mary performs F and G.

(7)



To keep their copies of the planning tree consistent, the parties must know at every stage who is responsible for each subgoal. If a goal is a joint responsibility, the plans affixed to it on the two copies of the tree must agree exactly, and there must also be agreement on who is responsible for each subgoal of the plan. As soon as a subgoal is assigned to one of the parties, the other party need not elaborate this subgoal further on his copy of the planning tree.

If a goal is a joint responsibility, there are two ways of ensuring that the plans attached to it by each party correspond. First, a plan may be so obvious that each party can be sure that the other will select it. If this is not so, then the second method is used: a plan is agreed, or announced by someone in authority. The program uses the second method only (although the setting is one in which people would probably use the first). Responsibility for the subgoals of a plan is not announced explicitly, but is decided by the following rule: a state of affairs goal is always a joint responsibility, while a direct action goal is the responsibility of the robot designated to perform the action. A consequence of this rule is that the robots' copies of the planning tree are always identical as well as consistent, because subgoals which have been assigned to a particular individual are never elaborated.

A special problem which arises in the case of the program (and also occasionally in real life) is that an individual who is not responsible for a subgoal may have to be told when it has been achieved. If Mary has been assigned the subgoal [MARY PUSH] in a setting in which the BOLT is DOWN, her action will have no effect on the world, and John won't know that this subgoal has been achieved and that it should therefore be removed from the tree. A robot who has just performed an action therefore informs his partner.

The planning procedures described in Section 3.1 are augmented in order to allow co-operative planning. Instructions 3 and 6 of ACHIEVEGOAL (Diagram 5) and 3 and 4 of FINDPLAN (Diagram 6) are revised as shown in Diagrams 8 and 9.

(8) Revision of instructions 3 and 6 of ACHIEVEGOAL (Diagram 5)

3. If G is an act to be performed directly, then:
   (a) if G is assigned to your partner, wait until he tells you he has performed it;
   (b) if G is assigned to you, perform it. Then, if the goal above G on the planning tree is a joint responsibility, inform your partner that you have performed G and initiate a conversation to assess its result.

   Finally, remove G from the tree and return to instruction 2.
6. If G is your responsibility, select a plan to achieve it; but if G is a joint responsibility, initiate a conversation to agree a plan to achieve it. Call the plan P.

(9) Revision of instructions 3 and 4 of FINDPLAN (Diagram 6)

3. Identify the action in part X of B.
   If G is a joint responsibility, find the name of a robot who can do X, and make it the value of R; if neither robot can do X, exit and report failure. If G is your responsibility, check that you can do X, and if so make your name the value of R; if you can't do X, exit and report failure.
4. Wherever the symbol ROBOT occurs in B, substitute R (i.e. the name of the robot who is to perform the action).

The revised versions of ACHIEVEGOAL and FINDPLAN are, of course, the versions used in the program. Note that FINDPLAN is called by instruction 6 of ACHIEVEGOAL. This can come about in two ways. If the current goal is a joint responsibility, instruction 6 of ACHIEVEGOAL calls a conversational procedure AGREEPLAN, which then calls FINDPLAN. FINDPLAN is executed by the first speaker of AGREEPLAN, whose job is to suggest a plan. The second speaker then evaluates the plan, and if he accepts it the plan is returned by AGREEPLAN to ACHIEVEGOAL. This happens in the minds of both robots, so that the values they give to P in instruction 6 of ACHIEVEGOAL, and therefore the plans they affix to their copies of the planning tree, correspond. If on the other hand the current goal is the responsibility of a particular robot, then instruction 6 of ACHIEVEGOAL calls FINDPLAN directly, without the mediation of AGREEPLAN.

3.3. *How Planning Procedures Call Conversational Procedures.* We have already noted that at various points during the execution of a planning procedure a robot must arrange for the planning procedure to call a conversational procedure. This naturally raises the complication that the conversational procedure must be called in both minds. The purpose of this section is to explain how this happens. I shall use the first six utterances of the transcript as an example.

To follow the beginning of the transcript it is necessary to know about another planning procedure, CHOOSEGOAL. This procedure selects a main goal and calls ACHIEVEGOAL in order to achieve it. It may call ACHIEVEGOAL twice, labelling the main goal an individual responsibility the first time and a joint responsibility the second. But before the main goal can be labelled a joint responsibility, the co-operation of the partner must be secured by means of the conversational procedure AGREEGOAL. CHOOSEGOAL is not called by another planning procedure; it is the fundamental planning procedure and is called by the executive procedure when the robot is first roused by the chairman.

(10) PLANNING PROCEDURE CHOOSEGOAL

1. Select a main goal. call it M.
2. Try to achieve M on your own. That is, label M as your responsibility and call ACHIEVEGOAL. ACHIEVEGOAL will return a report R.
3. If R is SUCCEED, go to instruction 5. If R is FAIL, try to secure the co-operation of the other robot, using AGREEGOAL. Call his answer A.

4. If A is YES, label M as a joint responsibility and call ACHIEVEGOAL. If A is NO, go to instruction 5.
5. Do nothing further; that is, return control straight back to the chairman.

Let us now follow the transcript. When the operator has put the world and the minds of the robots into their initial states, he calls the chairman program, which in turn calls the John program. When the chairman rouses a robot, control passes to an executive procedure which has several functions: it checks whether the other robot has just said something; it looks now and then at the world to update the world model; and it interprets planning procedures and conversational procedures. The interpretation of planning procedures and conversational procedures is accomplished with the aid of a control stack, which records (a) which procedures are currently running, (b) their local variable values, and (c) the calling relationships between them.

Finding the control stack empty, John's executive takes an initial look at the world and then calls the fundamental planning procedure CHOOSEGOAL. It does this by putting a record onto the control stack stating that CHOOSEGOAL is running, the current instruction is 1, and the local variables are currently undefined. It then executes instruction 1 of CHOOSEGOAL, this being the current instruction of the procedure on top of the control stack. The main goal [JOHN IN] is selected and becomes the value of local variable M. Then the current instruction is advanced to 2. Execution of this instruction causes ACHIEVEGOAL to be called, so another record is put onto the control stack stating that ACHIEVEGOAL is running, its input variable M having the value [JOHN IN (JOHN)]. (This means that the goal is [JOHN IN] and that the robot responsible for the goal is John.) We need not pursue further the adventures of this call of ACHIEVEGOAL; suffice to say that it returns FAIL. There is now just one procedure running, CHOOSEGOAL, the current instruction is 3, and local variable R has received the value FAIL.

We now encounter the problem mentioned at the beginning of this section. Execution of instruction 3 must cause the conversational procedure AGREEGOAL to be called, and the call must be recorded on Mary's control stack as well as John's. The most realistic solution is probably for John to proceed with his version of AGREEGOAL, taking the role of first speaker and producing the opening remark. Mary would then have to infer from this remark that AGREEGOAL had been called and that her role was second speaker. But this method raises complicated problems of inference which are beyond the scope of the model. These problems are therefore evaded by the method of announcing conversational procedures explicitly; hence the repeated occurrence of such remarks as *I want to suggest a goal* (U3), *Shall we make a plan?* (U7), and *May I ask you something?* (U10). These remarks, and responses to them, are produced by a conversational procedure named ANNOUNCE, which is called by a special device controlled by the robots' executive procedures. The robot wishing to call ANNOUNCE utters his partner's name, and the partner acknowledges. This exchange causes a call of ANNOUNCE to be recorded in both minds, the robot who uttered his partner's name taking the role of first speaker. I shall refer to this device as 'calling-by-name'. To avoid the continual use of calling-by-name it is assumed that if no conversational procedure is currently running, any remark other than a call-by-name or its acknowledgement must be the opening remark of ANNOUNCE. This rule only applies when the robots have agreed to co-operate on a goal, so that a call-by-name is necessary at U1. After U6, calling-by-name only occurs when a conversational procedure is already running and a robot wishes to call another conversational procedure from inside it.

Let us resume our account of the transcript. John has reached instruction 3 of CHOOSEGOAL and his executive must arrange the call of AGREEGOAL. To do this it must call ANNOUNCE, with John as first speaker, and to call ANNOUNCE it must say Mary's name. Hence U1. Control now returns to the chairman, and thence to Mary.

Since this is the first time she has been roused, Mary's executive first takes a look at the world (without result, since she is blind), and then calls CHOOSEGOAL. But before executing instruction 1 of CHOOSEGOAL it checks whether an utterance has been made. Finding a call-by-name, it immediately records a call of ANNOUNCE on Mary's control stack, making Mary second speaker, and then produces the acknowledgement U2. Control returns to the chairman and John is roused again.

Finding the call-by-name acknowledged, John's executive records the call of ANNOUNCE on his control stack, making John first speaker. Execution of ANNOUNCE now commences. Diagram 11 specifies this procedure. The three input variables are S1 (first speaker), S2 (second speaker), and C (the conversational procedure to be called). In this case, S1 is JOHN, S2 is MARY, and C is AGREEGOAL. ANNOUNCE has no output variable; its job is to call C, making S1 the first speaker of C and S2 the second speaker.

(11) CONVERSATIONAL PROCEDURE ANNOUNCE(S1,S2,C)
1. S1 composes a sentence U to announce C, and utters it.

2. S2 decodes U, obtaining a value for C. He then checks that he is ready to call C. (Some conversational procedures can only be called when a particular instruction in a planning procedure has either been reached or passed.) If not, he suspends ANNOUNCE and continues execution of the procedure he was running previously until he is ready to call C. Then he calls C in place of ANNOUNCE, taking the role of second speaker, and utters a phrase indicating readiness, such as GO AHEAD or OKAY.
3. S1 reads S2's response, checks that S2 is ready, and then calls C in place of ANNOUNCE, taking the role of first speaker.

Finding that instruction 1 is assigned to S1 and that he is S1, John executes it. The encoding (and decoding) of English sentences is done by specialist procedures which are of no interest. Having produced U3, John returns control. Mary now starts to execute her version of ANNOUNCE. Since she is S2 she passes over instruction 1 and proceeds to 2. She decodes U3 and obtains the value AGREEGOAL for C. Had John left no utterance, she would have returned control, remaining on instruction 2.

Continuing the execution of instruction 2, Mary's executive checks whether she is ready to call AGREEGOAL. the condition for calling AGREEGOAL is that the robot has passed instruction 1 of planning procedure CHOOSEGOAL. (The reason is that this instruction selects a robot's main goal; until this has been done the robot cannot say whether his goal agrees with his partner's.) Mary therefore holds ANNOUNCE in cold storage and executes instruction 1 of CHOOSEGOAL, selecting [JOHN IN] as her main goal. (Main goals are actually predetermined by the operator; how people select main goals is not the business of the model.) ANNOUNCE is then reinstated, and she completes the execution of instruction 2 by calling AGREEGOAL in place of ANNOUNCE (taking the role of second speaker) and producing U4. Control passes to John; John executes instruction 3 of ANNOUNCE; and the result is that AGREEGOAL is finally called in both minds.

John now executes instruction 1 of AGREEGOAL, producing U5, and Mary executes instruction 2, producing U6. Details of AGREEGOAL, and of some other conversational procedures, are given in the next section.

3.4. *Conversational Procedures.* There are seven conversational procedures; Diagram 12 lists their names and purposes. At least one example of each conversational procedure occurs in the transcript (Section 2.2).

(12) Name	Purpose	Example
ANNOUNCE	To explicitly announce the call of another conversational procedure.	U10–U11
AGREEGOAL	To secure the co-operation of the partner in achieving a goal	U5–U6
AGREEPLAN	To agree a plan to achieve a goal which is a joint responsibility.	U40–U41
ASK	To obtain information	U12–U13
TELL	To give information.	U44–U45
DISCUSS	To compare beliefs about the consequences of an action.	U33–U35
ASSESS	To assess the result of an action which has just been performed.	U54–U57

There is no need to specify all these procedures in this section. DISCUSS and ASSESS are artificial and best disregarded; ASK and ANNOUNCE have already been defined (Diagrams 3 and 11); and TELL is straightforward. I shall therefore define AGREEGOAL and AGREEPLAN only (Diagrams 13 and 14).

(13) CONVERSATIONAL PROCEDURE AGREEGOAL (S1,S2,G) → A

1. S1 composes a sentence U asking for help with goal G, and utters it.
2. S2 decodes U, obtaining a value for G, and compares G with his main goal. If they are the same he gives A the value YES, if not, NO. If A is YES he carries out some surgery on his control stack, giving variable A of CHOOSEGOAL the value YES and setting the current instruction of CHOOSEGOAL to 4. Finally S2 utters A and exits from AGREEGOAL.
3. S1 reads A and exits, returning A to the procedure which called AGREEGOAL.

(14) CONVERSATIONAL PROCEDURE AGREEPLAN (S1,S2,G) → P

1. S1 selects a plan to achieve G, using FINDPLAN, and calls the plan P. He translates P into a sentence U, and utters U.
2. S2 decodes U, obtaining a value for P, and evaluates P against his own world model and theory of the consequences of actions. If he disagrees with P he says why, using a new conversational procedure (either DISCUSS or TELL); in this event AGREEPLAN recom-

mences with P undefined. If on the other hand S2 agrees, he says so, and exits from AGREEPLAN returning P to ACHIEVEGOAL.

3. S1 reads S2's reply. If P is rejected AGREEPLAN returns a failure message; otherwise, P.

(15)

John	Mary
	ASK 2 S1 = MARY S2 = JOHN Q = [DOOR OPEN] A = undefined
ASK 1 S1 = MARY S2 = JOHN Q = undefined A = undefined	FINDPLAN 5 G = [JOHN IN] P = undefined T = ROBOT B = MOVE shifts JOHN iff [DOOR OPEN]
AGREEPLAN 1 S1 = MARY S2 = JOHN G = [JOHN IN] P = undefined	AGREEPLAN 1 S1 = MARY S2 = JOHN G = [JOHN IN] P = undefined
ACHIEVEGOAL 6 M = [JOHN IN (BOTH) = G = [JOHN IN (BOTH)] P = undefined R = undefined	ACHIEVEGOAL 6 M = [JOHN IN (BOTH)] G = [JOHN IN (BOTH)] P = undefined R = undefined
CHOOSEGOAL 4 M = [JOHN IN (BOTH)] R = FAIL A = YES	CHOOSEGOAL 4 M = [JOHN IN (BOTH)] R = undefined A = YES

I conclude this section by giving the control stacks of both robots after utterance U26, in order to illustrate how they keep track of complicated situations in the dialogue. Each box on Diagram 15 records a procedure call, by specifying the name of the procedure, the current instruction, and the current local variable values. Note the detailed correspondence between the two control stacks. The only substantial difference between them is that Mary has a more precise record of where her question came from.

Part II. Some Proposals for an Improved Model

4. *Criticism of the Method Used in the Program*

The only respect in which the program is an advance on previous language-using programs is that it is able to represent the point of an utterance. Its central idea is that dialogue is produced by *conversational procedures*, which control short conversational exchanges, and that these procedures are called by *planning procedures*, which select and achieve goals. The robots are able to keep track of the dialogue because they have *control stacks*, which are records of the planning procedures and conversational procedures currently running. The control stack is therefore the means by which the program represents the point of an utterance. A person familiar with the program could determine the point of utterance U26 (for example) by examining the state of Mary's control stack (Diagram 15); he would not need to consult the previous dialogue.

I shall confine my criticism of the program to a single issue: How well does it represent the point of an utterance? I shall ignore several other criticisms (e.g. the planning system is primitive; the world is oversimple; the method of explicitly announcing conversational procedures is artificial; the English utterances are not properly analysed) because I regard them as peripheral; it would be possible to improve the program in these respects while retaining the central idea of the conversational procedure. This section, therefore, discusses how well a control stack of planning procedures and conversational procedures represents the point of an utterance; and the remaining sections suggest improvements.

I want to begin by introducing a new notion: namely, the notion of the state of the dialogue, or *dialogue state*. The dialogue state is the situation which has arisen between two speakers as a result of their previous dialogue. (It might be compared with the board position which has arisen between two chess players as a result of their previous moves. This analogy is improved if the players are playing blindfold chess, in which the board position exists only in their minds.) A participant, or observer, may be said to have 'followed a dialogue' if he has kept his copy of the dialogue state correctly up to date, and to have 'got the point of an utterance' if he has updated his copy of the dialogue state in the way intended by the speaker.

In the program, a robot represents the dialogue state by his control stack in conjunction with his copy of the planning tree. I want to make two criticisms of the control stack as a representation of the dialogue state: (1) it is incomplete; (2) it is insufficiently explicit. To bring out these weaknesses, let us imagine that Mary were able to put her control stack of Diagram 15 into words. The resulting description might be as follows:

(16) English interpretation of Mary's control stack after utterance U26, in which Mary says: 'Is the door open? John and I are co-operating to achieve the goal [JOHN IN]. This goal is not yet achieved and we are trying to agree a plan. I am to propose a plan and John is to evaluate it. I am therefore trying to think of one. I have found an action, MOVE, which I believe will shift John provided the DOOR is OPEN, and I have verified that he can perform this action. I am now trying to find out whether the condition [DOOR OPEN] is satisfied, so that I can decide whether to include [DOOR OPEN] as a subgoal of the plan. I don't know whether the DOOR is OPEN, and I can't see it, so I have just asked John'.

It seems to me that this is an accurate description of the state of the dialogue after U26, as a human observer would perceive it. Nevertheless, it is insufficiently detailed, because it fails to explain how Mary expects to find out whether the door is open by uttering the words 'Is the door open?'. There is a gap in her understanding here; a gap which a human observer would not have. Note also that the account in Diagram 16 is an INTERPRETATION of Mary's control stack, not a straight description of it. Facts stated explicitly in the account are only implicit in the control stack. Let us now examine each of these faults in more detail.

There are probably a number of ways in which Diagram 16 is incomplete as a representation of the point of U26, but the most glaring omission is the gap between the goal of finding out whether the door is open, and the act of uttering 'Is the door open?'. It is by no means easy to spot a problem here, and the credit for doing so belongs to the philosopher J. L. Austin, who introduced the notion of a 'Speech Act' (1962). The problem of how utterances achieve goals has interested philosophers and linguists ever since. Particularly difficult is the problem of how much of this process depends on socially created facts (conventions) and how much on natural facts. Searle, (1969), Grice (1968), and Strawson (1964) have debated this problem. In Section 5 I review this literature, for the benefit of readers who are not linguists or philosophers, and in Section 6 I suggest a solution based on the idea of the planning tree (see Section 3.1, and also Hewitt's PLANNER [unpublished]).

If the robots do not know how utterances achieve goals, does it matter? do people know? Some arguments suggesting that people do not know how utterances achieve goals are given in Section 6. I shall now try

to explain by an example why I think that this knowledge matters.

Imagine that three people called John, Mary, and Bill are together in a room, and that John wants the fire on. In some circumstances he might attempt to achieve this goal by saying to Mary, 'Will you put the fire on?'. But he would be very unlikely to utter these words if the circumstances were any of the following:

(a) He has just asked Mary to put the fire on, and she has agreed.
(b) Bill has just asked Mary to put the fire on, and she has agreed.
(c) Mary has just announced that she will put the fire on.
(d) Mary is walking towards the fire with a box of matches in her hand.
(e) Mary has just said that she doesn't want the fire on.
(f) Mary, who is a child, is not allowed to put the fire on. John disagrees with this rule but is sure that Mary will obey it.

How does John refrain from asking Mary to put the fire on in these circumstances? Has he a separate rule for each: for example, a rule stating that one should not repeat a request that has already been accepted? This is conceivable, but it is more parsimonious to suppose that he knows the following fact: that a request for X to do A causes X to do A by first causing X to INTEND to do A. In conjunction with two general principles of rational behaviour, this fact explains why the request is inappropriate in all six circumstances just mentioned (and no doubt in numerous others). The general principles are these: (1) don't try to achieve goals which are already achieved; (2) don't try to achieve goals which are known to be unachievable. In cases (a) to (d), the goal that Mary should intend to put the fire on is already achieved; in cases (e) and (f) it is known to be unachievable.

Let us now turn to the second fault of the control stack as a representation of the dialogue state: namely, insufficient explicitness. What this means is that the relations between elements of the dialogue state are not represented systematically. Consider for example variables P of FINDPLAN and G of ACHIEVEGOAL. The relation between these elements should be that P is a candidate for the role of a plan to achieve G; but on the control stack the relation between P and G is merely that P is the output variable of a procedure P1, G is a local variable of a procedure P2, and P2 calls P1. The lack of an explicit notation for roles such as 'candidate plan' is important because it complicates the task of interpreting the dialogue state. The result is that the dialogue state can be interpreted just one way; it cannot be interpreted by several different procedures for several different purposes. The robots therefore cannot respond flexibly to unexpected turns in the conversation; an unexpected remark throws them completely. Section 7 discusses the problem of flexibility further, and sketches a method of representing the dialogue state in which relations such as 'candidate plan' are explicitly marked.

5. *A Brief Introduction to Speech Acts*

Speech Acts is a topic in the post-war philosophy of language which has interested a number of philosophers and linguists, the pioneer being J. L. Austin. Austin arrived at the notion in the following way. He began by examining a type of utterance which he called the *performative*, which he contrasted with the *constative* (1971). Constatives, like (17), are ordinary declaratives which may be true or false; while performatives, like (18), appear at first sight to be necessarily true, though it is more accurate to say that the notions true—false cannot be applied to them. The performative performs an action rather than indicating a fact.

(17) It will rain tomorrow.
(18) I bet you sixpence it will rain tomorrow.

Syntactically, performatives are usually marked by verbs of the first person singular present indicative active: *I promise* ..., *I name this ship* ..., *I advise you* ..., and so on. Although performatives cannot be true or false, they can be 'happy' or 'unhappy' in various ways: a promise may be insincere, for example, or the man saying *I name this ship* ... may be in no position to do so.

Austin later (1962) came to believe that even utterances like (17) have an implicit performative. According to this later view, every utterance is associated with a function which might be made explicit by a function-indicating device such as *I ask* or *I promise*, but in most cases the function is implicit and must be inferred by the perceiver. The intended function of an utterance (promise, warning, order, or whatever) Austin calls its *illocutionary force*. The perception of this force by the hearer is called *illocutionary uptake*.

If a speaker S produces an utterance with a certain illocutionary force (a promise, let us say) and this force is perceived by the hearer H, then S is said to have performed an *illocutionary act*. Many English verbs refer to illocutionary acts: examples are promise, request, warn. Austin tried to characterize the illocutionary act by distinguishing it from two closely related types of act, the *locutionary act* and the *perlocutionary act*. It is important to realise that this distinction is not a means of grouping individual utterances into three exclusive categories. A particular utterance may perform all three types of act simultaneously, just as a particular gesture by the President of the United States might simultaneously depress a red button, obliterate Moscow, and start World War III.

The distinction is roughly as follows. If a speaker S utters a sentence with the intention that the constituent words and phrases should have normal meanings and references, he performs a locutionary act. If S further intends that his utterance should accomplish a purpose, and the hearer H perceives the intended purpose, then S performs an illocutionary act. And if, further, S's utterance actually achieves the intended purpose, then S performs a perlocutionary act.

Suppose for example that S says to H, 'You'll be paid tomorrow', and consider the following five contexts in which this sentence might be uttered:

(a) S and H are both Germans and neither speaks a word of English. The words are quoted from an English phrase book.
(b) S speaks English, but H understands only German.
(c) S is H's boss and is promising to pay him.
(d) S tells H that B, H's boss, will pay up tomorrow, but H doesn't believe him.
(e) S tells H that B will pay up tomorrow, and H does believe him.

Comments:

(i) The same locutionary act is performed in all the contexts (b)–(e). No locutionary act is performed in context (a).
(ii) The same illocutionary act (telling) is performed in (d) and (e); a different illocutionary act (promising) is performed in (c). No illocutionary act is performed in (a) or (b).
(iii) The perlocutionary act of assuring (persuading, convincing) is performed in (e), but not in (d).

Why did Austin draw this threefold distinction? It seems that the distinction resulted from a thorough examination of the words which are used to report utterances. 'I *said* "Go home"' reports a locutionary act; 'I *advised* him to go home' reports an illocutionary act; 'I *persuaded* him to go home' reports a perlocutionary act. Austin believed that a careful and thorough examination of the words we use to describe a phenomenon would illuminate the phenomenon itself. The concept of 'illocution' is probably the most influential of the concepts he reached by this method. It is an important concept because it expresses the difference between genuine conversation and mere recitation. The concept of 'locution' is too weak to express this difference, while 'perlocution' is too strong.

J. R. Searle's book *Speech Acts* (1969) attempts to explicate the notion of illocution directly, by stating the necessary and sufficient conditions for the performance of a number of illocutionary acts. These conditions are of several kinds:

(a) *Propositional content conditions*: the propositional content of the utterance must be suitable.

(b) *Preparatory conditions*: the circumstances must be appropriate.

(c) *Sincerity conditions*: the speaker's intentions and beliefs must be appropriate.

In the case of a question, for example, the content of the utterance must be a proposition or propositional function, a preparatory condition is that S does not know the answer, and a sincerity condition is that S wants to know the answer. If these and all the other conditions for a question are satisfied, then the utterance counts as an attempt to elicit the specified information from the hearer. The rule which states what a speech act 'counts as' is called the *Essential* rule. Here is a summary of the conditions and Essential rules which Searle gives for questions and requests (adapted from p. 66 of *Speech Acts*):

(20) *Question*

Content:	Any proposition or propositional function.
Preparatory:	
	(a) S does not know the answer.
	(b) It is not obvious that H will provide the information without being asked.
Sincerity:	S wants the information.
Essential:	Counts as an attempt to elicit the information from H.

(Note: exam questions are a separate case with different rules.)

(21) *Request*

Content:	Future action A of H.
Preparatory:	(a) H can do A. S believes H can do A.
	(b) It is not obvious that H is going to do A anyway.
Sincerity:	S wants H to do A.
Essential:	Counts as an attempt to get H to do A.

The idea underlying Searle's explication of illocution is that the illocutionary act belongs to a special class of acts (including serving [in tennis], checkmating, selling, marrying) which are constituted by social

conventions. These acts are distinguished from acts like hitting a ball, moving a piece of wood across a chequered surface, and handing an object to another person, which exist independently of social convention. Acts of the former kind depend on *constitutive rules* of the form: X counts as Y in situation Z. Thus, hitting a tennis ball into the opposite court counts as serving, provided that it was your turn to serve, you were standing in the right place, and so on. And similarly, uttering a sentence containing a propositional function counts as an attempt to elicit information from the hearer provided that you do not know the answer, you want to know the answer, and so on.

If someone asks a question and doesn't want to know the answer, it seems unnatural to say that he has therefore failed to ask a real question. According to Alston (1971), Searle's rules should not be taken this way. What characterizes a real question is not that the conditions are all satisfied, but that the speaker accepts their relevance. If a tennis player serves into the net, he has still performed a real serve if he intends that his action should be judged by the rules which define a correct service. But if he was merely knocking up he will not accept that these rules apply; if his opponent cries "Double fault" and updates the score, he will protest that it wasn't a real serve.

The constitutive rules in (20) and (21) tell us the effect that a question or request is meant to produce in the hearer, but not how this effect is produced. H. P. Grice's article 'Meaning' (1971) shows that this is a more complicated matter than one might suppose. Grice's aim is to explicate some meanings of the word 'meaning', and in particular to explicate a sense which he calls 'non-natural meaning'. This sense is the one which occurs in sentences of the form 'S meant something by x', where x is an utterance. The concept of non-natural meaning is important because it clarifies the nature of linguistic communication. I shall try to explain it by first distinguishing several ways in which an act can communicate a meaning. Suppose that S and H attend a long lecture, and after a while S yawns and H concludes that S is bored. This apparently straightforward event might have occurred in four ways:

(i) S yawned unintentionally; H perceived the yawn as being unintentional.

(ii) S faked a yawn while H was looking, intending that H should perceive the yawn as being unintentional; and H did so perceive it.

(iii) As (ii), except that H detected the fake, but concluded nevertheless that S was bored.

(iv) S blatantly faked a yawn, intending that H should conclude that S was bored by recognising S's intention to tell him just that. H did recognise this intention, and believing S to be sincere, concluded that S was bored.

In each of these cases S's yawn means that he is bored, but only in case (iv) is this meaning non-natural. All linguistic communication is like case (iv). Thus according to Grice, 'S meant something by x' is roughly equivalent to 'S intended the utterance of x to produce some effect in the audience by means of the recognition of this intention'.

Putting together Grice's notion of non-natural meaning and his own constitutive rules, Searle (1971) characterises the illocutionary act as follows:

(22) In performing an illocutionary act, S intends:
(a) to produce a certain effect r.
(b) to produce r by getting H to recognise his intention to produce r (i.e. to recognise intention (a)).
(c) to get this recognition in virtue of the fact that the rules for using the expressions he utters associate these expressions with the production of the effect r (assuming S is using words literally).

I want to conclude this section by introducing the problem of *indirect speech acts*. An indirect speech act occurs when someone performs one illocutionary act under the guise of another. This device is often used to soften a request or command. Instead of the imperative 'Go now!', for example, the speaker might use one of the following (adapted from Searle (1975):

(a) Can you find your way out?
(b) I would like you to go now.
(c) Will you go now?
(d) Would you like to go now?
(e) You'd better go now.
(f) Might I ask if you wouldn't mind going now?

The problem, of course, is to explain how H assigns the force of a request to an utterance which superficially has the force of a question or statement. I will summarise two solutions to this problem, those of Gordon and Lakoff (1975) and Searle (1975).

Gordon and Lakoff suggest that in addition to the constitutive rules for illocutionary acts mentioned above, we have a set of rules which they call *conversational postulates*. A conversational postulate has this form: Defective illocutionary act X entails illocutionary act Y. An example is (23).

(23) ASK(S,H,CAN(H,Q))* → REQUESTS(S,H,Q)

In English this means that a defective (e.g. obviously unnecessary) question of the form 'Can H do Q' should be taken as a request for H to do Q.

Searle (1975) argues against this solution. His own suggestion (Searle, 1975) is that the hearer works out the entailment from other kinds of knowledge: from the utterance, the context, the constitutive rules, and the assumption that S is being co-operative. The new idea here is the *co-operative principle*, a concept introduced by Grice. (1968). Grice points out that contributions to a conversation are expected to conform to a number of general principles: (a) be relevant, (b) be clear, (c) be truthful, and so on. If two speakers trust one another, they can communicate more concisely by blatantly appearing to flout these principles, relying on the hearer to make whatever inferences are required to maintain the assumption of co-operativeness. In (24), for example, S flouts relevance, thereby informing H that his cousin is a dentist.

(24) H: Can you recommend a good dentist?
S: My cousin lives in your district.

Searle gives the following informal account of the process by which a hearer H perceives the utterance 'Can you pass the salt?' as a request. 'S has asked if I can pass the salt. He must know I can. He is therefore violating a preparatory condition for questions (see (20)). Since I can assume he is being co-operative, his remark must have some other force. His question concerns a preparatory condition for the request "Pass the salt" (see (21)) and since we are at dinner it is likely he wants the salt. He is therefore probably requesting me to pass the salt.' An important difference between this account and that of Gordon and Lakoff is that in Searle's account H must believe the request to be plausible. A defective 'Can you do A?' question is not AUTOMATICALLY reinterpreted as a request (see Searle (1975) for an example).

6. *Goal-Directed Speech Acts*

My reason for reviewing the Speech Acts literature, it will be recalled, was to seek a solution to the problem of how utterances achieve goals. I have given prominence to Searle's influential views, and I now want to propose an alternative solution which gives less importance to convention and more importance to ordinary goal-directed behaviour. The general problem, more precisely formulated, is this: how can a speaker S achieve a goal G by uttering a sentence X? I shall deal with three specific instances of this problem:

(25) a. How can S get a door open by saying to H, 'Open the door'?

b. How can S find out whether a door is open by saying to H, 'Is the door open'?
c. How can S get a door open by saying to H, 'Can you open the door'?

The general form of my solution is this: the utterance of X is regarded as a goal in the speaker's mind, and this goal is related to the higher goal G by a planning tree which specifies the intermediate goals. Diagram 26 illustrates this idea. It gives part of the planning tree which relates a goal G with a sentence R(G). R is a function which expresses a goal as an English imperative. In the case of problem (25a) G would be (Door open) and R(G) would be 'Open the door'. Each expression in round brackets, such as (H achieve G), represents a state of affairs, and the believing or intending or achieving of a state of affairs constitutes another state of affairs. This notation is informal: to keep it simple I have left the tenses of the verbs vague and ignored the distinction between states of affairs and events. Each goal in diagram (26) is a subgoal of the one immediately above it.

(26) g1 G
  g2 (H achieve G)
  g3 (H intend (H achieve G))
  g4 (H believe (S intend (H achieve G)))
  g5 (H believe (S utter R(G)))
  g6 (S utter R(G))
(In the case of problem (25a), G=(Door open), R(G)='Open the door')

I do not believe that this account of the relationship between g1 and g6 is complete; in particular, a lot more needs to be said about how g5 achieves g4. But I do believe that the goals g1-g6 exist in the head of a human speaker making a request, and that these goals are subordinated to one another as indicated by Diagram 26. To support this claim I shall show: (1) that people can distinguish a number of different ways in which an attempt to achieve g1 by g6 might fail, and that these ways correspond to different points of failure on the chain from g6 to g1; and (2) that people can devise alternative ways of achieving goals g1 to g5. Diagram 27 lists ways in which S might detect a failure at each of the goals g2 to g6, and diagram 28 gives alternative plans for achieving g1 to g5.

(27) g6: No sound comes out; S has lost his voice.
  g5: No response by H; he is not listening.

  g4: H replies 'Eliot?'; he thinks S is reciting poetry.
  g3: H realises that S wants him to achieve G, but refuses.
  g2: H tries to achieve G, but fails.
(28) g1: S achieves G himself.
  g2: S achieves the subgoal (H intend (H achieve G)), where G achieves G as a byproduct. For example S, a dope peddler, asks H to deliver a tin of 'talcum powder' to S's aunt in Copenhagen.
  g3: S convinces H that G is in H's interest, though not necessarily in S's.
  g4: S pretends to attempt G, making it plain that he neither expects nor intends to achieve G directly. For example S, a child, in the presence of an adult H, stretches for a light switch which he obviously cannot reach.
  g5: S, needing an alibi, persuades S to imitate his voice in a telephone conversation with H.

I shall give a fuller planning tree for requests later (Diagram 37). before doing so I want to show how such a tree might be generated. I shall assume that the speaker has a repertory of *goal-directed procedures* and *inference rules*. A goal-directed procedure associates a plan with a goal, the plan normally being a list of subgoals. An inference rule associates a list of premisses with a conclusion. (The idea of the goal-directed procedure, and indeed of the planning tree, is adapted from Hewitt [unpublished].) Here is an example of each:

(29) PROCEDURE P29
  Goal: (Door open)
  Plan: 1. (Door unlocked)
    2. (Someone push door)
(30) THEOREM T30
  Premisses: 1. (X has feathers)
    2. (X has a beak)
  Conclusion: (X is a bird)

For brevity I shall use the words PROCEDURE and THEOREM instead of 'goal-directed procedure' and 'inference rule'. Each PROCEDURE or THEOREM is given an arbitrary reference label (P29, T30). It is not necessary that the plan of a PROCEDURE always achieves the goal, or that the conclusion of a THEOREM is always true if the premisses are true. The premisses are merely good evidence for the conclusion.

If S can assume that H's repertory of THEOREMS is the same as his own, he can transform THEOREMS into PROCEDURES by the following manoeuvre: make the conclusion the goal, the premisses the plan, and replace each expression E by (H believe E). The PROCEDURE corresponding to T30 would be P 31:

(31) PROCEDURE P31
  Goal: (H believe (X is a bird))
  Plan: 1. (H believe (X has feathers))
     2. (H believe (X has a beak))

Although a planning tree is generated by PROCEDURES only, I shall regard a PROCEDURE as a transformed THEOREM if it seems plausible to do so. The transformation will be indicated by the letter 'a', so that T30a is identical to P31.

The PROCEDURES and THEOREMS required to generate the planning tree for a request are given in Diagrams 32–36.

(32) PROCEDURE P32
  Goal: G
  Plan: (H achieve G)
(33) PROCEDURE P33
  Goal: (H achieve G)
  Plan: 1. (H can achieve G)
     2. (H intend (H achieve G))
(34) PROCEDURE P34
  Goal: (H intend (H achieve G))
  Plan: 1. (H willing to help S)
     2. (H believe (H can achieve G))
     3. (H believe (S intend (H achieve G)))
(35) THEOREM T35
  Premisses: 1. (S utter R(G))
     2. (not (S quote R(G)))
  Conclusion: EITHER (S intend)H achieve G))
     OR (S pretend (S intend (H achieve G)))
(36) PROCEDURE P36
  Goal (H believe (S utter U))
  Plan: 1. (H listening to S)
     2. (S utter U)

The least satisfactory of these PROCEDURES and THEOREMS is T35, which has roughly the force of Searle's Essential rule (see (20)). A

lot more needs to be said about the relationship between (S utter R(G)) and (S intend (H achieve G)), and also about the concept of pretending. My reason for including pretending is that it is required in order to give an account of indirect speech acts.

I now give a planning tree for requests, a fuller version of Diagram 26. Since the usual notation does not fit comfortably on the page, indented tree notation is used. Each inverted L has a goal above the horizontal stroke, and the subgoals of this goal immediately to the right of the vertical stroke. For example, the subgoals of (H achieve G) are (H can achieve G) and (H intend (H achieve G)). The PROCEDURE used to elaborate a goal is given beneath the horizontal stroke; thus the procedure used to elaborate (H achieve G) is P33.

(37) Planning tree for requests
(In the case of problem (25a), G=(Door open), R(G)='Open the door'
G
P32 (H achieve G)
 P33 (H can achieve G)
   (H intend (H achieve G))
  P34 (H willing to help S)
    (H believe (H can achieve G))
    (H believe (S intend (H achieve G)))
  T35a (H believe (S utter R(G)))
   P36 (H listening to S)
     (S utter R(G))
    (H believe (not (S quote R(G))))
    (H believe (not (S pretend (S intend (H achieve G)))))

Note that most of the extra subgoals on diagram 37 (i.e. those omitted from Diagram 26 function as preconditions: they are not achieved by the utterance of R(G), but they must be satisfied if uttering R(G) is going to achieve G.

Let us now compare the account of requests given by Diagram 37 with Searle's constitutive rules (see (20)). Searle's preparatory conditions, in my notation, are: (H can achieve G), (S believe (H can achieve G), and (not (S believe (H intend (H achieve G)))). In my account these conditions follow from Diagram 37 together with two general principles of rational behaviour: (i) don't try to achieve goals which are already achieved, and (ii) don't use plans containing preconditions which are not already achieved. By combining these principles with Diagram 37 one can obtain two further preparatory conditions: (a) it must not be obvious that H is unwilling to help S; (b) H must not already believe that S intends him to achieve G. If for example a third party had just told H that S wanted him to achieve G, a request by S would violate condition (b).

Another difference between the two accounts is that in Diagram 37 the role of convention is confined to THEOREM T35. PROCEDURES P32, P33, P34 depend not on convention but on logic or nature. Nor is their use confined to conversation; goals like (H intend (H achieve G)) and (H believe (S intend (H achieve G))) can be attempted by plans which do not employ speech at all, as Diagram 28 demonstrates. My account therefore has the advantage that it relates speech acts to those non-speech acts with which they have an affinity.

A possible objection to Diagram 37 as a theory of how people produce requests is that it is too complicated; it seems implausible to suppose that a speaker generates a large tree like Diagram 37 every time he utters a simple request. My reply to this objection is as follows. Although I believe that people have a tree-like Diagram 37 (but far more complicated) in their heads when they utter requests, and that they also have procedures like P33, P34, etc which generate such trees, I do not suggest that a new tree is produced every time a request is uttered. In speech, as in all other behaviour, commonly used planning trees are presumably stored so that they can be applied quickly. When I type, I don't worry about how my finger movements cause letters to appear on the paper, but my (partial) understanding of this process is demonstrated when something goes wrong, e.g. when a key sticks. As it happens I am not good at repairing typewriters, but like everybody else I AM good at repairing conversations, and this suggests that we understand how they work. If in place of Diagram 37 we had merely a rule of thumb stating that to achieve G one should utter R(G), then we would not appreciate the distinctions made in Diagrams 27 and 28.

I shall now suggest how the goal-directed approach can be extended to deal with questions, like (25b), and indirect speech acts, like (25c). for questions, one more THEOREM and one more PROCEDURE are needed:

(38) THEOREM T38
Premisses: 1. (S utter Q(G))
2. (not (S quote Q(G)))
Conclusion: EITHER (S intend (H achieve (S know whether G)))
OR (S pretend (S intend (H achieve (S know whether G))))

(39) PROCEDURE P39
Goal: (H achieve (S know whether G))
Plan: 1. (H know whether G)
2. (H intend (H achieve (S know whether G)))

There is an obvious similarity between T38 and T35, and between P39 and P33; a better theory would derive T38 and T35 from a more basic THEOREM and P39 and P33 from a more basic PROCEDURE.

The goal of a question is (S know whether G), where G is again a state of affairs such as (Door open). Q is a function which converts a state of affairs into an English interrogative such as 'Is the door open'. The planning tree for questions is as follows:

(40) (S know whether G)
P32 (H achieve (S know whether G))
P39 (H know whether G)
(H intend (H achieve (S know whether G)))
P34 (H willing to help S)
(H believe (H can achieve (S know whether G)))
(H believe (S intend (H achieve (S know whether G))))
T38a (H believe (S utter Q(G))
P36 (H listening to S)
(S utter Q(G))
(H believe (not (S quote Q(G))))
(H believe (not (S pretend (S intend (H achieve (S know whether G))))))

Finally I offer a tentative solution to the problem of indirect speech acts such as (25c). Let us consider a question Q(C), where C=(H can achieve G). Thus if G=(Door open), C=(H can achieve (Door open)) and Q(C)='Can you open the door'. The problem is to explain how the utterance of Q(C) is supposed to achieve G.

The first step is to cause H to believe that S is pretending that he wants to know whether H can achieve G. In my notation this goal is (H believe (S pretend (S intend (S know whether C)))). The utterance of Q(C) in a context in which S obviously knows that C is true achieves this goal as follows. By THEOREM T38, if S utters Q(C) and is not quoting, then either he intends (H achieve (S know whether C)) or he pretends to intend this. Since (S know whether C) is obviously true, H will infer the latter. But if S is pretending that he intends (H achieve (S know whether

C)) he is thereby pretending that he intends (S know whether C). This inference is represented by T 41:

(41) THEOREM T41
Premisses: (S pretend (S intend (H achieve G)))
Conclusion: (S pretend (S intend G))

The second step is to cause H to believe that S intends H to achieve G; in my notation this goal is (H believe (S intend (H achieve G))). Once this goal is achieved we are on familiar territory; the rest of the planning tree coincides with Diagram 37. The only remaining problem is therefore to show how (H believe (S pretend (S intend (S know whether (H can achieve G))))) is supposed to achieve (H believe (S intend (H achieve G))).

The solution is roughly this. One reason for openly pretending to attempt X is to tell someone that you want X done. A child stretching for a light switch that he knows he cannot reach is telling the watching adult that he wants the light on. If S therefore openly pretends to attempt the goal (H achieve G), by pretending to check whether the subgoal (H can achieve G) is achieved, and it is plausible that S wants H to achieve G, then H can infer (S intend (H achieve G)). I shall give this inference rule in English rather than the previous notation:

(42) THEOREM T42
Premisses: 1. S pretends to attempt G. There are three ways in which he might do this:
(a) by pretending to attempt a subgoal of G;
(b) by pretending to check whether a subgoal of G has been achieved;
(c) by pretending to seek a plan to achieve G.
2. It is plausible in the given context that S intends G.
Conclusion: S intends G.

I can now give the planning tree for this type of indirect speech act. (See (43) on following page.)

This theory predicts a large class of indirect requests, because there are many ways in which S can pretend to attempt a goal. The first premise of T42 specifies these ways:

a. pretend to attempt a subgoal,
b. pretend to check whether a subgoal has been achieved,
c. pretend to seek a plan to achieve the goal or one of its subgoals.

In the case of a request, the goal S is pretending to attempt is (H achieve G), and on Diagram 37 the subgoals of this goal (excluding those which

(43) Planning tree for indirect speech act 'Can you ...'
C=(H can achieve G)

```
G
P32   (H achieve G)
 P33    (H can achieve G)
        (H intend (H achieve G))
  P34     (H willing to help S)
          (H believe (H can achieve G))
          (H believe (S intend (H achieve G)))
   T42a     (H believe (S pretend (S intend (S know whether C))))
    T41a      (H believe (S pretend (S intend (H achieve (S know whether C)))))
     T38a       (H believe (S utter Q(C)))
      P36         (H listening to S)
                  (S utter Q(C))
                  (H believe (not (S quoting Q(C))))
                  (H believe (not (S intend (H achieve (S know whether C)))))
```

do not mention G) are numbers 2 to 6 on the following list:

1. (H achieve G)
2. (H can achieve G)
3. (H intend (H achieve G))
4. (H believe (H can achieve G))
5. (H believe (S intend (H achieve G)))
6. (H believe (S utter R(G)))

I shall now generate a list of indirect speech acts by combining goals 1 to 6 with pretences a to c. To give the reader some relief from the door example, a different setting is used: a mother S is trying to get her son H to wash up.

(44) Some indirect speech acts predicted by the theory
1c. Mother to son: 'I wonder if those dishes would become clean if you immersed them in hot water'.
2a. 'Let me explain how to wash up ...' (son already knows).
2b. 'Can you wash up?'
2c. Mother to father: 'Perhaps I should show junior how to wash up'.
3a. Mother to son (sweetly): 'I'm sure you'd enjoy washing up'.
3b. 'Were you planning to wash up?'
3c. Mother to father: 'I wonder whether junior's attitude to washing up could be altered by an adjustment of his pocket money'.
4a. 'I'm sure you could manage the washing up, dear'.
4b. 'Do you think you can wash up?'

4c. Mother to father: 'How can we convince junior that he has the ability to wash up?'
5a. Having already said 'Wash up, junior!' mother says sweetly: 'I'd like you to wash up dear'.
5b. Having asked him twice already, mother says sweetly: 'Did you realise I wanted you to wash up dear?'
5c. Having asked her son to wash up, without result, mother says to father: 'How can I show junior that I want him to wash up? I can't seem to explain it properly'.
6a. Having just said 'WASH UP, JUNIOR!' mother says very clearly and distinctly, but without anger, 'Wash — up — junior'.
6b. Having just said 'WASH UP, JUNIOR!', mother says sweetly: 'Did you hear me say "Wash up", dear?'
6c. Having just said 'WASH UP, JUNIOR!', without result, mother says to father: 'I wonder if a loud-speaker would help'.

Some of these are too involved to be used much, but I think that in each case junior would get the message.

It might be objected, again, that this account is too complicated, and that 'Can you wash up' is merely a form of words. For the more common indirect speech acts I think this may be true: we probably use rules of thumb like the Gordon and Lakoff conversational postulates. But I think we also have a general understanding of indirect speech acts which enables us to appreciate unfamiliar varieties like some of the examples in (44). These have an element of humour which the familiar 'Can you do X' seems to have lost.

7. *The Problem of Flexibility*

In Section 4 I argued that the program was incapable of conversing flexibly because it lacked an explicit representation of the dialogue state. In this section I shall illustrate what I mean by 'conversing flexibly' and sketch a more explicit system of representation.

Suppose that robot John is conversing with a human operator (the program can be used this way) and that John has just said: 'I suggest we get the door open and then I move in'. There are in fact just three replies he will understand: consent, refusal, and a call-by-name. People would understand a very much larger set of replies, including the following:

(45)
a. The door IS open.
b. We can't get the door open.

c. You can't move.
d. You ARE in.
e. When I said you were out I was lying.
f. When I said the door was shut I was joking.
g. I can't really see the door, you know.
h. The bolt is down and I can't slide it.
i. Can you see the door?
j. If you move, you get in whether or not the door is open.
k. I know someone who got in by moving when the door was shut.
l. I thought you wanted to get out.

Many of these replies cannot be understood merely in terms of John's last remark: they must be related to the whole dialogue state. The point of a reply like (45f), for example, is that a specific part of John's copy of the dialogue state is wrong. To understand the point of the reply John must identify the erroneous part and perceive its connection to his candidate plan. And to do this he needs an explicit representation of the whole dialogue state.

Imagine a representation composed of the following objects: assertions; goals; plans; candidate plans; goal-directed procedures; inference rules; and justifications of assertions, goals and plans. Each assertion must be associated with a justification, which points to the assertions which are evidence for it and the inference rule used in inferring it from the evidence. Some assertions are based on direct perception; for simplicity let these be permanent. The survival of the others depends on the survival of their justifications; if an assumption made in an inference turns out to be false, a chain of assertions may be removed. The justification of a goal points to (i) an assertion that it is not yet achieved, (ii) one or more candidate plans that show it is possible, and (iii) a plan or candidate plan of which the goal is a part, unless it is the main goal. The justification of a plan or candidate plan points to the goal-directed procedure it is derived from, to its component goals, and to the goal it is meant to achieve. The survival of any object, whether assertion, goal, or plan, depends on the survival of the objects that its justification points to.

Suppose that the dialogue state is represented by a structure composed of these objects. The structure will have grown from the main goal, and wil be connected to the system's repertory of assertions, goal-directed procedures, and inference rules by pointers. When John says 'I suggest we get the door open and then I move in' he is connecting a candidate plan to the dialogue state, or perhaps highlighting one already there, it being understood that a positive reply by the operator means that this candidate is selected. The replies in (45) are all relevant because, if accepted, they would trigger a chain reaction which would sweep John's candidate plan away.

The relationship between John's candidate plan and (45f) as it would be represented by this apparatus, is as follows:

(46) 1. There is a candidate plan to achieve the goal (John in).
2. The first subgoal of this plan is (Door open).
3. This subgoal is necessary because the assertion (Door shut) is true.
4. (Door shut) is believed true because (Operator believe (Door shut)) is believed true. The inference rule is that if S believes P, and S is in a position to know whether P is true, then P is probably true.
5. (Operator believe (Door shut) is believed true because (Operator said 'The door is shut') was directly observed. The inference rule is that if S says A(P), where A(P) is P expressed as a declarative, and S is not joking, deceiving, or reciting, then S probably believes P.

So the point of (45f) is to contradict an assumption of the inference in 5, thereby contradicting (Operator believe (Door shut)), thereby contradicting (Door shut), thereby removing the justification for subgoal (Door open), thereby removing the justification of the candidate plan of which it is a part. Unless John perceives the whole of this connection, he fails to get the point of the reply.

The problem with the actual program is that although it uses plans, candidate plans, and statements about whether goals have been achieved, these are not explicitly marked and related in the manner suggested above. For example, local variable P of conversational procedure AGREEPLAN represents a candidate plan, but we only call it a candidate plan because of the way in which the surrounding instructions deal with it; it is not labelled in a way which a general intepreter could pick up, nor is it linked explicitly to its provisional goal and its justification. Indeed, its justification — the tests on whether the subgoals are necessary and possible, for example — is computed by planning procedure FINDPLAN and is lost when FINDPLAN exits and returns control to AGREEPLAN. All this information needs to be saved as part of a global data structure, the connections being explicitly marked rather than implicit in the history of the procedure calls.

Received 12 September 1978
Revised version received
17 December 1978

Via della Provvidenza 15
35030 Sarmeola di Rubano,
Padova
Italy

Note

* This research was carried out at the School of Artificial Intelligence, Edinburgh, and at the Laboratory of Experimental Psychology, University of Sussex. I would like to thank Christopher Longuet-Higgins, who supervised the Ph.D. project, and Max Clowes, Stephen Isard, Mark Steedman, Aaron Sloman, Stuart Sutherland, and Gerald Gazdar, for helpful comments. Financial support was provided by the SRC and the Royal Society.

References

Alston, W. (1964). *Philosophy of Language*. Englewood Cliffs: Prentice Hall.
Austin, J. L. (1971). Performative-Constative. In J. R. Searle (ed.), *The Philosophy of Language*. Oxford: Oxford University Press.
—(1962). *How To Do Things With Words*. London: Oxford University Press.
Burstall, R. M., Popplestone, R. J. and Collins, (1972). *Programming in POP-2*. Edinburgh University Press.
Cohen, P. R. (1977). Knowing what to say: planning speech acts in purposeful dialogues. Ph.D. thesis, Department of Computer Science, University of Toronto.
Cole, P. and Morgan, J. (eds) (1975). *Syntax and Semantics, Vol. 3, Speech Acts*. New York: Academic Press.
Gordon, D and Lakoff, G. (1975). Conversational postulates. In P. Cole and J. Morgan (eds), *Syntax and Semantics, Vol. 3, Speech Acts*. New York: Academic Press.
Grice. H. P. (1975). Logic and conversation. In P. Cole and J. Morgan (eds), *Synatx and Semantics, Vol. 3, Speech Acts*. New York: Academic Press.
—(1968). Utterer's meaning, sentence-meaning, and word-meaning. In *Foundations of Language* 4.
—(1971). Meaning. *Philosophical review*.
Hewitt, C. (n.d.) Description and theoretical analysis (using schemes) of PLANNER. Unpublished Ph.D. thesis, MIT Cambridge, MA.
Neisser, U. (1967). *Cognitive Psychology*. New York: Appleton-Century-Crofts.
Power, R. J. D. (1974). A computer model of conversation. Ph.D. thesis, University of Edinburgh.
Schegloff, E. and Sacks, H. (1973). Opening up closings. *Semiotica* 8.
Searle, J. R. (1969). *Speech Acts*. Cambridge: Cambridge University Press.
—(1971). What is a speech act? In J. R. Searle (ed.), *The Philosophy of Language*. Oxford: Oxford University Press.
—(ed.) (1971). *The Philosophy of Language*. Oxford: Oxford University Press.
—(1975). Indirect speech acts. In P. Cole and J. Morgan (eds), *Syntax and Semantics, Vol. 3, Speech Acts*. New York: Academic Press.
—(1975). Speech acts and recent linguistics. In D. Aaronson and R. W. Rieber (eds), *Annals of the New York Academy of Sciences* 263, *Developmental Psychology and Communication Disorders*.
Strawson, P. F. (1964). Intention and convention in speech acts. *Philosophical Review* 73 No. 4.
Winograd, T. (1971). *Understanding Natural Language*. Edinburgh: Edinburgh University Press.
Wittgenstein, L. (1953). *Philosophical Investigations*. Oxford: Blackwell.