6CCS3AIN Lecture on Machine Languages

SGT Tutorial Questions

Consider the following agent communications protocol:

2-AGENT DIVISION PROTOCOL

1. Description: This is a protocol for 2 autonomous agents, called A (Alice) and B (Bob).

2. Assumptions: The symbol p represents a fraction in the unit interval, [0, 1], such as 0.6.

3. Utterances: The following 6 types of utterances are permitted, for any value of p. (In these utterances, A and B are fixed agent names, while p is a variable.)

   START(A,B)

   DIVIDE(A=p, B=1-p)

   ACCEPT(A=p, B=1-p)

   REJECT(A=p, B=1-p)

   NO-COMMENT

   QUIT

4. Combination Rules:

   Either agent may utter START(A,B) at any time. This starts a new dialog between agents A and B.

   Either agent may utter NO-COMMENT at any time.

   Either agent may utter QUIT at any time. If any agent does, the dialog ends.

   Either agent may utter DIVIDE(A=p, B=1-p) at any time.

   Whenever one agent utters DIVIDE(A=p, B=1-p), the other agent must utter one of the following three utterances:

   ACCEPT(A=p, B=1-p) or REJECT(A=p, B=1-p) or NO-COMMENT

5. Termination Conditions:

   The dialog terminates whenever one agent utters QUIT.

Question 1

(a) What application do you think the protocol is intended for? What type of dialog is it?

(b) What do you think is the intended meaning of the utterance DIVIDE(A=p, B=1-p)?

(c) What do you think are the intended meanings of ACCEPT(A=p, B=1-p) and REJECT(A=p, B=1-p)?

(d) Draw a tree diagram to show the possible dialog trajectories over time (with time along the horizontal axes, and with possible dialogs extending from left to right). (The tree may be infinite, so in this case only draw the first part of the tree.)

(e) Is it possible for dialogs to terminate?

(f) Will dialogs always terminate?

(g) A cycle in a dialog is a situation where the same collection of statements are made over and over again. Are cycles possible under this protocol?

(h) Suppose one agent wants to never reach agreement with the other agent, but also wants the dialog to last as long as possible. What strategy should that agent adopt? What strategy can the other agent adopt to counter this?

(Note: In legal disputes, deliberately delaying the proceedings as long as possible is sometimes a strategy used by one party to the dispute. Usually, this is done by a richer party against a poorer party, in the hope that the poorer party will give up due to the rising costs.)

Question 2 (Continuation)

After experiment with this protocol, an additional termination rule and additional combination rule are added.

Termination rule:

The following sequence of utterances leads immediately to dialog termination:

DIVIDE(A=p, B=1-p)
ACCEPT(A=p, B=1-p)

Combination rule:

If either of the following sequences of utterances occurs in a dialog

DIVIDE(A=p, B=1-p)
REJECT(A=p, B=1-p)

or

DIVIDE(A=p, B=1-p)
NO-COMMENT

then the agent who uttered DIVIDE(A=p, B=1-p) must respond immediately with:

DIVIDE(A=q, B=1-q) where $q < p$, in the case that A uttered DIVIDE(A=p, B=1-p),

or where $q > p$, in the case that B uttered DIVIDE(A=p, B=1-p).

Question 2

(a) What situation do you think the new Termination rule is intended to deal with?

(b) What is the immediate impact of the new Combination rule?

(c) Will dialogs now always terminate?

(d) What additional condition must we place on the variable p to ensure that dialogs always terminate?

6CCS3AIN Lecture on Machine Languages

SGT Tutorial Questions

Consider the following agent communications protocol

2-AGENT DIVISION PROTOCOL

1. Description: This is a protocol for 2 autonomous agents, called A (Alice) and B (Bob).

2. Assumptions: The symbol p represents a fraction in the unit interval, [0, 1], such as 0.6.

3. Utterances: The following 6 types of utterances are permitted, for any value of p. (In these utterances, A and B are fixed agent names, while p is a variable.)

   START(A,B)

   DIVIDE(A=p, B=1-p)

   ACCEPT(A=p, B=1-p)

   REJECT(A=p, B=1-p)

   NO-COMMENT

   QUIT

4. Combination Rules:

   Either agent may utter START(A,B) at any time. This starts a new dialog between agents A and B.

   Either agent may utter NO-COMMENT at any time.

   Either agent may utter QUIT at any time. If any agent does, the dialog ends.

   Either agent may utter DIVIDE(A=p, B=1-p) at any time.

   Whenever one agent utters DIVIDE(A=p, B=1-p), the other agent must utter one of the following three utterances:

   ACCEPT(A=p, B=1-p) or REJECT(A=p, B=1-p) or NO-COMMENT

5. Termination Conditions:

   The dialog terminates whenever one agent utters QUIT.

Question 1

(a) What application do you think the protocol is intended for? What type of dialog is it?

To enable two agents to agree a division of some resource (eg, a cake).

(b) What do you think is the intended meaning of the utterance DIVIDE(A=p, B=1-p)?

The speaker is proposing that agent A is allocated the proportion p of the resource, while B is allocated 1-p.

(c) What do you think are the intended meanings of ACCEPT(A=p, B=1-p) and REJECT(A=p, B=1-p)?

The first utterance indicates acceptance of the proposed division of the resource, while the second utterances indicates rejection of the proposed division.

(d) Draw a tree diagram to show the possible dialog trajectories over time (with time along the horizontal axes, and with possible dialogs extending from left to right). (The tree may be infinite, so in this case only draw the first part of the tree.)

Draw a tree.

(e) Is it possible for dialogs to terminate?

Yes. One agent only needs to utter QUIT and the dialog will terminate.

(f) Will dialogs always terminate?

No. Dialogs may continue forever.

(g) A cycle in a dialog is a situation where the same collection of statements are made over and over again. Are cycles possible under this protocol?

Yes. An agent may keep uttering a proposal with a specific value of p which the other agent rejects or does not respond to, forever, as in:

DIVIDE(A=0.8, B=0.2)
REJECT(A=0.8, B=0.2)
DIVIDE(A=0.8, B=0.2)
REJECT(A=0.8, B=0.2)
DIVIDE(A=0.8, B=0.2)
REJECT(A=0.8, B=0.2)
DIVIDE(A=0.8, B=0.2)
REJECT(A=0.8, B=0.2)

etc

(h) Suppose one agent wants to never reach agreement with the other agent, but also wants the dialog to last as long as possible. What strategy should that agent adopt? What strategy can the other agent adopt to counter this?

(Note: In legal disputes, deliberately delaying the proceedings as long as possible is sometimes a strategy used by one party to the dispute. Usually, this is done by a richer party against a poorer party, in the hope that the poorer party will give up due to the rising costs.)

Suppose agent A wants to delay reaching agreement. This agent could make a proposal which B would find completely unacceptable, such as:

DIVIDE (A=1, B=0)

and keep repeating this forever, despite it being rejected by the other agent. If agent B makes a proposal instead, agent A could respond with rejection or with NO-COMMENT every time.

It would be hard for the other agent to counter this delaying strategy. The only effective means to counter it would be to utter QUIT.

Question 2 (Continuation)

After experiment with this protocol, an additional termination rule and additional combination rule are added.

Termination rule:

The following sequence of utterances leads immediately to dialog termination:

DIVIDE(A=p, B=1-p)
ACCEPT(A=p, B=1-p)

Combination rule:

If either of the following sequences of utterances occurs in a dialog

DIVIDE(A=p, B=1-p)
REJECT(A=p, B=1-p)

or

DIVIDE(A=p, B=1-p)
NO-COMMENT

then the agent who uttered DIVIDE(A=p, B=1-p) must respond immediately with:

DIVIDE(A=q, B=1-q) where q < p, in the case that A uttered DIVIDE(A=p, B=1-p),

or where q > p, in the case that B uttered DIVIDE(A=p, B=1-p).

Question 2

(a) What situation do you think the new Termination rule is intended to deal with?

The situation where the two agents reach agreement on a proposed division. There is no point continuing the dialog if the two agents reach agreement.

(b) What is the immediate impact of the new Combination rule?

The immediate impact is that an agent making a proposal that is not accepted, needs to then immediately offer a new proposal which is better for the other agent than the previous proposal.

(c) Will dialogs now always terminate?

No. An agent could make an infinite sequence of proposals which are increasingly better for the other agent, but never reach an allocation which the other agent finds acceptable. For example, suppose B will not accept any proposal that offers B less than 40% of the resource. If A knows (or suspects) that B has this threshold then A could make an infinite sequence of proposals each of which never goes above 40%.

For instance, this sequence of proposals:

DIVIDE(A=7/8, B=1/8)

DIVIDE(A=13/16, B=3/16)

DIVIDE(A=25/32, B=7/32)

DIVIDE(A=49/64, B=15/64)

DIVIDE(A=97/128, B=31/128)

DIVIDE(A=193/256, B=63/256)

etc

Each proposal is better for B than the ones before, but the new incremental amount proposed for B is smaller each round. Here, we are using the fact that the infinite series comprising the fractions:

1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, ...

sums to 1. Hence, the truncated series of fractions:

1/8, 1/16, 1/32, 1/64, 1/128, ...

sums to 1/4. Thus, with this sequence, the proposal being made to B at each round will always be less than 1/4, and so each proposal will always below B's threshold of 40% for B.

(d) What additional condition must we place on the variable p to ensure that dialogs always terminate?

We would need to ensure that the set of values for the variable p is finite, to preclude the use of an infinite sequence as in the previous question. For example, we could say that p must be one of the following eleven values:

{0, 0.1, 0.2, 0.3, 0.4, . . ., 0.8, 0.9, 1.0}

(e) Discussion question: Can the students think of ways to subvert the protocol (eg, to make it run forever) even with this extra restriction on the value of p?