Notre Dame Journal of Formal Logic
Volume XXI, Number 1, January 1980
NDJFAM

DISCOURSE BETWEEN PROCESSES

JAN BERGSTRA

Introduction and definitions Let $\mathcal{L}$ be a countable language. $\mathcal{L}$ contains a special word START. A discourse over $\mathcal{L}$ is an infinite sequence $k = \langle \text{START}, k_q^1, k_a^2, k_q^2, k_a^3 \ldots . \rangle$, where $k_a^1 = \text{START}$. The q-components of k are called questions; the a-components are answers. The word START is used to initiate the discourse and invokes a first question of the first speaker. It is assumed that $k_q^i \neq \text{START}$ $(i \geqslant 1)$, $k_a^{i+1} \neq \text{START}$ $(i \geqslant 1)$. We denote the set of discourses by D.

Before proceeding it may be useful to note that our considerations will be meaningful for finite discourses as well; the infinite case, however, is more general.

Now suppose that by some criterion we established that $SD \subseteq D$ consists of the sensible (meaningful) discourses. We ask the following question: Is there a set SP of sensible speakers such that:

1. for every $k \in SD$ there are p_1 and p_2 in SP such that the discourse determined by p_1 and p_2 (notation: $p_1 \square p_2$) is just k.
2. for all p_1 and p_2 in SP $p_1 \square p_2 \in SD$.

Of course we must specify exactly what a speaker can be to make the problem well-defined. We feel that if SD is to be the set of meaningful discourses in some sense there must exist a corresponding SP. The more natural the notion of a speaker is the more the existence of SP is a requirement for SD if it is to be a set of sensible discourses (in some sense which remains unspecified).

In this note we define the class of speakers as the class of deterministic processes with inputs in $\mathcal{L}$ and outputs in $\mathcal{L}' = \mathcal{L} - \{\text{START}\}$.

Definition A process is a function $p: \mathcal{L}^* \to \mathcal{L}'$, where $\mathcal{L}^*$ is the set of finite sequences of words in $\mathcal{L}$. Given processes p_1 and p_2 we define $p_1 \square p_2 = \langle \text{START}, k_q^1, k_a^2, k_q^2, \ldots . \rangle$ by means of the following recursion:

$$\begin{cases} k_q^1 = p_1(\langle \text{START} \rangle) \\ k_a^2 = p_2(\langle k_q^1 \rangle) \\ k_q^{i+1} = p_1(\langle \text{START}, k_a^2, \ldots, k_a^{i+1} \rangle) \\ k_a^{i+1} = p_2(\langle k_q^1, \ldots, k_q^i \rangle) \end{cases} .$$

Received April 30, 1978

Finally we define for $K \subseteq P$: $K \square K = \{p_1 \square p_2 | p_1, p_2 \in K\}$.

Theorem *For all $SD \subseteq D$ there exists $SP \subset P$ such that $SD = SP \square SP$.*

Comment: From the motivation as formulated in the introduction we must conclude that this is a negative result. It tells that the existence of a subset SP of P such that $SD = SP \square SP$ is a trivial condition. Therefore it cannot be used to specify, e.g., sets of meaningful discourses.

Proof: We use s to denote initial segments of discourses. If $ln(s)$, the length of s, is even then p_2 is the next to speak otherwise p_1. Let IS be the class of initial segments of discourses in D. We write $s < k$ if s is an initial segment of k. Let $SIS = \{s \in IS \mid \exists k \in SD\ s < k\}$. Let A be a countable subset of SD such that $\forall s\,[(\exists k \in SD\ s < k) \rightarrow (\exists k \in A\ s < k)]$. The existence of A follows from the fact that there are only countably many initial segments (although SD may well be uncountable). Let F be a bijective function from ω, the natural numbers, to A. We define a partial mapping $f: IS \rightarrow A$ with domain SIS as follows: $f(s) = F(n)$, where n is the least m, if any, such that $s < F(m)$. Now we define for all $k, t \in SD$ processes p^k, p^t in such a way that:

i. $\forall k, t \in SD\ p^k \square p^t \in SD$

ii. $\forall k \in SD\ p^k \square p^k = k$.

Then we may take SP: $\{p^k | k \in SD\}$.
We will give an algorithmic description of the p^k using the following information: (i) the characteristic function of SIS; (ii) f; and (iii) k. To present the algorithm we use a self explaining programming language for processes. Questions are input, answers are output. QUESTION is a word identifier which always has the value of the last question that has been received. NEWQUESTION is a statement asking for a new question. The result is an update of QUESTION. ANSWER(k) is a statement expressing that $k \in \mathcal{L}$ is answered. We first define $\overline{p}_1^k$ and $\overline{p}_2^t$ such that always $\overline{p}_1^k \square \overline{p}_2^k = k$ and $\overline{p}_1^k \square \overline{p}_2^t \in SD$ for $k, t \in SD$. The program for $\overline{p}_1^k$ has four main internal states: I, . . ., IV.

```
I    NEWQUESTION
     n := 1
     if QUESTION = START then s := ⟨START, k_q^1⟩
                              ANSWER(k_q^1)
                              GOTO II
                         else GOTO IV
     fi
```

(Comment: in state I $\overline{p}_1^k$ receives START, counter n is initialized as well as s which will denote the initial segment at any stage. n counts the number of questions that have been received. IV is the state which collects all errors.)

```
II   NEWQUESTION
     n := n + 1
```

```
s := s * QUESTION
if s < k then s := s * k_q^n
              ANSWER (k_q^n)
              GOTO II
         else GOTO III
fi
```

(Comment: as long as $s < k$ $\bar{p}_1^k$ answers consistent with k, if its partner does not follow k any longer a new strategy is followed in III.)

```
III  if s ∈ SIS then s := s * f(s)_q^n
                   ANSWER(f(s)_q^n)
                   NEWQUESTION
                   n := n + 1
                   s := s * QUESTION
                   GOTO III
              else GOTO IV
     fi
```

(Comment: $\bar{p}_1^k$ tries to follow $f(s)$ at any stage.)

```
IV   ANSWER(k_0) (Comment: k_0 is some fixed element of ℒ.)
     NEWQUESTION
     GOTO IV
```

The program for $\bar{p}_2^k$ is quite similar. In state I it only initializes n and s but does not read. In state II it gives answers of the form k_a^n and in state III of the form $f(s)_a^n$.

Now we must show for $k, t \in SD$:

1. $\bar{p}_1^k \Box \bar{p}_2^k = k$. Both $\bar{p}_1^k$ and $\bar{p}_2^k$ remain in their respective states II and k is the resulting discourse.

2. $\bar{p}_1^k \Box \bar{p}_2^k \in SD$. There are two cases (let $h = \bar{p}_1^k \Box \bar{p}_2^t$):

   i. $\bar{p}_1^k$ or $\bar{p}_2^t$ remains in its state II, then either k or t must be the resulting discourse. (Of course $k, t \in SD$.)

   ii. both $\bar{p}_1^k$ and $\bar{p}_2^t$ move to their respective states III after a (finite) part of the computation of h. Let this be the case after initial segment s^1 of h. With induction on the length of $s < h$ one proves $s \in SIS$, using that $s \in SIS$ implies $s * f(s)_q^{n+1} \in SIS$ if $ln(s) = 2n + 1$ and $s * f(s)_a^{n+1}$ if $ln(s) = 2n$. To see this note that $f(s)$ always extends s. We claim that in fact $h = f(s^1)$. This follows from the following equalities for $s^1 \leqslant s < h$:
   $f(s) = f(s * f(s)_q^{n+1})$ if $ln(s) = 2n + 1$ and
   $f(s) = f(s * f(s)_a^{n+1})$ if $ln(s) = 2n$.
   The reason for these equalities is that $f(s)$ is the minimal extension of s in SD (in the sense of F) which is clearly equal to the minimal extension of any longer initial segment of $f(s)$ in SD.

Now p^k is simply described as follows: If the first question received is START then it behaves like $\bar{p}_1^k$, otherwise like $\bar{p}_2^k$. This completes the proof of the theorem.

Conclusion As mentioned before our method works in the case of finite discourses too. If we look at games as discourses we can draw the following conclusion: Let SD be a collection of chess games, then there exists a collection of strategies SP such that $SD = SP \square SP$.

University of Leiden
Leiden, The Netherlands