SOME LINGUISTICAL PROBLEMS ABOUT COLLOQUIES

by

Fabio A. Schreiber
Istituto di Elettrotecnica ed Elettronica
Politecnico di Milano
P.zza Leonardo da Vinci, 32
20133 Milano (Italy)

KEYWORDS: communication protocols, computer communications, computer networks, dialogues, finite automata, formal languages, interlocutors, variable structure sequential machines.

1. INTRODUCTION

Lots of things have been written about the formal description of languages, seen as meaningful strings of symbols issued by a single "speaker", be he a human or a computer.

This set of studies has developed out of the need of having a very clear description and of finding some properties of languages useful to the design of translation algorithms. Considering the case of programming languages, a single speaker - the programmer - tells the computer how to operate in a symbolic, possibly high level, language and this description has to be compiled to produce machine code.

Commonly, however, a speaker is not alone playing long monologues, but he interacts with another speaker, giving rise to dialogues, or with more than one, generating complex colloquies. Today this consideration has become of actuality for computers as well as for humans. The growing interest in connecting computers to share resources has stressed the problem of describing and designing complex colloquy procedures.

This paper briefly reports on an attempt for formally describing computers interactions by means of automata and grammars, and it is meant more to be of stimulus to further research in the field than to draw any conclusion on the subject.

Firstly the case of two partners exchanging messages between them will be considered, then some considerations will be made on more complex form of interaction.

2. THE DIALOGUE

Before describing the interaction between two speakers, we call hereafter interlocutors, let us characterize the interlocutors themselves.

An interlocutor can be described as a black-box with six couples of ports. At ports m and μ (Message ports) information is exchanged with the partner in the colloquy; at ports c and η (Command ports) information is exchanged with the environment the interlocutor is embedded in; at ports t and τ (Text ports) possibly a part of the information carried by the message is extracted (inserted) to be forwarded to the environment (to the partner), fig. 1, /1/, /2/.

In this study, our interest is mainly focused on what is going on at the M-ports depending on the partner's message and on the environmental status, as they are seen at ports μ and η .

Some authors /3/ have described the colloquy procedures by means of grammars, treating the colloquy as a whole, the structure of the partners being not clearly identifiable.

Also for easing the implementation of the devices behaving as interlocutors inside a machine, we prefer to keep the individuality of the partners and to reflect this individuality on the grammatical description.

It has been shown /1/, /4/ that a suitable model for the black-box in fig. 1 is a Variable Structure sequential Machine (VSM), that is a Mealy machine with a set of transition functions and a set of output functions on the same set of internal states. The environment, selecting one transition and one output function out of their sets, introduce through port η a determinism in a machine that would have been otherwise non deterministic. In any case, the grammatical description of such a machine results in a regular grammar /5/.

Formally then a colloquy can be described by a grammar

$$G = (V_N, V_T, P, S)$$

where:

$V_N = \{A,\ldots,Z,A',\ldots,Z'\} = \{N \cup N'\} = \{\alpha_i \mid 1 \leq i \leq 52\}$ is the non terminal vocabulary

$V_t = V_M \cup V_S$ is the terminal vocabulary

$V_M = \{m_1, \ldots, m_n, m'_1, \ldots, m'_m\}$

is the set of the messages of the colloquy (the empty message included)

$V_S = \{\eta_1, \ldots, \eta_p, \eta'_1, \ldots, \eta'_q\}$

is the selector vocabulary, constituted by the commands from the environment

P is a set of productions of the form

$$S \rightarrow m_j \alpha_i$$

$$N_h \rightarrow |_{\eta_i}\ m_j \alpha_m |_{\eta_k}\ m_1 \alpha_n \quad \ldots$$

$$N'_h \rightarrow |_{\eta'_i}\ m'_j \alpha_m |_{\eta'_k}\ m'_1 \alpha_n \quad \ldots$$

S is the start symbol

The terminal vocabulary partitioned in two sets V_M and V_S, with the condition $V_M \cap V_S = \emptyset$, reflects the existence of two inputs, μ and η , with a different semantical meaning.

As it can be noticed, the terminal and non terminal vocabularies are constituted by symbols with and without a prime index. This distinction results from the presence of two interlocutors, and the productions, by specifying in their left side which interlocutor the non terminal belongs

to, give the rules of interaction. Would in all the productions the non terminal at left not belong to the same machine than those in the right side, a "thrust and riposte" dialogue is generated. On the other hand, if the non terminals in the same production can belong to the same machine, mechanisms as the message generation on a time-out can be modeled.

As to the problem of relating machines to symbols, we remember that a master/slave relation exists between the interlocutors. The relation can be of fixed type or, as in the most general case of a symmetrical colloquy with a contention procedure, that is when the two partners are identical and both can begin a colloquy, the master/slave relation exists in the sense that who opened the colloquy has to wait for the partner's availability. So we identify the slave machine putting the prime on its symbols. It can be noticed that if the relation is of fixed type the prime uniquely identifies one of the two machines, while in the case of symmetrical colloquy the condition of slave is attributed in a dynamical way so that all the productions are to be duplicated with and without the prime.

Let us give in fig. 2 a very simple example of a colloquy to achieve the transmission of a text from one machine to the other, both as a message-graph /1/ and as a grammar. In fig. 3 the two interlocutors implementing the colloquy are described

by means of their state-graphs.

3. AIM OF THE MODEL

We want now point out which problems can take advantage of such a representation of colloquies.

- DOCUMENTATION, descriptions of colloquy procedures, as they can be found in today's manuals, are very poor and fuzzy. The message graph and the syntactical model could supply a clear and complete standard form of representation.

- EQUIVALENCE, some well known properties of regular grammars and finite automata assure the decidability of the problem of determining if two automata (grammars) accept (generate) the same language /5/. These properties can be used to compare the description of different interlocutors for determining if they are equivalent, in order to be directly connected in a colloquy (possibly but for a transcoding of terminal vocabularies).

- COMPATIBILITY, this is a broader property than equivalence. In fact, it is not necessary for two interlocutors to colloquiate to be fully equivalent with respect to the colloquy procedure. It is enough to have a subset of the services requested by the slave equivalent to a subset of those offered by the master. In this case the resulting colloquy can be a degraded one, but some interaction can be kept alive by a non empty exchange of messages.

- IMPLEMENTATION, the colloquy protocol being given by means of a syntactic description as the one presented above, the implementation of the two interlocutors as VSMs is rather strightforward /6/, /7/. It must be noticed that it can be conceivable that communication procedures much more complicate than the one in the given example (for instance file transfer protocols, etc.) be given as "programs" in some high level language and these "programs" be compiled in order to obtain as final result the descriptions of the two interlocutors.

4. MULTIPLE INTERLOCUTORS COLLOQUIES

In this section we try to extend unformally the model to colloquies among several interlocutors, pointing out the changes this extension brings to the class of grammars used for the description.

The case of the dialogue between two interlocutors on a point-to-point link is the simplest one also from the point of view of the communication net topology. Let us examine a slightly more complicate one, that is the multipoint connection of several interlocutors sharing the same link. This case is generally applied when very asymmetrical colloquies are carried on among a master computer and several terminals acting all as slaves. The master computer polls the terminals in a prescheduled order to see if they have text to send. If it finds a terminal ready to transmit, a dialogue is initiated between the two in

terlocutors which terminates when the terminal sends an End Of Text character. At this moment the polling procedure is resumed. In a very similar way the master selects the terminal which it has text to send to, giving rise to a dialogue again. We can conclude therefore that for a multipoint connection, although the communication line is shared among several interlocutors, only one dialogue a time can take place between the master computer and one of the terminals, so it can be represented by the model of section 2. As to the scheduling policy of the master interlocutor, it can be modeled by a counter, which is in turn a finite state machine. Since the concatenation of finite states machines is still a finite state machine, all the considerations made for point-to-point dialogues can be applied also to multipoint connections.

As final case, we consider a colloquy procedure in a general packet-switched computer network. Let us state some hypotheses:

a - the nodes of the communication subnet (IMP) have dynamical routing policies

b - but for the routing policy, the IMPs are transparent to the colloquy between two HOST computers

c - packets are sequentially numbered by the sender interlocutor

d - packets are delivered with variable but finite time delays

e - packets not arrived within a predetermined time-out are to be considered lost.

With such hypotheses we are in the situation of a dialogue carried out on a packet-switched network, the only effect of is possibly to change the order of the packets sequence at the receiving node, fig. 4. The receiving interlocutor then, besides and before recognizing the input message and giving an answer, has to reorder the arriving packets following their sequence number and checking for possible packet lost. The sorting algorithm to reorder packets arrived in whatever sequence needs, besides a finite control, two Push Down Stores (PDS). The first one for storing out-of-sequence packets waiting for insertion into the right position of the receiving buffer, the other to allow a packet exchange in order to examine the contents of the PDS themselves; a very rough and partial flow chart for such an algorithm is reported in fig. 5.

A machine implementing such an algorithm is equivalent to a stack automaton, as defined in /5/, in which the contents of a single PDS is accessible in a read only mode.

It is difficult, at this point, to say which class of languages such a kind of colloquy belongs to, but we can say it lies between context-sensitive and type 0 languages. Since, for

all the languages generated by grammars which are not regular or at least not LR(k), the problems of section 3 are not decidable, we can conclude, at this stage of the research, that perhaps the range of applicability of the proposed model is restricted to dialogues on a single line or on communication networks not altering the messages (packets) sequence.

We think it would be interesting to examine even other kinds of communication links (e.g. broadcasting), broadening the field of problems the model could be of interest to.

ACKNOWLEDGMENT

I wish to thank Dr. Pierluigi Della Vigna of Politecnico di Milano for his helpful criticism and Dr. Lorenzo Giovacchini of Syntax S.p.A. for the interesting discussions we had on communication procedures.

REFERENCES

/1/ Mezzalira L., Schreiber F. "A proposal for a formal description of colloquies as a form of interaction of sequential machines", ALTA FREQUENZA, vol. XLII, n. 11, November 1973, pp. 594-603.

/2/ Le Moli G. "A theory of colloquies", 1st European workshop on computer networks, Arles April 1973, pp. 153-174, Publ. by IRIA.

/3/ Hoffman H.J. "On linguistic aspects of communication line control procedures", IBM Zurich Research Lab. RZ 345, February 1970.

/4/ Mezzalira L., Schreiber F. "Designing colloquies", 1st European workshop on computer networks, Arles; April 1973, pp. 351-363.

/5/ Hopcroft J.E., Ullman J.D. "Formal languages and their relation to automata", Addison-Wesley 1969.

/6/ Martucci R., Schreiber F. "An experiment in computer connection", Journées internationales sur les mini-ordinateurs et la transmission des données, Liége, Jan. 1975.

/7/ Mezzalira L., Schreiber F. "A microcomputerized interface for computer communication", Euromicro Newsletter, Vol. 1, n. 4, July 1975.

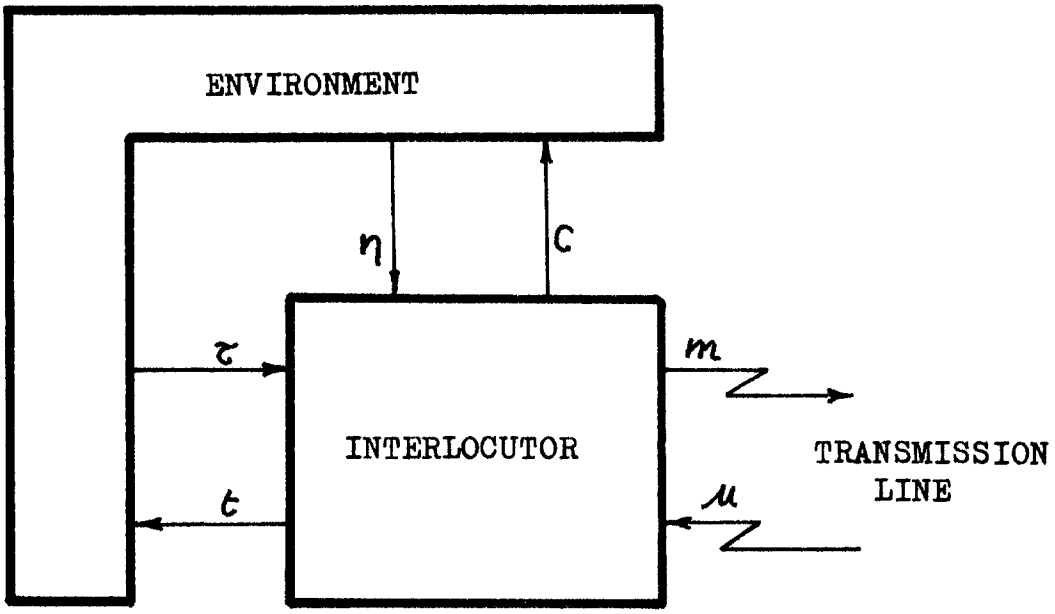
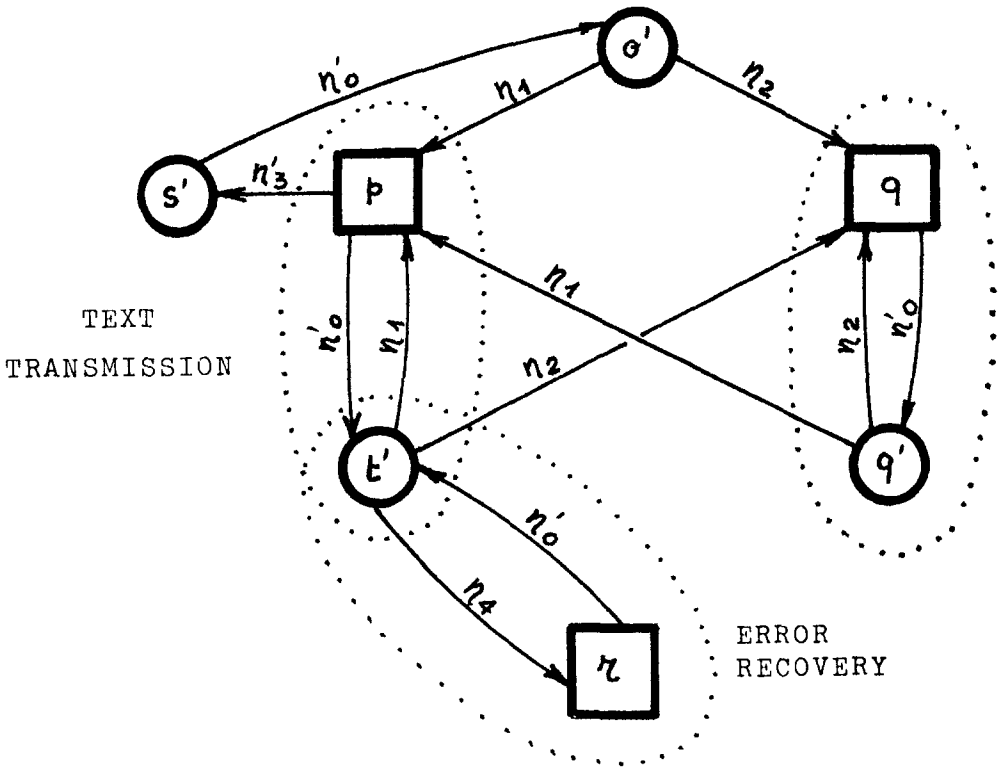

ENVIRONMENT
η
c
τ
m
INTERLOCUTOR
TRANSMISSION
LINE
t
u

Fig. 1


HOST
COMMUNICATION
NETWORK
HOST

Fig. 4



V_M = {o',p,q,q',r,s',t',Ø}
o' = request to send text
p = text acknowledged (ACK)
q = wait for sending
q' = waiting
r = request for repetition (NACK)
s' = end of text
t' = text transmission
Ø = empty message

V_S = $\{n'_o, n_1, n_2, n'_3, n_4\}$
n'_o = text to send
n_1 = ready to receive
n_2 = busy
n'_3 = no more text
n_4 = error flag
x = don't care condition

PRODUCTIONS

$$A' = |_{n'_o}\ o'B$$

$$B = |_{n_1}\ pC'\ |_{n_2}\ qD'$$

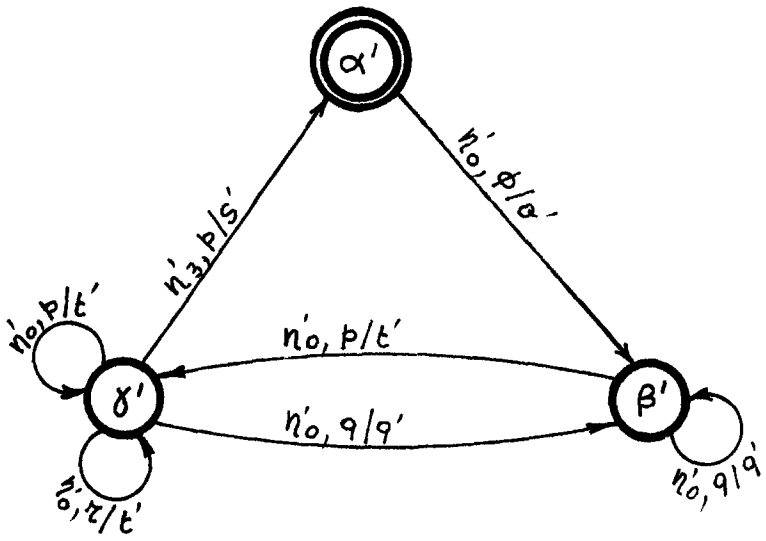
$$C' = |_{n'_o}\ t'E$$

$$D' = |_{n'_o}\ q'B$$

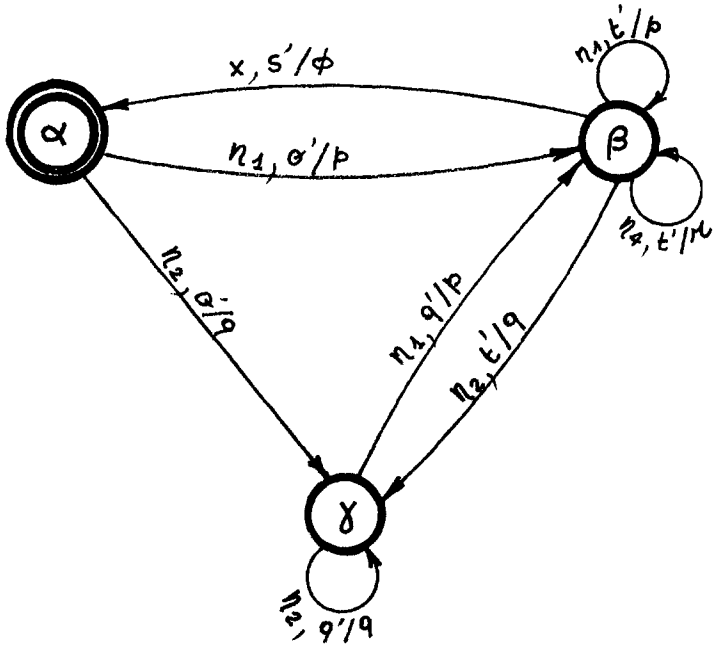
$$E = |_{n_1}\ pF'\ |_{n_2}\ qD'\ |_{n_4}\ rC'\ |_{x}\ ØB$$

$$F' = |_{n'_o}\ t'E\ |_{n'_3}\ s'A'$$

FIG. 2



SLAVE INTERLOCUTOR



MASTER INTERLOCUTOR

Fig. 3

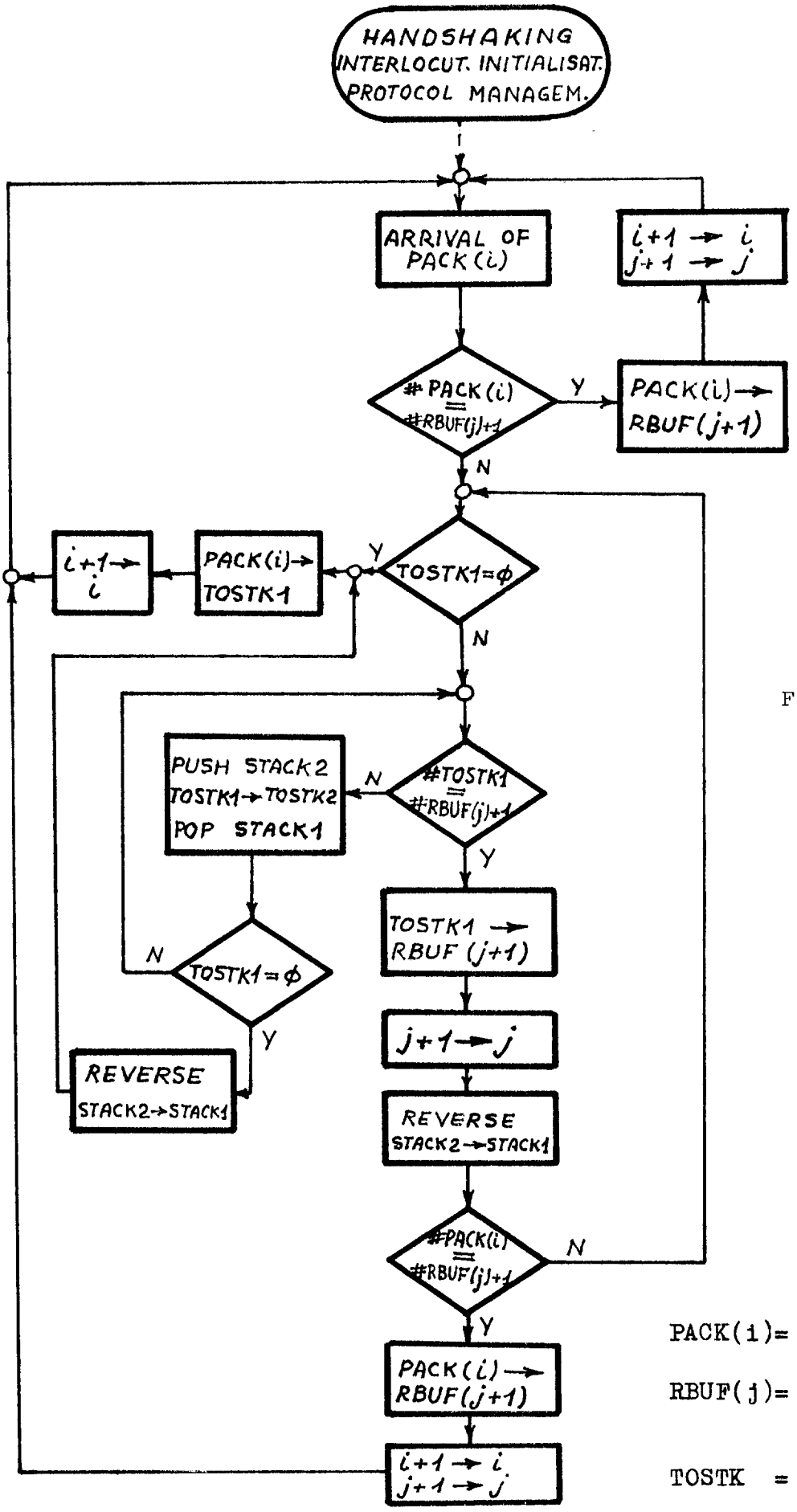


Fig. 5

PACK(i)= current packet
RBUF(j)= last filled position in rec. buffer
TOSTK = top element of PDS
\# = sequence n°