The notion and some applications of generalized initial segment

By Á. MAKAY

Introduction

In computational practice we are often confronted with situations where we have to handle strings composed of elements that can be put into two disjoint classes according to the way they influence the outcome of the operations to be performed on these strings. E.g. this happens in the case of translating higher-level programming languages. A programme written in such a language is a combination of terminators (operation symbols, parentheses, etc.) and quantities (identifiers and numbers, these are composed of letters and numerals). During translation the terminators control the compiler, while quantities play only a passive role: the translation process is not influenced e.g. by replacing each occurrence of an identifier with occurrences of another, while a similar statement for terminators is obviously false. Other types of formula-handling algorithms serve also as good examples in which such mixed-sequence situations arise.

In the present article we are going to define the notion of generalized initial segment for strings composed of elements of two disjoint sets. This notion simultaneously extends those of initial segment and of subsequence. We also present an algorithm which can decide whether or not any string is a generalized initial segment of another. The applicability in practice of the algorithm is illustrated with two examples. The second one of these deals with the Universal Decimal Classification (UDC), which is generally used in library practice. The ideas outlined in this context have been used in an information-retrieval system here working on the basis of UDC.

Definitions

Let V be a finite set and Σ a subset of V. The elements of V are usually called signs, the elements of Σ letters, and the elements of $V-\Sigma$ terminators.

Let V^* designate the set of all finite sequences which can be formed of elements of V (i.e. the free semigroup over the set V). We mean by the length $|x|$ of a sequence $x \in V^*$ the number of signs (including repeats) in x, while $\|x\|$ denotes the number of terminators (again including repeats) in x. Obviously $0 \leqq \|x\| \leqq |x|$ and $|x|$ is 0 if and only if $x=\varepsilon$, where ε is the empty sequence.

Assume $x, y \in V^*$. The notion of the generalized initial segment is defined by recursion on $\|x\|$. In case $\|x\|=0$, x is called generalized initial segment of y if

$$x=j_1 j_2 \ldots j_n, \quad y=j_1 j_2 \ldots j_n z,$$

where $n \geqq 0$, $z \in V^*$, $j_1, j_2, \ldots, j_n \in \Sigma$ (i.e. x is an initial segment of y and contains no terminators). Assume now $\|x\| = n+1$. We say that x is the generalized initial segment of y if there are sequences $x_1, x_2, y_1, y_2 \in V^*$ such that

$$x=x_1 x_2, \quad y=y_1 x_2 y_2, \quad x_2=t j_1 j_2 \ldots j_m,$$

where x_1 is the generalized initial segment of y_1, and, moreover, $m \geqq 0$, $t \in V-\Sigma$, and $j_1, j_2, \ldots, j_m \in \Sigma$.

In other words, x is a generalized initial segment of y if and only if the following holds: y contains any such sequence as is either an initial segment of x containing no terminators or is a subsequence of x such that its first element and only this is a terminator, moreover, choosing any set of such nonoverlapping sequences, these occur in y in the same order as in x.

The generalized initial segments of a sequence y consisting only of elements of Σ are simply the initial segments of y in the usual sense. Thus if $\Sigma = V$, the generalized initial segments are also initial segments. However if Σ is the empty set, then any subsequence of y is a generalized initial segment as well.

We will define an algorithm which decides for any $x, y \in V^*$ whether x is a generalized initial segment of y. By selecting Σ in a suitable way, the same algorithm can decide whether x is an initial segment or a subsequence of y.

We define the algorithm as an ALGOL-60 [1] Boolean function. The signs of the formal parameter-strings x and y are denoted by $x_1, x_2, \ldots, x_{|x|}$ and by $y_1, y_2, \ldots, y_{|y|}$, respectively. For lucidity's sake we use the nonstandard notations $|x|$, $\|x\|$, $|y|$, $\notin$ as well. The outcome of the procedure is the value *true* if x is a generalized initial segment of y, otherwise it is *false*.

```
boolean procedure GEN IN SEG (x, y); string x, y;
  begin integer array F, G [1: ||x||]; integer i, j, k;
    i:= j:= k:= 1;
    C: if j>|x| then GEN IN SEG:= true else
      if i>|y| then GEN IN SEG:= false else
        if x_j=y_i then
    begin if x_j ∉ Σ then begin F[k]:= i; G[k]:= j; k:= k+1 end;
      i:= i+1; j:= j+1; goto C
    end else if x_j ∉ Σ then begin i:= i+1; goto C end else
    begin k:= k-1; if k≠0 then
    begin i:= F[k]+1; j:= G[k]; goto C end else GEN IN SEG:= false
    end
  end GEN IN SEG;
```

Applications

The notion of the generalized initial segment and the algorithm defined above is quite widespread. Here we disregard the special cases of the initial segment and subsequence, and give two applications which are useful in information retrieval systems.

Example 1. Let us suppose that in our information system abstracts written in a natural language serve as descriptions of the content of documents. The request for retrieval is written in words or expressions consisting of several words. In the request the single words are given as root-words and in the abstracts in their inflected forms. Disregarding rootchanges, when comparing a request consisting of only one word and an abstract our task is to ascertain of the word in the request whether it agrees with the begginning of any of the words of the abstract. If the request is an expression we have to ascertain of several words whether all these words agree with the beginnings of some words in the abstract, that have the order same as given in the request. To sum up, if hyphens and spaces between words are regarded as terminators, it is to be decided whether the word or expression in the request is a generalized initial segment of any of the sentences of the abstract.

Example 2. The most widespread system of content classification of the library practice is the Universal Decimal Classification (UDC) [2]. The tremendous amount of time, money and spirit devoted to the system makes it imperative that these results and forms should be used by up-to-date automatic information systems. Below a formal definition is given for the following statement: the notion denoted by UDC number y belongs to the category denoted by UDC number x.

When setting up the UDC system mainly manual methods had been in mind, their application in automatic systems was not considered. Therefore certain transformations for computer information systems are required. This can be done by decomposing UDC-numbers into parts (general subject, facets separately) [3]. Another way (which we follow here) is the mechanical transformation of the UDC-numbers. In particular we omit redundant signs (.,right-hand",)) and, moreover, instead of sings consisting of several characters we use only a single character (−0,.0,.00,(0,(=). The schematic description [1] gives all these transformations. However, the categories of the general connective (+), the inclusive connective (/) and the relative connective(:) are left undefined: instead of them in the information system various UDC-numbers can be used or there are several other ways of excluding them [2].

⟨UDC-number⟩ ::= ⟨general subject⟩ ⟨subordinate facets⟩ ⟨facets⟩

⟨general subject⟩ ::= ⟨empty⟩|⟨decimal number⟩|⟨synthetic connectiv⟩

⟨synthetic connectiv⟩ ::= ⟨general subject⟩'⟨decimal number⟩

⟨subordinate facets⟩ ::= ⟨empty⟩|⟨subordinate facet⟩|⟨subordinate facets⟩ ⟨subordinate facet⟩

⟨subordinate facet⟩ ::= ⟨special auxiliary⟩|⟨point of view⟩

⟨special auxiliary⟩ ::= −⟨decimal number⟩|.0⟨non-0 decimal number⟩

⟨point of view⟩ ::=.00⟨decimal number⟩

⟨facets⟩ ::= ⟨empty⟩|⟨facet⟩|⟨facets⟩ ⟨facet⟩

⟨facet⟩ ::= ⟨language⟩|⟨form of work⟩|⟨place⟩|⟨race⟩|⟨time⟩

⟨language⟩ ::== ⟨non-0 decimal number⟩

⟨form of work⟩ ::= (0 ⟨decimal number⟩

⟨place⟩ ::=(⟨non-0 decimal number⟩

⟨race⟩ ::= (=⟨decimal number⟩

⟨time⟩ ::=″⟨decimal number⟩

⟨decimal number⟩ ::=⟨digit⟩|⟨decimal number⟩ ⟨digit⟩

⟨non-0 decimal number⟩ ::= ⟨non-0 digit⟩|⟨non-0 decimal number⟩ ⟨digit⟩

⟨digit⟩ ::= 0|⟨non-0 digit⟩

⟨non-0 digit⟩ ::= 1|2|3|4|5|6|7|8|9

⟨empty⟩ ::=

If the UDC-numbers are decomposed in parts in an information system (general subject, facets), the requests are also to be built of these units. Therefore, if we want to formulate the simultaneous existence of two elements, we are in need of the logical operation of "conjunction", which is a notion unknown in library practice.

The form of formulating requests is brought closer to the UDC system when complete UDC-numbers are used as units. In view of the above syntactic rules it is easy to see that if the numeral digits are regarded as letters and all the other signs as terminators, the notion denoted by an UDC number y belongs to the category denoted by an UDC number x if and only if x is a generalized initial segment of y.

Понятие обобщенного начального сегмента и некоторые его применения

В вычислительной технике довольно частый случай, когда занимаемся такими последовательностями, которые с точки зрения обработки, построены из элементов двух друг от друга хорошо различиных множеств. Такое же положение тогда, когда, например, изготавливаем транслятор языка для программировании более высокой степени, в этом случае числа и буквы служат для обозначения «величин», а последовательность скобок и знаков операций содержит структуру програмы.

Статья, с помощью понятия «обобщенного начального сегмента», одновременно и обобщает понятие начального сегмента и частичной последовательности таких «смещенных» последовательностей, а также публикуется алгоритм решение того, что последовательность является ли обобщенным начальным сегментом другой последовательности.

Вторая из двух сообщенных возможностей применения обращает внимание на свойство алгоритма, который может быть применен в системах обратного отыскания информации в системе Универсальной Десятичной Классификации (УДК).

RESEARCH GROUP ON MATHEMATICAL LOGIC
AND THEORY OF AUTOMATA OF THE HUNGARIAN
ACADEMY OF SCIENCES,
SZEGED, HUNGARY

References

[1] NAUR, P., et al., Revised Report on Algorithmic Language ALGOL 60, *Comm.* ACM., v. 6, 1963, pp. 1—17.
[2] *Universal Decimal Classification*, Auxiliary tables (in Hungarian), Közgazdasági és Jogi Könyvkiadó, Budapest, 1969.
[3] FREEMAN, R. R. & P. ATHERTON, *File organization and search strategy using the universal decimal classification in mechanized reference retrieval systems*, American Institute of Physics, Report No. AIP/UDC-5, 1967.

(Received March 11, 1972)