Serdica

Bulgariacae mathematicae publicationes

Сердика

Българско математическо списание

CAA — A TOOL FOR DATA BASE DESCRIPTION*

JACEK BANKOWSKI

The paper presents a mathematical model of data base. The fundamental elements of this model are: the set of user data names (N) and the set of access algorithms (A). There is defined the set of rules which allows to construct for every $n \in N$ a chain of access algorithms (CAA). The execution of operations from CAA allows to determine the data which is denoted by n. Every data base may be described in terms of that model.

1. Introduction. Data base may be, in general, considered as a blackbox. Whenever a data name is referenced on its input, the value of this particular data appears on its output. Such being the case, the blackbox represents itself all properties of data base at a given instant. Obviously, there exists a mapping from the set of user names into the set of data values. This mapping may be described in terms of an access algorithm. Such approach is quite correct but is impractical, because the description of data base in the form of — saying it in another way — access program would be in most cases practically unreadable. To solve this problem, it is proposed to split the access algorithm into smaller parts, which may be called and executed in a sequential way, forming a chain of access algorithms (CAA).

It is proposed that every data is an ordered pair $\langle t, s\rangle$, where t is a data name used for its identification and s is a pointer to algorithm, which transforms data name into new pair $\langle t', s'\rangle$. This definition of data is a generalization of Knuth's concept where data is an ordered pair ⟨name, value⟩. It should be noted that t' may be different from any user data name. Such names are used to denote the data that are inaccessible to the user.

2. Basic notions. Let

V — any finite alphabet,

V^* — set of all finite words over V,

$T \subset V^*$,

S — a finite set called pointer set. Then $D \subseteq T \times S$ is a data set.

Let A be the class of all deterministic algorithms which define partial functions $a: T \to D$ and p be the algorithm numbering function $p: S \to A$. The operation C of data successor is defined as:

$$C(d) = C(\langle t, s\rangle) = p(s)(t) = a(t) = d' = \langle t', s'\rangle$$

$$(d, d' \in D;\ t, t' \in T;\ s, s' \in S;\ a \in A).$$

Hence $C(d)$ is a new data which is computed by the algorithm pointed by s, operating on the input word t.

---

* Delivered at the Conference on Systems for Information Servicing of Professionally Linked Computer Users, May 23-29, 1977, Varna.

As a is, in general, a partial function (there exists such t that $t \notin \mathrm{Dom}(a)$) there may exist data with no successor. Such data will be referred to as data with no interpretation within data base. The pointer s_1 is distinguished in the set S and $C(\langle t, s_1\rangle)=\langle t, s_1\rangle$ for every $t \in \mathrm{Dom}(a_1)$, $p(s_1)=a_1$.

Intuitively, algorithm a_1 repeats the input data and enables to recognize when the CAA should be terminated.

3. Data base description and data generation. Data base is fully described by an ordered quintuple: $B=\langle N, S, P, \mathscr{A}, s_1\rangle$, where $N \subset T$ is the set of user data names and $\mathscr{A} \subset A$; other elements were defined formerly. Data base description may be used to generate the set D (all data existing in data base) in the following way:

1. $D_0=\{d: \bigvee_{n \in N} \bigvee_{s \in S} d=\langle n, s\rangle \cap C(d) \text{ exists}\}$, i. e. $D_0=(N \times S) \cap (\mathrm{Dom}(C))$.

2. $D_{j+1}=\{d': \bigvee_{d \in D_j} d'=C(d) \wedge C(d') \text{ exists}\} \setminus \bigcup_{k=0}^{j} D_k$, i. e. $D_{j+1}=(C(D_j)$

$$\cap (\mathrm{Dom}(C))) \setminus \bigcup_{k=0}^{j} D_k.$$

3. Process is terminated in j steps if $D_j \neq \emptyset$ and $D_{j+1}=\emptyset$. Then

$$D=\bigcup_{k=0}^{j} D_k.$$

D_0 is the set of data which are formed from data names accessible to users. The sets D_{j+1} include successors of data in D_j, provided that those successors are different from all data in the sets $D_0 \cdots D_j$.

It may not be stated that the process described above is, in general, finite. As in all practicall cases the set of data names is finite it is evident that in such a case the process is terminated in a finite number of steps.

Data of the type $\langle t, s_1\rangle$ are interpreted as simple data, i. e. the word t is interpreted directly as the value of simple data.

It should be noted that generated in this way set D includes all data that have interpretations within existing data base. In particular, for an empty data base there will be no simple data in the set D which does not imply that it is itself empty.

Two equivalent, from user's point of view, data bases may have descriptions using different algorithms and, in particular, similar algorithms may differ in domains of functions which they are computing.

The number of algorithms used in the description of data base depends mainly upon the application, so it does not seem wise to define algorithm other than a_1 at moment. Nevertheless, in many applications algorithm a_2 such that $C(\langle t, s_2\rangle)=\langle t', s_1\rangle$ for every $t \in \mathrm{Dom}(a_2)$ seems to be useful. In this case t might be interpreted as the word defining address(es) and length(s) of the field(s) in which simple data $\langle t', s_1\rangle$ is stored.

4. Conclusions. Since no conditions concerning the set of algorithms were assumed it therefore follows that every data base may be described in terms of the presented model. It should be pointed out that the description represents a static situation in the data base. Inserting (deleting new items to) from data base results in changing of its description. If the set $\mathscr{A}$ is properly formed then this change may be kept local, i. e. restricted to one algorithm and even to its domain. E. g. a new occurrence of a record accessed by key will change only the domain of the algorithm which locates the record by key. For practical applications a slight modification of the model seems to be reasonable. Namely, an algorithm a_0 (with corresponding pointer) is introduced such that $C(\langle t, s_0\rangle)=\langle t, s_0\rangle$ for every t. Then, the set of algorithms $\mathcal{A}$ is transformed to $\mathcal{A}'$ in such a way that

$$C'(\langle t, s\rangle)=\begin{cases} C(\langle t, s\rangle) \text{ if } t \in \mathrm{Dom}\,(a),\ (a=p(s)), \\ \langle t', s_0\rangle \text{ otherwise.} \end{cases}$$

The process of data evaluating may be performed in the following way:

1. $d: =\langle n, s\rangle$

(the value of s must be given as, in general, there may exist different access algorithms for a given data name n).

2. While the pointer of $d>1$ do $d: =C(d).2'.d: =C(d)$.

3. If the pointer of $d=1$ then the data name of d is the value of the data named n.

4. If the pointer of $d=0$ then the data named n is inaccessible in existing data base.

5. STOP

The main objective of the research currently done at ISTEI is the formulation of the set $\mathcal{A}$ that would be convenient in practical applications.

REFERENCES

1. D. E. Knuth. The Art of Computer Programming, vol. 1. Addison-Wesley, 1968.
2. E. F. Godd. A relation Model of Data for Large shared data Banks. *Communs. ACM,* **13,** 1970, No 6.

Institute for Scientific, Technical and Economic Information
00-950 Warszawa, P. O. Box 123, Poland

Received 11. 9. 1977