Access models and access times of relational data base systems

Theo Härder, TH Darmstadt, West-Germany

1. Introduction

The relational model according to E.F. Codd (Co 70) is characterized by explicit representation of relations, which are confined to the aggregation of attribute names on the user interface. No information concerning the physical realization of the relations is needed. A salient feature of the model is the definition of non-procedural data base sublanguages (Co 71). The completeness of the relational model on the logical level allows an implementation which is independent from distinct storage structures and access paths. In particular, there is no need to extract additional information from the access paths in order to complete the logical data model. These characteristics guarantee a high degree of data independence.
On the contrary , hierarchical and network models don't meet the important requirement concerning the completeness of the data model on the logical level. Distinct relations are often represented by suitable storage structures and implementation techniques, for example, pointer, repeating group, physical contiguity. As a consequence there is a need of procedural data manipulation languages demanding detailed knowledge of existing access paths from the user. The programmer navigates through the data structures (Ba 73) and selects the searched records in a "one record at a time" manner.
It is pointed out, what access facilities exist in relational data base systems and how the selection capability of non-procedural languages, often called a "multirecord at a time" logic, can be utilized in an optimal way by the data base system. The access time behavior of relational data base systems is investigated in a deterministic worst case analysis under consideration of real storage media (moveable head discs). The comparison of times accessing sets of records with single record search of navigatorial languages yields a substantial performance enhancement.

2. Model assumptions

Let A_1, A_2,..., A_n be nonempty sets, not necessarily distinct. A subset R of the Cartesian product $A_1 \times A_2 \times \ldots \times A_n$ is called a relation, and A_i is designated a domain or attribute of R. Let r be an element of R, then r is a n-tuple $(r_1, r_2,\ldots,r_n)$ where r_i belongs to A_i. The attribute or minimal group of attributes which guarantees the uniqueness of tuples is called primary key. The physical occurrences of a relation and of a tuple are denoted respectively as file and record.

We assume that the N_{REC} records of the file are randomly stored in secondary memory and are not clustered according to any criteria. The records are uniformly distributed over neighbouring cylinders of a disc, and access to them is performed with equal probability for all transactions. In case of random processing both assumptions yield the worst case conditions with respect to access arm movement.

We distinguish two classes of access paths. Obtaining access via primary key one record is found at the most. In comparison to this a secondary key qualifies n records. The corresponding access path is called secondary index. A primary key access path is necessary for an efficient support of batch processing. The following three operation primitives are important for its implementation:

- random access to any record,
- sequential processing of all records in a particular order,
- efficient maintenance.

To support the processing of ad hoc queries secondary indices are suitable. Non-procedural languages allow the selection of records based on their contents rather than according to the relative position within a storage structure. Consequently, the languages possess a high selection capability. This advantage should be preserved by appropriate implementation techniques for access paths. By separating access paths from data records the subsetting of qualified tuples is manageable merely within the access path data. The system can then choose the optimal way to allocate the records on external storages. By merging of records and access path data in one data stream only a few factors are open for optimization in evaluating queries, because the access sequence is determined by the storage structure. Therefore we assume an index implementation by inverted lists, which may contain primary keys, tuple identifiers (tids) or physical addresses.

The effectiveness of search operations is critically affected by the number of accesses to external storage media. If all attributes specified in a query are inverted, the target list T(L) constructed by merge operations contains exactly the entries for the qualified tuples. In other cases only a list S(L) comprising the entries of a superset of tuples is found. If there are no means to restrict the set of tuples according to the query predicate, the total file must be scanned.

3. Models for random processing

The time requirement of random processing depends mainly on the number of qualified records (N_{qual}) or accesses of a transaction. In order to describe the physical allocation of records we introduce the hit ratio of a transaction defined by

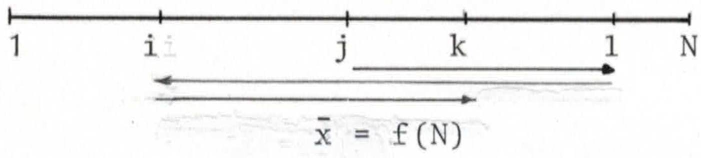
$$HR = \frac{N_{qual}}{N_{REC}} \cdot 100\% \qquad (3.1)$$

The total cost for random access increases according to the hit ratio, while it remains constant in case of scanning in a sequential process. It follows that random processing is not advantageous, if a critical hit ratio HR_o is exceeded. Even if there exists an access path, a total scan is superior.

3.1 Random processing according to random and random-sorted lists

Random access to specified records on moveable head discs is distinguished by two different cases.

1) The access arm motions x_i are not coordinated. Each cylinder of the file has equal prohability of being accessed by any request. The average access motion is a function of the number of occupied cylinders N.

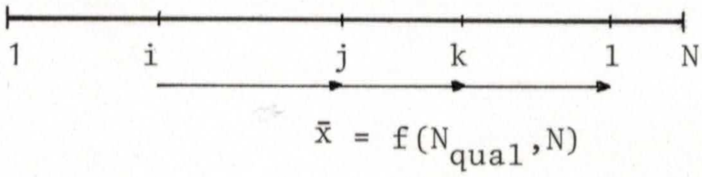


$x_1 = jl$
$x_2 = li$
$x_3 = ik$

An average access time $\bar{t}_r$ results independent of the hit ratio. The average access time for the processing according to random lists is given by

$$\bar{t}_{HR} = N_{qual} \cdot \bar{t}_r \qquad (3.2)$$

2) Since all record addresses are known to the data base system at the begin of a random transaction processing, the accessing pattern to the external storage may be determined in such a manner that the access motions are initialized only in one direction and thus become minimal. The average access motion is a function of N and N_{qual}.



$x_1 = ij$
$x_2 = jk$
$x_3 = kl$

The average access time $\bar{t}_{rs}$ to a record diminishes with increasing hit ratio.
The total access time for the processing according to random-sorted lists is given by

$$\bar{t}_{HR} = N_{qual} \cdot \bar{t}_{rs} (N_{qual}) \qquad (3.3)$$

3.2 Single and parallel access

Figure 1 shows the record by record allocation of the qualified set. A characteristic for this kind of processing is the necessity to start the access to the succeding record after the terminated transfer of the record already searched. This property of a data base system is also called "one record at a time"logic. During the single record processing no strategies can be applied by the data base system to decrease the total access time.
The initialization of access, the sequence of access arm motions, the look up and transfer of records being dependent on data organization are not further specified in figure 1.
The seek time can be reduced for large files extending over several discs. When search requests for several records are known to the system and the set of addresses represents distinct devices, access motions on different devices can be initialized and achieved concurrently. Therefore it is an essential requirement that the searched records are qualified by contents with appropriate data base sublanguages, i.e., predicate calculus type of language. Correspondingly, this is denoted as "multi record at a time" logic.
The individual time components of access - seek,rotational delay and data transfer - can be reduced by overlapping in case of suitable scheduling of resources. It is assumed that the data transfer is performed strictly sequential because of the availability of one channel. In case of disc units of type IBM 2311 and IBM 2314 only seeks can be overlapped,because the channel is occupied by a disc unit during the rotational delay. For the IBM 3330 seeks and portions of the rotational delay may be synchronized, because the channel is only busy short before data transfer due to rotational position sensing. Even transfer of records from the devices to main memory may be overlapped to the extend that the devices may be attached to different channels. In order to perform worst case estimations only the overlapping of seeks is considered.
The logical flow diagram in figure 1 shows only the principal scheme of parallel processing. The use of the terms FORK, JOIN and TERMINATE follows the terminology of (An 65) for parallel processes. The FORK statement starts n parallel processes Pi (i= 1,2,...,n), which perform the seeks on n devices and position the heads at the referenced cylinders. The JOIN statement deactivates

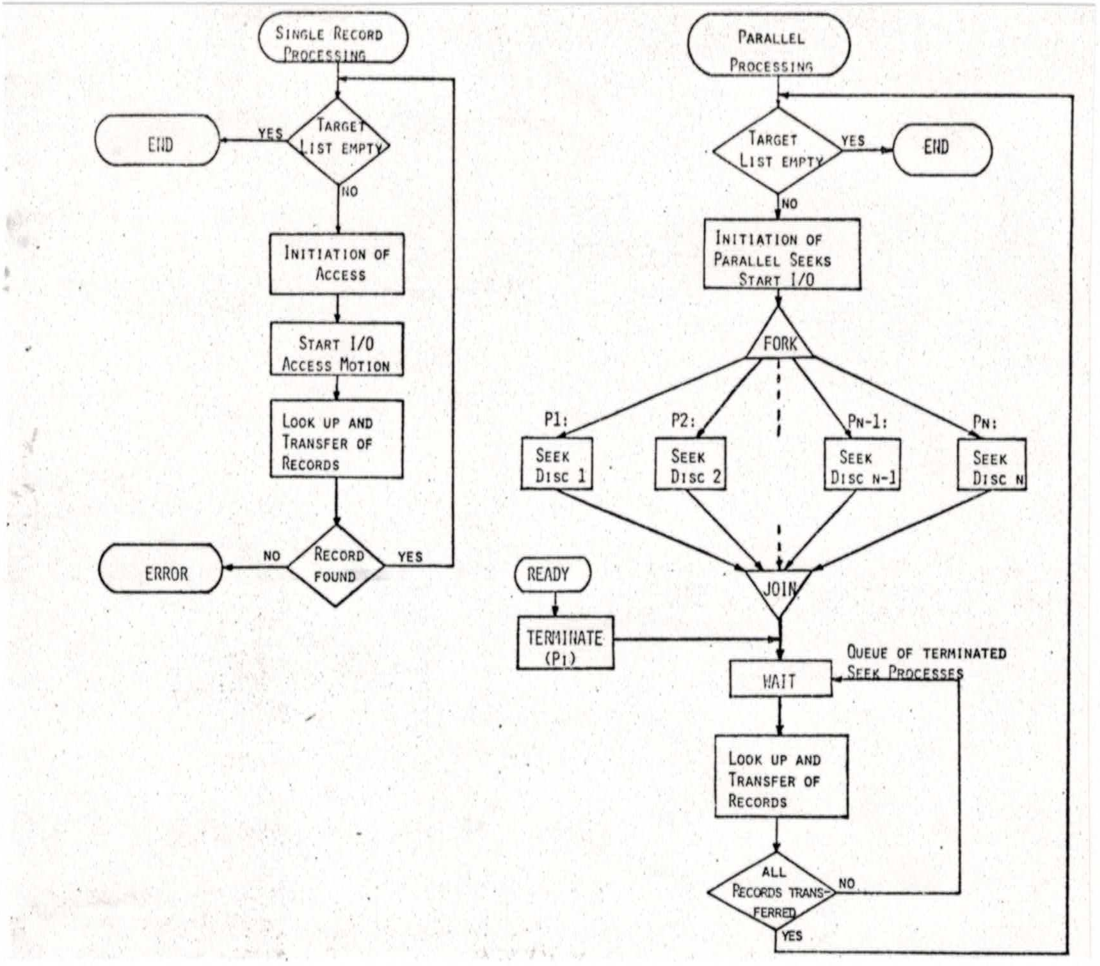


Figure 1: Logical flow diagram for single record and parallel processing

parallel processes, but it works as an AND-function, that is, the successor statement of JOIN is not reached before all processes are finished. In oder to start the look up and transfer of the first record as soon as some process P_i has done the positioning, each process can be finished by a TERMINATE statement independent of the other concurrent processes. This is achieved by an exit to the label READY after successful termination. The WAIT statement characterizes a queue of I/O-requests for devices on which the seek is completed. Since only one channel is available, the queue must be processed sequentially.

In case of parallel access two or more processes are initiated at a time and work completely independent and parallel in time. First of all the question arises, when the first process is finished. This means that the expected value for traversing a minimal number of cylinders has to be computed as a function of n discs. Thus, the minimal average seek time is obtained. Immediately after the termination of the minimal seek the look up and transfer of the first of n records can be started. To examine the question, whether delays can possibly occur during this sequential phase of processing, the expected value of the maximal seek is computed as a funtion of n discs. Delays happen when some processes have not finished their seek and the sequential process achieving look up and transfer finds an empty queue.

To solve these problems the "parallel combination" of random variables (IBM1) is used. The variables $X_1, X_2, \ldots, X_n$ and their distribution functions $F_1(x), F_2(x), \ldots, F_n(x)$ be stochastic independent from one another. Hence, the distribution functions of minima and maxima of all n random variables are given by

$$C_{\min}^{(n)}(x) = 1 - \left[1 - F_1(x)\right] \cdot \left[1 - F_2(x)\right] \cdot \ldots \left[1 - F_n(x)\right] \tag{3.4}$$

$$C_{\max}^{(n)}(x) = F_1(x) \cdot F_2(x) \cdot \ldots \cdot F_n(x). \tag{3.5}$$

If all distribution functions are equal, equations (3.4) and (3.5) can be reduced to (3.6) and (3.7). This holds in case of one device type, uniform distribution of all records and uniform access frequencies to all records.

$$C_{\min}^{(n)}(x) = 1 - \left[1 - F(x)\right]^n \tag{3.6}$$

$$C_{\max}^{(n)}(x) = \left[F(x)\right]^n \tag{3.7}$$

Considering equations (3.6) and (3.7) the expected values $\bar{x}_{\min}^{(n)}$ and $\bar{x}_{\max}^{(n)}$ will be derived in case of processing according to random lists. The density function f(x) of the distribution function F(x) is given by (We 73)

$$f(x) = \begin{cases} \frac{1}{N} & \text{for } x = 0 \\ \frac{2(N-x)}{N^2} & \text{for } 1 \leqslant x \leqslant N-1 \end{cases} \tag{3.8}$$

The distribution function is obtained by summing the density function over x:

$$F(x) = \frac{1}{N} + \frac{2}{N^2} \cdot \sum_{\xi=1}^{x} (N-\xi) = \frac{1}{N^2} \left[N + 2 \cdot N \cdot x - x^2 - x\right].$$

Hence, equations (3.6) and (3.7) yield

$$C_{\min}^{(n)}(x) = 1 - \frac{(N-x)^n \cdot (N-x-1)^n}{N^{2n}} \tag{3.9}$$

$$C_{\max}^{(n)}(x) = \frac{1}{N^{2n}} \cdot (N+2 \cdot N \cdot x - x^2 - x)^n. \tag{3.10}$$

For the computation of the mean values $\bar{x}_{\min}^{(n)}$ and $\bar{x}_{\max}^{(n)}$ the density functions of the joint distribution functions (3.9) and (3.10) are needed:

$$P_{o\min}^{(n)} = 1 - \frac{(N-1)^n}{N^n}; \quad P_{o\max}^{(n)} = \frac{1}{N^n}$$

$$P_{x\min}^{(n)} = C_{\min}^{(n)}(x) - C_{\min}^{(n)}(x-1) = \frac{1}{N^{2n}} \cdot (N-x)^n \cdot \left[(N-x+1)^n - (N-x-1)^n\right]$$

$$P_{x\max}^{(n)} = C_{\max}^{(n)}(x) - C_{\max}^{(n)}(x-1) = \frac{1}{N^{2n}} \cdot \left[(N+2Nx-x^2-x)^n - (2Nx-N-x^2+x)^n\right].$$

It follows:

$$\bar{x}^{(n)}_{min} = \sum_{x=1}^{N-1} x\cdot p^{(n)}_{xmin} = \frac{1}{N^{2n}}\cdot\sum_{x=1}^{N-1} x\cdot\Big[(N-x)^{n}\cdot\big\{(N-x+1)^{n}-(N-x-1)^{n}\big\}\Big]$$

$$= \frac{1}{N^{2n}}\sum_{k=1}^{N-1}\big[k\cdot(k+1)\big]^{n} \qquad (3.11)$$

$$\bar{x}^{(n)}_{max} = \sum_{x=1}^{N-1} x\cdot p^{(n)}_{xmax} = (N-1) - \frac{1}{N^{2n}}\cdot\sum_{x=o}^{N-2}\big[N(2x+1)-x(x+1)\big]^{n}. \qquad (3.12)$$

The following special cases are given, for example, with n = 1

$$\bar{x}^{(1)}_{min} = \bar{x}^{(1)}_{max} = \frac{N^2-1}{3N} = \frac{N}{3}\cdot\left(1-\frac{1}{N^2}\right)$$

and with n = 2

$$\bar{x}^{(2)}_{min} = \frac{N}{5}\cdot\left(1-\frac{5}{3N^2}+\frac{2}{3N^4}\right), \qquad \bar{x}^{(2)}_{max} = \frac{7}{15}\,N\cdot\left(1-\frac{5}{7N^2}-\frac{2}{7N^4}\right).$$

Density and distribution functions for the processing according to random-sorted lists are derived in detail in (Hä 75). Here, the results are summarized. If m accesses are performed on a single disc then the density function for an uniform distribution of accesses over N cylinders is

$$g(m,x) = \frac{\binom{N+m-x-2}{m-1}}{\binom{N+m-1}{m}} \qquad \text{for } 0 \leqslant x \leqslant N-1$$

and the corresponding distribution function

$$G(m,x) = 1 - \frac{\binom{N+m-x-2}{m}}{\binom{N+m-1}{m}} \qquad \text{for } 0 \leqslant x \leqslant N-1$$

With regard to equations (3.6) and (3.7) the joint distribution functions of minima and maxima are

$$G^{(n)}_{min}(m,x) = 1 - \left[\prod_{k=1}^{m}\frac{(N+m-x-k-1)}{(N+m-k)}\right]^{n} \qquad (3.13)$$

$$G^{(n)}_{max}(m,x) = \left[1 - \prod_{k=1}^{m}\frac{(N+m-x-k-1)}{(N+m-k)}\right]^{n} \qquad (3.14)$$

For the computation of the expected values $\bar{x}^{(n)}_{min}(m)$ and $\bar{x}^{(n)}_{max}(m)$ the joint density functions of the n parallel random variables are needed, which are derived by differentiation of equations (3.13) and (3.14) with respect to x. The evaluation of finite difference quotients and the mean value computation must be achieved numerically.

$$\bar{x}^{(n)}_{min}(m) = \sum_{x=o}^{N-1} \frac{\Delta C^{(n)}_{min}(m,x)}{\Delta x} \cdot x = \sum_{x=o}^{N-2} \left[\prod_{k=1}^{m} \frac{(N+m-x-k-1)}{(N+m-k)} \right]^{n} \qquad (3.15)$$

$$\bar{x}^{(n)}_{max}(m) = \sum_{x=o}^{N-1} \frac{\Delta C^{(n)}_{max}(m,x)}{\Delta x} \cdot x = N-1- \sum_{x=o}^{N-2} \left[1- \prod_{k=1}^{m} \frac{(N+m-x-k-1)}{(N+m-k)} \right]^{n} \qquad (3.16)$$

In the special case n = 1 the following holds:

$$\bar{x}^{(1)}_{min}(m) = \bar{x}^{(1)}_{max}(m) = \frac{N-1}{m+1} \quad .$$

3.3 Sequential processing

If no access path exists for a particular transaction or if the target list comprises so many entries that the critical hit ratio HR_o is exceeded, the total file should be scanned serially. This sequential scan can be achieved in logical order according to the access path of the primary key or in the physical order of the records. Since the order of accessing records in a normalized relation does not matter , a physical sequential processing will be chosen for economic reasons. The computation of the processing time of a total scan is based on the assumption that the physical blocks are transferred continuously from the external storage device at full channel speed. Applying synchronized buffer techniques the internal computing times for the selection of qualified records may be overlapped. Therefore it is assumed that speed and availability of the central processor do not affect the sequential scan. This serial scan may eventually be delegated to a peripheral specialized computer (Ka 75). In this model the total processing time depends linearly on N_{REC}:

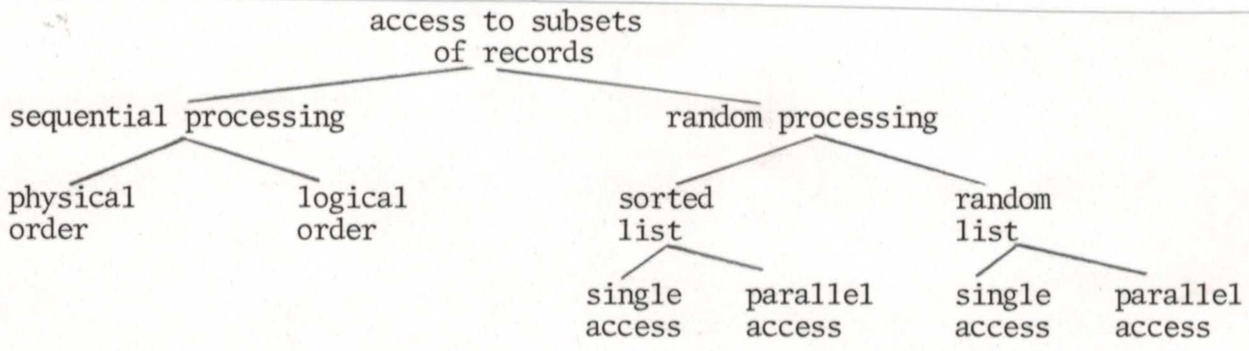
$$\bar{t}_{tot} = K_{seq} \cdot N_{REC}$$

K_{seq} is derived in the appendix. Considering (3.2) resp. (3.3) the hit ratio HR_o is determined by the following condition:

$$\bar{t}_{HR_o} = \bar{t}_{tot}$$

3.4 Access facilities in relational data base systems

Comprehendingly, it should be stressed that the strict separation of non-procedural language from aspects of access provides to the DBMS an additional degree of freedom. It enables the system to access subsets of records. Furthermore, it is important to choose an implementation technique for access paths, which separates access information from data records. The following classification scheme points out these advantages.

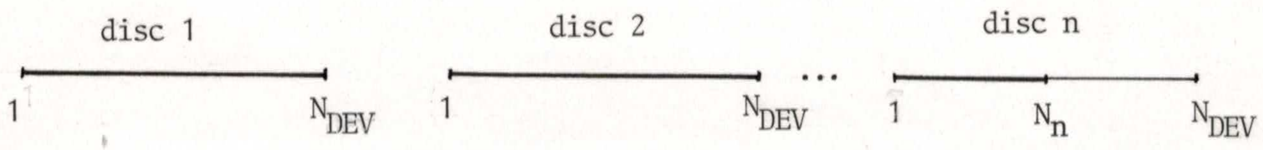


In case of a non-procedural language usually all facilities are available. Random processing based on a navigating language is only conceivable with regard to the classification scheme called single record access according to random lists. It is immaterial whether the access paths are embedded or not. In general, sequential processing must be performed in logical order for hierarchical data structures, because implicit relations represented by storage structures can only be recognized if the right access path is followed. If normalized relations are implemented without such implicit representation techniques, a faster physical sequential scan can be performed.

4. Access times for direct addressable files

For the access models outlined in section 3 the access times to be obtained are computed without queueing considerations. It is assumed in this case that sets of records are allocated. As an illustrative example we restrict ourselves to the very simple case of direct addressable files. Unique addresses are derived from the primary keys stored in the target list by an key to address transformation, which can be accessed without sequential search in a disc's track. Because overflow can't happen, each record is found in one access cycle. The same situation exists if actual physical addresses are stored in the target list.

The following allocation of a file on n devices is chosen for the time considerations:



The details are fully explained in the appendix. The occupied cylinders of the file N_{CYL} are stored on n discs without gaps. The covered cylinders of the n-th disc are given by

$$N_n = N_{CYL} - (n-1)\cdot N_{DEV} ,$$

where N_{CYL} is derived by

$$N_{CYL} = \frac{1}{\beta}\cdot\frac{N_{REC}}{b\cdot N_B\cdot T_{CYL}} \tag{4.1}$$

4.1 <u>Access according to random lists.</u>

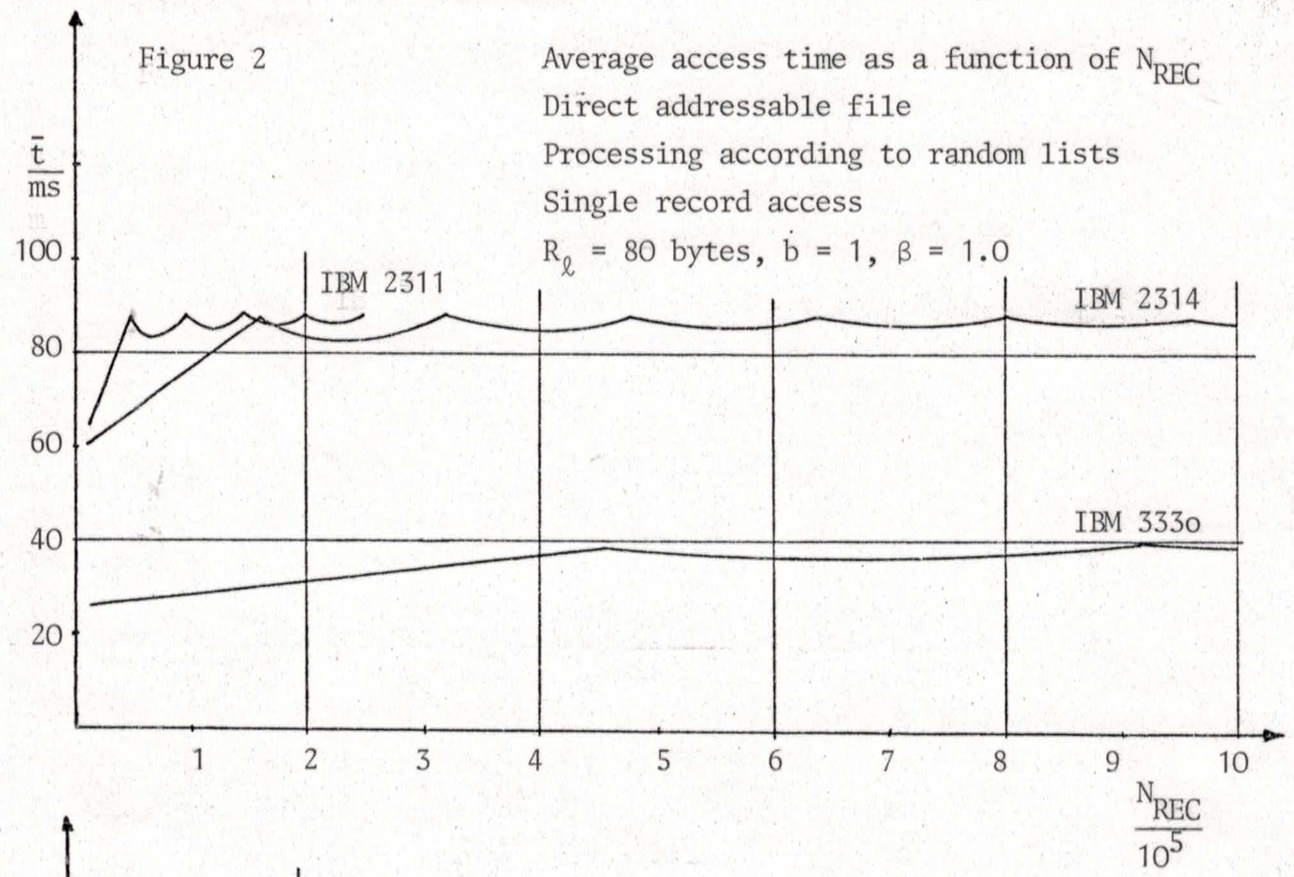
Considering equations (3.11) and (A.3) the average access time to a record is given with regard to equation (A.2) by

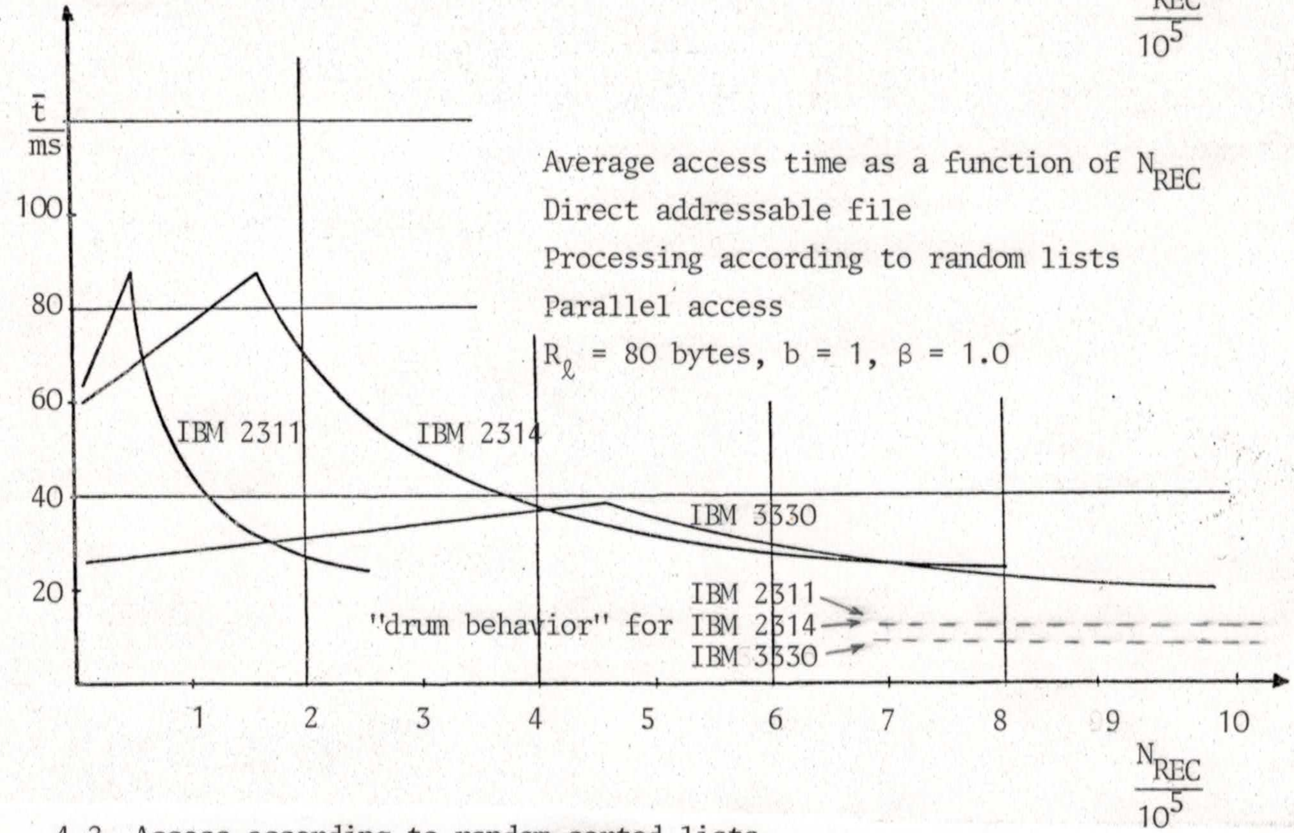
$$\bar{t}=\frac{1}{2}t_{rev}+\frac{1}{u}\cdot b\cdot R_\ell+(n-1)\cdot\frac{N_{DEV}}{N_{CYL}}\cdot(t_o+s_o\cdot\bar{x}^{(1)}_{min}(N_{DEV}))+\frac{N_n}{N_{CYL}}\cdot(t_o+s_o\cdot\bar{x}^{(1)}_{min}(N_n)) \tag{4.2}$$

In case of parallel access ($N_{CYL} > N_{DEV}$) only seek times to one fully occupied disc must be taken into consideration because of the chosen access strategy and the uniform distribution of access. Hence,

$$\bar{t} = \frac{1}{2}\cdot t_{rev} + \frac{1}{u}\cdot b\cdot R_\ell + \frac{N_{DEV}}{N_{CYL}}\cdot(t_o + s_o\cdot\bar{x}^{(n)}_{min}(N_{DEV})) \quad \text{for } N_{qual} > n \tag{4.3}$$

In figure 2 the equations (4.2) and (4.3) are evaluated as function of N_{REC} for given hardware constants and storage parameters. The influence of a load factor $\beta > 1$ can be estimated by multiplying the abscisse values with β and leaving the shape of the curves unaffected. The parameter b and N_B are linked together by equation (A.1). The choice of a blocking factor $b > 1$ improves the utilization of storage, because the number of necessary hardware gaps is reduced. Hence, the scale on the abscisse axis is condensed. Since larger physical records are to be transferred, the component of transfer time increases linearly with the blocking factor. As it is illustrated in figure 2 "drum behavior" can be reached approximately on discs in case of parallel access to large files, viz., by applying a suitable scheduling policy the component of the average seek time is nearly eliminated.





4.2 Access according to random-sorted lists

For this kind of processing the access mechanism is moved only in one direction. The average time required to position the head at the specified cylinder is dependent on the number of total accesses. Consequently, it is a function of the hit ratio. If $N_{qual} < N_{CYL}$, then a seek is required for each access. If at least one record is searched in each cylinder ($N_{qual} \geq N_{CYL}$), then the factor N_{CYL} / N_{qual} can be taken as the probability of a cylinder to cylinder positioning. Hence, we have to distinguish several cases to compute the average access times. The number of hits on a disc is approximately given by $m = \lceil N_{qual}/n \rceil$ or under consideration of equation (3.1) by

$$m = \left\lceil \frac{HR \cdot N_{REC}}{100 \cdot n} \right\rceil .$$

The single access is described by

$$\begin{aligned} \bar{t}(HR) &= \frac{1}{2} \cdot t_{rev} + \frac{1}{u} \cdot b \cdot R_{\ell} + \frac{N_{DEV}}{m} \cdot t_{s\,min} && \text{for } m \geqslant N_{DEV} \\ \bar{t}(HR) &= \frac{1}{2} \cdot t_{rev} + \frac{1}{u} \cdot b \cdot R_{\ell} + t_o + s_o \cdot \bar{x}^{(1)}_{min}(m) && \text{for } m < N_{DEV} \end{aligned} \tag{4.4}$$

In case of $N_{CYL} > N_{DEV}$ a concurrent access strategy can be applied. If the seek is overlapped only one weighted head motion $\bar{x}^{(n)}_{min}(m)$ must be considered in each parallel access cycle. In case of cylinder to cylinder progression $x^{(n)}_{min}(m)$ is again the probability to position anew. Hence,

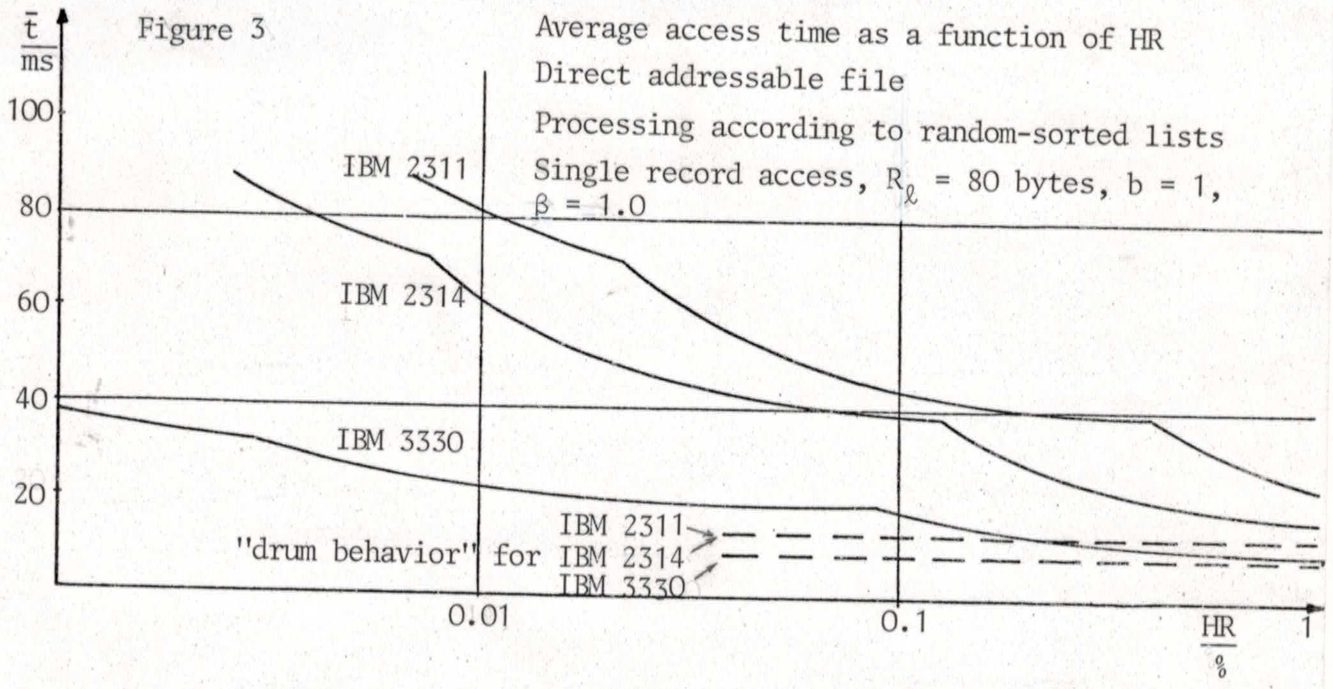
$$\begin{aligned} \bar{t}(HR) &= \frac{1}{2} t_{rev} + \frac{1}{u} \cdot b \cdot R_{\ell} + \frac{m \cdot 100}{HR \cdot N_{REC}} \cdot \bar{x}^{(n)}_{min}(m) \cdot t_{s\,min} && \text{for } 0 \leqslant \bar{x}^{(n)}_{min}(m) \leqslant 1 \\ \bar{t}(HR) &= \frac{1}{2} t_{rev} + \frac{1}{u} \cdot b \cdot R_{\ell} + \frac{N_{DEV}}{N_{CYL}} \cdot (t_o + s_o \cdot \bar{x}^{(n)}_{min}(m)) && \text{for } \bar{x}^{(n)}_{min}(m) > 1 \end{aligned} \tag{4.5}$$

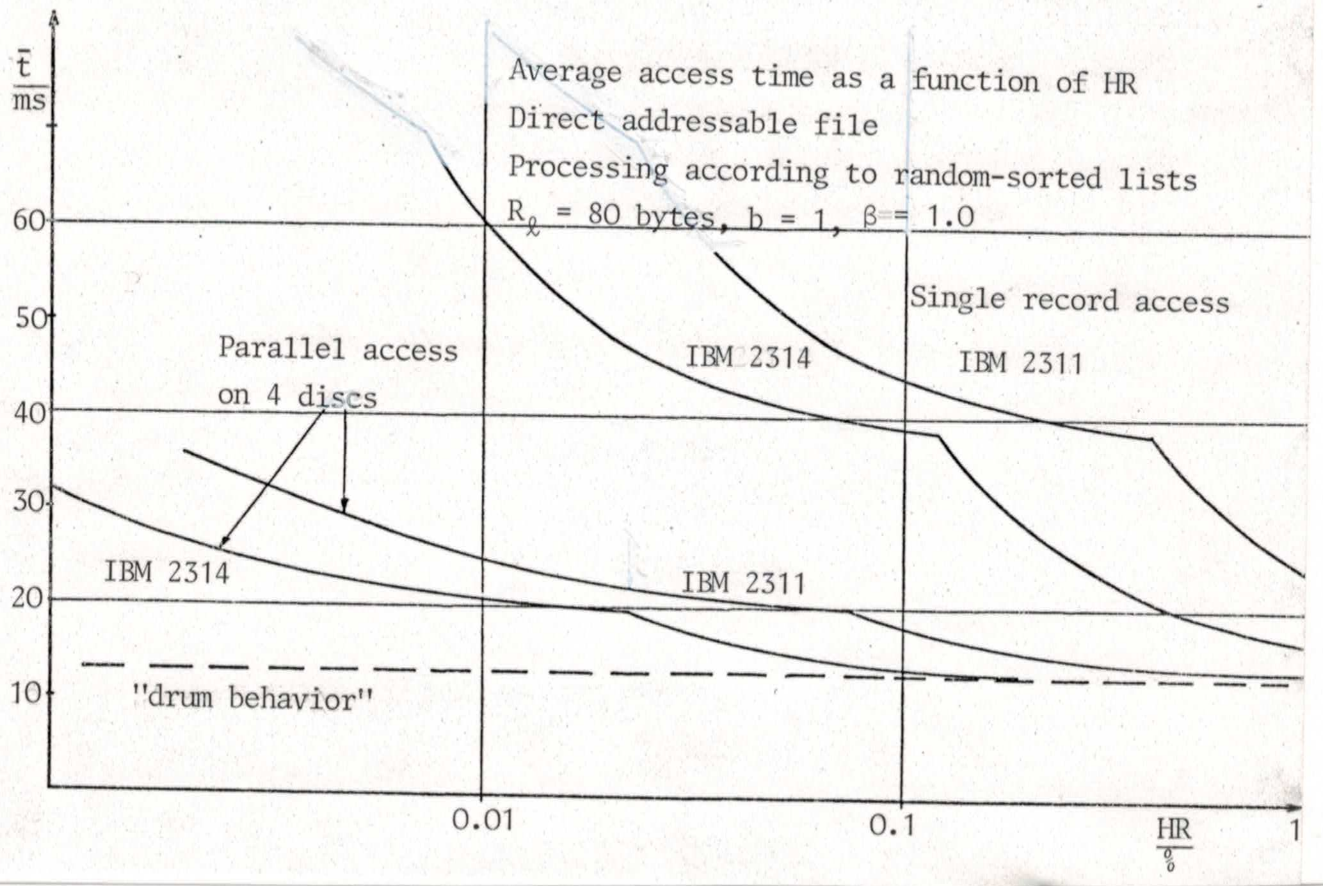
In figure 3 the equations (4.4) and (4.5) are evaluated for given parameters as functions of HR. For a single access the average access time is independent on N_{REC} , while N_{REC} determines the degree of parallelism in case of concurrent accesses. "Drum behavior" can be already reached for relatively small hit ratios. With regard to the particular characteristics of the used storage devices it is inherent with the access logic that a performance enhancement up to a factor of 7 is attainable when parallel access on the basis of random-stored lists is compared with "one record at a time" accesses.

This increase of effectiveness can be realized by the high selection capability of non-procedural languages and the separation of access path information from data records. In a navigating search the DBMS cannon coordinate the accesses because of the permanent disturbance of the application program. Embedding the access path by address chains impairs the performance optimization as well as the use of a procedural data manipulation language.

5. Conclusions

The access time behavior of relational data base systems has been investigated in a deterministic worst case analysis under consideration of real storage media. Given a certain hit ratio the processing and allocating of records has been considered according to random and random-sorted lists in case of parallel and single record access. Density and distribution functions of accessing cylinders on moveable head discs have been derived under the assumption of equal distribution of re-





cords and equal probability of accessing any cylinder. To compute mean values of arm movement in case of parallel access on n discs the method of "parallel combination" of random variables was used. It has been shown that "drum behavior" can be established on discs by accessing subsets of records of a file.
The strict separation of a non-procedural language from aspects of access provides to the DBMS an additional degree of freedom by accessing subsets of records.
It allows at any time to select optimal access paths and access strategies.
The embedding of access paths destroys the property of ordering the access according to arm movement of discs and the potential parallelism of accessing.

Appendix: Derivation of processing and access times

The following storage parameters and device characteristics are needed:

$\bar{t}$	= average access time to a record
$\bar{t}_s$	= average seek time
$\bar{t}_r$	= average rotational delay (latency)
$\bar{t}_{tr}$	= average data transfer time
t_o	= initial value of the approximation line of seek
s_o	= slope of the approximation line of seek
x	= number of traversed cylinders
N_{CYL}	= number of occupied cylinders of the file
N_B	= number of blocks per track
b	= blocking factor
R_ℓ	= logical record length
$b \cdot R_\ell$	= transferred data
β	= load factor (ratio of occupied and available storage)
KL	= key length of a separatly stored key

The technical characteristics of the considered magnetic disc units are comprised in the following table :

characteristic		disc unit		
		IBM 2311	IBM 2314	IBM 3330
t_{smin}	= seek (min.)	25 ms	25 ms	10 ms
t_{sav}	= " (av.)	75 ms	75 ms	30 ms
t_{smax}	= " (max.)	135 ms	135 ms	55 ms
t_{rev}	= revolution time	25 ms	25 ms	16.7 ms
C	= available track capacity	3625 bytes	7294 bytes	13030 bytes
T_{CYL}	= number of tracks per cylinder	10	20	19
N_{DEV}	= number of cylinders	200 (+3)	200 (+3)	404 (+7)
u	= transfer speed	$156 \frac{\text{bytes}}{\text{ms}}$	$312 \frac{\text{bytes}}{\text{ms}}$	$806 \frac{\text{bytes}}{\text{ms}}$
D_{ov}	= hardware overhead for data	61 bytes	101 bytes	135 bytes
K_{ov}	= hardware overhead for key	20 bytes	45 bytes	56 bytes
V_{ov}	= hardware overhead variable	0.05	0.04	0.04
t_a	= initial value of the approximation line of seek for $x \gg 1$	45 ms	45 ms	17.5 ms
s	= slope of the approximation line of seek for $x \gg 1$	0.45	0.45 ms	0.094
s_n	= slope of the approximation line of seek for $x \leqslant N_{DEV}/10$	1.6	1.6	0.325

a) The number of blocks per track is given by (Hä 75)

$$N_B = 1 + (C-K_{ov}-KL-b \cdot R_\ell)/(b \cdot R_\ell + D_{ov}+K_{ov}+KL+V_{ov} \cdot (b \cdot R_\ell + KL)) \quad \text{for } R_\ell \leqslant C \qquad (A.1)$$

If no separately stored key exists, then KL = 0 and $K_{ov} = 0$.

b) To determine the average access time $\bar{t}$ to a moveable head disc tree components - seek, latency and transfer - are to be considered:

$$\bar{t} = \bar{t}_s + \bar{t}_r + \bar{t}_{tr} \qquad (A.2)$$

Relational delay and transfer time for a block (page) are

$$\bar{t}_r = \frac{1}{2}\, t_{rev} \text{ and } \qquad \bar{t}_{tr} = \frac{1}{u} \cdot b \cdot R_\ell$$

The access motion time canbe described by an approximation line with sufficient precision for design considerations (Hä 75). The general approximation line is introduced by

$$\bar{t}_s = t_o + s_o \cdot x \qquad (A.3)$$

In order to better approximate the special seek time graph a case distinction is necessary with respect to x.
Equation (A.3) yields

$$\bar{t}_s = t_{smin} + s_n \cdot x \qquad \text{for } 1 < x < N_{DEV}/10$$

and

$$\bar{t}_s = t_a + s \cdot x \qquad \text{for } x > N_{DEV}/10$$

c) In case of physical sequential processing the head of the access mechanism has to be moved once over all tracks and cylinders of the file. If the transfer of blocks is achieved at full channel speed, the total processing time can be estimated by

$$\bar{t}_{tot} = N_{CYL} \cdot T_{CYL} \cdot t_{rev} + N_{CYL} \cdot t_{smin}$$

Considering equation (4.1) it follows because of $t_{smin} < t_{rev}$

$$\bar{t}_{tot} = \frac{T_{CYL} + 1}{\beta \cdot b \cdot N_B \cdot T_{CYL}} \cdot t_{rev} \cdot N_{REC} = K_{seq} \cdot N_{REC} \qquad (A.4)$$

References

(An65) Anderson, J. P.: Programming Structures for Parallel Processing, in: Comm. of the ACM, Vol. 8, No. 12, Dec. 1965, pp. 786-788.

(Co70) Codd, E. F.: A Relational Model of Data for Large Shared Data Banks, in: Comm. of the ACM, Vol. 13, No. 6, June 1970, pp. 377-387.

(Co71) Codd. E. F.: A Data Base Sublanguage Founded on the Relational Calculus, in: 1971 ACM SIGFIDET Workshop on Data Description, Access and Control, San Diego, Calif., Nov. 1971, pp. 35-68.

(Hä75) Härder, T.: Das Zugriffszeitverhalten von Relationalen Datenbanksystemen, Dissertation im Fachbereich Informatik der Technischen Hochschule Darmstadt, Juni 1975.

(IBM 1) IBM Data Processing Techniques; Analysis of some Queuing Models in Real-Time Systems, Form F20-0007-1.

(Ka75) Karlowsky, I, Leilich, H.-D., Stiege, G.: Ein Suchrechnerkonzept für Datenbankanwendungen, in: Elektronische Rechenanlagen 17 (1975), No. 3, pp. 108-118.

(We73) Wedekind, H.: Systemanalyse, Carl Hanser Verlag, München 1973.