HUGO VOLGER

The role of rudimentary relations in complexity theory

*Mémoires de la S. M. F. 2e série*, tome 16 (1984), p. 41-51

<http://www.numdam.org/item?id=MSMF_1984_2_16__41_0>
THE ROLE OF RUDIMENTARY RELATIONS IN COMPLEXITY THEORY

Volger, Hugo

Résumé:
On étudie dans cet article les classes R et XR des relations rudimentaires et faiblement rudimentaires qui se reposent sur la relation de la concaténation bornée. On obtient RUD et XRUD, les classes correspondantes des langages, comme l'union d'une hiérarchie linéaire resp. polynomiale. Ces hiérarchies utilisent des quanteurs alternants aux longueurs bornées ou également des machines alternantes de Turing avec alternance constante. Nous allons introduire une autre description utilisant des quanteurs alternants pour des oracles. En plus on obtiendra une chaîne nouvelle des hiérarchies pour tous les niveaux exponentiels, dont l'union sera ERUD, l'analogue exponentiel de la classe RUD. Et on va montrer que ERUD est la classe $E_3$ des langages élémentaires.

Abstract:
We shall study the classes R resp. XR of rudimentary resp. extended rudimentary relations which are based on the relation of bounded concatenation. The associated classes RUD resp. XRUD of languages are the union of a linear - resp. polynomial time hierarchy. It can be described either by means of alternating length bounded quantifiers or by means of Turing machines with constant alternation. We shall introduce another description based on alternating quantifiers for oracle sets. Extending these results we obtain a chain of hierarchies for the iterated exponential time levels, whose union is the class ERUD, the exponential analogue of RUD. Moreover, it will be shown that ERUD coincides with the class of elementary recursive languages.

Table of contents:
1. Introduction
2. Concatenation as a base of computability theory
3. The rudimentary relations
4. Turing machines with constant resp. linear alternation
5. The linear - and polynomial time hierarchies
6. A chain of exponential time hierarchies
7. Two logspace hierarchies
8. The theories of bounded concatenation
9. Conclusion
10. References

0037-9484/84 03 41 11/ $3.10/ © Gauthier-Villars
Introduction:

This paper is a survey on the classes R, XR, ER of rudimentary resp. extended rudimentary resp. exponential rudimentary relations and the corresponding classes RUD, XRUD, ERUD of languages. R and XR were introduced by Smullyan in 1961 resp. Bennett in 1962 (cf. [19], [1]), whereas ER is a new class. As we shall see later, a relation is rudimentary if it is definable from the concatenation relation by means of a first order formula where all quantifiers have linear length bounds. XR resp. ER will be the polynomially resp. exponentially bounded analogue of R.

The associated classes RUD, XRUD, ERUD may be obtained as the union of certain hierarchies. In her thesis in 1975 Wrathall [27] has shown that there are length bounded quantification hierarchies which yield \( L^p = RUD \) resp. \( \text{PH} = XRUD \) and have as first step \( \text{NUTIME} \) resp. \( \text{NPTIME} \). As length bounded quantification is closely related to time bounded alternation, these hierarchies can also be described as constant alternation hierarchies for \( L^p \) and \( \text{PH} \) (cf. Chandra, Stockmeyer [4], Kozen [10]).

Recently Orponen [16] has introduced a class \( \text{EH} \) as the union of an exponential time hierarchy involving oracle set quantification and having \( \text{NEXPTIME} \) as a first step. Extending his approach we are able to describe the hierarchies for \( L^p \) and \( \text{PH} \) as oracle set quantification hierarchies. Moreover, we shall introduce classes \( \text{EH}^{(i)} \) as the union of an analogous hierarchy involving the \( i \)-th iterate \( e_i \) of the exponential function, and we shall show that each of the three descriptions may be used. As a consequence we obtain that \( \text{ERUD} \) is the union of the classes \( \text{EH}^{(i)} \) and coincides with the class of elementary recursive languages. In addition, the alternating log-space hierarchy of Chandra, Kozen and Stockmeyer [5] may be viewed as step -1 of this chain of hierarchies.

The class \( \text{EH}^{(1)} \) which consists of languages requiring a constant number of alternations is contained in the class \( \text{LA}_1 \) the corresponding class with a linear amount of alternation. Recently we have shown that the decision problem of the theory \( e_i \)-bounded concatenation is complete in the class \( \text{LA}_1 \) w.r.t. polynomial time reductions for \( i \geq 1 \). In a certain sense these results for \( \text{EH}^{(1)} \) and \( \text{LA}_1 \) measure the power of \( e_i \)-bounded concatenation (cf. also Wilkie [24, 25, 26]). However, the question whether the inclusion \( \text{EH}^{(i)} \subseteq \text{LA}_1 \) is proper for some \( i \geq 0 \) remains open. A positive answer would imply that the inclusions \( \text{PH} \subseteq \text{APTIME} \) and \( \text{LH} \subseteq \text{ALTIME} \) are proper, thus solving important open problems in complexity theory.

 Concatenation as a base of computability theory:

In 1946 Quine [17] suggested to use the concatenation relation rather than addition and multiplication as a base of computability theory. Thus in 1961 Smullyan [19] introduced the class R resp. \( R_\delta \) of rudimentary resp. strictly rudimentary relations on \( \{1,2\}^* \). They consist of those relations which are definable from the concatenation relation by a first order formula where all quantifiers have a linear...
Rudimentary Relations

bound resp. are subword quantifiers. Smullyan has shown that $R_s$ is all we need to describe computations. Each language $L \subseteq \{1,2\}^*$ which is recursively enumerable i.e. accepted by some Turing machine $M$ can be obtained from a relation $Q$ in $R_s$ as follows: $x \in L$ iff $\exists y: (x, y) \in Q$, where $(x, y) \in Q$ expresses the fact that $y$ is an accepting computation sequence with input $x$. This shows that $R_s$ is large enough to enable us to describe Turing machine computations by means of words consisting of sequences of configuration words. On the other hand $R_s$ is quite small since the associated class $\text{RUD}_s$ of languages is contained in $\text{LOGSPACE}$ and does not contain $\{1^{n^2} \mid n \in N\}$ (cf. Nepomnjasiei [15], Meloul [11]). In addition, the $\text{NP}$-complete problem $\text{SAT}(x)$ is of the form $\exists y: |y| \leq |x| \land Q(x, y)$ with $Q$ in $R_s$ as Meloul [11] has shown. This may explain why the class $R_s$ and the related classes $R$ and $XR$ play an important role in complexity theory.

3. The rudimentary relations:

The class $R$ resp. $R_s$ of rudimentary resp. strictly rudimentary relations on $\{1,2\}^*$, introduced by Smullyan [19], is defined as the least class of relations which contains the concatenation relation $\text{Con}$ and which is closed under the boolean operations, explicit transformations and linearly bounded resp. subword quantification. The class $R^+$ of positive rudimentary relations on $\{1,2\}^*$, introduced by Bennett [1], is defined as the least class of relations which contains the relation $\text{Con}$ and which is closed under finite unions and intersections, explicit transformations, subword quantification and linearly bounded existential quantification.

$\exists y: y \leq x \land \cdots$, $\forall y: y \leq x \rightarrow \cdots$ subword quantification
$\exists y: |y| \leq |x| \land \cdots$, $\forall y: |y| \leq |x| \rightarrow \cdots$ linearly bounded quantification

Using the $k$-adic encoding words over $\{1, \ldots, k\}$ may be identified with natural numbers. Bennett [1] has shown that modulo the dyadic encoding $R$ coincides with the class $\text{CA}$ of constructive arithmetic relations on $N$, which is the analogue of $R$ on $N$ using $+$ and $\times$ rather than $\text{Con}$. In addition, $\text{CA}$ coincides with the class of bounded arithmetic relations of Harrow [6]. Moreover, the analogues of $R$ resp. $R_s$ resp. $R^+$ on $\{1, \ldots, k\}^*$ coincide with $R$ resp. $R_s$ resp. $R^+$ modulo the $k$-adic encoding and the dyadic decoding. Using the sequential encoding $\theta(Q)$ of a relation $Q$ one obtains the corresponding classes of languages on $\{1,2,\$\}: \text{RUD}, \text{RUD}_s, \text{RUD}^+$. It can be shown that these classes may be identified with the unary relations in $R, R_s, R^+$.

Replacing linearly bounded quantification by polynomially bounded quantification (i.e. $\exists y: |y| \leq |x|^k \land \cdots$ and $\forall y: |y| \leq |x|^k \rightarrow \cdots$) one obtains the classes of extended rudimentary resp. extended positive rudimentary relations, which were introduced by Bennett [1].

Going a step further we introduce the classes $\text{ER}$ resp. $\text{ER}^+$ of exponential rudimentary resp. exponential positive rudimentary relations. They are obtained from
R resp. \( R^+ \) by replacing linearly bounded quantification by exponentially bounded quantification (i.e. \( \exists y : |y| \leq e_1(|x|^k) \land \ldots \) and \( \forall y : |y| \leq e_1(|x|^k) \rightarrow \ldots \) with \( e_1(n) = 2^n \)).

Clearly, iterated exponential functions can be used as length bounds as well. The corresponding classes of languages are denoted by XRUD, XRUD' resp. ERUD, ERUD'.

These classes are related as follows: \( \text{RUD} \subseteq \text{RUD}^+ \subseteq \text{XRUD} \subseteq \text{XRUD}^+ \subseteq \text{ERUD}^+ \subseteq \text{ERUD} \).

It should be mentioned that Jones [8] has introduced sublinear analogues of the class R resp. RUD. In particular, he considered a subclass \( \text{RUD}_{\log} \) of LOGSPACE. It is not clear how this class fits into the above set up.

4. Turing machines with constant resp. linear alternation:

Chandra and Stockmeyer [4] and Kozen [10] have extended the concept of nondeterministic Turing machines (NTM's) to alternating Turing machines (ATM's). There is a close connection between alternation and quantification. In particular, hierarchies defined by bounded quantification are closely related to hierarchies defined by constant alternation using the same time bound.

An ATM \( M \) is a NTM which has 2 disjoint sets of states, the existential and universal states, and a distinguished accepting resp. rejecting state. Configurations and their successor relation are defined as for NTM's. An input \( w \) is accepted by \( M \) (i.e. \( w \in L(M) \)), if there exists a finite accepting subtree \( B \) of the computation tree of \( M \) for \( w \). \( B \) is accepting, if (1) the root of \( B \) is labeled with the input configuration for \( w \), (2) all leaves of \( B \) are labeled with accepting configurations, (3) if a node \( b \) of \( B \) is labeled with an existential (resp. universal) configuration \( C \) then at least one (resp. all) successor configurations \( C' \) of \( C \) must appear as labels of successors \( b' \) of \( b \) (cf. Berman [2]).

A language \( L \) belongs to the alternating class \( \text{STA}(s,t,a) \), if \( L \) is accepted by an ATM \( M \) such that each \( w \) in \( L \) possesses an accepting subtree \( B \) of depth \( \leq t(n) \) and alternation depth \( \leq a(n) \) and each configuration in \( B \) uses space \( \leq s(n) \), where \( n=|w| \).

We shall use the notation \( \text{STA}_3(s,t,a) \) resp. \( \text{STA}_\text{u}(s,t,a) \) to indicate that the input configuration is required to be existential resp. universal. As special cases we obtain the alternating time class \( \text{ATIME}(t) = \text{STA}(-,t,-) \) and the alternating space class \( \text{ASPACE}(s) = \text{STA}(s,-,-) \). The time class with constant alternation \( \text{CATIME}(t) \) is defined as \( \cup \text{STA}_3(-,t,k) : k \in \mathbb{N} \). Similarly the time class with linear alternation \( \text{LATIME}(t) \) is defined as \( \text{STA}_3(-,t,\text{id}) \).

Alternating time bridges the gap between nondeterministic time and deterministic space as Chandra, Kozen and Stockmeyer [5] have shown:

\[ (*) \quad \text{NTIME}(t) \subseteq \text{CATIME}(t) \subseteq \text{LATIME}(t) \subseteq \text{ATIME}(t) \subseteq \text{DSPACE}(t) \quad \text{for} \quad t \geq \text{id} \]

\[ (**) \quad \text{ALOGSPACE} = \text{PTIME}, \quad \text{APTIME} = \text{PSPACE}, \quad \text{APSPACE} = \text{EXPSPACE} \]
5. The linear - and polynomial time hierarchies:

Wrathall [27] has shown that the class XRUD is the union of the polynomial time hierarchy of Meyer and Stockmeyer [12], and that the class RUD is the union of a linear time analogue of this hierarchy. There are several descriptions of these two hierarchies as we shall see below.

Constant Alternation:

\[
\text{AP}_k = \bigcup_{i \in \mathbb{N}} \text{AP}_k = \bigcup_{i \in \mathbb{N}} \text{STAG}(-,0(n^i),k) : i \in \mathbb{N} \text{ for } k \geq 1, \text{AP}_0 = \text{PTIME}
\]

\[
\text{AL}_k = \bigcup_{i \in \mathbb{N}} \text{AL}_k = \text{STAG}(-,0(n),k) \text{ for } k \geq 1, \text{AL}_0 = \text{LTIME}
\]

Hence we have \( \text{AP} = \bigcup_{i \in \mathbb{N}} \text{CATIME}(O(n^i)) : i \in \mathbb{N} \), \( \text{AL} = \text{CATIME}(O(n)) \)

Length Bounded Quantification:

\[
\text{PH} = \bigcup_{i \in \mathbb{N}} \text{PH}_i \text{ and } m_1, ..., m_k \text{ such that:}
\]

\[
x \in L \text{ iff } \exists y_1: |y_1| \leq |x|^{m_1} \cdots Q_k y_k: |y_k| \leq |x|^{m_k}: (x,y_1,...,y_k) \in L'.
\]

\[
\text{LH} = \bigcup_{i \in \mathbb{N}} \text{LH}_i \text{ and } L_1, ..., L_k \text{ such that:}
\]

\[
x \in L \text{ iff } \exists y_1: |y_1| \leq m_1 |x|^{m_1} \cdots Q_k y_k: |y_k| \leq m_k |x|: (x,y_1,...,y_k) \in L'.
\]

Oracle Set Quantification:

\[
\text{OPH} = \bigcup_{i \in \mathbb{N}} \text{OPH}_i \text{ and } \text{OP} = \bigcup_{i \in \mathbb{N}} \text{STAG}(\log(n^i),-,k) : i, k \in \mathbb{N}
\]

\[
\text{LH} = \bigcup_{i \in \mathbb{N}} \text{LH}_i \text{ and } L_1, ..., L_k \text{ such that:}
\]

\[
x \in L \text{ iff } \exists A_1, ..., A_k: M \text{ accepts } x \text{ with the oracles } A_1, ..., A_k.
\]

Iterated Nondeterministic Oracles:

\[
\text{NP}_k = \bigcup_{i \in \mathbb{N}} \text{NP}_k \text{ for } k \in \mathbb{N}; \text{NP} = \text{PH}
\]

\[
\text{PH} = \text{XRUD}; \text{NP} = \text{NPTIME} = \text{XRUD}^+
\]

\[
\text{NL}_k = \bigcup_{i \in \mathbb{N}} \text{NL}_k \text{ for } k \in \mathbb{N}; \text{NL} = \text{LH}
\]

\[
\text{LH} = \text{RUD}; \text{NL} = \text{NLTIME} \subset \text{RUD}^+
\]

The proofs of (1) - (4) except \( \text{NL} \subset \text{RUD}^+ \) can be found in Wrathall [28, 29]. An application of a result of Book and Greibach [3] to the inclusion \( \text{CFL} \subset \text{RUD}^+ \) in Yu [30] yields the desired inclusion (cf. Meloul [11]).

The proof of the next proposition will be given in some detail since the result will be generalized later on.
Prop. 2: (1) $AP_k = \Pi^P_k$ for $k \in \mathbb{N}$; $AP_H = PH$
(2) $AL_k = L \Sigma_k$ for $k \in \mathbb{N}$; $AL_H = LH$
(3) $OP_k = \Pi^P_k$ for $k \geq 1$ in $\mathbb{N}$; $OP_H \subseteq AP$; $OP_H = AP_H$.

The result in (1) was mentioned in Chandra, Kozen and Stockmeyer [5] and the analogous result in (2) can be found in Volger [23]. (3) is a new result which constitutes an analogue of a result of Orponen [16] for EH, the union of an exponential time hierarchy.

(1) and (2) can be proved by the same method. Given the syntactic description of $L$ which uses at most $k$ alternations of length bounded quantifiers, it is easy to construct an ATM accepting $L$ with the corresponding time bound and at most $k$ alternations. This proves $P \Sigma_k \subseteq \Pi^P_k$ resp. $L \Sigma_k \subseteq \Pi^P_k$. Conversely, given an ATM accepting $L$ with at most $k$ alternations, one constructs a deterministic TM accepting a language $L'$ and having $k$ additional tapes with the following property. Simulating the $i$-th alternation phase the machine controls the choice of moves to be simulated by reading the $i$-th tape as long as necessary going from left to right. Hence $L$ can be obtained from $L'$ by an appropriate length bounded quantification with at most $k$ alternations, as desired. This should be compared with the incremental stack automata in Yu [30]. This proves $AP_k \subseteq P \Sigma_k$ resp. $AL_k \subseteq L \Sigma_k$.

To prove (3) we adapt Orponen's proof in [16]. The oracle free part of the constant alternation oracle TM $M$ for $L$ can be simulated by a DTM $M'$ working in polynomial time because of $\text{STA}^j(\log(n^j),-,k) \subseteq \text{ASPACE}(\log(n^j)) \subseteq \text{DTIME}(O(n^3))$ for some $j$. This inclusion can be found in Chandra, Kozen and Stockmeyer [5]. The $k$ quantifiers concerning the oracle sets $A_1, \ldots, A_k$ will be replaced by $k$ alternations of an ATM $M''$ extending $M'$, where each branch in the $j$-th alternation phase corresponds to an oracle set $A_j = A_j \land (1,2)^{< \log(n^j)}$. Because of the space bound of $M$ it suffices to consider $A_j$ instead of $A_j$. Moreover, each set $A_j'$ can be specified in $n^j$ steps. Thus $M''$ works in polynomial time. This shows $OP_k \subseteq AP_k$.

Conversely, let $L$ be accepted by a constant alternation TM $M$ working in polynomial time. The idea is to code a computation sequence $\alpha$ of configurations of $M$ by an oracle set $C(\alpha)$ which is coded characterwise. A sequence $\alpha$ of $d = n^1$ configurations of length $n^1$ is a word of length $\leq d^2$. It can be coded as follows: $C(\alpha) = \{(i,j,a_{i,j}):(i,j) \leq d^2\}$, where $a_{i,j}$ is the $j$-th character in the $i$-th configuration of $\alpha$. The indices $i, j$ are short because of $|i|, |j| \leq 2\log(n^1)$. Given $(i,j) a_{i,j}$ can be recovered from $C(\alpha)$ by at most a fixed number of queries. Since the successor relation is local, it is possible to construct a constant alternation oracle TM $M'$ working on space $\log(n^1)$ for some $i$ such that $(u,v)$ is accepted by $M'$ with oracle $C$ if and only if $C$ codes a computation sequence of $M$ starting with $u$ and ending with $v$ and having no alternation except at the last step. Similarly, the input configurations and the
accepting configurations can be handled by appropriate machines. In order to express acceptance by the given ATM \( M \) note that each alternation phase \( i \) gives rise to a quantification over an oracle \( C_i \) corresponding to it. By this method one obtains a constant alternation oracle \( TM^A \) working on space \( \log(n^i) \), which does the required job. It should be noted that \( A \) can be chosen to be universal. This shows \( AP \subseteq OP \).

The inclusion \( OP = \bigcup \text{STAA}^\log(n^i),k:k \in N \) follows from \( \text{PTIME} = \text{ALOGSPACE} \) which was proven in Chandra, Kozen and Stockmeyer [5].

6. A chain of exponential time hierarchies:

As mentioned above, Orponen [16] introduced a class \( EH \) as the union of an exponential time analogue of the hierarchy for \( AP = PH \). More generally, we shall consider iterated exponential time analogues of the hierarchy for \( PH \) and obtain a chain of classes \( EH(1) \) whose union is the class \( \overline{E} \) of elementary recursive languages.

Let \( e_i \) be the \( i \)-th iterate of the exponential function, i.e. \( e_0(n) = n \) and \( e_{i+1}(n) = \exp(2, e_i(n)) \), where \( \exp(2,m) = 2^m \). As before there are several ways of describing the hierarchies for \( EH(1) \).

The constant alternation hierarchy \( \text{AEH}(1) = \bigcup \text{AE}^i(1) : k \in N \) is obtained from \( APH \) by replacing everywhere \( \Theta(n^i) \) by \( e_i(O(n^i)) \). The length bounded quantification hierarchy \( \text{EH}(1) = \bigcup \text{EH}^i(1) : k \in N \) is obtained from \( PH \) by replacing everywhere \( \Theta(n^i) \) by \( e_i(O(n^i)) \). The oracle set quantification hierarchy \( \text{OEH}(1) = \bigcup \text{OEH}^i(1) : k \in N \) is obtained from \( OPH \) by replacing everywhere the space bound \( \log(n^i) \) by the time bound \( e_i-1(O(n^i)) \) and defining \( \text{OEH}^i(1) = \text{AE}^i(1) \).

Orponen [16] considered the hierarchies for \( \text{AEH}(1) \) and \( \text{OEH}(1) \) and proved \( \text{AEH}(1) = \text{OEH}(1) \). The hierarchy for \( \text{EH}(1) \) and all the other hierarchies for \( i \geq 2 \) seem to be new. In the case \( i = 0 \) we obtain the hierarchies for \( APH \), \( PH \) and \( OPH \) discussed earlier. The following proposition extends the results in proposition 2.

Prop. 3: For \( i > 1 \) we have:

1. \( \text{AE}^i(1) = \text{EH}^i(1) \) for \( k \in N \).
2. \( \text{OE}^i(1) = \text{AE}^i(1) \) for \( k \geq 1 \) in \( N \).

This can be proved by the same method which was used to prove (1) and (3) in proposition 2. To prove \( \text{EO}^i(1) = \text{AEH}^i-1(1) \subseteq \text{AE}^i(1) \) we use \( \text{STAA}^\log(n^i),k \subseteq \text{SPACE}(e_i-1(O(n^i))) \subseteq \text{DTIME}(e_i(O(n^i))) \) proved in [5]. Moreover, an oracle set of words of length \( e_i-1(O(n^i)) \) can be specified in \( e_i(O(n^i)) \) steps, whereas the code of a computation sequence of \( e_i(O(n^i)) \) configurations of length \( e_i(O(n^i)) \) uses words of length \( \leq e_i-1(O(n^i)) \). This shows that (1) and (2) can be proved as before.

The next proposition shows that \( \overline{E} \), the class of elementary recursive languages, coincides with \( \text{ERUD} \) and that the classes \( EH(1) \) form a new hierarchy for \( \overline{E} \).
We shall use the following abbreviations: \( IA_i = \cup \langle \text{LATIME}(e_i(0(n^i))) : i \in \mathbb{N} \rangle \) and \( AS_i = \cup \langle \text{ASPACE}(e_i(0(n^i))) : i \in \mathbb{N} \rangle \).

**Prop. 4:**

1. \( \text{AEH}^{(i)} \subseteq IA_i \subseteq AS_i \subseteq \text{AEH}^{(i+1)} \subseteq \text{AEH}^{(i+1)} \) for \( i \in \mathbb{N} \).
2. \( \cup \langle \text{AEH}^{(i)} : i \in \mathbb{N} \rangle = \cup \langle \text{LA}_i : i \in \mathbb{N} \rangle = \cup \langle \text{AS}_i : i \in \mathbb{N} \rangle = \mathbb{E} \).
3. For each \( L \in \text{AEH}^{(i)} \), there exists \( L' \in \text{IH} \) and \( i \in \mathbb{N} \) such that \( x \in \mathbb{X} \) iff
   \( \exists y : |y| = e_i(0(n^i)) : (x, y) \in L' \).
4. \( \hat{\mathbb{E}} = \mathbb{ERUD} = \text{ERUD}^+ \).
5. \( \text{AEH}^{(i)} \neq \text{AEH}^{(i+2)} ; \text{AEH}^{(1)} \neq \text{AEH}^{(1)} \) implies \( \text{AEH}^{(1)} \neq \text{AEH}^{(i+1)} \).

The inclusions needed for (1) can again be found in [5]. (2) is a consequence of (1) because of the well known fact \( \hat{\mathbb{E}} = \cup \langle \text{AE}(i) : i \in \mathbb{N} \rangle \). To prove the representation result in (3) which represents elements of \( \mathbb{E} \) with the help of elements of \( \text{IH} \) we show (cf. Wrathall [27] in the case \( i = 0 \)):

(*) For each \( L \in \text{STA}_{3}(-, e_i(0(n^i)), k) \), there exists \( L' \in \text{STA}_{3}(-, 0(n), k) \) such that:
   \( x \in L \) iff
   \( \exists y : |y| = e_i(|x|^1) \land (x, y) \in L' \).

\( L' = \{(x, y) : |y| = e_i(|x|^1) \land x \in L \} \) or \( \{x \in L \land |x|^m = e_i(|x|^1) \} \) will do the job.

\( \mathbb{ERUD} \) is contained in \( \hat{\mathbb{E}} \) since \( \hat{\mathbb{E}} \) contains Con and has the necessary closure properties. To prove the converse note that \( \mathbb{ERUD} \) as well as \( \hat{\mathbb{E}} \) are closed under length bounded quantification where any \( e_i \) is used as a length bound. Then the inclusion \( \hat{\mathbb{E}} \subseteq \mathbb{ERUD} \) follows by an application of (3) because of \( \text{IH} \subseteq \text{LSpace} \subseteq \hat{\mathbb{E}} \). This proves \( \mathbb{ERUD} = \hat{\mathbb{E}} \). To prove the equality \( \text{ERUD}^+ = \hat{\mathbb{E}} \) it suffices to show \( \text{DTIME}(e_i(0(n^1))) \subseteq \text{ERUD}^+ \) because of \( \cup \langle \text{AE}^{(1)} : i \in \mathbb{N} \rangle = \hat{\mathbb{E}} = \text{ERUD} \). However, for each \( L \) in \( \text{DTIME}(e_i(0(n^1))) \), there exists \( L' \) in \( \text{LOGSPACE} \) such that:
   \( x \in L \) iff
   \( \exists y : |y| = e_i(0(n^1)) \land (x, y) \in L' \).

\( (x, y) \in L' \) states that \( y \) is an accepting computation sequence with input \( x \). This proves (4). (5) follows from (1) and the well known fact \( \text{AE}^{(1)} \neq \text{AE}^{(1+1)} \).

It should be mentioned that the representation result in (3) can be used to lift equalities between complexity classes at the linear time level to higher levels, e.g. \( \text{IH} = \text{LSpace} \) implies \( \text{EH}^{(1)} = \cup \langle \text{DSpace}(e_i(0(n^1))) : i \in \mathbb{N} \rangle \).

7. Two logspace hierarchies:

In [5] Chandra, Kozen and Stockmeyer considered indexing ATM's, a variant of the
ATM's which permits the use of sublinear time bounds. An indexing ATM has an index
tape whose content may be interpreted as position of the input which can be accessed.
Let \( e_{-1}(n) \) be \( \log(n) \). The two logspace hierarchies defined below might both be con-
sidered as step -1 of the chain of hierarchies discussed earlier. The first hier-
archy was introduced in [5].

\[
\begin{align*}
\text{AEH}^{(-1)} &= \cup \langle \text{AE}^{(-1)}_k : k \in \mathbb{N} \rangle, \\
\text{AE}^{(-1)}_O &= \cup \langle \text{LOGSPACE}_k : k \in \mathbb{N} \rangle, \\
\text{AE}^{(-1)}_K &= \cup \langle \text{LOGTIME}_k : k \in \mathbb{N} \rangle,
\end{align*}
\]
Rudimentary Relations

We obtain another description of these logspace hierarchies if we replace in the definition of PH the bounds $O(n^k)$ by $\log(n^k)$ and PTIME by LOGSPACE resp. LOGTIME. This yields the hierarchies $\text{EH}^{(-1)} = \cup k \in \mathbb{N} \back E_k^{(-1)}$ and $\text{EH}^{(-1)} = \cup k \in \mathbb{N} \back E_k^{(-1)}$.

The following proposition shows that $\text{AEH}^{(-1)}$ is contained in the class $\text{RUD} = \text{LH}$ whereas $\text{AEH}^{(-1)}$ contains the class $\text{RUD} = \text{LH}$ of Jones [8]:

Prop. 5: (1) $\text{RUD} \subseteq \text{LOGSPACE} \subseteq \text{AEH}^{(-1)} \subseteq \text{RUD}^{+}$, $\text{AEH}^{(-1)} \subseteq \text{RUD}$
(2) $\exists k \in \mathbb{N} \back E_k^{(-1)}$ for $k$ in $\mathbb{N}$, $\text{AEH}^{(-1)} = \text{EH}^{(-1)}$
(3) $\text{RUD} \subseteq \text{AEH}^{(-1)} \subseteq \text{LOGSPACE}$
(4) $\exists k \in \mathbb{N} \back E_k^{(-1)}$ for $k$ in $\mathbb{N}$, $\text{AEH}^{(-1)} = \text{EH}^{(-1)}$

(1) was proved in Volger [23]. (3) follows since $\text{AEH}^{(-1)}$ has the closure properties of $\text{RUD}$. (2) and (4) can be proved as (1) resp. (2) in proposition 2.

8. The theories of bounded concatenation:

The question whether linear alternation is more powerful than constant alternation, i.e. whether the inclusions $\text{CATIME}(e_i) \subseteq \text{LATIME}(e_i)$ and $\text{EH}^{(i)} \subseteq \text{LA}_i$ are proper, remains open. The classes $\text{LA}_i = \cup\text{STA}(e_i(0(n^k)),n), i \in \mathbb{N}$ are closely related to the theories of bounded concatenation. They were introduced by A.R.Meyer in 1975 (cf. [22]) as a uniform method for proving lower bounds for the complexity of first order theories.

The $t$-bounded concatenation relation $\text{Con}_t$ for a given function $t:N \rightarrow N$ is defined as follows: $(u,v,w,x) \in \text{Con}_t$ iff $uv = wa \land |w| \leq t(|x|)$. The theory of $t$-bounded concatenation, is the theory $\text{Thr}((1,2)^*, \text{Con}_t, 1, 2))$. Viewed in this context the equality $\text{AEH}^{(i)} = \text{EH}^{(i)}$ implies that each $L$ in $\text{AEH}^{(i)}$ is first order definable in the structure $((1,2)^*, \text{Con}_t, 1, 2)$. Recently, we have proved a completeness result for the classes $\text{LA}_i$ which in some sense measures the power of bounded concatenation (cf. [22]).

Prop. 6: (1) for all $L$ in $\text{EH}^{(i)}$ there is a uniform polynomial time reduction to the decision problem of $\text{BCT}((1,2)e_i)$.
(2) For each $L$ in $\text{LA}_i$ there is a polynomial time reduction to the decision problem of $\text{BCT}((1,2)e_i)$.
(3) The decision problem of $\text{BCT}((1,2)e_i)$ belongs to $\text{LA}_i$ whenever $i \geq 1$. In the case $i = 0$ i.e. $\text{LA}_0 = \text{ATIME}(O(n))$ the problem remains open.

9. Conclusion:

The results presented in this paper show that the bounded concatenation relation as well as the different classes of rudimentary languages which are based on it play an important role in that part of complexity theory concerned with the classes LOGSPACE, PTIME, NPTIME etc. There is also a close connection with time classes
Hugo Volger

with constant resp. linear alternation which should be studied in more detail.

10. References:

Rudimentary Relations

[31] Yu, Y.Y.: Rudimentary relations and stack languages, Math. Systems Theory 10 (1977), 337-343

Hugo Volger
Mathematisches Institut
Universität Tübingen
Auf den Morgenstelle 10
D 7400 TÜBINGEN