Relational Games

Yde Venema

1 Introduction

This article originates with the observation that similar phenomena occur in the theory of binary relations and in the modal logic of time. In both fields of logic, we find that operators play a rôle which can be defined in first order logic, and that it is an object of study to compare the expressive strength of different sets of such operators. Especially important is the question whether there is a simple way of generating all operators definable in first order logic.
In the case of temporal logic, Gabbay reduced the question whether the set of all first order definable operators is finitely generated, to the following problem: Is there a $k \in \omega$ such that, over the involved class of temporal orders, $L_k(x)$ is as least as expressive as $L(x)$? (cf [G].) Here $L(x)$ is the set of first order formulas of which x is the only free variable, $L_k(x)$ the set of $L(x)$-formulas in which only k different variables may occur. By comparing the expressiveness of different sets of formulas we mean the following:

Definition 1.1.
Suppose we have a language L, two sets F and F' of first order formulas in L, and a class K of models for L. F' is said to be as least as expressive as F over K if there is, for all formulas $\phi(x_1, \ldots, x_k) \in F$, a $\phi'(x_1, \ldots, x_k) \in F'$, such that ϕ and ϕ' are equivalent over K, i.e. for every model $\mathbf{A}$ in K and every k-tupel of elements $a_1, \ldots, a_k$ in A:

$$\mathbf{A} \models \phi[a_i, \ldots, a_k] \Leftrightarrow \mathbf{A} \models \phi'[a_1, \ldots, a_k].$$

In [IK], Immerman and Kozen achieved some results by reformulating Gabbay's problem in terms of games. They introduced the 'k-pebble game', an adapted version of the well-known Ehrenfeucht game meant to characterize the k-variable fragment of first order logic, in the same way as the ordinary Ehrenfeucht game characterizes the set of all first order formulas.

In this paper, we apply this technique in the field of proper augmented relation algebras (cf. [J2]); we first use the k-pebble game in section 4 to give a new proof of a well-known result by Tarski, viz. the fact that the clone of logical

operations on binary relations is not finitely generated. In section 5 we give a similar treatment to the clone of Jónsson's Q-operations, for which we define our own adaptations of the rules of the Ehrenfeucht game. Andréka and Németi proved in [AN] that these Q-operations do not generate the clone of logical operations either. Independently, we came to the same result, providing a negative solution to the problem 1.5.5 in Jónsson's paper [J2]. The novelty of our proofs lies in their game-theoretical approach which we feel is very intuitive and can be generalized to other clones of operations.
In the last sections we establish related results concerning sets of formulas of which the definition is inspired by phenomena in natural language. In order to keep this paper self-contained, we first give the basic definitions and facts concerning Ehrenfeucht games and the theory of proper augmented relation algebras, resp. in the sections 2 and 3.

Throughout this paper, L is a first order language without constants or function symbols, with a sufficiently large, yet finite number of dyadic relation symbols and with individual variables $x_1, x_2, \ldots$; $\mathbf{A}$ and $\mathbf{A}'$ are two structures for L.

2 Ehrenfeucht games.

Definition 2.1.
A *partial valuation* over $\mathbf{A}$ is a partial function $u : \{x_1, x_2, \ldots\} \mapsto A$ with finite domain δu. For a partial valuation u, the partial valuation $u[x_i/a]$ is defined by $u[x_i/a](x_j) = u(x_j)$ if $i \neq j$, $u[x_i/a](x_i) = a$. The notion of satisfaction of a formula ϕ by a partial valuation u is usually denoted by $\mathbf{A} \models \phi[u]$, informally we will write: u satisfies ϕ. A *k-configuration* over $\mathbf{A}, \mathbf{A}'$ is a pair (u, u'), where u and u' are partial valuations over $\mathbf{A}, \mathbf{A}'$ such that $\delta u = \delta u' \subseteq \{x_1, \ldots, x_k\}$. For a set of first order formulas F in L, $F(x_1, \ldots, x_k)$ is the set of F-formulas of which the free variables are in $\{x_1, \ldots, x_k\}$. For a k-configuration (u, u'), u and u' are said to be F-equivalent if they satisfy the same formulas in $F(x_1, \ldots, x_k)$. (u, u') is a local isomorphism if u and u' satisfy the same atomic formulas.

We will now define the ordinary Ehrenfeucht game of n moves on a k-configuration (u, u'):

Definition 2.2.
Let (u, u') be a k-configuration over $\mathbf{A}$,$\mathbf{A}'$. The game $G_n(u, u')$ is played by two players, I and II. The first player has to show that the two structures $(\mathbf{A}, u)$ and $(\mathbf{A}', u')$ are different, the other one wants to make them appear isomorphic. There is an infinite number of pebbles, pairwise colored $x_1, x_2, x_3, \ldots$. For $x_i \in \delta u (= \delta u')$, at the beginning of the game the two pebbles with color x_i are situated on the elements $u(x_i)$ and $u'(x_i)$ of A resp. A'.
The game is played in n moves; in the m-th move player I takes a new pebble,

say (with color) x_{k+m}. (Intuitively, the old pebbles (i.e. in δu) correspond to *free* variables of a formula, the new pebbles to *bound* variables.) She then chooses a structure, say A and puts the pebble on an element a of A: $x_{k+m} \mapsto a$. The second player has to respond by placing the other x_{k+m}-pebble on an element a' of the other structure, in this case A': $x_{k+m} \mapsto a'$.
Thus after the m-th move an $k+m$-configuration (u_m, u'_m) is generated, so after n moves one has a sequence $(u, u') = (u_0, u'_0), \ldots, (u_n, u'_n)$ of configurations.
The second player is said to have a winning strategy in the game if he has a way of playing such that every generated configuration is a local isomorphism. The first player has a winning strategy in the game if the second player has not.

The ordinary Ehrenfeucht game can be said to characterize first order logic. What is meant by this is expressed in Lemma 2.4. First we need some notions concerning first order formulas:

Definition 2.3.
The quantifier depth $qd(\phi)$ of a first order formula ϕ is inductively defined as follows: atomic formulas have quantifier depth zero, $qd(\neg\phi) = qd(\phi)$, $qd(\phi \wedge \psi) = max(qd(\phi), qd(\psi))$ and $qd(\exists x_i \psi) = 1 + qd(\psi)$. $L_{\omega,n}$ is the set of all first order formulas of quantifier depth n. For a k-configuration (u, u'), u and u' are said to be equivalent if they are $L_{\omega,n}$-equivalent for all n.
Now let ϕ be a first order formula. For a subformula $\exists x \psi$ of ϕ, we call ψ the scope of the quantifier occurrence $\exists x$. Let E and E' denote quantifier occurrences in ϕ. E is said to be above E' if E' is in the scope of E.

Lemma 2.4.
For any k-configuration (u, u'), player II has a winning strategy in $G_n(u, u')$ iff u and u' are $L_{\omega,n}$-equivalent.

Proof.
By induction on n: for $n = 0$ the proposition follows by definition of a winning strategy in a game with no moves, as formulas of quantifier depth 0 are just the Boolean combinations of atomic first order formulas.
For the inductive case, suppose that II has a winning strategy in $G_{n+1}(u, u')$ and suppose that u satisfies ϕ where $\phi(x_1, \ldots, x_k) \in L_{\omega,n+1}$. Without loss of generality we may assume that ϕ has the form $\exists x_{k+1}\psi$, where $\psi(x_1, \ldots, x_{k+1})$ has quantifier depth n. We have to prove that u' satisfies ϕ.
As u satisfies $\exists x_{k+1}\psi$, there must be an a in A such that $u_1 = u[x_{k+1}/a]$ satisfies ψ.
Now suppose player I starts playing the game $G_{n+1}(u, u')$ by moving $x_{k+1} \mapsto a$. As player II has a winning strategy in the game, he has a countermove $x_{k+1} \mapsto a'$ such that he has a winning strategy in $G_n(u_1, u'_1)$.
By the induction hypothesis, u_1 and u'_1 are $L_{\omega,n}$- equivalent, whence $u'_1 = u'[x_{k+1}/a']$ satisfies ψ. This implies that u' satisfies ϕ.

For the other direction, suppose that the first player has a winning strategy in $G_{n+1}(u,u')$. We must show that u and u' are not $L_{\omega,n+1}$-equivalent. Now suppose the winning strategy of I has $x_{k+1} \mapsto a'$ as a first move, with a' in A'. Write $u'_1 = u'[x_{k+1}/a']$. Whatever II's response $x_{k+1} \mapsto a$ is, player I has a winning strategy in $G_n(u_a, u'_1)$, where $u_a = u[x_{k+1}/a]$. Then by the induction hypothesis there is a formula ψ_a in $L_{\omega,n}$ such that $\mathbf{A}' \models \psi_a[u'_1]$, $\mathbf{A} \not\models \psi_a[u_a]$. Now the set of $L_{\omega,n}$-formulas with free variables in $\{x_1,\ldots,x_{k+1}\}$ is finite modulo equivalence, so there is a finite conjunction ψ of $L_{\omega,n}$-formulas such that ψ is equivalent to $\bigwedge_{a\in A}\psi_a$.
Then ψ itself is in $L_{\omega,n}$ and u' satisfies $\exists x_{k+1}\psi$ while u does not. So u and u' are not $L_{\omega,n}$-equivalent.

3 Logical operations on binary relations.

Definition 3.1.
Let $Re(U) = \{R | R \subseteq U \times U\}$ be the set of all binary relations on a universe U. For $(s,t) \in R$ we will write sRt. The following three relations play an important rôle in the theory of binary relations: the universal relation $V = U \times U$, the null relation $\emptyset$ and the identity relation $Id = \{(t,t) | t \in U\}$.
A proper, augmented relation algebra or $PARA$ is an algebra $(Re(U), H)$ with H a set of operations on binary relations. (The above-mentioned relations can and will be seen as constant or nullary relations.) The set of all operations generated by H (in the usual algebraic sense) is denoted by $gen(H)$. A set C of operations is called a clone if $C = gen(C)$.
The simplest operations of a $PARA$ are the Boolean set-theoretical operations: intersection $(R \cap S)$, union $(R \cup S)$ and complementation (R^c). The Boolean clone $C_b(U)$ is defined as $gen\{\cap, \cup, ^c\}$. It is easy to show that $C_b(U) = gen(\{\cap, ^c\})$; one may even prove that the Boolean clone is generated by a single operation.
The classical clone $C_c(U)$ is defined as the set of operations generated by $\{\cap, ^c, \circ, \breve{}\}$, where the composition $\circ$ and converse $\breve{}$ are defined by

$$R \circ S = \{(s,t) | \exists u (sRu \text{ and } uSt)\}$$
$$R\breve{} = \{(s,t) | tRs\}$$

The composition operation can be generalized to operations Q_n of rank n^2. For an $n \times n$-matrix of relations $\vec{R}$, we define

$$u_1 Q_n(\vec{R}) u_2 \Leftrightarrow \exists u_3 \ldots \exists u_n (\bigwedge_{1\le i,j\le n} u_i R_{ij} u_j)$$

The Q-clone $C_Q(U)$ is defined as the set of operations generated by the union of the classical clone and the Q_n-operations.

We may equivalently describe the Q-clone as $gen\{\cap,^c, Id, Q_2, Q_3, \ldots\}$, as the composition and converse operations are in the Q-clone:
$S^{\smile} = Q_2(\vec{R})$ where $\vec{R}$ is the 2×2-matrix of relations with $R_{21} = S$ and all other R_{ij} are equal to the universal relation V, which is in the Boolean clone.
$S \circ T = Q_3(\vec{R})$ where $\vec{R}$ is the 3×3-matrix of relations with $R_{13} = S$, $R_{32} = T$ and $R_{ij} = V$ for the other pairs (i, j).

All the above-mentioned operations can be defined in terms of first-order logic; this inspires the following definition:

Definition 3.2.
Let U be a universe. Every formula $\phi \in L(x_1, x_2)$ using n relations symbols $P_1, \ldots, P_n$, defines an operation F_ϕ of rank n: for $R_1, \ldots, R_n$ binary relations on U, let $\mathbf{U}$ be the structure (U, I) with I an interpretation such that $I(P_i) = R_i$. Then F_ϕ is given by

$$sF_\phi(R_1, \ldots, R_n)t \Leftrightarrow \mathbf{U} \models \phi[s, t]$$

We call the set of all these operations the logical clone, $C_l(U)$.

Our definition of the logical clone is slightly different from, but equivalent to the one Jónsson gives. The proof of this equivalence can be found in [J2]; we reproduce the argument, as we need it to show that our logical clone is indeed a clone:

Lemma 3.3.
Every operation generated by the logical clone is defined by a first order formula.

Proof.
If an operation $H \in gen(C_l(U)$ is in the logical clone itself, there is nothing to prove. In the other case, suppose H is given by

$$H(R_1, \ldots, R_n) = F(G_1(R_{11}, \ldots, R_{1r(1)}), \ldots, G_m(R_{m1}, \ldots, R_{mr(m)}))$$

By the induction hypothesis, every $G_l(R'_1, \ldots, R'_{r(l)})$ has a defining formula $\psi(x_1, x_2, P'_1, \ldots, P'_{r(l)})$, and by assumption $F(R''_1, \ldots, R''_m)$ is defined by a formula $\phi(x_1, x_2, P''_1, \ldots, P''_m)$. Then H is defined by the formula ϕ_H which is obtained by taking ϕ and replacing every occurrence of a relation symbol P''_l in ϕ by $\psi_l(P_{l1}, \ldots, P_{lr(l)})$, making sure that the variables do not clash.

In the sequel we will only be concerned with logical operations, and drop the adjective 'logical'. Furthermore, we will frequently omit references to the universe U, and simply speak about 'the logical clone' without further ado. The justification for this sloppiness is that, when two formulas ϕ and ψ define the same operator on a universe U, they do so on every universe V. The easy proof for this well-known fact uses the Löwenheim-Skolem theorem (cf. Theorem 8.1

in [J1]).

In [J2], Jónsson raises the question whether a simple and natural generating set can be found for the logical clone. Some candidates are:
(1) finite sets of operations
(2) the Q-clone
(3) sets of operations whose rank is bounded by a $k \in \omega$.

For a *finite* universe U, Andréka and Németi show in [AN] that the set $\{\cap, ^c, Q_k\}$ generates $C_l(U)$ if $|U| = k$. So, from now on, we assume that the universe U is infinite. As early as 1941, Tarski announced in [T] a proof that a finite set of operations cannot generate the logical clone. (A proof of a more general version of this theorem can be found in [J].) Andréka and Németi proved in [AN] that the Q-clone does not generate $C_l(U)$ for infinite sets U either. In the next two sections we will give new proofs for these two facts. For an outline of our proofs, we need the following:

Definition 3.4.
Let S be a set of logical operations. Using the proof of lemma 3.3, we can easily show that every operation in $gen(S)$ has a first order formula that defines it. Let L_S be the set of $L(x_1, x_2)$-formulas ϕ such that ϕ defines an operation in $gen(S)$.

Theorem 3.5.
S generates $C_l(U)$ iff $L_S(x_1, x_2)$ is as expressive as $L(x_1, x_2)$.

Proof.
We only prove the direction from left to right (the other directions is straighforward).
Assume that S generates the logical clone. Let $\phi(x_1, x_2, P_1, \dots, P_n)$ be in $L(x_1, x_2)$. As F_ϕ is in $gen(S)$, there is a formula ψ in L_S defining F_ϕ. We then have, for any structure $\mathbf{U} = (U, I)$ with $I(P_i) = R_i$:

$$\begin{array}{ll} & \mathbf{U} \models \phi(x_1, x_2, P_1, \dots, P_n)[s, t] \\ \Leftrightarrow & sF_\phi(R_1, \dots, R_n)t \\ \Leftrightarrow & \mathbf{U} \models \psi(x_1, x_2, P_1, \dots, P_n)[s, t] \end{array}$$

So any $\phi \in L(x_1, x_2)$ has an equivalent in $L_S(x_1, x_2)$.

In the sections 4 and 5, we will define and use special Ehrenfeucht games to show that if S is an arbitrary finite set of operations, or the Q-clone, $L_S(x_1, x_2)$ is less expressive than $L(x_1, x_2)$. As was said before, these results are not new; the novelty of the present proof lies in its game-theoretical approach.

4 Finite generating sets and k-pebble games.

In this section we start with showing that if the logical clone is finitely generated, then there is a $k < \omega$ such that every L-formula $\phi(x_1, x_2)$ is equivalent to a formula $\phi'(x_1, x_2)$ in which only k different variables occur. (These variables may be bound by different quantifiers at different occurrences.) The proof we give is well-known: see e.g. [G] for the version concerning operators in temporal logics, or [TG], pp. 75-76, for the 3-variable fragment of a first order language with dyadic relation symbols.

Definition 4.1.
L_k is the set of L-formulas with at most k (possibly reused) variables $x_1, \ldots, x_k$. $L_{k,n}$ is the set of L_k-formulas of quantifier depth n.

Lemma 4.2.
If S is a finite set of operations on binary relations, then there is a k such that every operation in $gen(S)$ is defined by an $L_k(x_1, x_2)$-formula.

Proof.
Suppose $S = \{F_1, \ldots, F_s\}$, every F_i has rank $R(i)$ and is defined by ϕ_i, i.e.

$$(s, t) \in F_i(R_1, \ldots, R_{r(i)}) \text{ iff } \mathbf{U} \models \phi_i(x_1, x_2, P_1, \ldots, P_{r(i)})[s, t]$$

where $\mathbf{U} = (U, I)$ with $I(P_i) = R_i$. Let $k(i)$ be the number of variables occurring in ϕ_i and define $k = max\{k(i) | 1 \leq i \leq s\}$.
We prove that, for every operation $H \in gen(S)$, there is a formula $\phi_H \in L_k(x_1, x_2)$ defining H. The proof is by induction over the complexity of H.
If H is in S, say $H = F_i$, then H is defined by ϕ_i, and by definition of k, $\phi_i \in L_k(x_1, x_2)$.
So suppose

$$H(R_1, \ldots, R_m) = F_i(G_1(R_{11}, \ldots, R_{1m(1)}), \ldots, G_{r(i)}(R_{r(i)1}, \ldots, R_{r(i)m(r(i))}))$$

where every G_j has rank $m(j)$ and every R_{ij} is in $\{R_1, \ldots, R_m\}$.
The induction hypothesis is that every $G_j(R'_1, \ldots, R'_{m(j)})$ has a defining formula $\psi_j(x_1, x_2, P'_1, \ldots, P'_{m(j)})$ in $L_k(x_1, x_2)$.
It is not hard to see that we get a formula ϕ'_H defining H by taking ϕ_i and replacing every occurrence of a predicate P_j in ϕ_i by $\psi_j(P_{j1}, \ldots, P_{jm(j)})$.
To obtain an L_k-equivalent of ϕ'_H, we have to attune the free variables in the formulas ψ_j to the variables of P_j in ϕ_i:
Consider an occurrence p of P_j in ϕ_i. Suppose P_j occurs at p with the variables x_a and x_b (by definition of k, $1 \leq a, b \leq k$). Let σ be a permutation of $\{x_1, \ldots, x_k\}$ such that $\sigma(x_1) = x_a$, $\sigma(x_2) = x_b$. Now replace every occurrence of a (free or bound) variable y in ψ_j by $\sigma(y)$. So we get a formula $\psi_{j,p}(x_a, x_b)$ using only the variables $x_1, \ldots, x_k$.

The required formula ϕ_H is then obtained by simultaneously substituting in ϕ_i, for every occurrence p of every predicate P_j in ϕ_i, the formula $\psi_{j,p}$ for P_j. This finishes the proof of lemma 4.2.

We will now define a special version of the Ehrenfeucht game, in which there are only k pebbles for each player. This will mean that the players sometimes have to take pebbles from the board in order to continue the game, and that only k-configurations are generated. We reproduce the argument in [IK] that these k-pebble games exactly characterize the formulas with only k variables.

Definition 4.3.
Let (u, u') be a k-configuration over $\mathbf{A}$, $\mathbf{A}'$. The k-pebble game $G_{k,n}(u, u')$ is the Ehrenfeucht game $G_n(u, u')$ with the restriction that the first player has only k pebbles $x_1, \ldots, x_k$ at her disposal. The definition of a winning strategy in $G_{k,n}(u, u')$ is the same as for $G_n(u, u')$.

Theorem 4.4.
For any k-configuration (u, u'), u and u' are $L_{k,n}$-equivalent iff the second player has a winning strategy in $G_{k,n}(u, u')$.

Proof.
The equivalence is proved by induction over n. For $n = 0$, the claim is immediate by definition of a winning strategy. For $n > 0$, one can easily obtain the result by following the structure of the induction step in the proof of lemma 2.4. As both the variable bound by the newly added quantifier and the pebble with which player I starts the game must be in $\{x_1, \ldots, x_k\}$, everything runs smoothly.

Definition 4.5.
Consider the structure $\mathbf{Q} = (Q, <)$ of the rational numbers with the usual ordering. Let α be an ordinal with $\alpha \leq \omega$. An α-shuffle of $\mathbf{Q}$ is a partition $\{C_i | i < \alpha\}$ of $\mathbf{Q}$ into α subsets, from now on called the shuffle sets, each of which is dense in $\mathbf{Q}$. By this we mean that for all $i < \alpha$ and all elements p and r of Q such that $p < r$, there is a q in Q with $p < q < r$ and $q \in C_i$. A binary relation R on Q is α-shuffled on $\mathbf{Q}$, if R is an equivalence relation such that the set of R-equivalence classes forms a k-shuffle on $\mathbf{Q}$. By theorem 7.11 of [R], there is for every $\alpha \leq \omega$ an α-shuffled equivalence relation on Q.

Lemma 4.6.
There is no $k < \omega$ such that every $L(x_1, x_2)$-formula has an equivalent in $L_k(x_1, x_2)$.

Proof.
By lemma 4.4 it is sufficient to find two structures $\mathbf{A}$ and $\mathbf{A}'$ with a $2-$configura-

tion (u, u') such that u and u' are not equivalent, while the second player has a winning strategy in $G_{k,n}(u, u')$ for all n.
Define $\mathbf{A}$,$\mathbf{A'}$, u and u' as follows: $\mathbf{A} = (Q, <, R)$, $\mathbf{A'} = (Q, <, R')$ where R is $k+1$-shuffled and R' is k-shuffled; $u(x_1)$ and $u(x_2)$ belong to different shuffle-sets, as do $u'(x_1)$ and $u'(x_2)$.
Then u and u' are not equivalent, as u' satisfies $\forall x_3 \ldots \forall x_{k+1}(\bigvee_{1 \leq i,j \leq k+1} Rx_ix_j)$, while u does not.
The fact that the second player has a winning strategy in $G_{k,n}(u, u')$ is immediate from the following claim:

For any k-configuration (u, u') on $\mathbf{A}$,$\mathbf{A'}$, player II has a winning strategy in $G_{k,n}(u, u')$ if (u, u') is a local isomorphism.

The proof of the claim is by induction on n; the proof of the basic step is straightforward, so consider the case $n > 0$. Let (u, u') be a local isomorphism and assume that $\delta u = \delta u' = \{x_1, \ldots, x_k\}$, which means that at the beginning of the game all k pebbles are on the board. Assume also that player I starts playing in $\mathbf{A}$. (The other cases are simpler.)
Writing $S(i)$, resp. $S'(i)$ for the shuffle set $u(x_i)$ (resp. $u'(x_i)$) is in, one can easily show:

(†) (u, u') is a local isomorphism iff
for all i, j: $u(x_i) < u(x_j) \Leftrightarrow u'(x_i) < u'(x_j)$ and $S(i) = S(j) \Leftrightarrow S'(i) = S'(j)$

As she has only the k pebbles on the board at her disposal, with every possible move player I must lift a pebble x_i from the board, so for a moment only $k-1$ pebbles are left behind on structure $\mathbf{A}$.
If she places the selected pebble x_i on an already pebbled element $u(x_j)$ of A, the strategy for player II is clear: he should move $x_i \mapsto u'(x_j)$.
So suppose the pebble x_i is moved between $u(x_j)$ and $u(x_l)$ (the case in which she puts x_i on an element greater than all $u(x_j)$, can be treated likewise); let S be the shuffle set in which x_i is placed.
The second player, lifting pebble x_i from A', has to put it somewhere between the positions $u'(x_j)$ and $u'(x_l)$; this is well possible, though he has to be careful in which shuffle set S' to put x_i: if $S = S(m)$ for some $m \neq i$, x_i is to be put in $S'(m)$, of course. If S is different form all the $S(1)$,. . ., $S(i-1)$,$S(i+1)$,. . .,$S(k)$, then x_i must be placed in a shuffle set S' not appearing in the sequence $S'(1)$,. . .,$S'(i-1)$,$S'(i+1)$,. . .,$S'(k)$. Such a set S' exists, as there are k different shuffle sets partitioning A.
In both cases there is always an S'-element between $u'(x_j)$ and $u'(x_k)$, as each shuffle set is dense in $\mathbf{Q}$.
By the assumption that (u, u') is a local isomorphism and the characterization (†) of local isomorphisms, it will be clear that the new $k-$configuration (u_1, u'_1) is a local isomorphism as well. By the induction hypothesis then, II has a winning strategy in $G_{k,n-1}(u_1, u'_1)$.

So, the above sketched procedure yields a winning strategy for the second player in $G_{k,n}(u, u')$.
This finishes the proof of lemma 4.6.

Theorem 4.7.
No finite set of logical operations on binary relations generates the logical clone.

Proof.
By theorem 3.5 and the lemmas 4.2 and 4.6.

We refer to [V] for the closely related result that no finite set of temporal interval-operators can be functionally complete over a class of temporal structures, if that class contains the rationals with their usual ordering relation.

5 Q-formulas and Q-games.

In this section we first define the set of Q-formulas, which are the formulas defining an operation in the Q-clone (recall 3.1 for a definition of the Q-clone). Q-formulas will be defined inductively, in accordance with the fact that the Q-clone is defined as the set of operations generated by $\{\cap, ^c, Id, Q_i | i \leq \omega\}$: each inductive step in this definition is the syntactical counterpart to an application of one of the generating operations of the Q-clone. Along with the induction we define the existential rank $ER(\phi)$ and the Q-depth $QD(\phi)$ of a Q-formula ϕ. Intuitively, Q-formulas having existential rank k will correspond to operations in the clone generated by $\{\cap, ^c, Id, Q_1, \ldots, Q_k\}$; the meaning of the Q-depth will hopefully be clear by its definition.

Definition 5.1.
$L_Q(x_i, x_j)$, the set of Q-formulas in x_i and x_j, is inductively defined as follows, as is the existential rank and Q-depth of such formulas:
Any atomic L-formula Px_ix_j or $x_i = x_j$ is a Q-formula in x_i, x_j; its existential rank is 2 and its Q-depth is 0.
If ϕ and ψ are Q-formulas in x_i,x_j, then so are $\neg\phi$ and $\phi \wedge \psi$. Furthermore, $ER(\neg\phi) = ER(\phi)$,$QD(\neg\phi) = QD(\phi)$ and $ER(\phi \wedge \psi) = \max(ER(\phi), ER(\psi))$, $QD(\phi \wedge \psi) = \max(QD(\phi), QD(\psi))$.
If, for $1 \leq i, j \leq k$, ψ_{ij} is a Q-formula in x_i and x_j, then

$$\phi = \exists x_1 \ldots \exists x_{a-1} \exists x_{a+1} \ldots \exists x_{b-1} \exists x_{b+1} \ldots \exists x_k (\bigwedge_{1 \leq i,j, \leq k} \psi_{ij})$$

with $1 \leq a, b \leq k$ is in $L_Q(x_a, x_b)$, and
$ER(\phi) = \max(k, \max(\{ER(\psi_{ij})|1 \leq i, j \leq k\},$
$QD(\phi) = 1 + \max\{QD(\psi_{ij})|1 \leq i, j \leq k\}.$
The set of Q-formulas in x_i and x_j with existential rank k and Q-depth n is

denoted by $L_{Q,k,n}$.

Note that every Q-formula of existential rank k and Q-depth n is in $L_{k,(k-2)n}$. Note too that in Q-formulas, quantifier occurrences come in sequences; now an essential property of Q-formulas is that, for $\vec{p}$ such a sequence, only two variables occurring in the scope of (all the quantifiers of) $\vec{p}$ may be free, or bound by a quantifier occurrence above $\vec{p}$. This means that the following formula is a typical example of a non-Q-formula:

$$Pux \wedge \exists y(Pxy \wedge \neg\exists z(Puz \wedge Pxz \wedge Pyz))$$

because there are *three* variables in the scope of $\exists z$ which are free (viz. u and x) or bound by a quantifier occurence above $\exists z$ (viz. y).

Theorem 5.2.
(1) For $\phi \in L_{Q,k,n}$, the operation F_ϕ defined by ϕ is in $gen(\{\cap,^c, Q_1,\ldots, Q_k\})$.
(2) Every $F \in gen(\{\cap,^c, Q_1,\ldots, Q_k\})$ is defined by a $\phi \in L_{Q,k,n}(x_1, x_2))$.

Proof.
The tedious proof follows by a straightforward adaptation of 3.3 (cf. 4.2.), and is left to the reader.

Corollary 5.3.
(1) Every Q-formula defines an operation in the Q-clone.
(2) Every operation in the Q-clone is defined by a Q-formula.

Now we want a characterization of Q-formulas in terms of a special Ehrenfeucht game. This Q-game will consist of a number of rounds in each of which another special Ehrenfeucht game is played which will be defined first:

Definition 5.4.
$L_{E,k}(x_i, x_j)$ is the set of k-existential formulas in x_i and x_j, is defined as $L_{Q,k,1}(x_i, x_j)$, i.e. those formulas of the form

$$\exists x_1 \ldots \exists x_{i-1} \exists x_{i+1} \ldots \exists x_{j-1} \exists x_{j+1} \ldots \exists x_k (\bigwedge_{1\le a,b\le n} \psi_{ab})$$

where every ψ_{ab} is a Boolean combination of atomic formulas with free variables x_a, x_b.
Let (u, u') be a k-configuration with $\delta u = \delta u' = \{x_i, x_j\}$. The k-existential Ehrenfeucht game $G_{E,k}(u, u')$ is the ordinary Ehrenfeucht game in which the first player may only use the $k-2$ pebbles in $\{x_1, \ldots, x_k\}$ differing from x_i and x_j, and has to place them on $\mathbf{A}$. Furthermore, the second player waits until I has put all her $k-2$ pebbles on the board before he responds with his $k-2$ moves.

II has a winning strategy in this game if the configuration generated at the end of the game is a local isomorphism.

Lemma 5.5.
Let (u, u') be a k-configuration with $\delta u = \delta u' = \{x_i, x_j\}$. Then II has a winning strategy in $G_{E,k}(u, u')$ iff u' satisfies all k-existential formulas that u satisfies.

Proof.
Straightforward.

Definition 5.6.
Let (u, u') be a k-configuration with $\delta u = \delta u' = \{x_i, x_j\}$. For notational simplicity, assume $i = 1$ and $j = 2$. The Q-game $G_{Q,k,n}(u, u')$ is defined as follows:
If $n = 0$, there is nothing to play.
If $n > 0$, the game consists of n rounds of existential games. At the start of the first round, the first player chooses a structure, say $\mathbf{A}$, and a number $m \leq k$. Then the game $G_{E,k}(u, u')$ is played. At he end of the first round, the first player chooses two colors, say x_i and x_j, and removes all pebbles from the board except those colored x_i and x_j. This creates a k-configuration (v, v') with $\delta v = \delta v' = \{x_i, x_j\}$. Now the game $G_{Q,k,n-1}(v, v')$ is played.
A winning strategy in this game for the second player is a strategy in which every generated configuration is a local isomorphism.

Theorem 5.7.
Player II has a winning strategy in $G_{Q,k,n}(u, u')$ iff u and u' are $L_{Q,k,n}$-equivalent.

Proof.
The proof is by induction on n; we only consider the induction step.

From left to right:
Suppose for some $\phi \in L_{Q,k,n+1}(x_1, x_2)$, u satisfies ϕ, while u' does not; without loss of generality we may assume that ϕ has the form $\exists x_3 \ldots \exists x_k \psi$, where $\psi \equiv \bigwedge_{1 \leq i,j \leq k} \psi_{ij}$, with $\psi_{ij} \in L_{Q,k,n}(x_i, x_j)$.
Writing $u(x_1) = a_1$, etc. we have the following winning strategy for I in $G_{Q,k,n+1}(u, u')$: in the first round she puts the pebbles $x_3, \ldots, x_k$ on elements $a_3, \ldots, a_k$ such that $\mathbf{A} \models \psi[a_1, \ldots, a_k]$.
Let $x_3 \mapsto a'_3, \ldots, x_k \mapsto a'_k$ be the answer of II.
As u' does not satisfy ϕ, $\mathbf{A}' \not\models \psi[a'_1, \ldots, a'_k]$; so there is a pair a'_i, a'_j in A' with $\mathbf{A}' \not\models \psi_{ij}[a'_i, a'_j]$. Note that $\mathbf{A} \models \psi_{ij}[a_i, a_j]$. Player I then ends this round by removing all pebbles from the board, but she leaves x_i and x_j behind. Let (v, v') be the arisen configuration.
By induction hypothesis, I has a winning strategy in $G_{Q,k,n}(v, v')$; but then she has a winning strategy in $G_{Q,k,n+1}(u, u')$ as well.

From right to left:
Suppose u and u' satisfy the same $L_{Q,k,n+1}$-formulas, then we must define a winning strategy for player II in $G_{Q,k,n+1}(u,u')$. Assume that in the first round, player I chooses $\mathbf{A}$ as her structure and $k-2$ as the number of pebbles to move. To define II's strategy, it is convenient to extend the language L with (finitely many!) new relation symbols.
First have a look at $L_{Q,k,n}(x_1,x_2)$; as this set is contained in $L_{\omega,kn}$, it is finite modulo equivalence; let $\chi_1,\ldots,\chi_m \in L_{Q,k,n}(x_1,x_2)$ be such that every $\psi \in L_{Q,k,n}(x_1,x_2)$ has an equivalent in $\{\chi_1,\ldots,\chi_m\}$. We now introduce m new relation symbols $P_1,\ldots,P_m$ into our language and set

$$I(P_l) = \{(a,b) | \mathbf{A} \models \chi_l[a,b]\}$$

and likewise for $I'(P_l)$. It is straightforward to verify that for every ψ in $L_{Q,k,n}(x_1,x_2)$, there is a relation symbol P_l with $I(P_l) = \{(a,b)|\mathbf{A} \models \psi[a,b]\}$. By assumption that u and u' are $L_{Q,k,n+1}$-equivalent, we then know that with respect to the new language, u' satisfies all the k-existential formulas that u satisfies. As we have introduced finitely many new relation symbols, lemma 5.5 now gives us

(*) II has a winning strategy for $G_{E,k}(u,u')$ in the expanded structures.

Now let player II make his moves in the first round of $G_{Q,k,n+1}(u,u')$ according to this strategy. By (*), the k-configuration (v,v') reached at the end of this first round is a local isomorphism. So when I decides to leave the pebbles x_i and x_j on their place, we know that (v,v') satisfy the same atomic formulas in the extended language, whence they are $L_{Q,k,n}$-equivalent.
By induction hypothesis then, player II has a winning strategy for the remainder of the game. This means that altogether II has a winning strategy in $G_{Q,k,n+1}(u,u')$.

This finishes the proof of theorem 5.7.

Lemma 5.8.
$L_Q(x_1,x_2)$ is less expressive than $L(x_1,x_2)$.

Proof.
By the game-theoretical characterization of Q-formulas, it is sufficient to give a 2-configuration (u,u') such that u and u' are not equivalent, while I cannot win $G_{Q,k,n}(u,u')$ for any pair (k,n). First we define the structures $\mathbf{A}$ and $\mathbf{A}'$:

$\mathbf{A}'$ is defined as $(Q',<',R')$, where $(Q',<')$ is isomorphic to the ordering of the rationals and R' is ω-shuffled, i.e. R' is an equivalence relation on Q' having ω equivalence classes $E_1,E_2,\ldots$, each of which is dense in $(Q',<')$ (cf. definition

4.5).

The structure **A** is also based on a countable dense linear ordering without endpoints; let $(Q_1, <_1)$ and $(Q_2, <_2)$ be isomorphic copies of $(Q, <)$. The lexicographical ordering $<$ on $Q_1 \times Q_2$ is defined by $(q_1, r_1) < (q_2, r_2)$ if $q_1 <_1 q_2$ or $q_1 = q_2$ and $r_1 <_2 r_2$.
Define, for $q \in Q_1$, $A(q) = \{q\} \times Q_2$; then $(A(q), <)$, as a subordering of $(Q_1 \times Q_2,<)$, is isomorphic to $(Q, <)$. Let there be given an ω-shuffle $\{C_i | i \in \omega\}$ over $(Q_1, <_1)$. Next, let there be given, for every $q \in Q_1$, an ω-shuffle over $(A(q), <)$. If $i \in \omega$ is such that $q \in C_i$, name these shuffle sets $D_{q,0}$,$D_{q,1}$,. . .,$D_{q,i-1}$,$D_{q,i+1}$,$D_{q,i+2}$,. . .
Then, set for $i \in \omega$, $D_i = \bigcup_{q \in Q_1} D_{q,i}$. Now $\{D_i | i \in \omega\}$ is a partitioning of $Q_1 \times Q_2$ such that:

(†) if $q \in C_i$, then $A(q) \cap D_i = \emptyset$
(‡) if $q \notin C_i$, then D_i is dense in $A(q)$.

Finally, define $\mathbf{A} = (Q_1 \times Q_2, <, R)$ where R is the equivalence relation of which of which the D_i form the equivalence classes.

R is almost an ω-shuffle over $(A, <)$; yet not all equivalence classes D_i of R are dense: for example, consider (q, r_1) and (q, r_2) such that $r_1 <_2 r_2$ and $q \in C_i$. Let (q_3, r_3) be such that $q <_1 q_3$ and $(q_3, r_3) \in D_i$. Then by (†) no (q, r_4) with $r_1 <_2 r_4 <_2 r_2$ can be in D_i, so for no such element can we have $(q_3, r_3)R(q, r_4)$. So the partial valuation u defined by $u(x_1) = (q, r_1)$, $u(x_2) = (q, r_2)$ satisfies the following formula ϕ:

$$x_1 < x_2 \wedge \exists x_3 (x_2 < x_3 \wedge \forall x_4 (x_1 < x_4 < x_2 \leftrightarrow \neg R x_3 x_4))$$

Now look at any partial valuation u on $\mathbf{A}'$ satisfying $u'(x_1) < u'(x_2)$. As every equivalence class is dense in $\mathbf{A}'$, u' cannot satisfy ϕ. So u and u' are not $L(x_1, x_2)$-equivalent. But we can easily choose u' such that (u, u') is a local isomorphism, by taking $(u'(x_1), u'(x_2))$ in R' iff $(u(x_1), u(x_2))$ is in R. The fact that II then has a winning strategy in $G_{Q,k,n}(u, u')$ is an immediate consequence of the claim below.
First however, the reader is invited to act the part of the first player in an *ordinary* Ehrenfeucht game, in order to show that **A** and **A′** are different. She will notice that at a certain moment during the game, she has to switch from moving pebbles on **A′** to moving on **A**, and that she needs to leave *three* pebbles behind on **A′** before this step. Now this constitutes precisely a strategy which is forbidden in a Q-game, as switching boards means starting a new round in a Q-game, whence only *two* pebbles may be left behind. (Compare this to the 'essential property' of Q-formulas described after their definition 5.1).

We can now finish the proof of this lemma by proving the following claim:

II has a winning strategy in $G_{Q,k,n}(u,u')$ if (u,u') is a local isomorphism.

We only treat the induction step of the proof; assume that (u,u') is a local isomorphism and that I is about to move her $k-2$ pebbles in the first round of $G_{Q,k,n+1}(u,u')$. To prove the existence of a winning strategy for II, it is by the induction hypothesis sufficient to show that he can reach a situation after the first round such that the arisen k-configuration (v,v') is a local isomorphism.

The case in which I places her pebbles in $\mathbf{A}$ is simple for II by the fact that R' is ω-shuffled over $\mathbf{A'}$ (cf. the proof of lemma 4.6), so assume that I chooses to play in $\mathbf{A'}$. We will show that there is a subset B of A such that B contains $u(x_1)$ and $u(x_2)$ and the induced substructure $(\mathbf{B}, u(x_1), u(x_2))$ is isomorphic to $(\mathbf{A'}, u'(x_1), u'(x_2))$. If the second player then answers I's moves by putting pebbles on those B-elements which are the images of $v'(x_3), \dots, v'(x_k)$ under the isomorphism, he has a simple winning strategy by the isomorphism. Now, distinguish the following cases:

(1) For some $q \in Q_1$, $u(x_1)$ and $u(x_2)$ are both in $A(q)$. Then set $B = A(q)$, which is an ω-shuffled, countable, unbounded, dense, linear order. As (u,u') was a local isomorphism, this means $(\mathbf{B}, u(x_1), u(x_2))$ is isomorphic to $(\mathbf{A'}, u'(x_1), u'(x_2))$.
(2) Let $u(x_1) = (q_1, r_1)$ and $u(x_2) = (q_2, r_2)$ be such that $q_1 < q_2$. Take an m with $q_1, q_2 \notin C_m$ and $(q_1,r_1), (q_2,r_2) \notin D_m$. As C_m is dense in Q_1, there are elements p_1, p_2, p_3 in Q_1 with $p_1 <_1 q_1 <_1 p_2 <_1 q_2 <_1 p_3$ and $p_1, p_2, p_3 \in C_m$. The latter fact implies $D_m \cap A(p_i) = \emptyset$ by (†).
Define $B = A(p_1) \cup \{(q_1,r_1)\} \cup A(p_2) \cup \{(q_2,r_2)\} \cup A(p_3)$. Then B with the ordering induced by A, is isomorphic to $(Q', <')$. Further, notice that, as (q_1,r_1) and (q_2,r_2) are not in D_m, and $D_m \cap A(p_i) = \emptyset$, the set $\{D_l \cap B | l \neq m\}$ partitions B. Notice too that for $l \neq m$, p_1, p_2 and p_3 are not in C_l, so by (‡) D_l is dense in B for $l \neq m$. But then $\{D_l \cap B | l \neq m\}$ is an ω-shuffle over B. As (u,u') was a local isomorphism, $(\mathbf{B}, u(x_1), u(x_2)) \cong (\mathbf{A'}, u'(x_1), u'(x_2))$.

This finishes the proof of lemma 5.7.

Theorem 5.9.
The Q-clone does not generate the logical clone.

Proof.
By 3.5, 5.3 and 5.8.

6 Cyclic games for k-bound formulas.

In [vB], some fragments of first order logic are discussed which are to capture phenomena of natural language. As an example, it is argued that

> bindings in most natural languages cannot cross arbitrary layers of operators in a sentence.

In this section a syntactic and game-theoretic characterization is given of first order formulas that satisfy such a constraint. It comes out that these formulas bear a resemblance to the $L_{k,n}$-formulas of section 4. (Compare this with the fact that natural languages only have a finite number of variables!) The exact relation between both sets of formulas will be elaborated on in the the next section.

Definition 6.1.
In a formula ϕ, consider an occurrence p of a variable x which is bound by a quantifier occurrence Q. A quantifier occurrence Q' is between p and Q if p is in the scope of Q' and Q' is in the scope of Q. The binding depth of p is defined as the number of quantifier occurrences between p and Q. A formula ϕ is k-bound if all variable occurrences have a binding depth smaller than k.

As an example, in the formula $\forall v \forall x(Pxv \rightarrow (\exists y Pyx \wedge \forall z Qzx))$ the existential quantifier occurrence $\exists y$ is not between the x in Qzx and its binding quantifier occurrence $\forall x$; so ϕ is 2-bound. The bindingsdepth of a bound occurrence p of a variable in a formula ϕ may also be described using the construction tree of ϕ, where the binding depth is the number of quantifier nodes one encounters going up from p to the quantifier node where the variable of p is bound.
In the sequel, we will confine ourselves to k-bound *sentences*. The reason for this is that k-bound formulas are hard to define inductively, which makes their game characterization less elegant. (To give an indication of the problem: where ϕ : $\exists x(Pxx \wedge \exists y(Rxy \wedge Rxv))$ is 2-bound, $\exists v \phi$ is not.) We might solve this problem by treating the free variables of a k-bound formula as constants, but this makes things rather messy.

Here we will give a game-theoretic characterization of k-bound formulas. First we will prove that every k-bound formula has an alphabetical variant which is k-cyclic. Intuitively spoken, k-cyclic formulas are those first order formulas satisfying the following constraint: if Q and Q' are quantifier occurrences in ϕ such that Q' is in the scope of Q, while there is no quantifier occurrence between Q' and Q, then Q' binds x_i if Q binds x_{i+1}, and Q' binds x_k if Q binds x_1. For example: $\exists x_1 \forall x_3 \exists x_2 (Px_2x_3 \wedge \exists x_1 Px_1x_2 \wedge \exists x_1(Px_1x_3 \wedge \forall x_3 Px_1x_3)))$ is 3-cyclic, $\forall x_3(\exists x_2 Px_2x_3 \wedge \exists x_1 Px_2x_1)$ is not.

Definition 6.2.

Let k be a fixed natural number. For any natural number i, let $i*$ be the number j satisfying $1 \leq j \leq k$ and $j - i$ is divisible by k. Examples: $(i + k)* = i*$, $(k + 1)* = 1$, $0* = k$.
Now let i be in $\{1, \dots, k\}$. $C(k, i, n)$ will denote the set of k-cyclic formulas ϕ of quantifier depth n such that ϕ is a Boolean combination of formulas of which the uppermost quantifier binds x_i. Formally, by induction on n:
$C^0(k, i, 0)$ is the set of atomic formulas with free variables in $\{x_1, \dots, x_k\}$.
$C^0(k, i, n + 1) = \{\exists x_i \psi | \psi \in C(k, (i - 1)*, n)\}$.
$C(k, i, n)$ is the closure of $C^0(k, i, 0) \cup \ldots \cup C^0(k, i, n)$ under Boolean operators.

Lemma 6.3.
Every k-bound sentence of quantifierdepth n has an alphabetical variant in $C(k, n*, n)$.

Proof.
First consider the following alphabetical variant ϕ' of ϕ: any quantifier occurrence Q binding a variable y is replaced by a quantifier binding x_{n-i}, where i is the number of quantifiers above Q. Of course, the bound variables are renamed correspondingly. An example:

$$\psi : \forall u \exists v \forall w((\forall x(Rux \vee \exists y(Rvy \wedge Rwx)) \vee \exists z Ruz)$$

(which is 4-bound and has quantifier depth 5) turns into

$$\psi' : \forall x_5 \exists x_4 \forall x_3((\forall x_2(Rx_5x_2 \vee \exists x_1(Rx_4x_1 \wedge Rx_3x_2)) \vee \exists x_2 Rx_5x_2)$$

Then, rename every (bound or binding) variable x_m in ϕ' by x_{m*}, to obtain $\phi*$; the above ψ' then becomes

$$\psi* : \forall x_1 \exists x_4 \forall x_3((\forall x_2(Rx_1x_2 \vee \exists x_1(Rx_4x_1 \wedge Rx_3x_2)) \vee \exists x_2 Rx_1x_2)$$

It will be clear that in general, the obtained $\phi*$ is in $C(k, n*, n)$; we still have to check that it is an alphabetical variant of ϕ. Now the only conceivable problem is of the following kind: some variable $x_{(m+k)*}$ is in $\phi*$ bound by a quantifier Qx_{m*}, while its original x_{m+k} in ϕ' was bound by Qx_{m+k}. But in this case there would be k quantifier occurrences (viz. $Qx_{m+k-1}, \dots, Qx_{m+1}, Qx_m$) between the occurrence of x_{m+k} and its binding quantifier, and this is precisely the situation forbidden in k-bound formulas.
In the example: in ψ', x_5 cannot occur in the scope of $\exists x_1$, so in $\psi*$, x_{5*} can never get bound by $\exists x_{1*}$.

We will now give a game-theoretic treatment of the subject, defining the notion of a cyclic Ehrenfeucht game with k pebbles. The idea of this game is that the pebbles are used in a cyclic order, so that every pebble is reused after exactly k moves.

Definition 6.4.
For a k-configuration (u, u'), $G_{C,k,n}(u, u')$ is the k-pebble Ehrenfeucht game $G_{k,n}(u, u')$ with the following restriction: the first player must use the pebbles in the following order: $x_{n*}, x_{(n-1)*}, \ldots, x_2, x_1$.

This restriction is just the game-theoretical counterpart of the syntactic restriction to k-cyclic formulas:

Lemma 6.5.
For all k-configurations (u, u'), u and u' are $C(k, n*, n)$-equivalent iff player II has a winning strategy in $G_{C,k,n}(u, u')$.

Proof.
The equivalence is proved by induction on n. For $n = 0$, the claim is immediate by definition of a winning strategy. For $n > 0$, one obtains the result by following the structure of the proof of lemma 2.4 and observing that both the outermost quantified variable of each $C(k, (n+1)*, n+1)$-formula and the first pebble to be used in $G_{C,k,n+1}(u, u')$ are called $x_{(n+1)*}$.

The above lemmas 6.3 and 6.5 now immediately yield the following characterization of k-bound sentences:

Theorem 6.6.
Let **A** and **A'** be two structures. Then **A** and **A'** satisfy the same k-bound sentences of quantifierdepth n iff the second player has a winning strategy in $G_{C,k,n}(\emptyset, \emptyset)$.

7 Sentences with k variables and k-cyclic sentences.

As was said before, in this section we discuss the relation between formulas which have a bounded number of variables and k-bound formulas. If we confine ourselves to sentences, it turns out that a sentence in k variables does not need to have an equivalent which is k-bound, yet it does have an equivalent $(k+1)$-cyclic sentence.

Theorem 7.1.
Not every sentence in L_k has a k-cyclic equivalent.

Proof.
Consider the following structures $\mathbf{M}$ and $\mathbf{M}'$: both are connected, non-directed,

a-cyclic, countably infinite graphs. The difference is that in **M** two edges meet in every vertex, in $\mathbf{M}'$ three. This difference is captured by the following formula, holding in $\mathbf{M}'$ not in **M**:

$$\exists x_1 x_2 x_3 (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \wedge \exists y(Rx_1y \wedge Rx_2y \wedge Rx_3y)).$$

Because the graphs are a-cyclic, the property expressed by the above formula can be 'approximated' by the following L_3-sentence ϕ, which is true in $\mathbf{M}'$, not in **M**:

$$\begin{aligned} \exists x_1 \exists x_2 \exists x_3\ (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \\ \wedge\quad \exists x_1(Rx_1x_2 \wedge Rx_1x_3) \\ \wedge\quad \exists x_2(Rx_1x_2 \wedge Rx_2x_3) \\ \wedge\quad \exists x_3(Rx_1x_3 \wedge Rx_2x_3) \end{aligned}$$

We will now show that **M** and **M'** satisfy the same 3-cyclic sentences, by playing the cyclic game $G_{C,3,n}(\emptyset, \emptyset)$.
For the intuition behind II's winning strategy in this game, call the length of the shortest path between two elements of a graph, their distance. We denote the distance between $u(x_i)$ and $u(x_j)$ by $d(x_i, x_j)$ if no confusion arises concerning the partial valuation u involved. Now look at an arbitrary moment in the game $G_{C,3,n}(\emptyset, \emptyset)$, when we have a 3-configuration (v, v') with $\delta v = \delta v' = \{x_1, x_2, x_3\}$. If x_i is the pebble which I must move at this moment, call x_j and x_k the remaining pebbles. The winning strategy of II consists of creating only 3-configurations in which the distance of the remaining pebbles in **M** is the same as the distance of the remaining pebbles in $\mathbf{M}'$.
More formally, we will now prove the following claim:

If (u, u') is a local isomorphism and $d(x_{(n-1)*}, x_{(n-2)*}) = d'(x_{(n-1)*}, x_{(n-2)*})$, then II has a winning strategy in $G_{C,k,n}(u, u')$.

The basic step in the inductive proof of the claim is straightforward, as usual. For the inductive case $n > 0$, suppose for notational simplicity that $x_{n*} = x_3$, $x_{(n-1)*} = x_2$ and $x_{(n-2)*} = x_1$. By the induction hypothesis, it is sufficient for II to reach a configuration (u_1, u_1') after the first move which is a local isomorphism satisfying $d_1(x_1, x_3) = d_1'(x_1, x_3)$. Note that by our notational convention, x_1 and x_2 are the remaining pebbles, so $d_1(x_1, x_2) = d(x_1, x_2)$.
Now distinguish the following cases, according to I's move:

If I makes her move in **M**, consider a maximal path in $\mathbf{M}'$ through $u'(x_1)$ and $u'(x_2)$. As $d_1(x_1, x_2) = d_1'(x_1, x_2)$ by assumption, this path, considered as a subgraph $\mathbf{N}'$ of $\mathbf{M}'$, is such that $(\mathbf{N}', u'(x_1), u'(x_2))$ and $(\mathbf{M}, u(x_1), u(x_2))$ are isomorphic, so this move of I means no problem for II.

So consider the option in which I makes her move in $\mathbf{M}'$; distinguish the following cases:

(1) There is an acyclic maximal path in $\mathbf{M}'$ on which x_1, x_2 and x_3 are situated. As $d_1(x_1, x_2) = d'_1(x_1, x_2)$, this path (considered as a substructure $\mathbf{N}'$ of $\mathbf{M}'$), is just like the above described path. So again II has an easy job.
(2) If there is no such path, then observe that no pair of pebbles in M' can be direct neighbours; in particular, $u_1(x_3)$ and $u_1(x_2)$ are not adjacent.
Now look at $u_1(x_1)$. Let's say that $u_1(x_2)$ is situated on the *left* side of $u_1(x_1)$. ($\mathbf{M}$ may be considered as a copy of the integers with $Rz_1z_2 \Leftrightarrow |z_1 - z_2| = 1$.) As x_3 and x_2 are not neighbours in $\mathbf{M}'$, player II has the freedom to place x_3 on the *right* side of x_1. Now if he moves like this, and places his pebble x_3 so that $d_1(x_1, x_3) = d'_1(x_1, x_3)$, he creates the required local isomorphism.

An (almost) immediate consequence of this claim is that $\mathbf{M}$ and $\mathbf{M}'$ satisfy the same k-cyclic sentences, so we have proved lemma 7.1.

We have now come to our last result: we will show that every first order sentence in which only k variables are used, has a $k+1$-cyclic equivalent. The proof is based on the fact that a winning strategy for the first player in $G_{k,n}(\emptyset, \emptyset)$ can be modified into a winning strategy for her in the game $G_{C,k+1,kn}(\emptyset, \emptyset)$. To prove this, we need a more general definition of the k-cyclic game:

Definition 7.2.
Let $G_{C,k,n,i}(u, u')$ be the cyclic game in which the first player is obliged to start with the pebble x_i (and then move $x_{(i-1)*}, \ldots$).

Note that for sentences, this change does not effect the characterizing power of the game. Furthermore, we need the following definition:

Definition 7.3.
Let (u, u') be a k-configuration, (v, v') a $k+1$-configuration and $i < k+1$. We call (v, v') an i-extension of (u, u') if there is, for all $j \leq k$, a $j' \leq k+1$ such that $j' \neq i$, $u(x_j) = v(x_{j'})$ and $u'(j) = v'(x_{j'})$.

Intuitively, (v, v') is an i-extension of (u, u') if (v, v') contains all the information of (u, u'), even without the pair $(v(x_i), v'(x_i))$. We can now prove our last result:

Theorem 7.4.
Every sentence in $L_{k,n}$ has an equivalent in $L_{C,k+1,kn}$.

Proof.
By the above given game-theoretical characterization of $L_{k,n}$ and $L_{C,k+1,kn}$, and the remark below definition 7.2, it is sufficient to prove the following claim:

Suppose (u, u') is a k-configuration, (v, v') is an i-extension of (u, u') and player I has a winning strategy in $G_{k,n}(u, u')$. Then she has a winning strategy in

$G_{C,k+1,kn,i}(v, v')$.

The proof is, as always, by induction on n, and again as always, the basis step is trivial.
So suppose, for the induction step, that $x_j \mapsto a$ is the first move in the winning strategy of I in $G_{k,n+1}(u, u')$. Let j' be as in definition 7.3. I would like to start the cyclic game by moving $x_{j'} \mapsto a$, but unfortunately she must first use x_i, x_{i-1},…,$x_{j'+1}$. (we drop the * from the indices.) Her solution is the following trick: in $G_{C,k+1,k(n+1),i}(v, v')$, she moves:

$$x_i \mapsto v(x_{i-1}), x_{1-1} \mapsto v(x_{i-2}), \ldots, x_{j'+2} \mapsto v(x_{j'+1}).$$

Because she puts with every move a pebble on top of another one, II is forced to move the corresponding pebbles on $v'(x_{i-1}), v'(x_{i-2}), \ldots, v'(x_{j'+1})$. The resulting configuration is a $j' + 1$-extension of (u, u').
Now I plays: $x_{j'+1} \mapsto a$.
Suppose II answers with $x_{j'+1} \mapsto a'$.
Let (w, w') be the now arisen $k+1$-configuration in the $k+1$-cyclic game. Notice that (w, w') is a j'-extension of $(u[x_{j'}/a, u'[x_{j'}/a'])$.
By assumption I has a winning strategy in $G_{k,n}(u[x_{j'}/a], u'[x_{j'}/a'])$ so the induction hypothesis gives her a winning strategy in $G_{C,k+1,kn,j'}(w, w')$.
The claim then follows by the fact that it took at most k moves to reach (w, w').

This finishes the proof of theorem 7.4.

8 Literature.

References

[AK] Andréka, H. and Németi, I., *On the relation algebraic clone generated by Jónsson's Q's*, Mathematical Institute of the Hungarian Academy of Sciences, Budapest, 1985, 10 pp.

[vB] van Benthem, Johan, "Logical Syntax", *Theoretical Linguistics* , **14** (1987) 119-142.

[G] Gabbay, Dov M., "Expressive Functional Completeness in Tense Logic", in: U. Mönnich, ed. *Aspects of Philosophical Logic*, Dordrecht, 1981.

[IK] Immerman, Neil and Kozen, Dexter, "Definability with bounded number of bound variables", *Proceedings of the Symposium on Logic in Computer Science; Ithaka, New York*. Computer Society Press, Washington, 1987, 236-244.

[J1] Jónsson, Bjarni, "Varieties of relation algebras", *Algebra Universalis*, **15** (1982) 273-298.

[J2] Jónsson, Bjarni, "The theory of binary relations", *this volume*.

[R] Rosenstein, Joseph G., *Linear Orderings*, Academic Press, New York, 1982.

[T] Tarski, Alfred, "On the calculus of relations", *Journal of Symbolic Logic*, **6** (1941) 73-89.

[TG] Tarski, Alfred and Givant, Steven, *A formalization of set theory without variables*, AMS Colloquium Publications Vol **41**, Providence, Rhode Island, 1987.

[V] Venema, Yde, "Expressiveness and Completeness of an Interval Tense Logic", *Notre Dame Journal of Formal Logic*, to appear.