COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **19** (4) (1999) 507–532

EQUIVALENCE IN FINITE-VARIABLE LOGICS IS COMPLETE FOR POLYNOMIAL TIME

MARTIN GROHE*

Received March 23, 1998

How difficult is it to decide whether two finite structures can be distinguished in a given logic? For first order logic, this question is equivalent to the graph isomorphism problem with its well-known complexity theoretic difficulties. Somewhat surprisingly, the situation is much clearer when considering the fragments L^k of first-order logic whose formulas contain at most k variables (for some $k \geq 1$). We show that for each $k \geq 2$, equivalence in the logic L^k is complete for polynomial time. Moreover, we show that the same completeness result holds for the powerful extension C^k of L^k with counting quantifiers (for every $k \geq 2$).

The k-dimensional Weisfeiler–Lehman algorithm is a combinatorial approach to graph isomorphism that generalizes the naive color-refinement method (for $k \geq 1$). Cai, Fürer and Immerman [6] proved that two finite graphs are equivalent in the logic C^{k+1} if, and only if, they can be distinguished by the k-dimensional Weisfeiler-Lehman algorithm. Thus a corollary of our main result is that the question of whether two finite graphs can be distinguished by the k-dimensional Weisfeiler–Lehman algorithm is P-complete for each $k \geq 1$.

1. Introduction

Two finite structures are isomorphic if, and only if, they are equivalent in first-order logic (that is, satisfy the same first-order sentences). Based on this well-known observation, Immerman and Lander [19] suggested a "first-order approach to graph canonization". Instead of full first-order logic, they studied the fragments L^k of first-order logic consisting of all formulas with at most k variables and their extensions C^k by counting quantifiers such as "there are at least m elements x such that ...". For many important classes $\mathcal{C}$ of graphs there is a $k \geq 1$ such that two graphs in $\mathcal{C}$ are isomorphic if, and only if, they are equivalent in L^k (or C^k). Examples are the class of planar graphs [12] and all classes of graphs of bounded tree-width [13]. Immerman and Lander proved that for each fixed k there are polynomial time algorithms deciding whether two given graphs are equivalent in L^k or C^k, respectively. If equivalence in the respective logic coincides with isomorphism, this actually gives rise to a polynomial time canonization algorithm.

Mathematics Subject Classification (1991): 03C13, 68Q15, 05C60

* This research was done while the author visited the University of California in Santa Cruz; supported by a NATO Scientific Council grant administrated by DAAD.

Combinatorial algorithms for isomorphism testing very similar to the C^k-equivalence testing algorithm due to Immerman and Lander have been developed much earlier. Rather than testing isomorphism directly, these algorithms try to compute the orbits of the automorphism group of a given graph (which of course suffices for the purpose of isomorphism testing). The naive "color refinement method" starts by labeling each vertex of a graph by its degree. Then, in each iteration step, the labelling (or coloring) obtained after the previous step is refined by extending the label of each node by the multiset of labels of its neighbors. If the equivalence relation induced by the labelling is not refined in an iteration step, the algorithm stops after this step. On a finite graph, this eventually happens, yielding the so called stable coloring. The equivalence classes of vertices of the same color are called the color classes of the graph. Babai, Erdös, and Selkow [3] have shown that for almost all graphs the color classes have size one, and thus are the orbits of the automorphism group. However, there are obvious examples of graphs where this is not the case. For example, in a regular graph all vertices belong to the same color class. The method can be improved by coloring k-tuples of vertices rather than single vertices, for some $k \geq 1$. This yields the k-dimensional Weisfeiler-Lehman algorithm. It is far less trivial to find examples of graphs where this algorithm fails to produce the orbits of the automorphism group, but eventually Cai, Fürer, and Immerman [6] succeeded to do so. They could even find graphs of degree 3 and color class size 4 where the algorithm failed. Nevertheless, the Weisfeiler-Lehman algorithm is an important method to decide isomorphism of many graphs in polynomial time. For a thorough presentation of the background and history of this method we refer the reader to [6].

The connection between the Weisfeiler-Lehman algorithm and finite variable logics is made through the observation of Cai, Fürer, and Immerman [6] that two elements or tuples of elements of a finite graph can be distinguished by the k-dimensional Weisfeiler-Lehman algorithm if, and only if, they can be distinguished in the logic C^{k+1}.

In this paper, we analyze the complexity of equivalence checking in the finite variable logics. For each logic $\mathsf{L} \in \{\mathsf{L}^k, \mathsf{C}^k \mid k \geq 2\}$, we consider the following two problems:

> L-EQUIVALENCE
>
> *Instance:* Directed graphs $\boldsymbol{G}$ and $\boldsymbol{H}$.
>
> *Problem:* Are $\boldsymbol{G}$ and $\boldsymbol{H}$ equivalent in L?

> L-TYPE
>
> *Instance:* A directed graph $\boldsymbol{G}$ and vertices a,b of $\boldsymbol{G}$.
>
> *Problem:* Do a and b have the same L-type (that is, do they satisfy the same L-formulas with one free variable)?

The choice to consider directed graphs in these problems is inessential, but most convenient.

Theorem 1. *For logic* $\mathsf{L} \in \{\mathsf{L}^k, \mathsf{C}^k \mid k \geq 2\}$, *the problems* L-*EQUIVALENCE and* L-*TYPE are complete for polynomial time under uniform* AC_0*-reductions.*

The choice of uniform AC_0-reductions in the statement of the theorem is somewhat arbitrary, actually the statement remains true for the much weaker (but less known) quantifier-free reductions (see [10] for a proof), and of course our theorem implies the corresponding statement for logarithmic space or NC-reductions.

Remember that the question for the complexity of first-order equivalence, that is, isomorphism, is wide open, so it is quite remarkable that for its finite variable fragments we obtain such a clean picture.

There is another interesting aspect of the TYPE problem: We may view colored directed graphs as finite Kripke-structures. It is a well-known observation that two elements of a Kripke-structure having the same L^2-type are *bisimilar*. Actually, bisimilarity is very closely related to L^2-equivalence. On the other hand, C^k-equivalence for large k is close to isomorphism. So our finite variable logics give us a family of natural equivalence relations filling the gap between bisimilarity and isomorphism.

Let me remark that it has been proved by Balcázar, Gabarró, and Sántha [4] that bisimilarity is complete for polynomial time.

My original motivation for this work came from a different direction. Logics with finitely many variables have always been playing an important role in descriptive complexity theory (see, for example, [8], [11], [18]). One of the most important results in this area is the Abiteboul-Vianu Theorem [2] which translates the question of whether PTIME equals PSPACE to the purely logical question of whether *least fixed point logic* and *partial fixed-point logic* have the same expressive power. Abiteboul, Vardi, and Vianu [1] and Dawar [7] have obtained similar results for other complexity classes, but none of these involved classes below PTIME.

The proofs of all these results are based on the fact that equivalence in L^k can be defined in least fixed-point logic. An easy corollary of our theorem is that any logic able to define equivalence in L^k can already define all PTIME-queries on ordered structures. This offers an explanation why all attempts to get results analogous to the Abiteboul-Vianu Theorem for complexity classes below PTIME with similar methods failed.

2. Preliminaries

I assume that the reader is familiar with the fundamentals of complexity theory and logic. Below we review some basic notions and results needed here.

2.1. Complexity theory

For us, a *monotone circuit* is a tuple $\boldsymbol{C}=(C,E^{\boldsymbol{C}},A^{\boldsymbol{C}},T^{\boldsymbol{C}})$ such that the following holds: $(C,E^{\boldsymbol{C}})$ is a directed acyclic graph in which each vertex has in-degree 0 or 2. The vertices of in-degree 0 are called the *input nodes*; $T^{\boldsymbol{C}}$ is a subset of the set of input nodes. The vertices of in-degree 2 are called the *internal nodes*, $A^{\boldsymbol{C}}$ is a subset of the internal nodes.

For a monotone circuit $\boldsymbol{C}$ we inductively define a function $\mathrm{val}^{\boldsymbol{C}} : C \to \{\text{TRUE},\text{FALSE}\}$ by letting the input nodes in $T^{\boldsymbol{C}}$ be TRUE and the others FALSE, and treating the nodes in $A^{\boldsymbol{C}}$ as AND-nodes and the other internal nodes as OR-nodes.

Fact 2. (Ladner [20]) *The following problem is complete for polynomial time under* AC_0*-reductions:*

MONOTONE CIRCUIT VALUE (MCV)
Instance: A monotone circuit $\boldsymbol{C}$ *and a vertex* $c\in C$.
Problem: Is $\mathrm{val}^{\boldsymbol{C}}(c)=\text{TRUE}$?

(Ladner did not prove the result for AC_0-reductions, but this improvement is straightforward.)

2.2. Logic

A *vocabulary* is a set τ containing finitely many relation and constant symbols. A τ-*structure* $\boldsymbol{A}$ consists of a set A, called the *universe* of $\boldsymbol{A}$, an interpretation $R^{\boldsymbol{A}}\subseteq A^r$ of each r-ary relation symbol $R\in\tau$, and an interpretation $c^{\boldsymbol{A}}\in A$ of each constant symbol $c\in\tau$. We restrict our attention to structures whose universe is finite. For example, graphs are structures $\boldsymbol{G}=(G,E^{\boldsymbol{G}})$ where E is a binary relation symbol, colored graphs are structures $\boldsymbol{G}=(G,E^{\boldsymbol{G}},C_1^{\boldsymbol{G}},\dots,C_m^{\boldsymbol{G}})$ where E is a binary and $C_1,\dots,C_m$ are unary relation symbols, and monotone circuits are structures whose vocabulary consists of the binary relation symbol E and the unary symbols A,T.

Atomic formulas are of the form $Rt_1\dots t_r$ or $t_1=t_2$, where R is an r-ary relation symbol and the t_i are *terms*, that is, constant symbols or *variables*. The class of *first-order formulas* is the result of closing the atomic formulas under Boolean combinations and existential and universal quantification. The semantics of *first-order logic* is defined in the usual way. For example, the sentence $\forall x\forall y\forall z\neg(Exy\wedge Exz\wedge Eyz)$ says that a graph is triangle free.

Let $k \geq 2$. L^k denotes the fragment of first-order logic consisting of all formulas with at most k variables. For example,

$$\forall x \forall y \exists z(Exz \wedge \exists x(Ezx \wedge \exists z(Exz \wedge \exists x(Ezx \wedge Exy))))$$

is an L^3-sentence saying that for all vertices x, y of a graph there is a path of length 5 from x to y. C^k is the extension of L^k where still only k variables are allowed, but in addition to the usual existential or universal quantifiers also quantifiers of the form $\exists^{\geq m}$, for $m \geq 1$. The meaning of $\exists^{\geq m} x\, \phi(x)$ is "there exist at least m elements x such that $\phi(x)$ holds". Clearly, $\exists^{\geq m} x\, \phi(x)$ is equivalent to the first-order formula $\exists x_1 \ldots \exists x_m (\bigwedge_{i \neq j} \neg x_i = x_j \wedge \bigwedge_i \phi(x_i))$, thus C^k is a fragment of first-order logic, but the counting requires many variables, even the formula $\exists^{\geq k+1} x(x = x)$ is not equivalent to an L^k-formula. Hence C^k is more expressive than L^k.

Let L be a logic. Two structures $\boldsymbol{A}, \boldsymbol{B}$ are L-*equivalent* if for all sentences $\phi \in \mathsf{L}$, structure $\boldsymbol{A}$ satisfies ϕ if, and only if, $\boldsymbol{B}$ satisfies ϕ. Two l-tuples $\bar{a} \in A^l, \bar{b} \in B^l$ in structures $\boldsymbol{A}, \boldsymbol{B}$ have the same L-*type*, if for all L-formulas $\phi(\bar{x})$ with l free variables, $\boldsymbol{A}$ satisfies $\phi(\bar{a})$ if, and only if, $\boldsymbol{B}$ satisfies $\phi(\bar{b})$.

2.3. Pebble games

Equivalence in our logics can be characterized by the following combinatorial games.

Definition 3. Let $k \geq 1$ and τ a vocabulary. The *k-pebble game* is played by two players, a *spoiler* and a *duplicator*, on a pair $(\boldsymbol{A}, \boldsymbol{B})$ of τ-structures.

A *position* of the game is a subset of $A \times B$ of size at most k. The game starts in an initial position and consists of a sequence of rounds.

In each round of the game the spoiler first removes a pair from the current position if its size is k. Let P be the resulting position (of size at most $(k-1)$). Now the spoiler either selects an $a \in A$ and the duplicator answers by selecting a $b \in B$, or the spoiler selects a $b \in B$ and the duplicator answers by selecting an $a \in A$. The new position is $P \cup \{ab\}$.

The duplicator *wins* the game if each position P that occurs is a partial isomorphism between $\boldsymbol{A}$ and $\boldsymbol{B}$ (that is, an isomorphism whose domain is the substructure of $\boldsymbol{A}$ with universe $\{a \in A \mid \exists b \in B : ab \in P\}$).

If $\boldsymbol{A} = \boldsymbol{B}$, we refer to the game on $(\boldsymbol{A}, \boldsymbol{B})$ as the game on $\boldsymbol{A}$.

Theorem 4. (Barwise [5], Immerman [17]) *Let $k \geq 1$ and τ a vocabulary. Furthermore, let $\boldsymbol{A}, \boldsymbol{B}$ be τ-structures and $\bar{a} = a_1 \ldots a_l \in A^l$, $\bar{b} = b_1 \ldots b_l \in B^l$ l-tuples, for an $l \leq k$. Then we have:*

(i) *$\boldsymbol{A}$ and $\boldsymbol{B}$ are L^k-equivalent if, and only if, the duplicator has a winning strategy for the k-pebble game on $(\boldsymbol{A}, \boldsymbol{B})$ with initial position $\emptyset$.*

(ii) *$\bar{a}$ and $\bar{b}$ have the same L^k-type if, and only if, the duplicator has a winning strategy for the k-pebble game on $(\boldsymbol{A},\boldsymbol{B})$ with initial position $\{a_1b_1,\dots,a_lb_l\}$.*

Definition 5. Let $k\geq 1$ and τ a vocabulary. The *k-bijective game* is played by two players, a *spoiler* and a *duplicator*, on a pair $(\boldsymbol{A},\boldsymbol{B})$ of τ-structures.

A *position* of the game is a subset of $A\times B$ of size at most k. If $|A|=|B|$, the game starts in an initial position and consists of a sequence of rounds. If $|A|\neq|B|$, the spoiler wins immediately.

In each round of the game the spoiler first removes a pair from the current position if its size is k. Let P be the resulting position (of size at most $(k-1)$). Now the duplicator defines a bijection β between A and B. The spoiler chooses an $a\in A$ and the new position is $P\cup\{a\beta(a)\}$.

The duplicator *wins* the game if $|A|=|B|$ and each position P that occurs is a partial isomorphism between $\boldsymbol{A}$ and $\boldsymbol{B}$.

Theorem 6. (Hella [14]) *Let $k\geq 1$ and τ a vocabulary. Furthermore, let $\boldsymbol{A},\boldsymbol{B}$ be τ-structures and $\bar{a}=a_1\dots a_l\in A^l$, $\bar{b}=b_1\dots b_l\in B^l$ l-tuples, for an $l\leq k$. Then we have:*

(i) *$\boldsymbol{A}$ and $\boldsymbol{B}$ are C^k-equivalent if, and only if, the duplicator has a winning strategy for the k-bijective game on $(\boldsymbol{A},\boldsymbol{B})$ with initial position $\emptyset$.*

(ii) *$\bar{a}$ and $\bar{b}$ have the same C^k-type if, and only if, the duplicator has a winning strategy for the k-bijective game on $(\boldsymbol{A},\boldsymbol{B})$ with initial position $\{a_1b_1,\dots,a_lb_l\}$.*

3. The main theorem

Instead of considering the L-EQUIVALENCE and L-TYPE problem separately for the logics L^k and C^k, we consider the following combined problems.

E(k)

Instance: Directed graphs $\boldsymbol{G}$ and $\boldsymbol{H}$ which are either C^k-equivalent or not L^k-equivalent.

Problem: Are $\boldsymbol{G}$ and $\boldsymbol{H}$ C^k-equivalent?

T(k)

Instance: A directed graph $\boldsymbol{G}$ and vertices $a,b\in A$ which either have the same C^k-type or distinct L^k-types.

Problem: Do a and b have the same C^k-type?

As we said in the introduction, the choice of directed graphs as the input structures in our problems is the most convenient. For $k \geq 3$ we could use undirected graphs as well. For $k = 2$ undirected graphs do not work, since there is only a finite number of L^2-equivalence classes of undirected graphs. But instead of directed graphs we could also work with colored undirected graphs.

Theorem 7. *For each $k \geq 2$ there is an AC_0-reduction from MCV to E(k) and T(k).*

Theorem 1 obviously follows. Note that Theorem 7 is much stronger than Theorem 1; it implies the corresponding result for all logics whose expressive power is between that of L^k and C^k, for example, for the extension of L^k by modular counting quantifiers.

4. Bisimilarity

Bisimilarity is an equivalence relation on Kripke structures, which we can simply consider as colored directed graphs. One way to introduce bisimilarity is via the following game:

Definition 8. The *bisimilarity game* is played by a *spoiler* and a *duplicator* on a Kripke structure $\boldsymbol{K}$. A *position* of the game is a pair of elements of K.

The game starts with an initial position. In each position ab of the game the spoiler replaces either a or b by one of its successors, that is, he either replaces a by an a' such that $E^{\boldsymbol{K}}aa'$ or b by a b' such that $E^{\boldsymbol{K}}bb'$. The duplicator answers by replacing the other element by one of its successors.

The duplicator wins the game, if she can always answer the spoiler's move and in each position the two elements have the same colors.

Definition 9. Two elements a, b of a Kripke structure $\boldsymbol{K}$ are *bisimilar* if the duplicator wins the bisimilarity game on $\boldsymbol{K}$ with initial position ab.

It is easy to see (and well-known) that two elements of a Kripke structure that have the same L^2-type are bisimilar.

Balcázar, Gabarró and Sántha [4] showed that bisimilarity is complete for polynomial time under NC-reductions. We give an alternate proof of this result on which the proof of our main theorem is based.

Proposition 10. *There is an AC_0-reduction from MCV to the following problem:*

> BISIMILARITY
>
> *Instance: A directed graph $\boldsymbol{G}$ and vertices a, b of A.*
>
> *Problem: Are a and b bisimilar?*

To prove the Proposition we define a simple concept, which is also quite important in the proof our main theorem:

Definition 11. Let $\boldsymbol{A},\boldsymbol{B}$ be structures of the same vocabulary and let P,Q be two positions of the k-pebble game, the k-bijective game, or the bisimilarity game on $(\boldsymbol{A},\boldsymbol{B})$. We say that the spoiler can *reach* Q from P if he has a strategy for the game with initial position P such that he either wins the game or position Q eventually occurs.

Conversely, we say that the duplicator can *avoid* Q from P if she has a winning strategy for the game with initial position P in which Q never occurs.

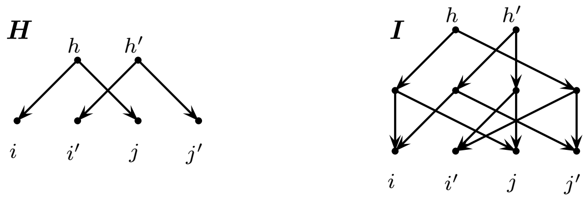

Fig. 1. The structures $\boldsymbol{H}$ and $\boldsymbol{I}$. In both structures, h,h' are colored green, i,i' are colored red, and j,j' are colored blue.

Proof (of Proposition 10). In our reduction, we use two little gadgets $\boldsymbol{H}$ and $\boldsymbol{I}$ (see Figure 1) to simulate AND and OR gates, respectively, of a circuit. ($\boldsymbol{I}$ has been introduced by Immerman [16] in a different context.) $\boldsymbol{I}$ has three nontrivial automorphisms denoted by fix_h, fix_i, fix_j. Each of them fixes one of the pairs hh', ii', jj' (fix_i fixes i and i') and switches the other two (*switching* ii' means mapping i to i' and vice versa). $\boldsymbol{H}$ only has one nontrivial automorphism, denoted by swi, that switches all three pairs. On the other hand, in the bisimilarity game on $\boldsymbol{I}$ the duplicator cannot avoid both $\{ii'\}$ and $\{jj'\}$ at the same time from $\{hh'\}$, and in the game on $\boldsymbol{H}$, the spoiler can reach both $\{ii'\}$ and $\{jj'\}$ from $\{hh'\}$.

Let $\boldsymbol{C}$ be a monotone circuit. We construct a colored directed graph $\boldsymbol{D}$ that contains, for all $a \in \boldsymbol{C}$, two vertices a, a' such that

$$\text{val}^{\boldsymbol{C}}(a) = \text{TRUE} \iff a \text{ and } a' \text{ are bisimilar.} \tag{4.1}$$

We say that we *connect* a pair aa' with a pair bb' if we draw an edge from a to b and an edge from a' to b'. We start building our graph $\boldsymbol{D}$ by taking two vertices a, a' for each $a \in C$. For each OR-node a with parents b and c we add a copy of $\boldsymbol{I}$ and identify aa' with hh'. Furthermore, we connect ii' with bb' and jj' with cc'. For each AND-node a with parents b and c, we add a copy of $\boldsymbol{H}$ and identify aa' with hh'. We connect ii' with bb' and jj' with cc'. For all input nodes a that are not contained in $T^{\boldsymbol{C}}$, we color a (but not a') white.

Observe that, by the last clause of the definition, for input nodes $a \in C$ the vertices a and a' have the same colors if, and only if, $\mathrm{val}^{\boldsymbol{C}}(a) = \textsc{True}$. This implies (4.1) for input nodes. Now an easy induction on the height of a shows that actually (4.1) holds for all nodes.

This gives us a reduction from MCV to a variant of BISIMILARITY where the input consists of a colored directed graph (and two of its vertices) instead of a plain directed graph. But the reduction can easily be modified in order to obtain a plain directed graph.

The reduction is in AC_0 because we just replace each vertex a of $\boldsymbol{C}$ by some gadget only depending on the color of a and on whether a is an internal node ore not. ∎

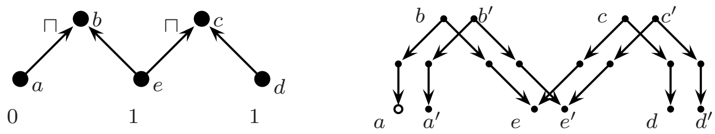

Fig. 2. A circuit $\boldsymbol{C}$ and the corresponding graph $\boldsymbol{D}$. Of course the colors of the two copies of $\boldsymbol{H}$ in $\boldsymbol{D}$ are not displayed.

5. Proof of the main theorem

Why does the reduction of MCV to BISIMILARITY given in the proof of Proposition 10 not reduce MCV to L^2-TYPE? Let us consider the example given in Figure 2. The value of node c in $\boldsymbol{C}$ is 1, thus the duplicator is supposed to win the bisimilarity game on $\boldsymbol{D}$ with initial position cc'. And indeed, she does. However, she does not win the 2-pebble game on $\boldsymbol{D}$ with initial position $\{cc'\}$, since the spoiler can reach position $\{aa'\}$ from $\{cc'\}$. The problem is that once position $\{ee'\}$ is reached, the game is not over, as the bisimilarity game in position ee', but the spoiler may start to move "backwards", and then after he reaches $\{bb'\}$ again forwards until he reaches $\{aa'\}$.

To handle backwards moves we introduce *one-way switches*, as shown in Figure 3. They are obtained by sticking a gadget $\boldsymbol{I}$ on top of an $\boldsymbol{H}$. (Because in the pebble or bijective game the spoiler can move backwards anyway, we no longer need our graphs to be directed. In particular, we henceforth use undirected versions of the gadgets $\boldsymbol{H}$ and $\boldsymbol{I}$.)

It not hard to see that in the 2-pebble game on $\boldsymbol{O}$ the spoiler can reach position $\{yy'\}$ from $\{xx'\}$, whereas the duplicator can avoid $\{xx'\}$ from $\{yy'\}$ even in the 2-bijective game on $\boldsymbol{O}$. The latter follows from the fact that $\boldsymbol{O}$ has an automorphism that fixes x and maps y to y'

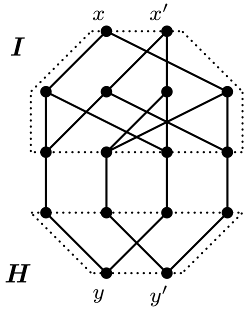

Fig. 3. A one-way switch. The vertices of ***I*** and ***H*** inherit their colors.

Using this switch in the reduction of the bisimilarity case we can obtain a reduction from MCV to L^2-TYPE. We are not going to present this reduction formally, since it will be subsumed by the following proof of Theorem 7. However, Figure 4 shows how to extend the example of Figure 2. It should be no problem for the reader to generalize this example.

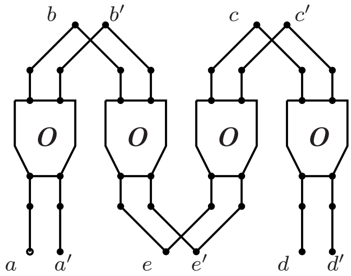

Fig. 4. Example of Figure 2 continued

The main technical difficulty of our proof is to extend this idea to the case of arbitrarily many variables. In the next subsection we describe an important technical lemma that has been proved in [9] in a different context. We apply it in Subsection 5.2 to construct so-called threshold switches. In a sense, they can be used to "reduce" the k-variable case to the 2-variable case. Thus they play a central role in our proof. In Subsection 5.3 we introduce a generalization of the one-way switches. In Subsection 5.4 we give a reduction from MCV to $\mathrm{T}(k)$, for each $k \geq 2$, along the lines of the reduction just sketched for $k = 2$. Finally, in Subsection 5.5 we modify our reduction and reduce MCV to $\mathrm{E}(k)$.

5.1. A combinatorial lemma

Lemma 12 states the existence of finite graphs with a certain property of homogeneity. It is a slight modification of Theorem 2.15 of [9], which is the core of the proof of several hierarchy results. For a full proof, which is based on a construction due to Hrushovski [15], we refer the reader to [9]. We only give a brief sketch that indicates which changes are necessary to obtain the version of the lemma stated here.

An l-*path* between two l-tuples $\overline{a}, \overline{b}$ in a graph $\boldsymbol{G}$ is a sequence $\overline{a}_1 = \overline{a}, \overline{a}_2, \ldots, \overline{a}_n = \overline{b}$ of l-tuples of vertices of $\boldsymbol{G}$ such that for all $i \leq n-1$, the $2l$ vertices in $\bar{a}_i$ and $\bar{a}_{i+1}$ induce a $2l$-clique.

Lemma 12. *For all $n, l \geq 2$ there exists a finite graph $\boldsymbol{G}^{l,n}$, an automorphism s of $\boldsymbol{G}^{l,n}$, and two l-tuples $\bar{c}, \bar{d}$ of vertices of $\boldsymbol{G}^{l,n}$ such that:*

(i) *$G^{l,n}$ can be partitioned into n disjoint rows in such a way that there are only edges between elements of the same row or succeeding rows.*

(ii) *The automorphism s preserves the rows and satisfies $s^{-1} = s$.*

(iii) *There is an l-path from $\bar{c}$ to $\bar{d}$, but not from $\bar{c}$ to $s(\bar{d})$.*

(iv) *For each set $B \subseteq G^{l,n}$ that contains at most $(l-1)$ elements of each row (that may, however, contain elements of several rows) there exists an automorphism f of $\boldsymbol{G}^{l,n}$ such that*
- *$f^{-1} = f$*
- *f preserves the rows*
- *$\forall b \in B\colon f(b) = s(b)$*
- *For all $a \in G^{l,n}$ of distance greater than one to B (that is, $\forall b \in B\colon |\mathrm{row}(a) - \mathrm{row}(b)| > 1$) we have $f(a) = a$.*

Proof (outline). We start by defining a graph $\boldsymbol{A} = \boldsymbol{A}^{l,n}$ that consists of two disjoint l-paths of length n. More precisely, we let

$$A = \{1, \ldots, n\} \times \{-1, \ldots, -l, 1, \ldots, l\}$$

and define the edge relation $E^{\boldsymbol{A}}$ by

$$E^{\boldsymbol{A}}(I,a)(J,b) \iff \big(|I-J| = 1 \wedge \mathrm{sgn}(a) = \mathrm{sgn}(b)\big) \\ \vee \big(I = J \wedge \mathrm{sgn}(a) = \mathrm{sgn}(b) \wedge a \neq b\big),$$

where $\mathrm{sgn}(a)$ is defined to be 1 if $a \geq 0$ and -1 otherwise. The Ith *row* of $\boldsymbol{A}$ is the subset $\{(I,a) \mid 1 \leq |a| \leq l\}$ of A.

For each set $C \subseteq A$ that contains at most $(l-1)$ elements of each row (that may, however, contain elements of several rows) we define a partial isomorphism

p_C of $\boldsymbol{A}$ by

$$p_C((I,a)) = \begin{cases} (I,-a) & \text{if } (I,a)\in C \text{ or } (I,-a)\in C \\ (I,a) & \text{if for all } (J,b)\in C \text{ we have } |I-J|>1 \\ (\text{undefined} & \text{otherwise}) \end{cases}$$

Let $p_1,\dots,p_m$ be an enumeration of all these partial isomorphisms. *This definition of the partial isomorphisms $p_1,\dots,p_m$ slightly differs from the corresponding definition in [9]. Actually, this is the only place where the proof given in [9] has to be modified.*

Let Γ be the group $(\mathbb{Z}_2^m,+)$. Note that there is a natural homomorphism h from Γ to the monoid of partial isomorphisms of $\boldsymbol{A}$ generated by $p_1,\dots,p_m$ that maps the ith unit vector $(0,\dots,0,1,0,\dots,0)$ to p_i and 0 to the identity on A.

We define an equivalence relation $\sim$ on the set $A\times\Gamma$ by

$$(a,\gamma)\sim(b,\delta) \iff a=b^{h(\delta-\gamma)}.$$

Here $b^{h(\delta-\gamma)}$ denotes the image of b under the partial isomorphism $h(\delta-\gamma)$.

The universe of our graph $\boldsymbol{G}=\boldsymbol{G}^{l,m}$ is $A\times\Gamma/_\sim$, and the edge relation $E^{\boldsymbol{G}}$ is defined by

$$E^{\boldsymbol{G}}(a,\gamma)/_\sim(b,\delta)/_\sim \iff \exists\eta,a',b' :\ (a,\gamma)\sim(a',\eta)\wedge(b,\delta)\sim(b',\eta)\wedge E^{\boldsymbol{A}}a'b'.$$

Now we state a sequence of lemmas. The proofs of Lemmas A and B can be found in Section 2.1 and the proofs of Lemmas C–F in Section 2.2 of [9]. Except for the last one, they are all quite simple.

Lemma A. *For all $\gamma\in\Gamma$, the mapping π_γ defined by $a\mapsto(a,\gamma)/_\sim$ is an injective homomorphism of $\boldsymbol{A}$ into $\boldsymbol{G}$.*

Lemma B. *For all $\gamma\in\Gamma$, the mapping f_γ defined by $(a,\delta)/_\sim\mapsto(a,\delta+\gamma)/_\sim$ is an automorphism of $\boldsymbol{G}$.*

Observe that in a sense the automorphism f_γ extends the partial isomorphism $h(\gamma)$, since for all a in the domain of $h(\gamma)$ we have $(a,\delta+\gamma)/_\sim=(a^{h(\gamma)},\delta)/_\sim$.

We define a mapping $\mathrm{row}: G\to\{1,\dots,n\}$ by $\mathrm{row}(((I,a),\gamma)/_\sim)=I$ (for $(I,a)\in A$, $\gamma\in\Gamma$).

Lemma C. *The mapping* row *is well-defined. Furthermore, there are only edges between elements of the same row or succeeding rows.*

The reason for this is that all partial isomorphisms p_i $(1\le i\le m)$ preserve the rows. Symmetry yields the following lemma:

Lemma D. *The mapping s defined by $((I,a),\gamma)/_\sim \mapsto ((I,-a),\gamma)/_\sim$ is an automorphism of $\boldsymbol{G}$ with $s^{-1}=s$.*

We say that an l-tuple $\bar{\alpha}=\alpha_1\ldots\alpha_l\in G$ belongs to the *right component* if there is an $I\leq n$, a $\gamma\in\Gamma$, and a permutation π of $\{1,\ldots,l\}$ such that for all $i\leq l$ we have $\alpha_{\pi(i)}=((I,i),\gamma)/_\sim)$. A tuple $\bar{\alpha}$ belongs to the *left component* if $s(\bar{\alpha})$ belongs to the right component.

Lemma E. *No tuple belongs to the left and the right component.*

Lemma F. *Let $\bar{\alpha},\bar{\beta}\in G$ induce a $2l$-clique. Then $\bar{\alpha}$ belongs to the right component (left component) if, and only if, $\bar{\beta}$ belongs to the right component (left component, respectively).*

Now the proof of Lemma 12 is easy. Obviously, $\boldsymbol{G}$ is a finite graph. Statement (i) of Lemma 12 follows from Lemma C. The mapping s defined in Lemma D is an automorphism that satisfies (ii). We let $c_i=(1,i,0)/_\sim$ and $d_i=(n,i,0)/_\sim$ (for $i\leq l$). Lemma A implies that there is an l-path from $\bar{c}$ to $\bar{d}$ in $\boldsymbol{G}$. Since $\bar{c}$ belongs to the right component and $s(\bar{d})$ belongs to the left component, by Lemmas E and F there is no l-path from $\bar{c}$ to $s(\bar{d})$.

Finally, (iv) follows from Lemma B as follows: Suppose $B\subseteq G$ contains at most $(l-1)$ elements of each row. Let $C=\{(I,a)\in A\mid\exists\gamma\in\Gamma:((I,a),\gamma)/_\sim\in B\}$. Then C is a subset of $\boldsymbol{A}$ that contains at most $(l-1)$ elements of each row. Choose $\gamma\in\Gamma$ such that $h(\gamma)=p_C$ and let $f=f_\gamma$. Clearly, f preserves the rows, and $f^{-1}=f_{-\gamma}=f_\gamma=f$ since $\Gamma=\mathbb{Z}_2^m$. By the remark following Lemma B we have for all $((I,a),\delta)/_\sim\in G$:

- If $((I,a),\delta)/_\sim\in B$ then $f_\gamma(((I,a),\delta)/_\sim)=((I,a)^{p_C},\delta)/_\sim$
  $=((I,-a),\delta)/_\sim=s(((I,a),\delta)/_\sim)$.
- If $|I-\mathrm{row}(\alpha)|>1$ for all $\alpha\in B$ then $f_\gamma(((I,a),\delta)/_\sim)=((I,a)^{p_C},\delta)/_\sim=((I,a),\delta)/_\sim$.

5.2. Threshold switches

We construct colored graphs, called *k-threshold switches*, serving as the basic building blocks of the structures to be constructed in the following steps. The "interface" of a threshold switch to the rest of a bigger structure which it is built in consists of two pairs tt' and uu' of vertices. Its basic property is that the spoiler can reach $\{uu'\}$ from $\{tt'\}$ in the k-pebble game, but not in the $(k-1)$-bijective game. Thus if the spoiler wants to make any progress on a threshold switch during the game, he needs to concentrate all his resources to the switch, and in some sense he loses the advantage of playing with k pebbles rather than 2.

A *distinguished pair* is a pair aa' of distinct vertices of a colored graph such that a and a' are the only two vertices of a certain color. For example, the pairs hh', ii', and jj' are distinguished pairs of the gadgets $\boldsymbol{H}$ and $\boldsymbol{I}$ (cf. Figure 1).

The "positions" $P\cup\{tt',uu\}$ and $P\cup\{tt,uu'\}$ in statement (vi) of the following lemma may contain more than k elements. In this case they are not really positions of the k-bijective game. The statements about them are meant to hold for all positions of size k they contain. Without explicitly mentioning it, whenever we talk about "positions" of the k-bijective or k-pebble game containing more than k elements in the following, we mean "all positions of size k they contain".

Lemma 13. *For each $k\geq 2$ there exists a colored graph $\boldsymbol{T}^k$ (a k-threshold switch) with two distinguished pairs tt' and uu' such that the following holds:*

(i) *In the k-pebble game on $\boldsymbol{T}^k$, the spoiler can reach position $\{uu'\}$ from $\{tt'\}$.*

(ii) *There is an automorphism s of $\boldsymbol{T}^k$ with $s^{-1}=s$ such that $tt',uu'\in s$ (that is, $s(t)=t'$ and $s(u)=u'$).*

Furthermore, there is a set of positions of the k-bijective game on $\boldsymbol{T}^k$, called switched positions, such that:

(iii) *Switched positions are partial isomorphisms.*

(iv) *The duplicator can avoid positions that are not switched from switched positions.*

(v) *Positions $\{tt',uu\}$, $\{uu,tt'\}$ are both switched.*

(vi) *For each switched position P, either $P\cup\{tt',uu\}$ or $P\cup\{tt,uu'\}$ is switched.*

We call strategies for the duplicator in which only switched positions occur switched strategies.

By (iii), switched strategies are winning strategies for the duplicator. Thus by (iv), switched positions are winning positions for the duplicator.

Observe that the spoiler cannot reach $\{uu'\}$ from $\{tt'\}$ in the $(k-1)$-bijective game on a k-threshold switch: Suppose for contradiction that he could. Then he could reach $\{uu',uu\}$ from the switched position $\{tt',uu\}$ in the k-bijective game. Now $\{uu',uu\}$ is clearly not a partial isomorphism, which is a contradiction to (iii) and (iv).

Proof. For $k=2$ the lemma is easy. We just let $\boldsymbol{T}^2$ consist of two paths of length 4, one from t to u and one from t' to u' and color t and t' orange and u and u' purple. Since the proof resembles some of the ideas of the following general proof, let us nevertheless give it. Let $t=a_0,a_1,\dots,a_4=u$ and $t'=a'_0,a'_1,\dots,a'_4=u'$ be the two paths of $\boldsymbol{T}^2$. Statement (i) is obvious. For (ii) we let $s(a_i)=a'_i$ and $s(a'_i)=a_i$. A position P is switched, if there is an $r\in\{1,2,3\}$ such that:

(1) If $ab\in P$ then $a,b\in\{a_i,a'_i\}$ for an $i\in\{0,\dots,4\}$.

(2) If a_ia_i or $a'_ia'_i$ are in P, then a'_ia_i and $a_ia'_i$ are not in P. In this case, we call i *fixed*. If a'_ia_i or $a_ia'_i$ are in P, then a_ia_i and $a'_ia'_i$ are not in P. In this case, we call i *switched*. If $\{a_i,a'_i\}^2\cap P=\emptyset$, we call i *empty*. We consider an empty i as both fixed and switched.

(3) r is empty, and either all $i<r$ are fixed and all $i>r$ are switched or vice versa.

It is straightforward to verify (iii)–(vi).

In general, we have to distinguish between even and odd k.

Assume first that $k>2$ is even. Let $l=\frac{k}{2}$ and $n=k+4$. $\boldsymbol{T}^k$ consists of a copy $\boldsymbol{G}$ of $\boldsymbol{G}^{l,n}$ (satisfying Lemma 12) and the four additional vertices t,t',u,u'. The vertices t and t' are colored orange and u, u' are colored purple. Furthermore, there are edges between t and every node c_i, between t' and every node $s(c_i)$, between u and every node d_i, and between u' and every node $s(d_i)$ (for $1\leq i\leq l$), where the automorphism s and the l-tuples $\overline{c},\overline{d}$ are chosen according to Lemma 12.

We extend the automorphism s from $\boldsymbol{G}$ to $\boldsymbol{T}^k$ by letting $s(t)=t'$, $s(t')=t$, $s(u)=u'$, $s(u')=u$. This proves (ii). Statement (i) follows from Lemma 12 (iii).

We extend the partition of $\boldsymbol{G}$ into rows to $\boldsymbol{T}^k$ by adding t and t' to the first row and u and u' to the last row. In a position P of the k-bijective game on $\boldsymbol{T}^k$ we call a row r *empty* if P does not contain any elements of row r. We call row r *critical* for P if P contains at least l pairs of elements of row r. Note that there are at most two critical rows for any position P.

We call a position P *switched* if there exists an automorphism h and a row r_s such that:

(1) h preserves the rows.
(2) h is the identity on the first and last row.
(3) $2\leq r_s\leq n-1$.
(4) Row r_s is empty,
(5) For each pair $ab\in P$ we either have $h(a)=b$ or $s(h(a))=b$. In the first case we call the row of a and b *fixed*, in the second case *switched*. An empty row is defined to be both fixed and switched. No non-empty row is both fixed and switched.
(6) Either all rows below r_s are fixed and all rows above r_s are switched (then we call position P *top-switched*), or all rows below r_s are switched and all rows above r_s are fixed (then we call position P *bottom-switched*).
(7) If there is exactly one critical row q for P then either $q\leq l+2=\frac{n}{2}$ and $r_s>q$ or $q>l+2$ and $r_s<q$.
(8) If there are two critical rows $q_1<q_2$ then
    (a) If $r_s<q_1<q_2$ then either $q_1=l+2$ and $q_2=l+3$ or $q_1>l+2$.
    (b) If $q_1<r_s<q_2$ then $q_1\leq l+2$ and $q_2>l+2$.
    (c) If $q_1<q_2<r_s$ then either $q_1=l+2$ and $q_2=l+3$ or $q_2\leq l+2$.

Statements (iii), (v), and (vi) are obvious. We have to prove (iv).

Suppose that P is a switched position satisfying (1)–(8) via the automorphism h and the row r_s. Without loss of generality we assume that P is top-switched.

Suppose first that $|P|=k$ and the spoiler removes a pair from P. Unless there are two critical rows for P, statements (1)–(8) remain true in the new position via the same h and r_s.

So assume that P contains precisely two critical rows $q_1 < q_2$. Since $|P| = k$, all of its elements are contained in rows q_1 and q_2. Say, the spoiler removes a pair ab of elements of row q_2. Let $P' = P \setminus \{ab\}$.

- If P and r_s satisfy (8)(a) with $q_1 > l+2$ then P' remains top-switched via the same h and r_s.
- If they satisfy (8)(a) with $q_1 = l+2$, $q_2 = l+3$ then P' is bottom-switched via h and $r'_s = l+4$.
- If they satisfy (8)(b) or (8)(c) then P' remains top-switched via the same h and r_s.

Therefore, from now on we can assume without loss of generality that $|P| = k-1$. We have to define a bijection β for the duplicator. Furthermore, for each $a \in T^k$ we have to define a mapping h' and a row r'_s witnessing (1)–(8) in position $P \cup \{a\beta(a)\}$.

For $r = 1, \ldots, n$ we we define a row-preserving automorphism h^r of $\boldsymbol{T}^k$ and a row r_s^r. Furthermore, we declare row r to be fixed or switched. Then we define the bijection β by letting

$$\beta(a) = \begin{cases} h^r(a) & \text{if } r \text{ has been declared fixed,} \\ s(h^r(a)) & \text{if } r \text{ has been declared switched.} \end{cases}$$

for each $a \in T^k$ in row r.

Furthermore, we show that h^r and r_s^r witness (1)–(8) in position $P \cup \{a\beta(a)\}$ for each $a \in T^k$ in row r.

So let $1 \leq r \leq n$.

Case 1. $r \neq r_s$, and row r does not become critical by adding one more pair.
In this case we can just let $h^r = h$ and $r_s^r = r_s$, and we declare row r to be switched if, and only if, it is switched in the current position P. Then clearly for each $a \in T^k$ in row r position $P \cup \{a\beta(a)\}$ remains top-switched (recall that we assumed P to be top-switched).

Case 2. $r = r_s$.
By the pigeonhole principle, there is an empty row $p \in \{2, \ldots, n-1\} \setminus \{r\}$. If there is a critical row q in the interval $\{l+3, \ldots, n\}$ then we can choose $p \in \{2, \ldots, l+2\} \setminus \{r\}$, since q already contains at least l elements. If there is a critical row q in the interval $\{1, \ldots, l+2\}$ then we can choose $p \in \{l+3, \ldots, n-1\} \setminus \{r\}$.

Without loss of generality we can further assume that $r < p$.

Let $Q \subseteq P$ be the set of pairs in P which are contained in a row between r and p and $B = \{s^{-1}(b) \mid ab \in Q\}$. Note that B contains at most $(l-1)$ elements per row. By Lemma 12 (iv) there exists an automorphism f of $\boldsymbol{G}$ such that for all $c \in B$ we have $f(c) = s(c)$ whereas f is the identity on all rows below r or above p. We extend f to $\boldsymbol{T}^k$ by letting $f(t) = t$, $f(t') = t'$, $f(u) = u$, and $f(u') = u'$.

We let $h^r = f \circ h$ and $r_s^r = p$. Furthermore, we declare row r fixed.

Then for any $a \in T^k$ in row r position $P \cup \{a\beta(a)\}$ is top-switched via h^r and r_s^r. Statements (1)–(4) are immediate. (7) remains valid by the choice of $r_s^r = p$ and the hypothesis of (8) does not apply. To see (5) and (6), note that for all pairs $ab \in P \setminus Q$ we have $h^r(a) = h(a)$, whereas for all $ab \in Q$ we have

$$h^r(a) = f(h(a)) = f(s^{-1}(b)) = s(s^{-1}(b)) = b.$$

Thus all rows between r_s and r_s^r have become fixed, as they were supposed to.

Case 3. By adding another pair to row r it becomes critical (that is, row r contains precisely $l-1$ pairs of P).

(a) No row is critical for P.

Without loss of generality we can assume that $r \le l+2$. If $r_s > r$, we stick with the old h and r_s. So suppose $r_s < r$. By the pigeonhole principle we can find an empty row $r_s^r \in]r, n-1]$. Now we can argue as in Case 2, turning the switched rows between r_s and r_s^r into fixed rows by a suitable automorphism.

(b) Row q is critical for P.

Then all elements of P are contained in row q or row r. Let us assume without loss of generality that $q \le l+2$. Then $r_s > q$ by (7). If $r \le l+2$ and $r_s > r$ or $q = l+2$ and $r = l+3$ or $q < r_s < r$ can stick with the old h and r_s. Otherwise we let $r_s^r = q+1$. If r is fixed in position P, we have to adjust things with a suitable automorphism, similarly as in Case 2.

To complete the proof we still have to consider the case that $k \ge 3$ is odd. We let $\boldsymbol{T}^k$ consist of a copy of $\boldsymbol{T}^{2k}$, together with a new vertex x_{ab} for every pair ab of vertices such that a and b are either in the same or in adjacent rows. Each vertex x_{ab} has an edge to both a and b and is colored brown. We can easily extend the automorphism s in such a way that (ii) holds.

Each positions of the k-bijective game on $\boldsymbol{T}^k$ corresponds to a position of the $2k$-bijective game on $\boldsymbol{T}^{2k}$ which is obtained by taking the projections of the new vertices x_{ab}. By replacing each move of the k-bijective game on $\boldsymbol{T}^k$ by two moves of the $2k$-bijective game on $\boldsymbol{T}^{2k}$, winning strategies for the duplicator for the $2k$-bijective game on $\boldsymbol{T}^{2k}$ can be translated to winning strategies for the k-bijective game on $\boldsymbol{T}^k$. The projections of switched positions satisfy (iii)–(vi).

However, we cannot necessarily translate every strategy for the spoiler for the $2k$-pebble game on $\boldsymbol{T}^{2k}$ to a strategy for the k-pebble game on $\boldsymbol{T}^k$. Thus (i) still has to be checked. The spoiler's strategy to reach uu' from tt' basically remains the same, namely to go along an k-path from the first row to the last and use that there exist such paths from $\overline{c}$ to $\overline{d}$ and from $s(\overline{c})$ to $s(\overline{d})$, but not from $\overline{c}$ to $s(\overline{d})$. Let $\overline{a}_1 = \overline{c}, \ldots, \overline{a}_n = \overline{d}$ be a k-path in $\boldsymbol{T}^{2k}$. Starting in position $\{tt'\}$, the spoiler selects the following $(k-1)$ elements:

$$x_{a_{11}a_{12}}, x_{a_{13}a_{14}}, \ldots, x_{a_{1k}a_{21}}, x_{a_{22}a_{23}}, \ldots, x_{a_{2(k-3)}a_{2(k-2)}}.$$

If she does not want to lose right away, the duplicator has to answer by selecting:

$$x_{s(a_{11})s(a_{12})}, x_{s(a_{13})s(a_{14})}, \dots, x_{s(a_{1k})s(a_{21})},$$
$$x_{s(a_{22})s(a_{23})}, \dots, x_{s(a_{2(k-3)})s(a_{2(k-2)})}$$

(maybe in a different order). Then the spoiler removes the pair tt' and selects $x_{a_{2(k-1)}a_{2k}}$ in an $\exists$-move; after that he removes the pair $x_{a_{11}a_{12}}x_{s(a)_{11}s(a)_{12}}$ and selects $x_{a_{21}a_{22}}$, et cetera. ∎

5.3. One-way switches

Lemma 14. *For each $k \geq 2$, there exists a k-one-way switch $\boldsymbol{O}^k$ with two distinguished pairs xx' and yy' of vertices such that:*

(i) *The spoiler can reach $\{yy'\}$ from $\{xx'\}$ in the k-pebble game on $\boldsymbol{O}^k$.*

Furthermore, there are two disjoint sets of positions of the k-bijective game on $\boldsymbol{O}^k$, called pre-trapped *and* trapped*, such that:*

(ii) *All pre-trapped and trapped positions are partial isomorphisms of $\boldsymbol{O}^k$.*
(iii) *Position $\{xx'\}$ is pre-trapped.*
(iv) *If P is pre-trapped, then $P \cup \{xx', yy\}$ is pre-trapped.*
(v) *The duplicator can avoid positions that are neither pre-trapped nor trapped from pre-trapped positions.*
(vi) *Positions $\{yy\}$ and $\{yy'\}$ are trapped.*
(vii) *If P is trapped, then $P \cup \{xx\}$ is also trapped.*
(viii) *The duplicator can avoid positions that are not trapped from trapped positions.*

We call a strategy for the duplicator trapped *if all positions that occur are either pre-trapped or trapped, and if once a trapped position has occurred the following positions remain trapped.*

Proof. $\boldsymbol{O}^k$ is built as in Figure 5 from the structures $\boldsymbol{I}$ and $\boldsymbol{H}$ (cf. Figure 1) and two copies $\boldsymbol{T}_{\text{red}}$ and $\boldsymbol{T}_{\text{blue}}$ of the k-threshold switch $\boldsymbol{T}^k$. Note that a 2-one-way switch is essentially the same as a one-way switch introduced in Figure 3, and that k-one-way switches are straightforward generalizations.

To reach $\{yy'\}$ from $\{xx'\}$ in the k-pebble game, the spoiler uses the fact that the duplicator cannot avoid both $\{ii'\}$ and $\{jj'\}$ from $\{hh'\}$ $(=\{xx'\})$ in the game on $\boldsymbol{I}$. From both of these positions, the spoiler can reach the corresponding position in $\boldsymbol{H}$ via the threshold switch, and from there he can reach $\{yy'\}$. This proves (i).

For any subset $B \subseteq O^k$ and position P we let $P_B = P \cap B^2$. All pre-trapped and trapped positions P respect the building blocks of the structure, that is, we have $P = P_I \cup P_{T_{\text{red}}} \cup P_{T_{\text{blue}}} \cup P_H$.

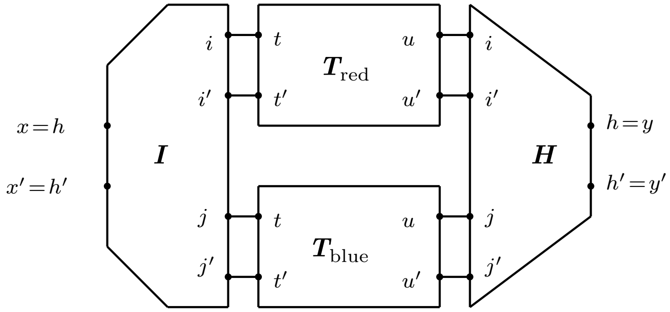

Fig. 5. A k-one-way switch

Before we define pre-trapped and trapped positions, let me informally give the cornerstones of the duplicator's strategy. She always plays in such a way that the subpositions $P_I, P_{T_\text{red}}, P_{T_\text{blue}}, P_H$ of the current position P are "good" positions in the game on the respective building block. P_I and P_H are good if they are contained in an automorphism of $\boldsymbol{I}$ and $\boldsymbol{H}$, respectively. P_{T_red} and P_{T_blue} are good if they are either contained in one of the automorphisms id or s of the k-threshold gadget or if they are switched positions of the k-bijective game on the k-treshold gadget. Thus good positions are always winning positions for the duplicator in the game on the respective building block. However, this alone does not suffice. The positions also have to be "consistent". This means, for example, that if the pair ii' is contained in P_I then tt should not be contained in P_{T_red}. To preserve consistency we distinguish between two cases. Either there is a building block B such that $P_B = P$. Then the subpositions associated with all other building blocks are empty and we do not have to worry about consistency. Or $P_B \subset P$ for all building blocks B. Then there is some room left in each subposition P_B that we can use to add the information needed from the subpositions on other building blocks. Say, for example, we have $P_{T_\text{red}} \subset P$, and $ii' \in P_I$. Then we do not only take care that P_{T_red} is a good position, but actually that $P_{T_\text{red}} \cup \{tt'\}$ is.

Let us now turn to the formal proof. Recall the definition of the automorphisms swi and fix_h, fix_i, fix_j of the gadgets $\boldsymbol{H}$ and $\boldsymbol{I}$, respectively.

A position P is *trapped* if one of the following conditions (Tr1)–(Tr4) holds:

(Tr1) $P \subseteq id$.

(Tr2) $P_I \subseteq id$, $P_H \subseteq swi$, and both $P_{T_\text{red}} \cup \{tt, uu'\}$ and $P_{T_\text{blue}} \cup \{tt, uu'\}$ are switched positions of the k-bijective game on the k-threshold gadget.

(Tr3) $P_I \subseteq fix_h$, $P_H \subseteq id$, and both $P_{T_\text{red}} \cup \{tt', uu\}$ and $P_{T_\text{blue}} \cup \{tt', uu\}$ are switched.

(Tr4) $P_I \subseteq fix_h$, $P_H \subseteq swi$, and both $P_{T_\text{red}} \subseteq s$ and $P_{T_\text{blue}} \subseteq s$, where s is chosen according to Lemma 13 (ii).

A position P is *pre-trapped* if it is not trapped and one of the following conditions (Pre1), (Pre2) holds:

(Pre1) $P_I \subseteq fix_j$, $P_H \subseteq id$, $P_{T_\text{red}} \cup \{tt', uu\}$ is switched and $P_{T_\text{blue}} \subseteq id$.

(Pre2) $P_I \subseteq fix_i$, $P_H \subseteq id$, $P_{T_\text{red}} \subseteq id$ and $P_{T_\text{blue}} \cup \{tt', uu\}$ is switched.

Obviously, a subposition of a trapped position is trapped and a subposition of a pre-trapped position is either pre-trapped or trapped. Thus we have to show that in a pre-trapped position P of size at most $k-1$ the duplicator can find a bijection β such that for all $a \in \boldsymbol{O}^k$, position $P \cup \{a\beta(a)\}$ is pre-trapped or trapped, and in a trapped position P of size at most $k-1$ she can find a bijection β such that for all $a \in \boldsymbol{O}^k$, position $P \cup \{a\beta(a)\}$ is still trapped.

Note that in positions (Tr1) and (Tr4) the duplicator can just play according to automorphisms and thus preserve the respective property.

Case 1. P satisfies (Tr2).

We let $\beta\restriction_I = id$ and $\beta\restriction_H = swi$. Then clearly $P \cup \{a\beta(a)\}$ is trapped for all $a \in I \cup H$. The restriction $\beta\restriction_{T_\text{red}}$ is defined as follows:

- If $|P_{T_\text{red}}| < k-1$ then we let $\beta\restriction_{T_\text{red}}$ be chosen according to a switched strategy on $\boldsymbol{T}^k$ in position $P_{T_\text{red}} \cup \{tt\}$.

  Then for any $a \in T_\text{red}$ position $P_\text{red} \cup \{a\beta(a), tt\}$ is switched and thus, by Lemma 13 (iii) and (vi) position $P_\text{red} \cup \{a\beta(a), tt, uu'\}$ is switched. Thus Position $P \cup \{a\beta(a)\}$ still satisfies (Tr2).
- If $|P_{T_\text{red}}| = k-1$ then we let $\beta\restriction_{T_\text{red}}$ be chosen according to a switched strategy on $\boldsymbol{T}$ in position P_{T_red}. Let $a \in T_\text{red}$, then $P_{T_\text{red}} \cup \{a\beta(a)\}$ is switched. By Lemma 13 (vi) one of the positions $P_{T_\text{red}} \cup \{a\beta(a), tt, uu'\}$ and $P_{T_\text{red}} \cup \{a\beta(a), tt', uu\}$ is switched. Since P_{T_blue}, P_I, and P_H are all empty in this case, position $P \cup \{a\beta(a)\}$ satisfies (Tr2) or (Tr3), respectively.

  $\beta\restriction_{T_\text{blue}}$ can be defined analogously.

Case 2. P satisfies (Tr3).

This case is symmetric to the previous one.

Case 3. P satisfies (Pre1), but is not trapped.

We let $\beta\restriction_I = fix_j$, $\beta\restriction_H = id$, and $\beta\restriction_{T_\text{blue}} = id$. Then $P \cup \{a\beta(a)\}$ is pre-trapped for all $a \in I \cup H \cup T_\text{blue}$. The restriction $\beta\restriction_{T_\text{red}}$ is defined as follows:

- If $|P_{T_\text{red}}| < k-1$ then we let $\beta\restriction_{T_\text{red}}$ be chosen according to a switched strategy on $\boldsymbol{T}^k$ in position $P_{T_\text{red}} \cup \{tt'\}$.

  Then for any $a \in T_\text{red}$ position $P_\text{red} \cup \{a\beta(a), tt'\}$ is switched and thus, by Lemma 13 (iii) and (vi) position $P_\text{red} \cup \{a\beta(a), tt', uu\}$ is switched. Thus Position $P \cup \{a\beta(a)\}$ still satisfies (Pre1).
- If $|P_{T_\text{red}}| = k-1$ then P satisfies (Tr3) and thus is trapped, in contradiction to our assumption.

Case 4. P satisfies (Pre2).
This case is symmetric to the previous one. ∎

Our one-way switches are quite flexible. The following lemma gives a different class of strategies for the duplicator.

Lemma 15. *Let $k \geq 2$. There is a set of positions of the k-bijective game on a k-one-way switch $\boldsymbol{O}^k$, called* twisted *positions, such that:*
(i) *Each twisted position is a partial isomorphism.*
(ii) *The duplicator can avoid non-twisted positions from twisted positions.*
(iii) *Positions $\{yy\}$ and $\{yy'\}$ are twisted.*
(iv) *If P is twisted, then $P \cup \{xx'\}$ is also twisted.*
A twisted strategy is a strategy that only contains twisted positions.

Note that this lemma implies that the duplicator cannot reach $\{xx\}$ from $\{yy\}$ on a one-way switch.

Proof. Using the notation of the previous proof, we call a position P *twisted* if $P = P_I \cup P_{T_{\text{red}}} \cup P_{T_{\text{blue}}} \cup P_H$ and it satisfies one of the following conditions:

(Tw1) $P_I \subseteq fix_i$, $P_H \subseteq id$, $P_{T_{\text{red}}} \subseteq id$, $P_{T_{\text{blue}}} \cup \{tt', uu\}$ is switched.
(Tw2) $P_I \subseteq fix_i$, $P_H \subseteq swi$, $P_{T_{\text{red}}} \cup \{tt, uu'\}$ is switched, $P_{T_{\text{blue}}} \subseteq s$.
(Tw3) $P_I \subseteq fix_j$, $P_H \subseteq id$, $P_{T_{\text{red}}} \cup \{tt', uu\}$ is switched, $P_{T_{\text{blue}}} \subseteq id$.
(Tw4) $P_I \subseteq fix_j$, $P_H \subseteq swi$, $P_{T_{\text{red}}} \subseteq s$, $P_{T_{\text{blue}}} \cup \{tt, uu'\}$ is switched.

Then (i), (iii), and (iv) are immediate, and (ii) can be proved similarly to the previous proof. ∎

5.4. The reduction

We proceed similarly as in the prove of Proposition 10.

Given a monotone circuit $\boldsymbol{C}$, we define a colored graph $\boldsymbol{D}$ as follows (cf. Figure 6): For each $a \in C$ we take 2 vertices a, a'. For each OR-node a with parents b and c we add a copy $\boldsymbol{I}_a$ of $\boldsymbol{I}$ and two copies $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$ of the k-one-way switch. We identify aa' with hh' of $\boldsymbol{I}_a$. We connect ii' and jj' of $\boldsymbol{I}_a$ with xx' of $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$, respectively, and yy' of $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$ with bb' and cc', respectively. Analogously, for each AND-node a with parents b and c we add a copy $\boldsymbol{H}_a$ of $\boldsymbol{H}$ and two copies $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$ of the k-one-way switch. We identify aa' with hh' of $\boldsymbol{H}_a$. We connect ii' and jj' of $\boldsymbol{H}_a$ with xx' of $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$, respectively, and yy' of $\boldsymbol{O}_a^{\text{red}}$ and $\boldsymbol{O}_a^{\text{blue}}$ with bb' and cc', respectively. For all input nodes a that are not contained in $T^{\boldsymbol{C}}$, we color a (but not a') white.

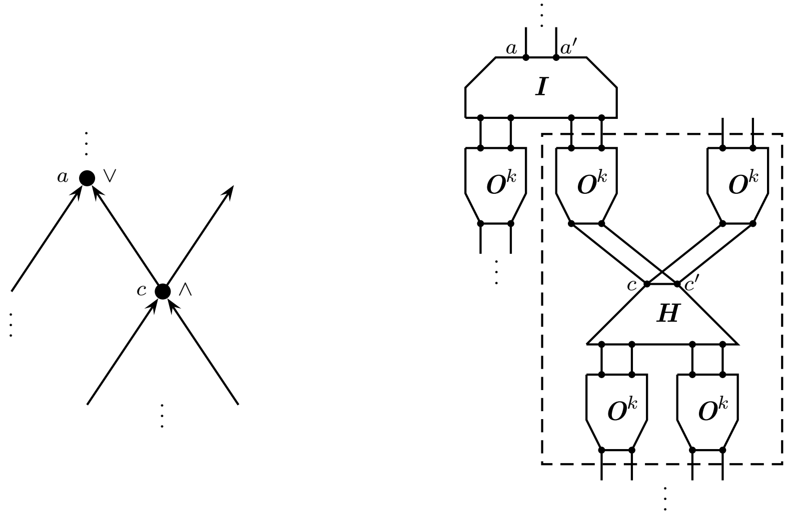

Fig. 6. A part of a circuit $\boldsymbol{C}$ and its translation to $\boldsymbol{D}$. The dashed box contains the neighborhood of c.

We shall prove that for all $a \in C$ we have

(5.1) $$\mathrm{val}^{\boldsymbol{C}}(a) = \text{FALSE} \Longrightarrow a \text{ and } a' \text{ have a distinct } \mathsf{L}^k\text{-type in } \boldsymbol{D}$$

(5.2) $$\mathrm{val}^{\boldsymbol{C}}(a) = \text{TRUE} \Longrightarrow a \text{ and } a' \text{ have the same } \mathsf{C}^k\text{-type in } \boldsymbol{D}.$$

An easy induction on the height of a in $\boldsymbol{C}$, using Lemma 14 (i) to proceed through the one-way gadgets, shows that if $\mathrm{val}^{\boldsymbol{C}}(a) = \text{FALSE}$ then the spoiler wins the k-pebble game on $\boldsymbol{D}$ with initial-position $\{aa'\}$. This proves (5.1).

To see (5.2), we have to prove that if $\mathrm{val}^{\boldsymbol{C}}(a) = \text{TRUE}$ then the duplicator has a winning strategy for the k-bijective game on $\boldsymbol{D}$ with initial position $\{aa'\}$.

Therefore we partition the positions of the k-bijective game on $\boldsymbol{D}$ into *safe* and *dangerous* positions and prove:

(i) $\{aa'\}$ is safe for each node $a \in C$ such that $\mathrm{val}^{\boldsymbol{C}}(a) = \text{TRUE}$.
(ii) All safe positions are partial isomorphisms of $\boldsymbol{D}$.
(iii) The duplicator can avoid dangerous positions from safe positions.

As usual, safe positions will preserve the building blocks of the structure. For any position P and subgraph $\boldsymbol{B} \subseteq \boldsymbol{D}$ we let $P_B = P \cap B^2$.

We say that a one-way switch $\boldsymbol{O}$ in $\boldsymbol{D}$ is *adjacent* to an $a \in C$, if the pair yy' in $\boldsymbol{O}$ is connected with aa'.

The *neighborhood* $\boldsymbol{N}(a)$ of a node $a \in C$ is the subgraph of $\boldsymbol{D}$ consisting of aa', all gadgets with subscript a, and all one-way switches that are adjacent to a. For example, in Figure 6 the neighborhood of c consists of the parts of $\boldsymbol{D}$ in the dashed box.

A node $a \in C$ is *critical* in a position P if $\mathrm{val}^{\boldsymbol{C}}(a) = \text{TRUE}$ and one of the following two conditions is satisfied:

(1) For all one-way switches $\boldsymbol{O}$ adjacent to a, position $P_O \cup \{yy\}$ (of the k-bijective game on $\boldsymbol{O}$) is trapped and

$$P_{N(a)\setminus\bigcup\{O \mid \boldsymbol{O} \text{ is adjacent to } a\}} \subseteq id.$$

(2) For all one-way switches $\boldsymbol{O}$ adjacent to a, position $P_O \cup \{yy'\}$ is trapped and one of the following three conditions is satisfied:

(a) a is an AND-node, and $P_{H_a} \subseteq swi$ and $P_{O_a^{\mathrm{red}}} \cup \{xx', yy\}$, $P_{O_a^{\mathrm{blue}}} \cup \{xx', yy\}$ are pre-trapped.

(b) a is an OR-node, say with parents b and c and either $\mathrm{val}^{\boldsymbol{C}}(b) = \text{TRUE}$, $P_{I_a} \subseteq fix_j$, $P_{O_a^{\mathrm{red}}} \cup \{xx', yy\}$ is pre-trapped, and $P_{O_a^{\mathrm{blue}}} \subseteq id$ or $\mathrm{val}^{\boldsymbol{C}}(c) = \text{TRUE}$, $P_{I_a} \subseteq fix_i$, $P_{O_a^{\mathrm{red}}} \subseteq id$, and $P_{O_a^{\mathrm{blue}}} \cup \{xx', yy\}$ is pre-trapped.

(c) a is an output node contained in $T^{\boldsymbol{C}}$ and $aa \notin P$.

A position P is safe if there is a critical $a_c \in C$ such that $P_{D\setminus N(a_c)} \subseteq id$. Then (i) and (ii) are obvious.

To see (iii), first note that in safe positions P only nodes a of value TRUE can be adjacent to a one-way switch $\boldsymbol{O}$ such that P_O is pre-trapped. The moment P_O becomes trapped the node a becomes the new critical node.

Removing pairs from a safe position leaves it safe. So let P be a safe position of size at most $k-1$. We have to define a bijection β for the duplicator such that for each $d \in D$ position $P \cup \{d\beta(d)\}$ is still safe.

Let a be the critical node. We let $\beta\restriction_{D\setminus N(a)} = id$. To define β on $N(a)$ we distinguish between two cases:

Case 1. P and a satisfy (1).
We let

$$\beta\restriction_{N(a)\setminus\bigcup\{O \mid \boldsymbol{O} \text{ is adjacent to } a\}} = id.$$

Furthermore, for all one-way switches $\boldsymbol{O}$ adjacent to a, if $|P_O| = k-1$ we choose $\beta\restriction_O$ according to a trapped strategy on $\boldsymbol{O}$, and if $|P_O| < k-1$ we choose $\beta\restriction_O$ according to a trapped strategy on $\boldsymbol{O} \cup \{yy\}$.

Case 2. P and a satisfy (2).

If a is an AND-node, say with parents b and c, we let $\beta\restriction_{H_a} = swi$. If, furthermore, $|P_{O_a^{\mathrm{red}}}| = k-1$ we choose $\beta\restriction_{O_a^{\mathrm{red}}}$ according to a trapped strategy on $\boldsymbol{O}_a^{\mathrm{red}}$ in position $P_{O_a^{\mathrm{red}}}$. Note that this may cause b to become the critical node of the next position. If $|P_{O_a^{\mathrm{red}}}| < k-1$ we choose $\beta\restriction_{O_a^{\mathrm{red}}}$ according to a trapped strategy on $\boldsymbol{O}_a^{\mathrm{red}}$ in position $P_{O_a^{\mathrm{red}}} \cup \{xx'\}$. The restriction of β to O_a^{blue} is defined similarly.

If a is an OR-node, say with parents b and c, let us assume without loss of generality that $P_{I_a} \subseteq fix_i$ and $P_{O_a^{\text{red}}} \subseteq id$. We let $\beta_{I_a} = fix_i$ and $\beta_{O_a^{\text{red}}} = id$. If, furthermore, $|P_{O_a^{\text{blue}}}| = k-1$ we choose $\beta\restriction_{O_a^{\text{blue}}}$ according to a trapped strategy on $\boldsymbol{O}_a^{\text{blue}}$ in position $P_{O_a^{\text{blue}}}$. This may cause c to become the critical node of the next position. If $|P_{O_a^{\text{blue}}}| < k-1$ we choose $\beta\restriction_{O_a^{\text{blue}}}$ according to a trapped strategy on $\boldsymbol{O}_a^{\text{blue}}$ in position $P_{O_a^{\text{blue}}} \cup \{xx'\}$.

If a is an output node in $T^{\boldsymbol{C}}$ we let $\beta(a) = a'$ and $\beta(a') = a$.

Furthermore, for all one-way switches $\boldsymbol{O}$ adjacent to a, if $|P_O| = k-1$ we choose $\beta\restriction_O$ according to a trapped strategy on $\boldsymbol{O}$, and if $|P_O| < k-1$ we choose $\beta\restriction_O$ according to a trapped strategy on $\boldsymbol{O} \cup \{yy'\}$.

(5.1) and (5.2) yield a reduction of MCV to a variant of T(k) where the input consists of a colored graph and two of its vertices instead of a directed graph. As in the case of bisimilarity (Proposition 10), it is easy to modify the reduction to obtain a directed graph instead of a colored graph.

The reduction is in AC_0 for the same reason as the reduction in the bisimilarity case: We just replace each vertex a of $\boldsymbol{C}$ by some gadget only depending on the color of a and on whether a is an internal node or not.

5.5. Extension to the equivalence query

Given a monotone circuit and a node $c \in C$ we have seen how to produce a colored graph $\boldsymbol{D}$ such that

(5.3) $$\text{val}^{\boldsymbol{C}}(c) = \text{FALSE} \Longrightarrow c \text{ and } c' \text{ have a distinct } \mathsf{L}^k\text{-type in } \boldsymbol{D}$$

(5.4) $$\text{val}^{\boldsymbol{C}}(c) = \text{TRUE} \Longrightarrow c \text{ and } c' \text{ have the same } \mathsf{C}^k\text{-type in } \boldsymbol{D}.$$

We further reduce $\boldsymbol{D}$ to a pair $\boldsymbol{G}, \boldsymbol{H}$ of colored graphs such that

(5.5) $$c \text{ and } c' \text{ have a distinct } \mathsf{L}^k\text{-type in } \boldsymbol{D} \Longrightarrow \boldsymbol{G} \text{ and } \boldsymbol{H} \text{ are not } \mathsf{L}^k\text{-equivalent}$$

and

(5.6) $$c \text{ and } c' \text{ have the same } \mathsf{C}^k\text{-type in } \boldsymbol{D} \Longrightarrow \boldsymbol{G} \text{ and } \boldsymbol{H} \text{ are } \mathsf{C}^k\text{-equivalent}$$

The problem is that the fact that c and c' have the same C^k-type in $\boldsymbol{D}$ does not necessarily imply that the structures $(\boldsymbol{D}, c)$ and $(\boldsymbol{D}, c')$ are C^k-equivalent. (Here $(\boldsymbol{D}, c^{(')})$ denotes the expansion of $\boldsymbol{D}$ by the constant $c^{(')}$.)

Therefore, we need to define $\boldsymbol{G}$ and $\boldsymbol{H}$ in a slightly more complicated manner: They both consist of a copy of $\boldsymbol{D}$ and a k-one-way switch. The pair yy' of the one-way switch is connected to cc'. Furthermore, in structure $\boldsymbol{G}$ the vertex x of the one-way switch is colored pink, whereas in $\boldsymbol{H}$ vertex x' is colored pink.

Then the spoiler can win the k-pebble game on $\boldsymbol{G}, \boldsymbol{H}$ if he can win the game on $\boldsymbol{D}$ with initial position cc'. (In the first move he selects x, and the duplicator has to answer by selecting x' as the only pink vertex of $\boldsymbol{H}$. From $\{xx'\}$ he can reach $\{yy'\}$ and then $\{cc'\}$, and he wins.) This proves (5.5).

On the other hand, if the duplicator has a winning strategy for the game on $\boldsymbol{D}$ with initial position $\{cc'\}$ she can easily extend it to a strategy for the game on $\boldsymbol{G}, \boldsymbol{H}$ by playing a twisted strategy (see Lemma 15) on the one-way switch.

This gives a reduction from MCV to E(k) and thus completes the proof of Theorem 7. ∎

References

[1] S. Abiteboul, M.Y. Vardi, and V. Vianu: Fixpoint logics, relational machines, and computational complexity, *Journal of the ACM*, **44** (1997), 30–56.

[2] S. Abiteboul and V. Vianu: Generic computation and its complexity, in: *Proceedings of the 23rd ACM Symposium on Theory of Computing*, 209–219, 1991.

[3] L. Babai, P. Erdős, and S. Selkow: Random graph isomorphism, *SIAM Journal on Computing*, **9** (1980), 628–635.

[4] J. Balcázar, J. Gabarró, and M. Sántha: Deciding bisimilarity is P-complete, *Formal Aspects of Computing*, **4** (1992), 638–648.

[5] J. Barwise: On Moschovakis closure ordinals, *Journal of Symbolic Logic*, **42** (1977), 292–296.

[6] J. Cai, M. Fürer, and N. Immerman: An optimal lower bound on the number of variables for graph identification, *Combinatorica*, **12** (1992), 389–410.

[7] A. Dawar: A restricted second order logic for finite structures, in: *Logic and Computational Complexity: International Workshop, LCC '94*, (D. Leivant, ed.), volume 960 of *Lecture Notes in Computer Science*, 1995.

[8] H.-D. Ebbinghaus and J. Flum: *Finite Model Theory*, Springer-Verlag, 1995.

[9] M. Grohe: Arity hierarchies, *Annals of Pure and Applied Logic*, **82** (1996), 103–163.

[10] M. Grohe: The complexity of finite-variable theories, 1997, Habilitationsschrift at the Albert-Ludwigs-Universität Freiburg.

[11] M. Grohe: Finite-variable logics in descriptive complexity theory, *Bulletin of Symbolic Logic*, **4** (1998), 345–399.

[12] M. Grohe: Fixed-point logics on planar graphs, in: *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*, 6–15, 1998.

[13] M. Grohe and J. Mariño: Definability and descriptive complexity on databases of bounded tree-width, in: *Proceedings of the 7th International Conference on Database Theory*, (C. Beeri and P. Buneman, eds.), volume 1540 of *Lecture Notes in Computer Science*, 70–82, Springer-Verlag, 1999.

[14] L. Hella: Logical hierarchies in PTIME, *Information and Computation*, **129** (1996), 1–19.

[15] E. Hrushovski: Extending partial isomorphisms of graphs, *Combinatorica*, **12** (1992), 411–416.

[16] N. Immerman: Number of quantifiers is better than number of tape cells, *Journal of Computer and System Sciences*, **22** (1981), 384–406.

[17] N. Immerman: Upper and lower bounds for first-order expressibility, *Journal of Computer and System Sciences*, **25** (1982), 76–98.

[18] N. Immerman: *Descriptive Complexity*, Springer-Verlag, 1998.

[19] N. Immerman and E. Lander: Describing graphs: A first-order approach to graph canonization, in: *Complexity theory retrospective*, (A. Selman, ed.), 59–81. Springer-Verlag, 1990.

[20] R.E. Ladner: The circuit value problem is logspace complete for P, *Journal of the ACM*, **22** (1975), 155–171.

Martin Grohe

Institut für Mathematische Logik
Universität Freiburg
Eckerstr. 1
79104 Freiburg
Martin.Grohe@mathematik.uni-freiburg.de