Logic and Automata over grids and graphs : A Survey

MSc. Thesis Report

by

Abdullah Abdul Khadir

under the guidance of

Prof. Narayan Kumar
Chennai Mathematical Institute
14^{th} May, 2010

Contents

Chapter 1

Hanf's Theorem

Hanf's theorem plays a very interesting role in model theory as it provides connections between logical equivalence and locality. It gives an upper bound to the extent to which a formula in first order logic can distinguish two structures.

1.1 Definitions and Notations

Any first order formula contains one or more relational symbols and constants. We first fix a set of relational symbols and constant symbols and then analyze the first order formulae built up from these symbols. The set of relational and constant symbols fixed in advance is called the vocabulary.

The notations that we will use in the remainder of this chapter are as follows:-

- The vocabulary is σ=(R_1,...,R_m,c_1,...,c_s) where, $\forall i$, 1$\leq i \leq$m, R$_i$ is a relation symbol of arity k$_i$, for some k$_i \in \mathbb{N}$ and $\forall i$, 1$\leq i \leq$s, c$_i$ is a unique constant symbol.
- $\mathcal{A}$=(A,R_1^A,...,R_m^A,c_1^A,..., c_s^A) and $\mathcal{B}$=(B,R_1^B,...,R_m^B, c_1^B,...,c_s^B) are two finite structures interpreting σ over the domains A and B respectively.

Definition 1.1.1. *If the two structures $\mathcal{A}$ and $\mathcal{B}$ as considered above satisfy the same first order formulae upto a quantifier depth of r for some r $\in \mathbb{N}$, then we say that $\mathcal{A}$ and $\mathcal{B}$ are logically r-equivalent, denoted by $\mathcal{A} \equiv_r \mathcal{B}$.*

Simply stated, $\mathcal{A} \equiv_r \mathcal{B}$, if given any formula ϕ of quantifier depth r, $\mathcal{A} \models \phi$ iff $\mathcal{B} \models \phi$.

Definition 1.1.2 (The Gaiffman Graph). *Given a structure $\mathcal{A}$=(A,R_1^A,...,R_m^A, c_1^A,...,c_s^A), the Gaiffman graph is the undirected graph $\mathcal{G}_A$ = (A,E) where*

- A *is the domain of the structure, $\mathcal{A}$.*
- *E is a binary relation on* A *such that for any two elements a,b $\in$ A, E(a,b) holds iff there is a relation R_i^A of arity k_i and k_i elements $\{a_1, ..., a_{k_i}\} \subseteq$ A such that R($a_1, ..., a_{k_i}$) and {a,b} $\subseteq \{a_1, ..., a_{k_i}\}$*

Example : $\mathcal{A}$ = ({1,2,3,4},$\leq$)

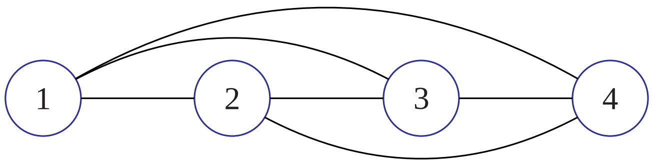


Figure 1.1: *The gaiffman graph, $\mathcal{G}_{\mathrm{A}}$ of $\mathcal{A}$=({1,2,3,4},$\leq$)*

We also need some way of referring to a subgraph of $\mathcal{G}_{\mathcal{A}}$, in which, for a particular node in the gaiffman graph, say a $\in$ *A*, designated as centre, every other node in the subgraph is at a distance $\leq$ d, for some d $\in \mathbb{N}$. Given an element a $\in$ *A*, we denote this subgraph by **N($\mathcal{G}_{\mathcal{A}}$,a)$\restriction$d**, the d-neighbourhood of a $\in \mathcal{G}_A$. Also, we use the alias **N(a,d)** instead of **N($\mathcal{G}_{\mathcal{A}}$,a)$\restriction$d** for the sake of brevity.

Definition 1.1.3. *Given any two elements from each structure, say a $\in$ A and b $\in$ B, and for some d $\in \mathcal{N}$ we say that a and b are **d-equivalent**, iff there exists a bijective function h, given by, h : N(a,d) $\to$ N(b,d) such that h(a) = b and for any relation R_i^A of arity k_i and k_i elements {a_1,...,a_{k_i}} $\subseteq$ N(a,d), R_i^A(a_1,...,a_{k_i}) holds iff R_i^B(h(a_1),...,h(a_{k_i})) holds. We denote this by a $\sim_d$ b.*

Definition 1.1.4. *We can extend d-equivalence to σ-structures by saying that two σ-structures, $\mathcal{A}$ and $\mathcal{B}$ are d-equivalent, again, denoted by $\mathcal{A} \sim_d \mathcal{B}$ iff there is a bijection g : A $\longrightarrow$ B such that, $\forall a \in$ A, a $\sim_d$ g(a).*

Note that $\sim_d$ between elements of two different structures as well as between two structures are equivalence relations. If we fix the maximum number of nodes possible in a subgraph of radius d, to be m, for some m$\in$ $\mathbb{N}$, then, we have only finitely many equivalence classes for $\sim_d$, say n equivalence classes. We then fix an ordering of the equivalence classes as (Type_1,Type_2,...Type_n) and given any structure $\mathcal{A}$=(*A*,R_1^A,...,R_m^A,c_1^A,..., c_s^A), we compute the **d-type signature** of *A* as (#Type_1^A,#Type_2^A,...,#Type_n^A) where #Type_i^A is the number of elements, a $\in$ *A* such that N($\mathcal{G}_{\mathcal{A}}$,a)$\restriction$d $\cong$ Type_i.

Now, notice that if we constrain the number of possible nodes in any subgraph of size d to be the maximum of $|$ *A* $|$ and $|$ *B* $|$, $\mathcal{A} \sim_d \mathcal{B}$ holds iff the d-type signatures of *A* and *B*, (#Type_1^A,#Type_2^A,...,#Type_n^A) and (#Type_1^B,#Type_2^B,...,#Type_n^B), are component-wise equal.

1.2 Hanf's theorem

Hanf's Theorem. *For any two structures $\mathcal{A}$ and $\mathcal{B}$ and for any r,d $\in \mathbb{N}$,*

$$if \;\; d \geq 3^{r-1} \; and \; \mathcal{A} \sim_d \mathcal{B}, \; then \; \mathcal{A} \equiv_r \mathcal{B}.$$

Proof Sketch. In order to prove the above result, we assume that $\mathcal{A} \sim_d \mathcal{B}$, for d$\geq 3^{r-1}$. Now, in order to prove that $\mathcal{A} \equiv_r \mathcal{B}$, we describe a winning strategy for the Duplicator in the r-round EF-game. In this strategy, the Duplicator maintains a partial isomorphism defined by induction on i (1$\leq$i$\leq$ r), where i denotes the number of rounds that have been played in the EF-game so far. Thus, after i rounds, this partial isomorphism is given by the bijection

$$\mathbf{h} : (\cup_{\mathbf{j=1}}^{\mathbf{i}} \mathbf{N}(\mathbf{a_j}, \mathbf{3^{r-i}})) \longrightarrow (\cup_{\mathbf{j=1}}^{\mathbf{i}} \mathbf{N}(\mathbf{b_j}, \mathbf{3^{r-i}})) \qquad \textbf{(I-1)}$$

In addition to being a bijection, it has to satisfy the following :-

(a) $\forall$j, $1 \leq$ j $\leq$ i, h(a$_j$) = b$_j$, where (a$_j$,b$_j$) are the elements selected in round j.
(b) for any R$_i^A$, of arity k and for any k elements {a$_1$,a$_2$, ... a$_k$} $\subseteq$ ($\cup_{j=1}^{i}$N(a$_j$,3^{r-i})),

$$\mathbf{R_i^A}(\mathbf{a_1}, \mathbf{a_2}, ...\mathbf{a_k}) \Longleftrightarrow \mathbf{R_i^B}(\mathbf{h}(\mathbf{a_1}), \mathbf{h}(\mathbf{a_2}), ...\mathbf{h}(\mathbf{a_k}))$$

If the Duplicator is able to maintain **(I-1)**, then, at the end of the rth round, i.e., when i = r, we have the following partial isomorphism

$$\begin{aligned} & h : (\cup_{j=1}^{r} N(a_j, 3^{r-r})) \longrightarrow (\cup_{j=1}^{r} N(b_j, 3^{r-r})) \\ \Longrightarrow \quad & h : (\cup_{j=1}^{r} N(a_j, 1)) \longrightarrow (\cup_{j=1}^{r} N(b_j, 1)) \\ \Longrightarrow \quad & h : \{a_1, a_2, ..., a_r\} \longrightarrow \{b_1, b_2, ..., b_r\} \end{aligned}$$

Furthermore, the conditions (a) and (b) of **(I-1)** implies that $\forall$ i, h(a$_i$)=b$_i$ and for any R$_i^A$, of arity k and for any {a$_1$,a$_2$, ... a$_k$} $\subseteq$ {a$_1$,a$_2$, ... a$_r$}, chosen during the game,

$$\mathbf{R_i^A}(\mathbf{a_1}, \mathbf{a_2}, ...\mathbf{a_k}) \Longleftrightarrow \mathbf{R_i^B}(\mathbf{h}(\mathbf{a_1}), \mathbf{h}(\mathbf{a_2}), ...\mathbf{h}(\mathbf{a_k}))$$

Thus, if the Duplicator is able to maintain the invariant **(I-1)**, then the Duplicator can win the r-round EF-game. Inductively, the Duplicator can maintain the invariant **(I-1)** after every round and will be described below.

1.2.1 Round 1

The Spoiler picks some vertex in one of the structures, say $a_1 \in \mathcal{A}$. The Duplicator considers the graph, $N(a_1,d)$ and picks a vertex from the other structure, here $b_1 \in \mathcal{B}$ such that $a_1 \sim_d b_1$. We are guaranteed to have a point b_1, since $\mathcal{A} \sim_d \mathcal{B}$.

By the definition of d-equivalence, we are guaranteed to have a bijection h that maintains invariant (**I-1**) and thus after Round 1, (**I-1**) is maintained.

1.2.2 Round i+1

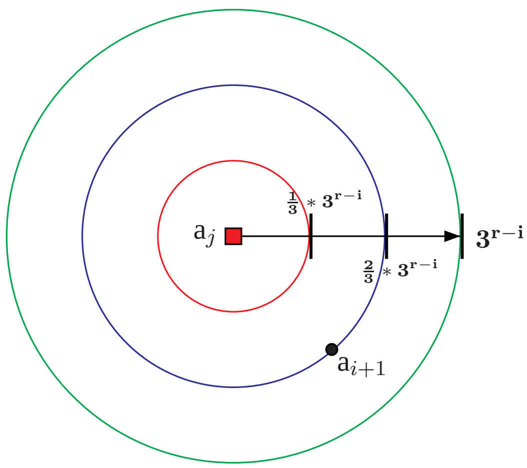
For the inductive case we assume that for all i, $1 \leq i < r$, we have the partial isomorphism h, given below that satisfies the requirements of (**I-1**)

$$\mathbf{h} : (\cup_{\mathbf{j=1}}^{\mathbf{i}} \mathbf{N}(\mathbf{a_j}, \mathbf{3^{r-i}})) \longrightarrow (\cup_{\mathbf{j=1}}^{\mathbf{i}} \mathbf{N}(\mathbf{b_j}, \mathbf{3^{r-i}}))$$

Given h we now need to give a strategy such that we get a new partial isomorphism, h' that satisfies (**I-1**) after round i+1. For this, there are two cases to be considered.

1.2.2.1 Case 1

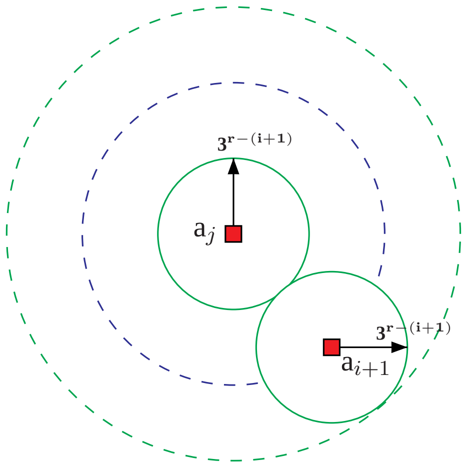
Figure 1.2: *At the beginning of Round i+1*



In this case, the element that the Spoiler picks is within the $\frac{2}{3}$*3^{r-i} (or 2*$3^{r-(i+1)}$) radius of a previously selected element. More formally, the Spoiler picks an element from one of the structures, say $a_{i+1} \in A$, and $\exists$ j, 1≤j≤i, such that $a_{i+1} \in N(a_j, 2*3^{r-(i+1)})$.

Now, the Duplicator can select the element $h(a_{i+1})$ as b_{i+1} and the new partial isomorphism

Figure 1.3: *At the end of Round i+1*



can be given by,

$$h' : (\cup_{j=1}^{i+1} N(a_j, 3^{r-(i+1)})) \longrightarrow (\cup_{j=1}^{i+1} N(b_j, 3^{r-(i+1)}))$$
$$and, \quad h'(a) = h(a)$$

The above function is well-defined as $(\cup_{j=1}^{i+1} N(a_j, 3^{r-(i+1)})) \subseteq (\cup_{j=1}^{i} N(a_j, 3^{r-(i)}))$ which implies that Domain(h') ⊆ Domain(h).

1.2.2.2 Case 2

The final case occurs when the Spoiler picks an element from one of the structures, say a_{i+1} from *A*, such that ∀ j, 1≤j≤i, $a_{i+1} \notin$ N(a_j,2*$3^{r-(i+1)}$).

This in turn implies that,

$$(N(a_{i+1}, 3^{r-(i+1)})) \cap (\cup_{j=1}^{i} N(a_j, 3^{r-(i+1)})) = \phi \tag{R2a}$$

Now, to prove that the duplicator can pick an element b_{i+1} and maintain **(I-1)**, we use a counting argument. Recall that we can fix a tuple of types $(Type_1, ...Type_n)$ by considering only the spheres of radius d which contain only max(*A*,*B*) nodes. Then, $\mathcal{A} \sim_d \mathcal{B}$ implies that the type signatures as considered above are component-wise equal.

Let N(a_{i+1},d) be of the type, Type_i. The Duplicator picks a point b_{i+1} such that $a_{i+1} \sim_d b_{i+1}$ and

$$(N(b_{i+1}, 3^{r-(i+1)})) \cap (\cup_{j=1}^{i} N(b_j, 3^{r-(i+1)})) = \phi \tag{R2b}$$

Such a point will exist and it can be proved by contradiction using the fact that the type signatures of the 2 structures are component-wise equal. As for the new isomomorphism, h', we first consider the isomorphism $g : N(a_{i+1}, 3^{r-(i+1)}) \rightarrow N(b_{i+1}, 3^{r-(i+1)})$ which satisfies the requirements (a) and (b) of **(I-1)**. The new isomorphsim h' is

$$h' : (\cup_{j=1}^{i+1} N(a_j, 3^{r-(i+1)})) \longrightarrow (\cup_{j=1}^{i+1} N(b_j, 3^{r-(i+1)}))$$

$$where, \quad h'(a) = \begin{cases} h(a) & \text{if } a \in \cup_{j=1}^{i} N(b_j, 3^{r-i}) \\ g(a) & \text{if } a \in N(a_{i+1}, 3^{r-(i+1)}) \end{cases}$$

Once again, h' is well-defined due to (R2a) and (R2b). It also satisfies **(I-1)** as g and h satisfy the invariant **(I-1)**.

□

1.3 Concluding Remarks

In the theorem above, we defined d-equivalence in two ways. One of the ways of defining d-equivalence was by stating that for any two structures $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{A} \sim_d \mathcal{B}$ iff the signatures $(\#Type_1, ...\#Type_n)$ are component-wise equal. In this definition, we are counting the exact number of times that any type occurs in the Gaiffman graph. However, we need not maintain the exact count and it is enough to count upto a threshold t and the result will still hold.

Notice that in the proof the maximum number of nodes that will ever be a part of any isomorphism is bounded by the number of rounds, r. Thus, the number of times we need to count any particular type is $\Sigma_{i=1}^{r} 3^{r-i}$ which is bounded by above by t=$r \times 3^r$. The d-type signatures with counting upto a threshold t are then denoted as $(Type_1, ...Type_n) \restriction t$ and two structures $\mathcal{A}$ and $\mathcal{B}$ are said to be d,t-equivalent if they are d-equivalent with threshold t, denoted by $\mathcal{A} \sim_{d,t} \mathcal{B}$. This might merge many d-equivalence classes but never breaks them up. If any two structures are d-equivalent, they remain d,t-equivalent.

Thus, in this chapter we have discussed local equivalence and first-order logical equivalence and how the two are related. In the remainder of my thesis, I studied papers that utilised Hanf's theorem to obtain some really amazing results pertaining to connections between logic and automata. Besides the connections with Hanf's theorem, which is the major theme of my thesis, the papers I read also had other ground-breaking results.

Chapter 2

MSO and automata over Pictures

In this chapter, we discuss pictures, which are basically extensions of words to the two-dimensional realm. A tiling system is defined over these objects and used as automata over these 2D-words. Finally, the expressiveness of these automata is studied. For the complete proofs of the main theorems, please refer [2].

2.1 Preliminaries

Definition 2.1.1 (Pictures, Picture Languages)**.**

A picture, p, over an alphabet Σ is basically a function of the form

$$p : \{1, 2, ..., n\} \times \{1, 2, ..., m\} \to \Sigma,$$
$$for\ some \quad n, m \in \mathbb{N}$$

The set of all pictures (over Σ) is the set of all possible functions p for every n,m $\in \mathbb{N}$. It is denoted by Σ^{**}. A picture language, L, is a subset of Σ^{**}

Thus, a picture is an extension of words to two-dimensions while Σ^{**} is the universe of all two-dimensional pictures. Some sample pictures are shown below ,

$$p : \begin{array}{cc} a & b \\ b & a \end{array} \qquad r : \begin{array}{ccccccc} a & a & a & a & a & a & a \\ b & b & b & b & b & b & b \\ a & a & a & a & a & a & a \\ b & b & b & b & b & b & b \\ a & a & a & a & a & a & a \end{array}$$

$$q : \begin{array}{ccc} a & a & b \\ a & b & a \end{array}$$

Figure 2.1: *Sample Pictures*

For convenience of defining automata over pictures, we use modified pictures $\widehat{p}$, obtained from a picture p by surrounding it by a special boundary symbol '#' $\notin \Sigma$. Note that $\widehat{p}$ is also a picture. Some examples of these special pictures obtained from the pictures p,r from 2.1 are shown below,

$$\hat{p}: \begin{array}{cccc} \# & \# & \# & \# \\ \# & a & b & \# \\ \# & b & a & \# \\ \# & \# & \# & \# \end{array} \qquad \hat{r}: \begin{array}{ccccccccc} \# & \# & \# & \# & \# & \# & \# & \# & \# \\ \# & a & a & a & a & a & a & a & \# \\ \# & b & b & b & b & b & b & b & \# \\ \# & a & a & a & a & a & a & a & \# \\ \# & b & b & b & b & b & b & b & \# \\ \# & a & a & a & a & a & a & a & \# \\ \# & \# & \# & \# & \# & \# & \# & \# & \# \end{array}$$

Figure 2.2: *Pictures surrounded by hash symbols*

Definition 2.1.2 (Concatenation of Pictures)**.**

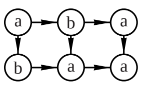
Let p and q be two pictures given by $p : [n_1] \times [m_1] \to \Sigma$ *and* $q : [n_2] \times [m_2] \to \Sigma$*. The row concatenation of p and q denoted p*$\ominus$*q is the new picture*

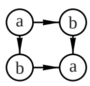
$$p \ominus q : [n_1 + n_2] \times [m_1] \to \Sigma$$

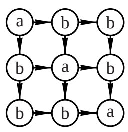
where p$\ominus$*q(i,j) = p(i,j) if i*$\leq n_1$ *else p*$\ominus$*q = q(i,j). It is well-defined only if* $m_1 = m_2$*.*

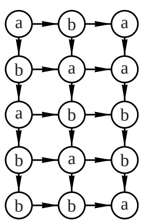
Column concatenation denoted by p$\oslash$*q is defined similarly. The example for row concatenation is given at Figure 2.3 and the one for column concatenation at Figure 2.4.*

Figure 2.3: *Examples for Row concatenation*

p : 

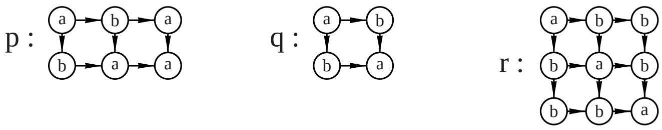
q : 

r : 

p $\ominus$ r : 

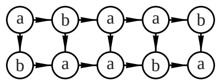
p $\ominus$ q : Undefined
q $\ominus$ r : Undefined
(As the number of columns are incompatible)

Figure 2.4: *Examples for Column concatenation*



p $\oslash$ r : Undefined
q $\oslash$ r : Undefined
(As the number of rows are incompatible)

p $\oslash$ q:



The concatenation operations can be applied on picture languages too.
Let L,L_1 and L_2 be 3 picture languages (subsets of Σ^{**}). Then,

(a) $L_1 \ominus L_2 = \{x \ominus y \mid x \in L_1$ and $y \in L_2\}$. Similarly for $L_1 \oslash L_2$.
(b) $L^{\ominus 1} = L$; $L^{\ominus n} = L^{\ominus(n-1)} \ominus L$. Similarly for $L^{\oslash n}$.
(c) Row Kleene Closure of L, $L^{*\ominus} = \bigcup_i L^{\ominus i}$.
(d) Column Kleene Closure of L, $L^{*\oslash} = \bigcup_i L^{\oslash i}$.

Definition 2.1.3 (Projections)**.**

Let Σ_1 and Σ_2 be two finite alphabets such that $\mid \Sigma_1 \mid \geq \mid \Sigma_2 \mid$ and $\pi : \Sigma_1 \rightarrow \Sigma_2$ is a mapping. Then given p $\in \Sigma_1^{**}$, $\pi(p)$ is the picture p' $\in \Sigma_2^{**}$ such that p'(i,j) = $\pi(p(i,j))\ \forall i, j$ such that, $1 \leq i \leq l_1(p), 1 \leq j \leq l_2(p)$ where l_1(p)is the number of rows of p and l_2(p) is the number of columns.

Similarly, given a picture language $L \subseteq \Sigma_1^{**}$, the projection of L by $\pi : \Sigma_1 \rightarrow \Sigma_2$ is defined as $\pi(L) = \{\pi(p) \mid p \in L\} \subseteq \Sigma_2^{**}$.

2.1.1 The Families LOC and REC

In this section, we describe the automata given as tiling systems over pictures. First, of all given a picture p of size (m,n), if $h \leq m$ and $k \leq n$, we denote by $T_{h,k}(p)$ the set of all subpictures (contiguous rectangular subblocks) of p of size (h,k).

Definition 2.1.4 (Local Picture Languages(LOC))**.**

A picture language $L \subseteq \Gamma^{**}$ is local if there exists a set Δ of pictures (or "tiles") of size ***(2,2)*** *over $\Gamma \cup$ {#}, such that L={p$\in \Gamma^{**} \mid T_{2,2}(\widehat{p}) \subseteq \Delta$} and Δ is a* local representation by tiles *for the language L.*

The family of all local languages is denoted by LOC. As an example of a picture language in LOC, consider $L_0 \subseteq \{0,1\}^{**}$ of square pictures (of size at least (2,2)) in which all nondiagonal positions carry symbol 0 whereas the diagonal positions carry symbol 1.

An appropriate set of tiles for L_0 consists of the 16 different (2,2)-subblocks of the picture displayed below in Figure 2.5

Figure 2.5: Δ *for* L_0 *is* $T_{2,2}(\widehat{p})$

$$\widehat{p}: \quad \begin{array}{cccccc} \# & \# & \# & \# & \# & \# \\ \# & 1 & 0 & 0 & 0 & \# \\ \# & 0 & 1 & 0 & 0 & \# \\ \# & 0 & 0 & 1 & 0 & \# \\ \# & 0 & 0 & 0 & 1 & \# \\ \# & \# & \# & \# & \# & \# \end{array}$$

Definition 2.1.5 (Recognizable Picture Language (REC))**.**
A picture language $L \subseteq \Sigma^{**}$ *is recognizable if there exists a local language L' over an alphabet* Γ *and a mapping* $\pi : \Gamma \rightarrow \Sigma$ *such that L=*$\pi(L')$

The family of all recognizable languages is denoted by REC. An example of a language in REC is the set of squares over $\Sigma = \{a\}$ and a suitable local language would be L_0 considered previously and the mapping, $\pi : \{0,1\} \rightarrow \{a\}$. The above language is nothing but the set of all square pictures labelled with the letter a.

Henceforth, we restrict our attention to only those local languages given by L'=(Γ, Δ) where $\Gamma = \Sigma\times$ Q is the alphabet of L'. The recognizable language, L $\subseteq \Sigma^{**}$ is obtained from the local language L' by the canonical projection $\pi : \Sigma\times$ Q $\rightarrow \Sigma$. It is sufficient to consider Local languages of the type above as every local language L' given in the definition, with an arbitrary alphabet may be modified into a local language with the alphabet, $\Gamma = \Sigma\times$ Q.

Under the above considerations, the tiling System is denoted by the triple $(\Sigma$,Q$, \Delta)$.

2.2 Some properties of the family REC

REC is closed with respect to

- projection,
- row and column concatenation,
- row and column closure,
- Boolean union and intersection.

Theorem 2.2.1. *REC is not closed with respect to Boolean complementation.*

Proof idea. Let Σ be an alphabet and let L be a language over Σ given by

$$L = \{p \in \Sigma^{**} \mid p = s \ominus s, where\ s\ is\ a\ square\}$$

The claim is that L $\notin$ REC while $\overline{L} \in$ REC. □

2.3 Logical definability of Picture Languages

Given a picture p $\in \Sigma^{**}$, we can identify the structure associated with the picture as

$$\underline{p} = (dom(p), S_1, S_2, (P_a)_{a\in\Sigma}),$$

In the logic x,y,z,x_1,x_2, ..., denote first-order variables for points of dom(p) while the variables X,Y,Z,X_1,X_2, ..., are MSO variables denoting sets of positions.

Atomic formulas are of the form **x=y, x$\mathbf{S_i}$y, X(x) and $\mathbf{P_a}$(x)** interpreted as equality between x and y, (x,y) $\in S_i$, x$\in X$,x$\in P_a$ respectively. Formulas are built up from atomic formulas by means of the Boolean connectives and the quantifiers $\exists$ and $\forall$ applicable to first-order as well as second-order variables.

If ϕ(X_1, ..., X_n) is a formula with at most X_1, ..., X_n occurring free in ϕ, p is a picture, and Q_1, ...,Q_n are subsets of dom(p), we write

$$((\underline{p}), Q_1, ..., Q_n) \models \phi(X_1, ..., X_n)$$

if p satisfies ϕ under the above mentioned interpretation where Q_i is taken as interpretation of X_i. If ϕ is a sentence we write $\underline{p} \models \phi$.

Definition 2.3.1 (MSO-definable)**.** *A picture language L is monadic second-order definable (L $\in$ MSO), if there is a monadic second-order sentence ϕ with L = L(ϕ).*

Definition 2.3.2 (FO-definable)**.** *A picture language L is first-order definable (L $\in$ FO), if there is a sentence ϕ containing only first-order quantifiers such that L = L(ϕ).*

Definition 2.3.3 (EMSO-definable)**.**

Finally, A picture language L is existential monadic second-order definable (L $\in$ EMSO), if there is a sentence of the form

$$\phi = \exists X_1 ... \exists X_n \psi(X_1, ..., X_n)$$

where ψ contains only first-order quantifiers such that L = L(ϕ).

2.4 Equivalence theorem for REC and EMSO

This is the main result of this chapter and of the article [2] with respect to picture languages.

Theorem 2.4.1. *For any picture language L, L$\in$ REC iff L $\in$ EMSO.*

Proof Sketch. (REC $\Longrightarrow$ EMSO)

The direction (REC $\Longrightarrow$ EMSO) is the easy one. Let L $\in$ REC and let (Σ,Q, Δ) be the tiling system accepting L. Now, we know that p$\in$ L iff $\exists$ a picture c $\in$ Q** of the same size as p such that $\widehat{p \times c}$ is tilable by Δ.

Hence, in order to prove theorem 2.4.1, it is sufficient to construct an EMSO formula ϕ such that given a picture p, ϕ $\models$ p iff $\exists$ a picture c $\in$ Q** of the same size as p such that $\widehat{p \times c}$ is tilable by Δ. The formula ϕ basically guesses a picture c $\in$ Q** and then verifies if $\widehat{p \times c}$ is tilable by Δ as given below,

$$\begin{aligned} \phi = \exists X_1 ... \exists X_k (\phi_{partition} \wedge \forall x_1 ... x_4 (\chi_m \wedge \chi_t \wedge \chi_b \wedge \chi_l \wedge \chi_r \\ \wedge \chi_{tl} \wedge \chi_{tr} \wedge \chi_{bl} \wedge \chi_{br})) \end{aligned}$$

where ,

$$\begin{aligned} \phi_{partition}(X_1, ..., X_k) = \forall z (X_1(z) \vee ... \vee X_k(z)) \\ \wedge \bigwedge_{i \neq j} \neg (X_i(z) \wedge X_j(z)). \end{aligned}$$

While χ_m,χ_t,χ_b,χ_l,χ_r,χ_{tl},χ_{tr},χ_{bl},χ_{br} refer to the formulae describing (2,2) local neighbourhoods that are a part of Δ.

This proves that REC $\Longrightarrow$ EMSO.

$\square$

The proof that EMSO $\Longrightarrow$ REC follows as a result of the following three theorems.

Theorem 2.4.2.

If L $\in$ EMSO then L is a projection of an FO-definable language.

Theorem 2.4.3.

A picture language is FO-definable iff it is locally threshold testable.

Theorem 2.4.4.

Any locally threshold testable language is the union of projections of local languages.

From theorems 2.4.2 , 2.4.3 , 2.4.4 and using the fact that REC is closed under union we can conclude that if L $\in$ EMSO then L is a union of projections of local languages and hence we get that L $\in$ REC.

Proof sketch of Theorem 2.4.2.

Let L $\in$ EMSO and let ϕ be an EMSO formula such that L=L(ϕ), given by,

$$\phi = \exists X_1 ... \exists X_n \psi(X_1, ..., X_n)$$

where ψ is a pure first order formula. If the alphabet of L is Σ_1, then, we consider the extended picture models of L where the alphabet is $\Sigma_2 = \Sigma_1 \times \{0,1\}^k$ and the atomic formula x $\in$ X_i is true over Σ_2 iff x = (x',t) and the i^{th} component t is 1.

Now it makes sense to speak of satisfiability of ψ over the alphabet Σ_2 with the semantics as described above, and let L' be the set of all pictures over Σ_2 that satisfy ψ. Now if we use the canonical mapping $\pi : \Sigma_1 \times \{0,1\}^k \longrightarrow \Sigma_1$ then we get L=π(L'). Thus L is the projection of an FO-definable language. $\square$

Before we discuss the other theorems, we need to define what we mean by locally threshold testable languages and locally strictly threshold d-testable languages which are slight variants of what we discussed in chapter 1. In the definitions below, we use squares instead of spheres for defining locality. Also, recall that, given a picture p, $T_{(i,j)}$(p) is the set of all sub-pictures of p of i rows and j columns.

(a) Given a picture p, $\#T_{(i,j)}(p)$ is the multiset of $T_{(i,j)}(p)$ which keeps track of the exact number of occurrences of each sub-picture of i rows and j columns in p, while, $\#T_{(i,j)}(p)\restriction$ t is the same as $\#T_{(i,j)}(p)$ counted upto the threshold t.

(b) Given two pictures, p_1,p_2 and d,t $\in \mathbb{N}$, if

$$\forall i,j \leq d, \#T_{(i,j)}(p_1) \restriction t = \#T_{(i,j)}(p_2) \restriction t$$

then, we say that p_1 is d,t-equivalent to p_2 denoted $p_1 \sim_{d,t} p_2$.

(c) For the same parameters as above, if

$$\forall i,j = d, \#T_{(i,j)}(p_1) \restriction t = \#T_{(i,j)}(p_2) \restriction t$$

then, we say that p_1 is exactly d,t-equivalent to p_2 denoted $p_1 \simeq_{d,t} p_2$.

(d) If L, is a union of $\sim_{d,t}$-equivalence classes then it is **locally d-testable with threshold t**.

(e) Given d, if it holds for some t,we say that L is **locally threshold d-testable**.

(f) If it holds for some d and t then we say L is **locally threshold testable**.

(g) If L is a union of $\simeq_{d,t}$-classes for some t, L is called **locally strictly threshold d-testable**.

The above definitions deal with locality conditions and they form the bridge for transferring a language from EMSO to the tiling systems. The proof makes crucial use of Hanf's theorem in making the transition.

Proof sketch of Theorem 2.4.3. (FO-definable $\Longleftrightarrow$ Locally threshold testable).

The proof for the direction $\Longrightarrow$ is by an adaptation of Hanf's theorem to pictures. We can use the bound as d=2·3^r+1 ard t=r·3^{2r} for r-equivalence. The reason for these values is that we are dealing with squares whereas the results proved in chapter 1 dealt with spheres. Hence, we accomodate the square into the sphere by providing more generous values.

For the other direction, we can actually describe a sub-picture of radius at most d in first order logic and say that it occurs exactly i times or less than i times or greater than i times. Thus, from this it follows that a locally threshold testable language is FO-definable. □

Thus, what we have learnt so far, in summary, is that if L∈ EMSO, then it is the projection of a locally threshold testable language. Now, for the last part which is the outline of the proof that any locally threshold testable picture language is the projection of a local language. Before that, we need one definition that forms a sort of transition step from local threshold systems to tiling systems.

Definition 2.4.1 (d-local picture language)**.**
Let d≥2 be a positive integer. A picture language L⊆ Σ^{**}_{d-2} is d-local if there exists a set $\Delta_{(d)}$ of pictures of size (d,d) (or "d-tiles") over Σ∪ {#}, such that L={p∈ $\Sigma^{**} \mid T_{d,d}(\widehat{p}) \subseteq \Delta^{(d)}$}

Proof sketch of theorem 2.4.4. (Locally threshold testable to Tiling systems)

This follows as a result of the following theorems.

Theorem 2.4.5. *Each locally threshold d-testable language L can be decomposed into L_0∪ L_1∪ ... ∪ L_{d-2} where $L_i \subseteq \Sigma_i^{**}$ (0≤i≤d-2) is locally strictly threshold (i+2)-testable.*

Theorem 2.4.6. *Let $d \geq 3$ be a positive integer. A locally strictly threshold d-testable picture language $L \subseteq \Sigma^{**}_{d-2}$ is the projection of a d-local language.*

Theorem 2.4.7. *A d-local picture language is a projection of a local language.*

□

Thus, we learn that expressibility of EMSO logic and recognizability of tiling systems coincides in the case of picture languages.

Chapter 3

Alternation hierarchy of MSO over grids and graphs

In this chapter, we discuss the MSO alternation hierarchy which is defined based on the number of second order quantifier alternations of an MSO formula. This hierarchy over grids and graphs is infinite and an overview of this fact will be given here. For further details and complete proofs see [3]. It should be noted that we count only the quantifier alternations of the variables denoting sets of positions and we can have arbitrary nesting of first order quantifier after the last second order quantification.

The inclusion results are as shown the diagram below with undirected edges indicating strict inclusion.

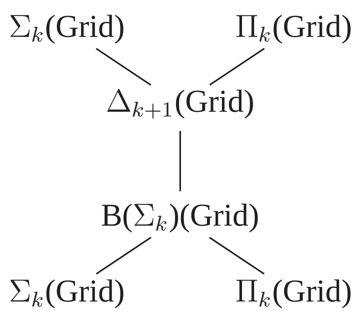


Figure 3.1: *Hierarchy of MSO over grids*

3.1 Preliminaries

The signature of grids is given by τ_{Grid} = ([m,n],$S_1^{m,n}$,$S_2^{m,n}$) where [m,n] = [m] $\times$ [n]. The relations $S_1^{m,n}$ denotes the horizontal successor and $S_2^{m,n}$ denotes the vertical successor. More formally, $\forall h, i,\ 1 \leq h, i \leq m$ and $\forall j, k,\ 1 \leq j, k \leq n$, the following holds

$$S_1^{m,n}((i,j),(h,k)) \Longleftrightarrow h = i+1 \ \ and \ \ k = j$$
$$S_2^{m,n}((i,j),(h,k)) \Longleftrightarrow h = i \ \ and \ \ k = j+1$$

Besides the logical structure of grids, we also use another related structure called a t-bit grid. The signature of t-bit grids for some t$\in$ $\mathbb{N}$ is given by τ_{t-Grid} = ([m,n],$S_1^{m,n}$,$S_2^{m,n}$,X_1,...,X_t) where the relation X_i is a unary relation.

The logic for discussing formulae over grids is similar to the one we have seen in the second chapter. The atomic formulae consist of x=y, X_iy (where X_i is a unary relation) and $S_i^{m,n}$(x,y). The first order formulae consist of the atomic formulae and closure with respect to the boolean combinations and first order quantifications.

The MSO formulae are built up like the first order formulae by closing with respect to boolean combinations and quantifications but the quantification is arbitrary and can be applied to first order as well as second order variables.

The MSO formulae belonging to Σ_k are defined inductively. The base case is when there are no second order quantifiers involved and thus no quantifier alternation. It is the set of all first order formulae denoted by Σ_0. Then, for every k $\geq$ 0, Σ_{k+1} is the smallest set of formulae that contains negations of all formulae in Σ_k and is closed under existential monadic second-order quantification. Π_k denotes the set of negations of formulae in Σ_k.

3.2 The MSO alternation hierarchy over grids

In this section, the main theorem of [3] is stated and discussed. Most of the hard work is done to show that the alternation hierarchy over grids is infinite. Once that is done, the results are then transferred to directed graphs by encoding grids into directed graphs and then all directed graphs are encoded into undirected graphs.

Thus the main theorem of this chapter is,

Theorem 3.2.1.

Boolean combinations of Σ_k-formulas over grids is a strict subset of Δ_{k+1} over grids.

$$\forall k \geq 1, B(\Sigma_k)(Grids) \subsetneq \Delta_{k+1}(Grids)$$

The above theorem implies that the hierarchy of MSO over grids is infinite. It is proved by using definability results for sets of grids. For a function f :$\mathbb{N} \rightarrow \mathbb{N}$ we denote by L_f the set of grids whose size is (m,f(m)) for m $\geq$ 1. A formula ϕ over τ_{Grid} defines the function f:$\mathbb{N} \rightarrow \mathbb{N}$ iff $Mod_{Grid}(\phi)$ = L_f.

Further, we say that a function is at most k-fold exponential if f(m) is $s_k(O(m))$, where,

$$s_0(m) = m$$
$$s_{k+1}(m) = 2^{s_k(m)}.$$

Proof Sketch.

The proof of Theorem 3.2.1 follows as a result of the following theorems,

Theorem 3.2.2. *Every B(Σ_k)-definable function is at most k-fold exponential.*

Theorem 3.2.3. *Given any k $\in \mathbb{N}$, we inductively define the function $f_k : \mathbb{N} \to \mathbb{N}$ as follows,*

$$f_1(m) = 2^m,$$
$$f_{k+1}(m) = f_k(m)2^{f_k(m)}.$$

Then, $\forall k \geq 1$,the function f_k is definable in Σ_k and Π_k over τ_{Grid}.

The function f_k is more than (k-1)-fold exponential and hence due to theorem 3.2.2, it cannot be described by a formula with $\leq$ k-1 second order quantifier alternations. Besides, the family of functions is infinite and for each k $\in \mathbb{N}$ there ***is*** a function f_k in the family such that it lies in the k^{th} level of the quantifier alternation hierarchy. These claims follow from the above two theorems and thus prove that the MSO quantifier hierarchy over grids is infinite. ☐

Proof Sketch for Theorem 3.2.2.

Here, we outline the various steps needed to show that B(Σ_k)-definable function is at-most k-fold exponential. The manner of doing it is by using the tiling systems and other results from standard automata theory. Also, we use t-bit grids to extend the logical structure of the grid with additional unary relations that provide the semantic interpretation to the second order free variables.

In chapter 2, given an alphabet Σ, we defined the set Σ^{**} as the universe of all pictures over the alphabet Σ. Here, given an alphabet Σ we denote by Σ^m, the subset of Σ^{**} with exactly m rows while $\Sigma^{m,1}$ is the subset of Σ^{**} of size m$\times$1.

- For a picture language L over alphabet Γ and an integer m $\geq$ 1, we denote by L(m) the language L restricted to Γ^m. Now, instead of considering L(m) as a picture language, we view it as a word language over the alphabet $\Gamma^{m,1}$ by merging all the rows along a particular column into a single letter in the alphabet $\Gamma^{m,1}$. We do this in order to make use of standard automata-theoretic results.

- $\forall$ t $\geq$ 0 and for every $\phi \in \Sigma_1$ with free variables among X_1, ... ,X_t, $\exists$c $\geq$1 such that for all m $\geq$1, there is an NFA with 2^{cm} states that recognises the word language $\mathrm{Mod}_t(\phi)$(m) over $(\{0,1\}^t)^{m,1}$.

- $\forall$ k $\geq$ 1 and for every $\phi \in \Sigma_k$ with free variables among X_1, ... ,X_t, $\exists$c $\geq$1 such that for all m$\geq$1, there is an NFA with s_k(cm) states that recognises the word language $Mod_t(\phi)$(m) over ({0,1}^{t})^{m,1}.

- Let N $\subseteq \mathbb{N}$ be recognizable by some n-state NFA. Then $\exists$ k $\leq$ (n+2)^2 and an integer p such that N is recognized by a DFA A with states 0,...,k+(p-1) such that A reaches the state k+((l-k) mod p) after reading an input of length $l \geq$k.

- Let ϕ be a B(Σ_k)-sentence. There is a constant c$\geq$1 such that for every m$\geq$1 the set $Mod_0(\phi)$(m) is s_k(cm)-periodic.

The end result of the above statements is that, given a formula $\phi \in$ B(Σ_k), we have an NFA, say $\mathcal{A}_\phi$, that is $s_k(cm)$-periodic accepting the language $Mod_0(\phi)$(m) when considered as a word Language over $\Gamma^{m,1}$. Now, if the formula $\phi \in B(\Sigma_k)$ defines a function f : $\mathbb{N} \rightarrow \mathbb{N}$, then for any m$\in \mathbb{N}$, f(m) is unique and hence there is only one picture, p such that $\phi \models$ p. If f(m) was more than $s_k(cm)$, then due to the $s_k(cm)$-periodicity of $\mathcal{A}_\phi$, it would accept more than one word and thus there would be more than one picture p, such that $\phi \models$p. Hence f is at most k-fold exponential. $\square$

Finally, we describe below how to construct a formula in MSO to describe a function f_k and thus prove the theorem 3.2.3. In fact, there are two ways of describing the function f_k, one of them is by a Σ_k formula while the other one is a Π_k formula.

Proof Sketch for Theorem 3.2.3.

Given k $\in \mathbb{N}$, a procedure exists whereby given a grid of height (number of rows) m, we can decide whether the width (or number of columns) is f_k(m). This procedure is in essence a method of counting over the grid. We will describe this method of counting and then hint as to how this method is captured by a Σ_k formula.

The method of counting over the grid is described inductively. The base case is when k=1, then the procedure simply consists of checking whether the number of columns is equal to 2^m. This is done by considering the 1-bit grid which is the original grid marked with the alphabet from {0,1} such that each column represents the binary number of the column when viewed as a word from top to bottom (the most significant bit on top and the least significant at the bottom). See Figure 3.2 to get an idea as to how the marking is done.

However, what is displayed in Figure 3.2 is not exactly the base case. Rather the Figures 3.2 and 3.3 indicate how to check the inductive case, i.e., whether the grid is of the size (m,f_k(m)). The manner we do that is by first marking the whole grid with {0,1} such that when considering the binary number represented by a column, with the most significant bit at the top row and the least significant bit at the bottom-most row, the value of the number will be the number of the column modulo 2^m. At the end of this marking, if the binary number along the last column

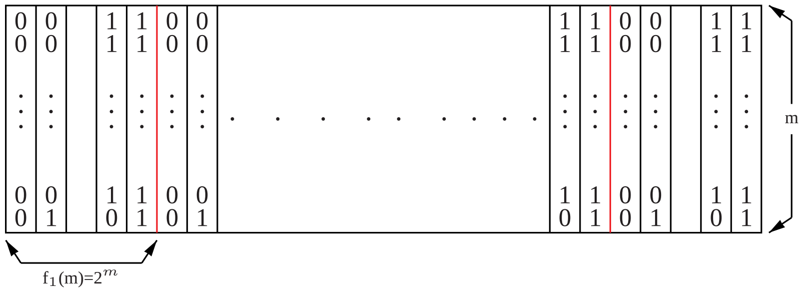


Figure 3.2: *The Complete column-numbering of the grid*

is not 2^{m-1}, then we can reject this grid as for the grid to be of the type (m,f_k(m)) it has to be divisible by f_i(m), $\forall$i, $1 \leq$ i < k. In particular, when i = 1, for the grid to be of the form (m,$f_k(m)$) the width of the grid has to be divisible by f_1(m) = 2^m. This initial marking is called the Complete column-numbering of the grid and is illustrated in Figure 3.2.

Now, for the second type of marking which verifies that the grid is actually of the form (m,f_k(m)), we mark only the top row by an additional layer of alphabets from {0,1}. Assuming that we have split the grid into f_i(m) pieces and we have marked the starting and the ending positions of these pieces. Now, the way to proceed from f_i(m) to f_{i+1}(m) is by adding an additional layer of {0,1} by using the f_i(m) pieces as the number of binary digits and counting the number of f_i(m) pieces from left to right as shown in Figure 3.3. If during any stage we do not have the last f_i piece to consist of all ones, then, we reject the grid. At the k^{th} level of the numbering we additionally have to check if there is exactly one sequence starting from all zeroes and ending in all ones, and only then do we accept the grid.

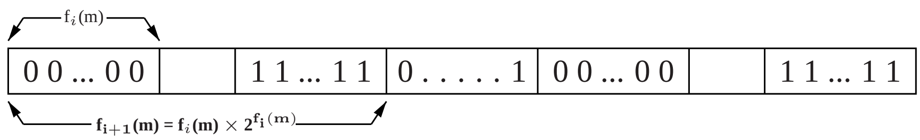


Figure 3.3: *Complete f_i-numbering along the top row of the grid*

The above procedure can be captured by formulae in Σ_k as well as Π_k. The Σ_k formula is constructed by second order variables that first describe the complete column numbering and then there is an inductive procedure to compute the complete f_{i+1}-numbering given the f_i-numbering. $\square$

3.3 Reduction from Grids to Graphs

In order to transfer the non-inclusion results from grids to graphs, we define and use a type of strong first-order reductions. This tool can be used to transfer separation results from one class of structures to another class of structures.

3.3.1 Strong First-order Reductions

This section first defines the strong first-order reductions and then, we state the theorems that will be used in the later part of this section. There are many kinds of reductions and the main idea in any reduction is to be able to translate a formula over one structure to a similar formula in the other structure without adding much to the logical complexity. What we mean by strong first order reduction is that there is only addition of some first order formulae in going from one structure to another.

Definition 3.3.1 (Strong First-order reduction)**.**

Let C be a class of structures over the relational signatures τ, C' a class of structures over the relational signature τ' and n≥1. Then,a strong first order reduction from C to C' with rank n is an injective mapping $\Phi : C \to C'$ such that

1. *For every structure M $\in$ C the universe of Φ(M) is given by $\bigcup_{i \leq n}(\{i\} \times dom(M))$, i.e., a disjoint union of n copies of the universe of M.*

2. *There is a first-order formula $\psi(x_1, ..., x_n)$ over τ' such that for all structures M in C, all $u_1, ..., u_n \in$ dom(M) and all $i_1, ..., i_n \leq$ n:*

$$\Phi(M) \models \psi[(i_1, u_1), ..., (i_n, u_n)] \iff \forall j \leq n : i_j = j \wedge u_j = u_1.$$

   (For structures in Φ(C), the formula ψ describes the n-tuples of the form ((1,u), . . . , (n,u)), which serve as representations of M-elements u.)

3. *For every relation symbol r' from τ' , say of arity l, and every $\kappa : [l] \to [n]$ there is an FO- formula $\phi_\kappa^{r'}(x_1, ..., x_l)$ over τ such that for all structures M in C and all $u_1, ..., u_l \in$ dom(M) we have*

$$M \models \phi_\kappa^{r'}[u_1, ..., u_l] \iff \Phi(M) \models r'[(\kappa(1), u_1), ..., (\kappa(l), u_l)]$$

4. *For every relation symbol r from τ, say of arity l, there is a first-order formula $\phi^r(x_1, ..., x_l)$ over τ' such that for all structures M in C and all $u_1, ..., u_l \in$ dom(M) we have*

$$M \models r[u_1, ..., u_l] \iff \Phi(M) \models \phi^r[(1, u_1), ..., (1, u_l)]$$

Theorem 3.3.1.

Let C, C' be classes of structures over the relational signatures τ and τ' , respectively. Let Φ be a strong first-order reduction from C to C' , and let L $\subseteq$ C. Then,

$$L \in \Sigma_k(C) \Longleftrightarrow \exists L' \in \Sigma_k(C') \;\; with \;\; \Phi(L) = L' \cap \Phi(C)$$

If, additionally, $\Phi(C)$, is Σ_k-definable, then

$$L \in \Sigma_k(C) \Longleftrightarrow \Phi(L) \in \Sigma_k(C')$$

Thus, strong first-order reductions is a type of embedding that has the required properties for transferring the separation results from one class of structures to another.

We say that C is *strongly first-order reducible* to C' iff there is a strong first-order reduction from C to C'. By the above theorem 3.3.1, it is sufficient to give strong first-order reductions from grids to other structures, to prove that the hierarchy over the other structure is also infinite. This is what will be done to show that the hierarchy over graphs is infinite.

3.3.2 Reduction from Grids to Directed Graphs

Now, to transfer the hierarchy results from grids to graphs, we give a reduction from grids to graphs of the kind described in Definition 3.3.1. The image is Π_1-definable. For every R $\in$ Grid we associate the graph $\Phi(R) := (\{1,2\} \times dom(R), E)$ with

$$\begin{aligned} E = \{((1,x),(1,x)) \mid x \in domR\} \\ \cup \{((1,x),(2,x)) \mid x \in domR\} \\ \cup \{((2,x),(1,y)) \mid (x,y) \in S_1^R\} \\ \cup \{((2,x),(2,y)) \mid (x,y) \in S_2^R\} \end{aligned}$$

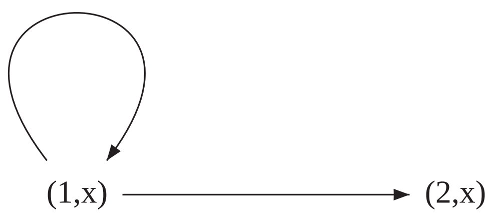


Figure 3.4: *gadget to embed grids into Directed Graphs*

Let's define the various formulae that are required of a first order reduction given in Definition 3.3.1 to show that the above reduction Φ is indeed a Strong First Order reduction. First, we give the first order formula for Definition 3.3.1 (2), of arity n=2, where n is the number of copies, that allow us to pick a particular element in the various copies. It is given by, $\psi(x_1, x_2)$=$E(x_1, x_1) \wedge E(x_1, x_2) \wedge \neg E(x_2, x_2)$. Next, the various formula for (3) $\forall f : [l] \to [n]$, and since l is 2(arity of E) and n is also 2 (number of copies), the four formulae for each function $f : [2] \to [2]$ is given by,

$$\begin{aligned}
\phi^E_{(1,1)} &= x_1 = x_2 \\
\phi^E_{(1,2)} &= x_1 = x_2 \\
\phi^E_{(2,1)} &= S_1(x_1, x_2) \\
\phi^E_{(2,2)} &= S_2(x_1, x_2)
\end{aligned}$$

Now,finally, to describe the first-order formulae in a graph that captures the relations over grids, to fulfill (4), the formulae are,

$$\begin{aligned}
\phi^{S_1}(x_1, x_2) &= \exists y_1 \exists y_2 (\psi(x_1, y_1) \wedge \psi(x_2, y_2) \wedge E(y_1, x_2)) \\
\phi^{S_2}(x_1, x_2) &= \exists y_1 \exists y_2 (\psi(x_1, y_1) \wedge \psi(x_2, y_2) \wedge E(y_1, y_2))
\end{aligned}$$

Thus, the reduction Φ from grids to directed graphs satisfies all conditions stipulated in Definition 3.3.1 and hence we have a strong reduction from grids to directed graphs.

3.3.3 Reduction from Directed Graphs to Undirected Graphs

Finally, we give a reduction from directed graphs to undirected graphs. For this reduction too, the image is Π_1-definable. Given a directed graph G = (V,E), we associate it with the undirected graph given by $\Phi(G)$=(V', E'), where $V' = [6] \times V$ and the edge set E' among the six copies is as shown in Figure 3.5 plus an edge between (1,x) and (6,y) whenever (x,y)$\in E$.

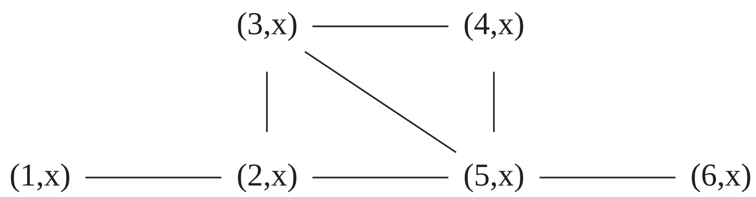


Figure 3.5: *Gadget for embedding directed graphs into undirected graphs*

Formally, let

$$M = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,2\},\{3,5\},\{5,6\}\}$$
$$N = \{\{1,3\},\{1,4\},\{1,5\},\{2,4\},\{2,6\},\{3,6\},\{4,6\}\}$$

The edge set of $\Phi(G)$ is given as

$$E' = \{((i,x),(j,y)) \mid (x = y \wedge \{i,j\} \in M) \vee ((i,j) = (1,6) \wedge (x,y) \in E(G))\}.$$

The formulae required to show that Φ is indeed a strong first-order reduction that satisfy the conditions stipulated in Definition 3.3.1 are described below,

$$\psi(x1,...,x6) = \forall z(E(z,x_4) \rightarrow (x_3 = z \vee x_5 = z)) \wedge \bigwedge_{\{i,j\}\in M} (E(x_i,x_j)) \wedge \bigwedge_{\{i,j\}\in N} \neg(E(x_i,x_j)).$$

ψ fixes the six nodes of a gadget (in the sense of Definition 3.3.1 – (2)).

In order to satisfy Definition 3.3.1 – (3), for any relation r in $\Phi(G)$, we need first order formulae $\forall f$ such that $f : [l] \rightarrow [n]$ where l is the arity of the relation r and n is the number of copies used in Φ(G). There is only one binary relation in an undirected graph, say E', and let the only binary relation we are allowed to use in the Directed graph be E. Then,

$$\phi^E_{(i,j)}(x,y) = \begin{cases} x = y & \text{if } \{i,j\} \in M \\ E(x,y) & \text{if } (i,j) = (1,6) \\ E(y,x) & \text{if } (j,i) = (6,1) \\ \neg(x = x) & \text{else} \end{cases}$$

Finally, for Definition 3.3.1 – (4) we have,

$$\phi^E(x,y) = \exists x_2...x_6 \exists y_1...y_5(\psi(x,x_2,...,x_6) \wedge \psi(y_1,...,y_5,y) \wedge E(x,y)).$$

Thus, we have managed to produce Strong First Order reductions from grids to Directed graph and from Directed graphs to Undirected graphs and hence by appealing to Theorem 3.3.1, the quantifier alternation hierarchy of MSO over grids, directed graph and undirected graphs is infinite.

Chapter 4

Message-Passing Automata and EMSO over MSC

The main objects of discussion in this chapter is Message Sequence charts(MSC) and Message-Passing automata over MSCs which are used in distributed systems development. An MSC is basically a drawing scenario as shown in Figure 4.1 describing how we intend the behaviour of the system to be. A collection of MSCs may be designed to depict the various ways in which concurrent processes may react with one another during any run of the system. The message-passing automata helps in realizing a set of desirable MSCs.

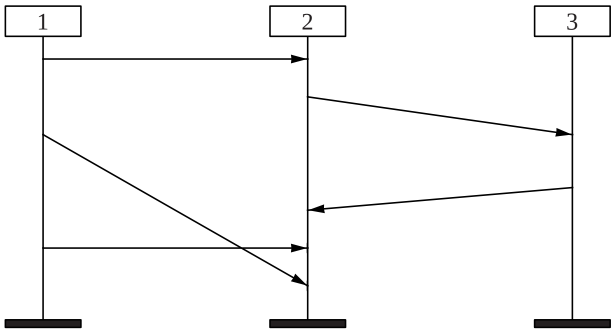


Figure 4.1: *A sample Message Sequence Chart for three processes*

The main result of this chapter is that MPA and EMSO$_{MSC}$ are expressively equivalent. Similar to the proof for the equivalence of EMSO over pictures and tiling systems discussed in Chapter 2 , in this proof too, we use Hanf's theorem as an intermediate step in arriving at the result. Then, finally we prove that EMSO is strictly less powerful than MSO by proving the more general theorem that the quantifier alternation hierarchy of MSO over MSCs is infinite.

This last result is obtained by embedding the grids into MSC and appealing to the theorem discussed in Chapter 3.

For complete proofs and additional details of any theorem or definition please see [1] which is the primary reference for this chapter.

4.1 Preliminaries

In this section we will formalize the definitions of Message Sequence charts and Message passing automata. An MSC can be represented in many ways. In the paper [1], it is described as a partial order on the events that take place. Further, any partial order may be represented as a special kind of graph, called a directed acyclic graph and this is explained first.

Let Σ, C be two sets of alphabets. Then, a ***(directed) graph*** over (Σ, C) is represented by the structure G = $(E, \{\lhd_c\}_{c \in C}, \lambda)$ where E is its nonempty finite set of nodes, the $\lhd_c \subseteq$ E $\times$ E are disjoint binary relations on E, and λ : E $\rightarrow \Sigma$ is a (node-)labeling function.

Let the set of all ***acyclic*** graphs of the type above be denoted by $\mathbb{DG}(\Sigma, C)$. Given two processes p and q, p!q denotes the event "p sends to q " while q?p denotes "q receives from p". Let Act = {p!q} $\cup$ {p?q} where p,q are processes from a set P=[n], where n$\in \mathbb{N}$. Finally, the set P_c = P $\uplus$ {c} is the set of labels over the edges. The labels from P will label subsequent events of a particular process (along a process line) while the label 'c' is marked along the channels (across processes). Ch(P) is a subset of P$\times$P such that (p,q) $\in$ Ch(P) iff there is a channel from p to q.

Having given enough background, I will now introduce the definitions for MSCs as well as MPA. The definitions are not completely formal as presented in [1], but they convey the essence of what is given in that paper.

Definition 4.1.1 (Message Sequence Charts)**.**

An MSC (over P) is a graph M = (E,$\{\lhd_p\}_{p \in P}$,$\lhd_c$,λ) $\in \mathbb{DG}$(Act,P_c) such that

- $\lhd_p$ *is a total order that connects any event along a process line with its successor event (if one exists).*
- $\lhd_c \subseteq E \times E$ *is the set of edges denoting messages such that for any e, e' $\in$ E, e$\lhd_c$e' iff*
  - $\lambda(e)$ *= p!q and* $\lambda(e')$*=q?p and (p,q) $\in$ Ch(P)*
  - *For some n$\in \mathbb{N}$, if the number of messages of type p!q sent by p before e is (n-1), then, the number of messages of type q?p received by q is also (n-1).*
- $|\ \{e \in E \mid \lambda(e) = p!q\}\ |$ = $|\ \{e \in E \mid \lambda(e) = q?p\}\ |$ *(Basically, the channel is reliable, in the sense that every message that is sent is received).*

The figure 4.2 illustrates an example of a sample MSC graph where the set of processes is given by P={1,2,3,4,5}. Notice the labelling over any edge of the form e$\lhd_p$e' is p while the edges of the form e$\lhd_c$e' is labelled by 'c'.

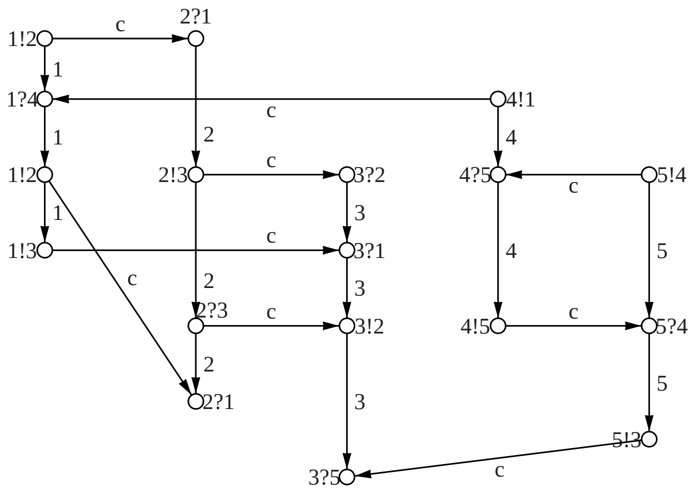


Figure 4.2: *Example of an MSC graph*

Definition 4.1.2 (Message-Passing Automata)**.**
MPA (over P) is a structure (A) = $(((A)_p)_{p\in P}$,$\mathcal{D}$,s^{-in},F) *such that*

- $\mathcal{D}$ *is a set of synchronization messages.*
- *for each p∈P ,*$(\mathcal{A})_p$ *is a pair* (S_p, δ_p) *where*
  - S_p *is a set of local states*
  - $\delta_p \subseteq S_p\times$ *Act*$_p \times \mathcal{D} \times S_p$
- s$^{-in} \in \Pi_{p\in P}S_p$ *is the global initial state.*
- $F \subseteq \Pi_{p\in P}S_p$ *is the set of global final states.*

Thus, the message passing automaton is given by a set of local automata, one for each process. The run of an automaton over an MSC is by running each local automaton along its process line and guessing the next state from its local transitions. The initial state of a local automaton is given by the projection of s^{-in} onto the particular process. The final state of any run is the global final state along the maximal events. We say that the MPA $\mathcal{A}$ accepts an MSC if there is an accepting run over it.

An example of a message-passing automata with two processes and with the synchronization messages as {o,x} is illustrated in Figure 4.3. The diagram only shows the behaviour of the two local automata.

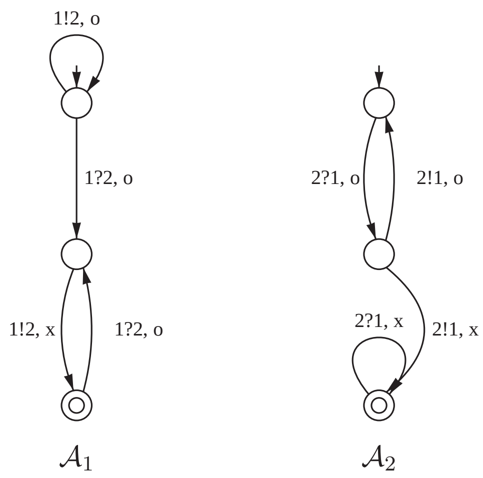


Figure 4.3: *Example of an MPA*

Now, we move over to the logical characterisation of MSCs. For this, we fix the supply of first-order variables to be Var = {x,y, ...} denoting events in a graph while the supply of Second-Order variables are from VAR={X,Y,...} denoting subset of events in a graph.

Definition 4.1.3 (MSO over $\mathbb{DG}(\Sigma, C)$)**.**
MSO(Σ,C) over the class $\mathbb{DG}(\Sigma, C)$ are built up from the atomic formulas λ(x) = a (for a $\in \Sigma$), x$\lhd_c$y (for c$\in$ C), x$\in$ X and x=y and then, as usual, we close under boolean operations and quantifications of first-order as well as second-order variables.

Now, given a graph G = $(E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}$, and two nodes e,e' $\in$ G, we define d$_G$(e,e') as k, the minimum natural number such that there is a sequence of elements e$_0$,...,e$_k$ $\in$ E with e$_0$=e, e$_k$=e' and e$_i \lhd_c$e$_{i+1}$ or e$_{i+1}\lhd_c$e$_i$ for each i $\in$ {0, . . . , k - 1}. If there is no such natural number k, then d$_G(e, e') = \infty$. For some R$\in \mathbb{N}$, an R-sphere $\in \mathbb{DG}(\Sigma, C)$ is a graph denoted by H = $(E, \{\lhd_c\}_{c \in C}, \lambda, \gamma)$, where γ is the distinguished element called the center such that for any node e $\in E$, d$_H(\gamma, e) \leq$ R.

Definition 4.1.4 (Graph Acceptor)**.**
A graph acceptor over (Σ,C) is a structure $\mathcal{B}$ = ($\mathcal{Q}, \mathcal{R}$,$\mathbb{S}$,Occ) where

- *$\mathcal{Q}$ is its nonempty finite set of states*
- *$\mathcal{R} \in \mathbb{N}$ is the radius*
- *$\mathbb{S}$ is a finite set of R-spheres over ($\Sigma \times$ $\mathcal{Q}$,C) and*
- *Occ is a boolean combination of conditions of the form "sphere H $\in \mathbb{S}$ occurs at least n times" where n $\in \mathbb{N}$.*

Any formula ϕ in EMSO(Σ, C) is of the form $\exists X_1 \exists X_2 ... \exists X_k \psi(X_1, X_2, ..., X_k)$, where ψ is a first order formula. Now, if we consider graphs wherein each node in addition to the alphabets from Σ is also labeled with the k-tuple vector $\{0,1\}^k$, then, checking for satisfiablility of EMSO over these graphs with the canonical semantics reduces to checking the satisfiability of the inner core FO sentences.

The graph acceptor is an automata-theoretic approach to the local equivalence described in the discussion of Hanf's theorem. A run of the graph acceptor on any graph G = $(E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$, is given by $\rho : E \to Q$ such that for any e $\in$ E, the R-sphere of G around e is isomorphic to some sphere H $\in \mathbb{S}$. The run ρ is said to be accepting if it satisfies OCC.

4.2 Equivalence between MPA AND EMSO$_{MSC}$

Theorem 4.2.1. *For any class $\mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$ of bounded-degree , EMSO$_{\mathcal{K}}$ = $\mathcal{GA}_{\mathcal{K}}$*

Note that for any graph G = $(E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{MSC}$, the degree of G is bounded by 3 and hence it holds that for any EMSO formula ϕ over $\mathbb{MSC}$, we can find a corresponding graph-acceptor B that accepts the same language as that accepted by ϕ.

Theorem 4.2.2. *MPA $\Longleftrightarrow$ EMSO$_{MSC}$*

Proof Sketch of Theorem 4.2.2.

The proof for the direction ($\Longrightarrow$) is done using the standard translation of an automata to an EMSO formula.This method involves guessing a set of states from the MPA to label the MSC with and then, checking that the guessed set of states for each second order variable is a valid run of the MPA. Finally, we have to add a clause that the states labelling the maximal nodes belongs to the set of final states of the MPA.

The other direction (MPA $\Longleftarrow$ EMSO$_{MSC}$) is the harder one and we use graph acceptors as an intermediate stage before constructing the actual MPA from the EMSO formula. By Theorem 4.2.1, and the fact that an MSC is a graph of maximum degree 3, it follows that for any $\phi \in$ EMSO$_{MSC}$, we have an equivalent graph acceptor $\mathcal{B}$ = ($\mathcal{Q}, \mathcal{R}$,$\mathbb{S}$,Occ) that accepts the same language of MSCs that are accepted by ϕ.

Having obtained an equivalent graph acceptor $\mathcal{B}$ = ($\mathcal{Q}, \mathcal{R}$,$\mathbb{S}$,Occ), from the formula ϕ, we construct an MPA $\mathcal{A}$ that accepts the same language as that accepted by $\mathcal{B}$. The idea behind the construction is to create a message-passing automaton that mimics the behaviour of the graph acceptor. For this, the state of the automaton will be of the form $2^{\mathbb{S}}$, a subset of the possible spheres used in the graph acceptor alongwith some extra information.

Thus, for defining the states of $\mathcal{A}$, we introduce what are called extended spheres, denoted by $\mathbb{S}^+$={$((E, \{\lhd_c\}_{c\in C}, \lambda, \gamma, e), i) \mid (E, \{\lhd_c\}_{c\in C}, \lambda, \gamma) \in \mathbb{S}, e \in E, i \in \{0, 1 ... 4 \cdot maxE^2 + 1\}$}. The maximum number of nodes of any sphere in $\mathbb{S}$ is denoted by maxE and the extra distinguished node in the sphere is the active or current node that is being read by the MPA $\mathcal{A}$. Also we denote by $\mathbb{S}_p \subset \mathbb{S}$, such that $\lambda(\gamma)$ = pθq while $\mathbb{S}^+_p \subset \mathbb{S}^+$ is the set of extended spheres such that $\lambda(e)$ = pθq, where $\theta \in$ {!,?}, a send or a receive.

Now, any local automaton ($\mathcal{A}_p$) of $\mathcal{A}$, is given by (S_p, δ_p) where each $s \in S_p$, is of the form (X,ν) where X $\subseteq \mathbb{S}^+$ such that

(a) $\exists$ exactly one sphere $\in$ X where the active node and the center coincide (γ = e).

(b) For any two spheres (($E, \{\lhd_c\}_{c\in C}, \lambda, \gamma$,e),i), (($E', \{\lhd'_c\}_{c\in C}, \lambda', \gamma'$,e'),i') $\in$ X,

- $\lambda(e)$ = $\lambda(e')$ $\in$ Act$_p\times$ Q is the same.
- If $((E, \{\lhd_c\}_{c\in C}, \lambda, \gamma, e), i) \cong ((E', \{\lhd'_c\}_{c\in C}, \lambda', \gamma', e'), i')$ and i = i' then e = e'.

(c) ν is a mapping $\mathbb{S}_p \rightarrow \{0, ..., max(Occ)\}$ and let ν_p^0 be a function that sets every sphere R$\in \mathbb{S}_p$ to 0.

The set of messages is D $\subseteq 2^{\mathbb{S}^+} \times 2^{\mathbb{S}^+}$ wherein, the first component of a message contains obligations the receiving state/event has to satisfy, while the second component imposes requirements that must not be satisfied by the receiving process to ensure isomorphism. Moreover, s^{-in}= ((ϕ, ν_p^0))$_{p\in P}$ and, for (S_p, ν_p) $\in$ S_p ,((S_p, ν_p))$_{p\in P}$ $\in$ F if the union of mappings ν_p satisfies Occ and, for all p $\in$ P and $((E, \lhd, \lambda, \gamma, e), i) \in S_p$, e is maximal in ($E_p, \leq p$).

Let (S, ν) and (S', ν') be two states in S_p and let $((E, \lhd, \lambda, \gamma, e), i)$ and $((E', \lhd', \lambda', \gamma', e'), i')$ be two arbitrary spheres belonging to S and S' repectively. Then, the definition of the p-local transition relation Δ_p is such that if ($(S, \nu), \sigma, (P, N), (S', \nu')$) $\in \Delta_p$ then the following hold:

1. $\lambda(S') = (\sigma, q)$, for some q $\in$ Q.
2. If $(E, \lhd, \lambda, \gamma) \cong (E', \lhd', \lambda', \gamma')$, and $i = i'$, then e$\lhd_p$e'.
3. If e' is not minimal in ($E', \leq_p$), then $\exists$ (($E', \lhd', \lambda', \gamma', e^-), i'$)) $\in S$ such that $e^- \lhd_p$ e.
4. If e is not maximal in ($E, \leq_p$), then $\exists$ (($E, \lhd, \lambda, \gamma, e^+), i'$)) $\in S'$ such that e $\lhd_p$ e^+.
5. If S$\neq \phi$ and e' is minimal in ($E'_p, \leq_p$) then, d(e',γ')=R.
6. If e is maximal in ($E_p, \leq_p$), then, d(e,γ)=R.
7. (i) In case that σ = p!q for some q $\in$ P :
   (a) For any e" $\in E'$, if e'$\lhd_c$e", then (($E', \lhd', \lambda', \gamma', e''$),i') $\in$ P.
   (b) For any e" $\in E'$, if $\neg(e' \lhd_c e'')$, then (($E', \lhd', \lambda', \gamma', e''$),i') $\in$ N.
   (c) If (($E, \lhd, \lambda, \gamma$,e),i) $\in$ P $\wedge \exists e' \in E(e' \lhd_c e) \Rightarrow ((E, \lhd, \lambda, \gamma$,e'),i) $\in$ S'.

   (ii) In case that σ = p?q for some q $\in$ P :
   (a) P $\subseteq$ S'
   (b) N $\cap$ S' = ϕ, and
   (c) For any e" $\in E'$, if e"$\lhd_c$e',then (($E', \lhd', \lambda', \gamma', e'$),i') $\in$ P.

8. $\nu' = \nu[c(S')/min\{\nu(c(S')) + 1, max(Occ)\}]$ (where c(S') is a sphere in S' such that the center is the active node, ν' is basically the same as ν except for the value of c(S')).

Thus, from the definition of B, we have described the corresponding automaton $\mathcal{A}$. Now, to complete the proof for the Theorem 4.2.2, all we need to show is that L(B) = L($\mathcal{A}$). To show that any MSC,M = ($E, \lhd, \lambda, \gamma$), accepted by B will also be accepted by $\mathcal{A}$, we first assign $\rho : E \to Q$ to be the accepting run of B on M.

Given ρ, it is easy to break up the marking into pieces and distribute them and mark each event in the MSC by a subset of extended spheres, $2^{\mathbb{S}^+}$ such that each sphere in the set reflects the local structure of the MSC. Also we fix the values of i in each extended sphere such that there are no isomorphic copies denoting different centres in the same set with the same i value. Then, it is easy to argue that this marking is a valid run of $\mathcal{A}$ on M.

The other direction is slightly more involved as we have to prove that anything that $\mathcal{A}$ accepts is also accepted by B. For this, we assume an accepting run of $\mathcal{A}$ on M. As each set has a distinct center, we obtain the mapping ρ as the projection of the state of the set labelling the sphere with center as the current event onto the event.

Now, that we have the mapping, we have to show that this mapping is actually acepting for B. That is done by showing that the extended sphere actually simulates the MSC and the MSC simulates the extended sphere. The heart of the matter is that if the MPA finds a run in the MSC then, the local spheres actually represent the local neighbourhood in the MSC and vice versa. This then entails that the run ρ will also be accepting for B.

$\square$

4.3 Infinite hierarchy of MSO over MSCs

Theorem 4.3.1. *The monadic quantifier alternation hierarchy over MSC is infinite.*

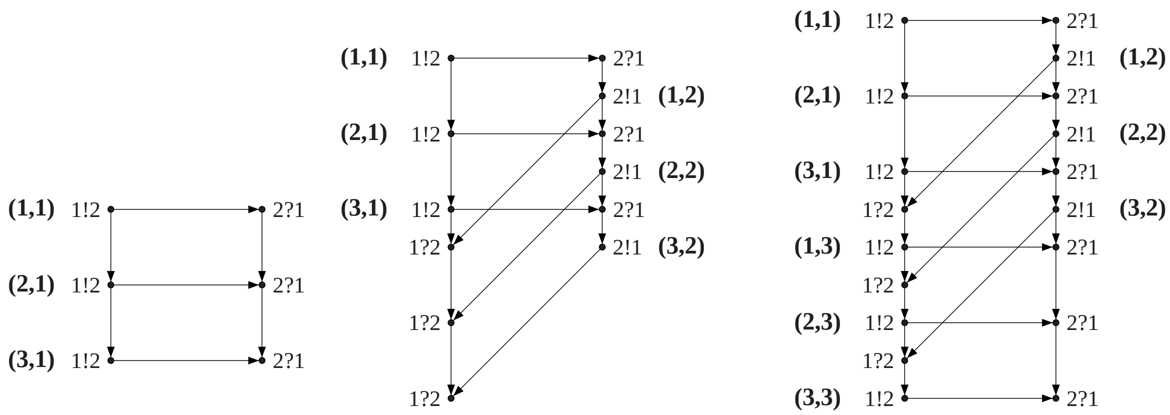
Proof Sketch of Theorem 4.3.1.

The idea is to use the theorems from Chapter 3 with some modifications to obtain similar results for MSCs. First, the emebddings of the grids into the MSC are discussed. Then, we show that the set of MSCs that are a valid embedding of some grid (denoted M(n,m) for some n,m $\in \mathbb{N}$) can be expressed in EMSO. Having done that, we show how to translate any Σ_k formula, ϕ, over grids to an equivalent Σ_k formula, ϕ' over the family M(n,m), $\forall n, m \in \mathbb{N}$.

Finally, the main proof of Theorem 4.3.1 is in the same spirit as that discussed over grids. However, here we construct an MPA instead of an NFA to show that any function g: $\mathbb{N} \to \mathbb{N}$ definable in Σ_k is at most $s_k(O(n))$ exponential. Then, we show that there is a particular function $f_{k+1} : \mathbb{N} \to \mathbb{N}$ in Σ_{k+1} that is not $s_k(O(n))$ exponential where $f_k : \mathbb{N} \to \mathbb{N}$ and $s_k(n)$ are as defined in the Matz, Thomas paper over grids. To recall, $s_0(n) = 2^n$ and $s_{k+1}(n) = 2^{s_k(n)}$ while $f_0(n) = n$ and $f_{k+1}(n) = f_k(n) \cdot 2^{f_k(n)}$. Clearly $f_{k+1}(n) \notin s_k(O(n))$ but $f_k(n) \in \Sigma_k(n)$

(by the formula over grids and our embedding) and thus we can say that the MSO hierarchy over MSCs is also infinite.

Now, to describe the embedding, we start of with a few examples as illustrated for some sample grids in Figure 4.4. We need an MSC with only two processes to embed any arbitrary grid and the figure illustrates the embedding for the grids 3×1, 3×2 and 3×3.



(a) 3×1 **(b) 3×2** **(c) 3×3**

Figure 4.4: *Embedding of grids into MSCs*

The grids are basically "folded" such that any point on the grid is represented by a send event and any send event in the MSC corresponds to a point on the grid. The MSC, by definition has to have the receive events for these sends as the channels are lossless and these receive events are also used to ensure proper alignment and to capture the family of M(n,m) by an EMSO formula. Formally, M(n,m) is given by its projections as follows,

$$M(n,m) \upharpoonright \{Act_1, \{1\}\} = \begin{cases} (1!2)^n[(1?2)(1!2)]^{(n\cdot((m-1)/2))} & \text{if } m \text{ is odd} \\ (1!2)^n[(1?2)(1!2)]^{(n\cdot((m/2)-1))} & \text{if } m \text{ is even} \end{cases}$$

$$M(n,m) \upharpoonright \{Act_2, \{2\}\} = \begin{cases} [(2?1)(2!1)]^{(n\cdot((m-1)/2))}(2?1)^n & \text{if } m \text{ is odd} \\ [(2?1)(2!1)]^{(n\cdot(m/2))} & \text{if } m \text{ is even} \end{cases}$$

Theorem 4.3.2. *For each k $\in \mathbb{N}$, the MSC language,$L(f_k)$ is Σ_{2k+3}-definable.*

Theorem 4.3.3. *Let $f : \mathbb{N} \to \mathbb{N}$ be a function. If $L(f)$ is (Σ_k)-definable over MSC for some k, where k$\geq$ 1, then f (n) is in $s_k(O(n))$.*

The EMSO-sentence that defines the set of all grid foldings can be defined by $\phi_{GF} = \exists\overline{X}\psi_{GF}(\overline{X})$ (over MSCs) with first-order kernel $\psi_{GF}(\overline{X})$ by saying that there is a chain of events iterating between process 1 and 2 and each chain consists of alternating send and receive events.The second variables $\overline{X} = \{X_1, X_2\}$ is enough for expressing the above formula.

From the Σ_k formula $\phi = \exists\overline{Y_1}\forall\overline{Y_2}...\exists/\forall\overline{Y_k}\phi'(\overline{Y_1},...,\overline{Y_k})$ with a core first order formula ϕ' over grids, we obtain the equivalent Σ_k formula Ψ_ϕ, over M(n,m), given by

$$\exists Z\exists\overline{X}\exists\overline{Y_1}\forall\overline{Y_2}...\exists/\forall\overline{Y_k}(\psi_{bottom}(Z) \wedge \psi_{GF}(\overline{X}) \wedge \|\phi'(\overline{Y_1},...,\overline{Y_k})\|_Z)$$

where $\psi_{bottom}(Z)$ ensures that Z refers only to the points at the end of any column and it can be constructed by taking the chain of elements from the last maximal receive on any process 1 or 2. This is done to ensure that the vertical successor does not refer to any point that is not actually a vertical successor. Then, obtaining the corresponding inner formula $\|\phi'(\overline{Y_1},...,\overline{Y_k})\|_Z$ is simply a matter of redefining the atomic formulae $S_1(x,y), S_2(x,y), \|\exists\phi\|_Z$ to refer to the appropriate formulae in M(n,m).

$\square$

Thus, in summary, MPA=EMSO$_{MSC}$ $\subsetneq$ MSO$_{MSC}$ and hence it entails that MPA is not closed under complementation, which was the original motivation for the paper [1].

Bibliography

[1] Benedikt Bollig and Martin Leucker. Message-passing automata are expressively equivalent to emso logic. In *Theoretical Computer Science*, pages 146–160. Springer, 2004.

[2] Dora Giammarresi, Antonio Restivo, Sebastian Seibert, and Wolfgang Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Information and Computation*, 125(18):32–45, 1996.

[3] Oliver Matz, Nicole Schweikardt, and Wolfgang Thomas. The monadic quantifier alternation hierarchy over grids and graphs. In *Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 236–244. IEEE, 1997.