RECOGNIZABILITY EQUALS DEFINABILITY FOR PARTIAL K-PATHS

by

Valentine Kabanets

B.Sc. University of Kiev, Kiev, Ukraine, 1993

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
June 1996

National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

ISBN 0-612-16940-5

Canadä

SIMON FRASER UNIVERSITY

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

Recognizability Equals Definability for Partial k-paths..

Author:

(signature)

Valentine Kabanets

(name)

June 14, 1996

(date)

APPROVAL

Name: Valentine Kabanets

Degree: Master of Science

Title of thesis: Recognizability Equals Definability for Partial k-Paths

Examining Committee: Dr. Slawomir Pilarski
Chair

Dr. Arvind Gupta
Assistant Professor
Senior Supervisor

Dr. Pavol Hell
Professor
Supervisor

Dr. Michael Fellows
Professor
Computer Science
University of Victoria
External Examiner

Date Approved: June 12, 1996

Abstract

It is well-known that a language is recognizable iff it is definable in a monadic second-order logic. The same holds for sets of finite ranked trees (or finite unranked trees, in which case one must use a counting monadic second-order logic).

Courcelle initiated research into the problem of definability vs. recognizability for finite graphs. Unlike the case of words and trees, recognizability does not equal definability for arbitrary families of graphs. Courcelle and others have shown that definability implies recognizability for partial k-trees (graphs of bounded tree-width), and conjectured that the converse also holds.

The converse implication was proved for the cases of $k = 0, 1, 2, 3$. It was also established for families of k-connected partial k-trees.

In this thesis, we show that a recognizable family of partial k-paths (graphs of bounded path-width) is definable in a counting monadic second-order logic (CMS), thereby proving the equality of definability and recognizability for families of partial k-paths.

This result is of both theoretical and practical significance. From the theoretical viewpoint, it establishes the equivalence of the algebraic and logical approaches to characterizing yet another recursively defined class of objects, that of partial k-paths. This also adds validity to Courcelle's conjecture on partial k-trees. From the practical viewpoint, since a partial k-path is recognizable in linear time, it establishes that a problem on partial k-paths is solvable in linear time using a finite automaton iff this problem is definable in CMS.

Acknowledgements

I am very grateful to my senior supervisor Arvind Gupta who suggested the topic for my research and was extremely helpful at every stage of my work. Without his limitless patience and constant encouragement this thesis would not have been written.

I want to thank the members of my Examining Committee, staff at the General Office (especially, Kersti Jaager), and all my friends for their unfailing support and faith in me. My warmest thanks go to my parents whose absolute love I could feel even being thousands of miles away.

Contents

List of Figures

Chapter 1

Introduction

In 1960, Büchi [8] showed that a language is regular iff it is definable by some formula in monadic second-order logic (MS). Here, MS is the extension of the first-order logic that allows quantification over sets of objects. A set of objects is definable by an MS-formula if the formula is true exactly on the members of the set. Thus Büchi established that recognizability is equivalent to MS-definability for words. Doner [13] then extended this result to ranked trees (tree representations of algebraic terms). A regular set of ranked trees is recognizable by a tree automaton, the extension of a finite automaton to algebraic terms.

Graphs are algebraic structures since any graph can be constructed from smaller graphs using certain graph operations. They are also logical structures since any graph is completely determined by the set of its vertices and the adjacency relation on this set. Thus the notions of recognizability and definability can be extended to finite graphs. Courcelle [10] proved that every MS-definable set of finite graphs is recognizable, but not conversely. However, he was able to extend the result of Doner to unordered unbounded trees using a counting monadic second-order logic (CMS), an extension of MS that allows modular counting.

The question remained whether there was a sufficiently large class of graphs for which recognizability would imply CMS-definability. In their study of graph minors, Robertson and Seymour [19] introduced the notion of the tree-width of a graph. A graph of tree-width k exhibits certain tree-like structure: Such a graph can be decomposed into subgraphs of size $k+1$ arranged as nodes of a tree (tree-decomposition) so that the nodes containing a given vertex form a subtree.

The class of graphs of tree-width at most k coincides with that of partial k-trees. Among other classes of graphs of bounded tree-width are trees and forests (tree-width ≤ 1), series-parallel graphs and outerplanar graphs (≤ 2), and Halin graphs (≤ 3).

For graphs of tree-width at most k, recognizability is defined using a tree automaton working on the corresponding tree-decompositions: A set $\mathcal{G}$ of partial k-trees G is recognizable if there is a tree automaton that accepts any tree-decomposition of each graph $G \in \mathcal{G}$, and rejects tree-decompositions of graphs not in $\mathcal{G}$. Since the size of each node in such tree-decompositions is bounded, to check if a partial k-tree (given with its tree-decomposition) is recognized takes linear time. Bodlaender [5] gave a linear-time algorithm for constructing tree-decompositions of tree-width at most k for partial k-trees. Thus it is possible to check in linear time whether a partial k-tree is recognized, even if its tree-decomposition is not part of the input.

It follows from the above-mentioned Courcelle's result that every CMS-definable set of partial k-trees is recognizable in *linear* time by a corresponding tree automaton. (In [1], the explicit construction of a tree automaton for a given CMS-formula is presented.) This is particularly interesting since many NP-complete problems on graphs can be described as properties expressible in MS, and therefore these problems become linear-time solvable on the class of partial k-trees. In [4] and [7], a formalism different from MS-definability is proposed so that the properties expressible using this formalism can be solved in linear time on recursively defined families of graphs.

The class of graphs of bounded tree-width plays an important role for another reason. The monadic second-order theory (MS-theory) of a class of graphs $\mathcal{G}$ is the set of all MS-formulas that are true on each element of $\mathcal{G}$. This theory of $\mathcal{G}$ is decidable if it is recursive (i.e., there is an algorithm that decides whether any given MS-formula holds for all the elements of $\mathcal{G}$). Courcelle showed in [10] that the MS-theory of the class of partial k-trees is decidable. Seese [20] proved that if the MS-theory of a class of finite graphs M is decidable, then the graphs in M have uniformly bounded tree-width. Thus, tree-width "characterizes" classes of finite graphs having decidable MS-theories.

Courcelle [11] showed that a recognizable set of partial k-trees is CMS-definable for $k = 1$ and $k = 2$, and conjectured that recognizability implies CMS-definability of partial k-trees for an arbitrary k. Kaller [17] proved the case of $k = 3$ and the case of k-connected partial k-trees.

In this thesis, we establish that every recognizable set of partial k-paths is CMS-definable. A partial k-path (graph of bounded path-width) is a partial k-tree for which the corresponding tree-decomposition is a path-decomposition. Partial k-paths are recognized by finite automata working on the corresponding path-decompositions.

In their solutions, Courcelle and Kaller show how to define in MS some tree-decomposition of a given partial k-tree. Simulating (in CMS) the behaviour of a tree-automaton on this tree-decomposition is then a fairly straightforward task.

To solve the problem for partial k-paths, we do the reconstructing of some path-decomposition of a given partial k-path G and simulating of the corresponding finite automaton A "in parallel." In fact, we do not define in CMS any path-decomposition of G, but only check if some of its path-decompositions is accepted by A.

First, we show how every partial k-path G can be coloured so that some of its path-decompositions can be reconstructed (although not in CMS) from this colouring. The structure that we can define in CMS given such a coloured graph is sufficient for us to verify (in CMS) whether this path-decomposition of G is accepted by the automaton A.

Thus we prove that recognizability implies CMS-definability for properly coloured partial k-paths. To establish this implication for uncoloured partial k-paths, we show then that the required coloured graph can be defined in MS.

A graph H is a minor of a graph G if H becomes a subgraph of G after a series of contractions of edges of G. A family of graphs is minor-closed if every minor of every member of that family also belongs to the family. Robertson and Seymour [18] proved that every minor-closed family of finite graphs can be characterized by a finite set of obstructions, graphs outside the family.

Using the fact that the class of partial k-trees is minor-closed, Courcelle [12] proved that this class is MS-definable. However, to construct the corresponding MS-formula, one needs to know the finite set of obstructions of the corresponding class of graphs. These sets are known only for $k = 1$, $k = 2$ ([21, 2]), and $k = 3$ ([3]).

On the other hand, the obstruction set of the class of partial k-trees can be determined from the MS-formula defining that class. To find such a set, one can use graph grammars (as suggested in [12]) or congruences of finite index (see [14]). This reasoning also applies to the class of partial k-paths.

In this thesis, we describe how to construct the MS-formula defining the class of partial k-paths for every given k. As a consequence, we can now compute the obstruction sets of the classes of partial k-paths for each k.

The remainder of this thesis is organized as follows: In Chapter 2, we give the necessary background material from graph theory, logic, and automata theory. We define partial k-paths, path-decompositions, and definability and recognizability for partial k-paths. We also give a proof of Büchi's result that a recognizable set of words is MS-definable. In Chapter 3, we show that recognizability implies CMS-definability for a certain generalization of the class of connected partial k-paths, the class of $(k, 1)$-paths. This will be a base case of our solution for arbitrary partial k-paths. In Chapter 4, we prove the case of connected partial k-paths first, and then extend our proof to possibly disconnected partial k-paths. In the last chapter, we give concluding remarks and discuss a possible approach to solving the problem on partial k-trees.

Chapter 2

Preliminaries

The problem studied in this thesis arose at the intersection of three areas of theoretical computer science: graph theory, logic, and automata theory.

In this chapter, the necessary material from each of those areas is presented. In the first section, we recall some basic terminology from graph theory and define partial k-paths. In the second section, we describe the concept of definability in counting monadic second-order logic. In the third section, we explain what it means for a graph family to be recognizable. In the last section, we prove that every recognizable language is MS-definable. (We will use the technique from this proof when showing the corresponding implication for partial k-paths.)

2.1 Graphs

2.1.1 Basic Definitions

The majority of definitions in this subsection can be found in any standard reference book on graph theory (see, e.g., [15] or [6]).

Our graphs are finite and simple. They can be undirected or directed (digraphs). For a graph $G = (V_G, E_G)$, V_G and E_G are its vertex and edge sets, respectively. (Whenever this leads to no confusion, we shall drop the subscript.)

If G is undirected, an edge $e \in E$ connecting vertices u and v ($u, v \in V$) is denoted by $e = \{u, v\}$. If G is directed, an edge $e \in E$ from u to v ($u, v \in V$), called an *arc*, is denoted by $e = (u, v)$. In both cases, the vertices u and v are called the *ends* of the edge e. They are also said to be *adjacent* to each other and *incident* to the edge e.

A graph $H = (V_H, E_H)$ is called a *subgraph* of $G = (V_G, E_G)$, denoted by $H \subseteq G$, iff $V_H \subseteq V_G$ and $E_H \subseteq E_G$. If H is a subgraph of G, G is called a *supergraph* of H.

For a non-empty subset V' of V, the subgraph of G with the vertex set V' and the edge set containing those edges of G that have both ends in V' is called the subgraph of G *induced* by V' and is denoted by $G[V']$.

Two undirected graphs G and G' are called *isomorphic*, denoted by $G \cong G'$, iff there is a bijection $\eta : V_G \to V_{G'}$ such that for any $u, v \in V_G$, $\{u, v\} \in E_G$ iff $\{\eta(u), \eta(v)\} \in E_{G'}$. The definition for directed graphs is similar.

A *path of length* s ($s \geq 0$) in an undirected graph G going from u to v ($u, v \in V$) is a sequence of vertices $\langle v_1, \ldots, v_{s+1} \rangle$ of G such that $v_1 = u$, $v_{s+1} = v$, and $\{v_i, v_{i+1}\} \in E$ for all $i \in \{1, \ldots, s\}$. The definition for a directed graph is similar.

A *chain of length* s ($s \geq 0$) in a directed graph G going from u to v ($u, v \in V$) is a sequence of vertices $\langle v_1, \ldots, v_{s+1} \rangle$ of G such that $v_1 = u$, $v_{s+1} = v$, and $(v_i, v_{i+1}) \in E$ or $(v_{i+1}, v_i) \in E$ for all $i \in \{1, \ldots, s\}$ (i.e., it is a path in the corresponding undirected graph).

An undirected (directed) graph G is called *connected* iff, for every two distinct vertices $u, v \in V$, there is a path (chain) going from u to v. The maximal connected induced subgraphs of G are called the *components* of G. Clearly, a connected graph has at most one component.

An undirected graph in which every two distinct vertices are adjacent is called a *complete graph*. The complete graph on n vertices (which is unique up to isomorphism) is denoted by K_n.

A *clique* of an undirected graph G is a subset S of V such that $G[S]$ is a complete graph.

For a graph G and an equivalence relation ρ on its set of vertices V, the *quotient graph* $G/\rho = (V_\rho, E_\rho)$ is defined as follows: $V_\rho = V/\rho$ is the quotient set of V with respect to ρ, and any two distinct equivalence classes are adjacent in the quotient graph iff so are at least a pair of the corresponding vertices in G.

For a graph G and (possibly empty) sequences of vertex subsets $S_v = \langle V_1, \ldots, V_{n_v} \rangle$ and edge subsets $S_e = \langle E_1, \ldots, E_{n_e} \rangle$, the triple $G^c = (G, S_v, S_e)$ is called a *coloured graph*. A vertex $v \in V_i$ ($1 \leq i \leq n_v$) and an edge $e \in E_i$ ($1 \leq i \leq n_e$) are called *coloured with colour* i. The graph G is said to be the *underlying graph* of the coloured graph G^c.

Two coloured graphs are called *isomorphic* iff their underlying graphs are isomorphic and the bijection realizing that isomorphism preserves the colouring.

2.1.2 Partial k-Paths and Path-Decompositions

Definition 2.1.1 The class of *k-terminal k-paths* ($k \geq 0$) is defined inductively as follows:

i. $G = K_k$ is a k-path with the set of k terminals $S = V_G$.

ii. Let G be a k-path with the set of k terminals $S = \{v_1, \ldots, v_k\}$. Any graph G' obtained from G by adding a new vertex $v \notin V_G$ and edges $\{v, v_i\}$ ($i \in \{1, \ldots, k\}$) is a k-path. An arbitrary subset of k vertices $S' \subset S \cup \{v\}$ can be chosen as the set of terminals of G'.

iii. No other graphs are k-paths.

Remark 2.1.2 After a k-terminal k-path is constructed, its k terminals can be "forgotten." This underlying graph will be called a k-path.

Definition 2.1.3 Any subgraph of a k-path is called a *partial k-path.*

Example 2.1.4 Graphs G_0 (see Fig. 2.1), G_1 (see Fig. 2.2), and G_2 (see Fig. 2.3) are 0-path, 1-path, and 2-path, respectively. (Note that the vertices in Fig. 2.1, Fig. 2.2, and Fig. 2.3 are numbered according to the order in which they have been added to form the corresponding k-path.) The subgraph G'_2 of G_2 (see Fig. 2.4) is a partial 2-path.

Figure 2.1: A 0-path G_0.

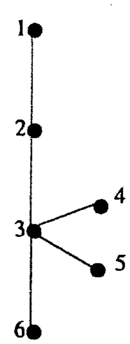

Figure 2.2: A 1-path G_1.

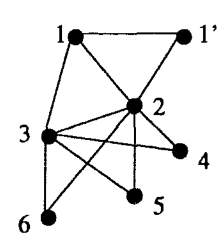

Figure 2.3: A 2-path G_2.

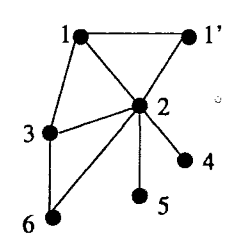

Figure 2.4: A partial 2-path G'_2.

Definition 2.1.5 [Robertson and Seymour [19]] A *path-decomposition* of a graph $G = (V, E)$ is a sequence of vertex-subsets (called *bags*) $B = \langle B_1, \ldots, B_m \rangle$ such that

i. every vertex $v \in V$ belongs to some bag B_i $(1 \leq i \leq m)$,

ii. for every edge $e \in E$, there is a bag B_i $(1 \leq i \leq m)$ that contains both ends of e,

iii. for any $i, l, j \in \{1, \ldots, m\}$ such that $i \leq l \leq j$, $B_i \cap B_j \subseteq B_l$.

Notation 2.1.6 Instead of path-decomposition, we will often use the term decomposition.

Example 2.1.7 Here are possible decompositions of the graphs G_0, G_1, G_2, and G_2' (see Example 2.1.4): $B(G_0) = \langle\{1\},\{2\},\{3\},\{4\},\{5\}\rangle$, $B(G_1) = \langle\{1,2\},\{2,3\},\{3,4\},\{3,5\},\{3,6\}\rangle$, $B(G_2) = \langle\{1,1',2\},\{1,2,3\},\{2,3,4\},\{2,3,5\},\{2,3,6\}\rangle$, and $B(G_2') = B(G_2)$.

Definition 2.1.8 The *path-width of a decomposition* $B = \langle B_1, \ldots, B_m\rangle$ is

$$\max_{1 \leq i \leq m} \{|B_i|\} - 1.$$

A decomposition of path-width at most k will be called a *k-decomposition.*

Definition 2.1.9 The *path-width of a graph G* is the minimum path-width over all decompositions of G.

It was not coincidental that the 0-path G_0, 1-path G_1, and 2-path G_2 turned out to have the path-widths 0, 1, and 2, respectively. The following claim is easy to prove.

Fact 2.1.10 *A graph G is a partial k-path iff it is of path-width at most k.*

Notation 2.1.11 We denote by $\mathcal{B}_r$ $(1 \leq r \leq m)$ the union of the first r bags in the sequence $B = \langle B_1, \ldots, B_m\rangle$, i.e., $\mathcal{B}_r = \cup_{i=1}^{r} B_i$ for any $r \in \{1, \ldots, m\}$.

Notation 2.1.12 For a partial k-path $G = (V, E)$ with a decomposition $B = \langle B_1, \ldots, B_m\rangle$, first$(v)$ is the number of the bag where a vertex $v \in V$ appears for the first time (i.e., first$(v) = \min_{1 \leq l \leq m}\{l | v \in B_l\}$), new$(B_i)$ $(i \in \{1, \ldots, m\})$ is the set of vertices in B_i that appear in the decomposition for the first time (i.e., new$(B_i) = \{u \in B_i | \text{first}(u) = i\}$), and old$(B_i)$ is the set of vertices in B_i that also appear in some earlier bag (i.e., old$(B_i) = B_i \setminus \text{new}(B_i)$).

Definition 2.1.13 For a partial k-path $G = (V, E)$ with a decomposition $B = \langle B_1, \ldots, B_m\rangle$, a vertex $u \in B_r$ $(1 \leq r \leq m)$ is called a *drop vertex* of B_r iff for every $w \in V \setminus \mathcal{B}_r$, $\{u, w\} \notin E$.

Notation 2.1.14 The set of all drop vertices of B_r $(1 \leq r \leq m)$ will be denoted by drop(B_r).

Definition 2.1.15 A vertex $v \in B_r$ $(1 \leq r \leq m)$ that is not a drop vertex of B_r is called a *non-drop vertex* of B_r.

Notation 2.1.16 The set of all non-drop vertices of B_r $(1 \leq r \leq m)$ will be denoted by non-drop(B_r), i.e., non-drop(B_r)=$B_r \setminus$drop(B_r).

Definition 2.1.17 A decomposition $\bar{B} = \langle B_1, B_1^-, \ldots, B_m, B_m^- \rangle$ is called *extended* iff dropping old vertices and adding new vertices occur separately, i.e., B_i^- = non-drop(B_i) for each $i \in \{1, \ldots, m\}$.

Example 2.1.18 Here is an extended 1-decomposition of the graph G_1:

$$\bar{B}(G_1) = \langle \{1,2\}, \{2\}, \{2,3\}, \{3\}, \{3,4\}, \{3\}, \{3,5\}, \{3\}, \{3,6\}, \{\} \rangle.$$

2.1.3 Nice Decompositions

In this subsection, we define a special kind of decomposition. It has many useful properties to be fully exploited later in the thesis.

Definition 2.1.19 A decomposition $B = \langle B_1, \ldots, B_m \rangle$ of $G = (V, E)$ is called *nice* iff all of the following conditions hold:

i. for any vertex $v \in V$, if $v \in$ old(B_i) $(i \in \{2, \ldots, m\})$, then v could not be dropped earlier, i.e., $v \in$ non-drop(B_{i-1}),

ii. new(B_i) $\neq \emptyset$ for every $i \in \{1, \ldots, m\}$,

iii. drop(B_i) $\neq \emptyset$ for every $i \in \{1, \ldots, m\}$,

iv. for any $i \in \{2, \ldots, m\}$, if $|$new(B_i)$| > 1$, then the following two conditions are satisfied:

   (a) for an arbitrary vertex $v \in V \setminus B_{i-1}$, any decomposition $\langle B_1, \ldots, B_{i-1},$ old(B_i) $\cup \{v\}, C_1, \ldots, C_s \rangle$ of G is such that drop(old(B_i) $\cup \{v\}$) $= \emptyset$ (i.e., for each $u \in$ old(B_i) $\cup \{v\}$, there is $w \in V \setminus (B_{i-1} \cup \{v\})$ such that $\{u, w\} \in E$),

(b) for an arbitrary subset $S \subset \text{new}(B_i)$, any decomposition $\langle B_1, \ldots, B_{i-1}, \text{old}(B_i) \cup S, C_1, \ldots, C_s \rangle$ of G is such that $\text{drop}(\text{old}(B_i) \cup S) = \emptyset$ (i.e., for each $u \in \text{old}(B_i) \cup S$, there is $w \in V \setminus (B_{i-1} \cup S)$ such that $\{u, w\} \in E$).

Here condition (i) says that vertices are dropped from a bag as soon as possible, condition (ii) that at least one new vertex is always added to form the next bag, condition (iii) that each bag contains at least one drop vertex, and condition (iv) that if more than one new vertex is added to form the bag B_i, then both (a) there is no single non-added vertex that could be chosen instead of the set $\text{new}(B_i)$ so that the new bag contains a drop vertex and (b) the set $\text{new}(B_i)$ is a minimal one (with respect to set inclusion) such that B_i contains a drop vertex.

Remark 2.1.20 By definition, nice decompositions cannot be extended decompositions.

Definition 2.1.21 A contiguous subsequence $\langle B_i, \ldots, B_{i+l} \rangle$ $(1 \leq i, i + l \leq m)$ of a nice decomposition $\langle B_1, \ldots, B_m \rangle$ is called *monotonic* iff $|\text{new}(B_i)| > 1$ and $|\text{new}(B_r)| = 1$ for each $i < r \leq i + l$.

Remark 2.1.22 A nice decomposition is defined so that it is monotonic as long as possible, then there is a "jump" (more than one new vertex is added to a bag) which starts a new monotonic piece, and so on.

Theorem 2.1.23 *Every k-decomposition can be converted into a nice k-decomposition.*

Proof. A given k-decomposition $B = \langle B_1, \ldots, B_m \rangle$ induces a linear order on vertices of the original graph through the order in which they are added to that decomposition. That is, we take the sequence $\langle \text{new}(B_1), \ldots, \text{new}(B_m) \rangle$ of sets of vertices and, for each $\text{new}(B_i)$ $(1 \leq i \leq m)$, order the vertices in $\text{new}(B_i)$ arbitrarily. This gives us a sequence S of vertices of the graph.

Step 1. We choose the first bag of the given decomposition as the first bag of the nice decomposition to be constructed.

Step i (i > 1). A sequence of bags $\langle B'_1, \ldots, B'_{i-1} \rangle$ has been constructed. We define the set B''_i as B'_{i-1} without its drop vertices. We go through our sequence of vertices looking for a non-added vertex v such that $B''_i \cup \{v\}$ contains at least one drop vertex (with respect to the already constructed sequence of bags). If we can find such a vertex, the bag B'_i is defined as $B''_i \cup \{v\}$.

If no such vertex exists, we choose the sequence S_i of the first p non-added vertices (where $p = k + 1 - |B'_{i-1}|$), take its shortest prefix S'_i such that $B''_i \cup S'_i$ contains at least one drop vertex, and form the minimal-size subset S''_i of S'_i such that $B''_i \cup S''_i$ contains at least one drop vertex. The bag B'_i is then defined as $B''_i \cup S''_i$.

It is not difficult to see that the new sequence $\langle B'_1, \ldots, B'_{i-1}, B'_i \rangle$ can be completed to some k-decomposition in both cases.

Indeed, after adding v or S''_i, we just drop all the drop vertices of this new bag (there is at least one such vertex) and then continue the decomposition by adding the non-added vertices in the order they appear in our sequence S. □

Definition 2.1.24 We call a nice k-decomposition $B = \langle B_1, \ldots, B_m \rangle$ of a partial k-path G a (k,p)-*decomposition* (for some $p \in \{1, \ldots, k\}$) iff $|\text{new}(B_i)| \leq p$ for all $i \in \{2, \ldots, m\}$. A partial k-path allowing a (k,p)-decomposition will be called a (k,p)-*path*.

Remark 2.1.25 By the above definition, every $(k,1)$-decomposition is monotonic.

Example 2.1.26 The decompositions $B(G_0)$, $B(G_1)$, and $B(G_2)$ (see Example 2.1.7) in the previous subsection) are $(k,1)$-decompositions.

2.1.4 Directed Partial k-Paths

In this subsection, we define a certain labeled digraph induced by a nice decomposition B of a k-path G. This labeled digraph will allow us to define a partial order on the set V_G, so that if u is less than v (for any $u, v \in V_G$), then u appears in B no later than v.

A nice decomposition $B = \langle B_1, \ldots, B_m \rangle$ of a partial k-path $G = (V, E)$ induces the following directed graph $G^d_B = (V, E^d)$:

Given a bag $B_r = \text{old}(B_r) \cup \text{new}(B_r)$ $(1 < r \leq m)$, where $\text{old}(B_r) = \{u_1, \ldots, u_s\}$ and $\text{new}(B_r) = \{v_1, \ldots, v_t\}$, if $\{v_i, u_j\} \in E$, then $(v_i, u_j) \in E^d$ (i.e., we direct the edges from new to old vertices).

If $t > 1$, we order the vertices in $\text{new}(B_r)$ so that the drop vertices of B_r (if any) come before the non-drop vertices (the order within each of the two sets of vertices $\text{new}(B_r) \cap \text{drop}(B_r)$ and $\text{new}(B_r) \cap \text{non-drop}(B_r)$ is arbitrary). Then any existing edges are directed from "last" to "first" vertices, i.e., for $v, u \in \text{new}(B_r)$ such that $\{v, u\} \in E$, $(v, u) \in E^d$ iff v follows u in the above-defined sequence of new vertices.

Notation 2.1.27 To simplify the notation, we will often omit the superscript in E^d and the subscript in G^d_B.

Remark 2.1.28 Using the definition of a nice decomposition, we can prove that if an old vertex u_j ($1 \leq j \leq s$) is a drop vertex of B_r ($1 < r \leq m$), then for each new vertex v_i, $i \in \{1, \ldots, t\}$, there is an arc $(v_i, u_j) \in E$ in G^d. This holds simultaneously for all the vertices in $\text{old}(B_r) \cap \text{drop}(B_r)$.

Similarly, if some new vertex v_i ($1 \leq i \leq t$) is a drop vertex of B_r and is the first vertex in $\text{new}(B_r)$ (according to the linear order defined above), then for each new vertex v_j, $j \in \{1, \ldots, t\} \setminus \{i\}$, there is an arc $(v_j, v_i) \in E$ in G^d.

Now we label our digraph G^d as follows:

For a vertex $u \in \text{old}(B_r) \cap \text{drop}(B_r)$ ($1 < r \leq m$), we choose arbitrarily some vertex $v \in \text{new}(B_r)$ and colour the arc $v \to u$ with some new colour. This coloured arc will be denoted as a double arrow $v \Rightarrow u$. If there are other drop vertices in $\text{old}(B_r)$, the arc from v to each of them also becomes a double arrow.

Notation 2.1.29 The subset of double arrows in E will be denoted by $E_{\Rightarrow}$.

If $\text{new}(B_r)$ is a singleton set containing some vertex v which is the only drop vertex of B_r (i.e., $\text{new}(B_r) = \text{drop}(B_r)$), we colour v with some new colour (the same colour for all such vertices). Each such coloured vertex will be denoted by having a loop arrow.

We also colour (with some new colour) the vertices of V that form the first bag B_1 of the nice decomposition B.

Example 2.1.30 For the 2-path G_2 and the partial 2-path G'_2 defined earlier, the nice decomposition $B(G_2) = B(G'_2)$ induces the labeled digraphs G^d_2 and G'^d_2 shown in Figs. 2.5 and 2.6, respectively. (Note that double arrows are shown as *thick* single arrows and that the labeling of the vertices in the first bag is not shown.)

Remark 2.1.31 In general, the same labeled digraph G^d can be induced by different nice decompositions of G. (For example, vertices 4 and 5 can be interchanged in the decomposition $B(G'_2)$, but the new decomposition will induce the same labeled digraph G'^d_2.)

Notation 2.1.32 Below, whenever we speak of a digraph G^d, we mean the correspondingly *labeled* digraph G^d.

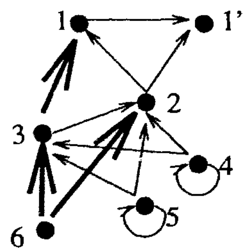

Figure 2.5: The labeled digraph G_2^d.

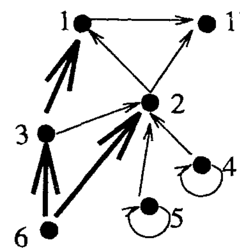

Figure 2.6: The labeled digraph $G_2'^d$.

Definition 2.1.33 Given the digraph G^d induced by a nice decomposition B of a partial k-path $G = (V, E)$, we define the following binary relation of *strong precedence*, denoted by $\overset{s}{\prec}$, on the set V:

For any $u, v \in V$, $u \overset{s}{\prec} v$ iff either $(v, u) \in E$ or there is some $w \in V$ such that $(u, w) \in E$ and $(v, w) \in E_{\Rightarrow}$ (see Fig. 2.7).

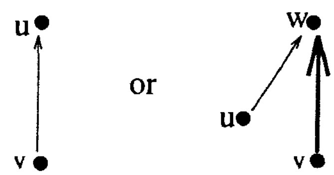

Figure 2.7: The definition of $u \overset{s}{\prec} v$.

The reflexive and transitive closure of $\overset{s}{\prec}$, denoted by $\preceq$, is called *precedence*.

Remark 2.1.34 Semantically, $u \prec v$ means that $\text{first}(u) \leq \text{first}(v)$.

Notation 2.1.35 For a partial k-path $G = (V, E)$ given with its nice decomposition $B = \langle B_1, \ldots, B_m \rangle$, we denote by $\bar{V}$ the set $V \setminus B_1$.

Example 2.1.36 For the 2-path G_2, the relation $\preceq$ induced by G_2^d is, actually, a linear order on $\bar{V}$ yielding the sequence $\langle 3, 4, 5, 6 \rangle$.

This is not true for the partial 2-path $G_2'^d$, however. Vertices 3, 4, and 5 are pairwise incomparable with respect to the precedence $\preceq$ induced by $G_2'^d$.

2.2 Logic and Definability

2.2.1 Basic Definitions

Let A be an arbitrary set and let $n \geq 1$ be any number. A subset of A^n is called an *n-ary relation on A*. A map from A^n to $\{\textbf{True}, \textbf{False}\}$ is called an *n-ary predicate on A*.

Each n-ary relation $R \subseteq A^n$ can be uniquely associated with a predicate r so that for any n-tuple $(a_1, \ldots, a_n)$ of elements of A, $(a_1, \ldots, a_n) \in R$ iff $r(a_1, \ldots, a_n) = \textbf{True}$.

Conversely, any n-ary predicate r can be uniquely associated with the set of n-tuples on which r assumes the value **True**. (We call such n-tuples the *truth-values* of r.)

2.2.2 Logics

Let $\Pi = \{p_i\}_{i \in I}$ (I some countable set) be a family of $n(i)$-ary relation symbols p_i ($i \in I$).

Definition 2.2.1 A *relational* Π*-structure* is a pair $\mathcal{P} = (D, \pi)$, where

i. $D \neq \emptyset$ is a set called the *domain* or *universe* of $\mathcal{P}$,

ii. π is a map defined on Π such that $\pi(p_i) = \mathbf{p}_i$ is an $n(i)$-ary predicate on D.

Let $\mathcal{K}$ be a class of relational Π-structures.

Definition 2.2.2 The *first-order language corresponding to* $\mathcal{K}$ (denoted by $L_1(\mathcal{K})$) has the usual logical connectives: $\neg$ ("not"), $\wedge$ ("and"), $\vee$ ("or"), $\Rightarrow$ ("if-then"), and $\Leftrightarrow$ ("if and only if"), universal ($\forall$) and existential ($\exists$) quantifiers, equality symbol $=$, a sequence $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, ..., of individual variables, and an $n(i)$-ary predicate $\mathbf{p}_i$ for each $i \in I$.

Definition 2.2.3 The $L_1(\mathcal{K})$-formulas are defined inductively as follows:

i. Atomic formulas

   (a) If $\mathbf{u}$ and $\mathbf{v}$ are individual variables, then $\mathbf{u}=\mathbf{v}$ is an $L_1(\mathcal{K})$-formula.

   (b) If $\mathbf{u}_1, \ldots, \mathbf{u}_{n(i)}$ are individual variables and $\mathbf{p}_i$ is an $n(i)$-ary predicate symbol ($n \geq 0$), then $\mathbf{p}_i(\mathbf{u}_1, \ldots, \mathbf{u}_{n(i)})$ is an $L_1(\mathcal{K})$-formula.

ii. Compound formulas

   (a) If ϕ is an $L_1(\mathcal{K})$-formula, so is $\neg\phi$.

(b) If ϕ and ψ are $L_1(\mathcal{K})$-formulas, so are $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \Rightarrow \psi)$, and $(\phi \Leftrightarrow \psi)$.

(c) If ϕ is an $L_1(\mathcal{K})$-formula and $\mathbf{v}$ is an individual variable, then $\forall \mathbf{v}\phi$ and $\exists \mathbf{v}\phi$ are $L_1(\mathcal{K})$-formulas.

iii. No other formulas are $L_1(\mathcal{K})$-formulas.

Monadic second-order logic (MS) is the extension of the first-order logic (F) allowing quantification over monadic (unary) predicates. Since unary predicates can be identified with the sets of their truth-values, one can extend the corresponding first-order language with *set variables* and the membership symbol $\in$ (with the usual interpretation), and allow quantification over set variables.

Definition 2.2.4 The *monadic second-order language corresponding to* $\mathcal{K}$ (denoted by $L_{m2}(\mathcal{K})$) is the extension of $L_1(\mathcal{K})$ by adding a sequence of set variables $\mathbf{U}, \mathbf{V}, \mathbf{W}, \ldots$, and the membership symbol $\in$.

Definition 2.2.5 The class of $L_{m2}(\mathcal{K})$-formulas is the extension of the class of $L_1(\mathcal{K})$-formulas by allowing the atomic formulas $\mathbf{v} \in \mathbf{V}$ as well as quantification over set variables.

Counting monadic second-order logic (CMS) was defined by Courcelle [10] as the extension of MS by the unary predicate symbols $\mathbf{mod}_{p,q}$ ($p < q$ are non-negative integers), with the intended meaning: $\mathbf{mod}_{p,q}(\mathbf{V}) = \mathbf{True}$ iff $|S| = p \bmod q$, where S is the set denoted by the set variable $\mathbf{V}$. The corresponding language is denoted by $L_{cm2}(\mathcal{K})$.

If Φ is a formula (in some language appropriate for $\mathcal{K}$) with free variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$, we indicate this by writing $\Phi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$. (Recall that free variables in a formula are those that are not bound by any quantifier.)

For $\Phi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, we denote by $\Phi[d_1, \ldots, d_n]$ the result of substituting the elements $d_1, \ldots, d_n$ of the domain D for the variables $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Let L be some language appropriate for $\mathcal{K}$. For an L-formula Φ and a Π-structure $K \in \mathcal{K}$, we write $K \models \Phi$ to denote that Φ is satisfied by K (i.e., K is a model for Φ).

2.2.3 Definability and Colourability

Definition 2.2.6 Let P be some property over a class $\mathcal{K}$ of relational Π-structures. The property P is called ***definable in F (MS, or CMS)*** over $\mathcal{K}$ iff there is an $L_1(\mathcal{K})$-formula ($L_{m2}(\mathcal{K})$-formula, or $L_{cm2}(\mathcal{K})$-formula) Φ such that for each $K \in \mathcal{K}$, K satisfies the property P iff $K \models \Phi$.

Notation 2.2.7 We call the properties definable in F, MS, and CMS, F-definable, MS-definable, and CMS-definable, respectively.

Definition 2.2.8 Let R be some n-ary relation over the domain D of a Π-structure K. Relation R is called *F-definable (MS-definable, or CMS-definable) over* K if there is an F-formula (MS-formula, or CMS-formula) $\phi(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ such that for any $d_1, \ldots, d_n \in D$, the tuple $(d_1, \ldots, d_n) \in D$ iff $\phi[d_1, \ldots, d_n]$.

Fact 2.2.9 (Courcelle [10]) *Let ρ be a binary relation over the domain D of a Π-structure K. If ρ is MS-definable over K, then so is its reflexive and transitive closure ρ^*.*

An undirected (directed) graph $G = (V, E)$ can be considered as a relational $\{p_v, p_e, p_{\text{inc}}\}$-structure with the domain $D = V \cup E$, where $\mathbf{p}_v$ and $\mathbf{p}_e$ are unary predicates such that for any $d \in D$, $\mathbf{p}_v(d) = \mathbf{True}$ iff $d \in V$, $\mathbf{p}_e(d) = \mathbf{True}$ iff $d \in E$, and $\mathbf{p}_{\text{inc}} = \mathbf{Inc}$ is the ternary incidence predicate, i.e., for any $e \in E$ and $u, v \in V$, $\mathbf{Inc}(e, u, v) = \mathbf{True}$ iff $e = \{u, v\}$ $(e = (u, v))$.

Thus we can have F-definable (MS-definable, or CMS-definable) properties over an arbitrary class $\mathcal{G}$ of graphs.

Example 2.2.10 Connectedness of a graph is an MS-definable property. Here are the corresponding MS-formulas for a graph $G = (V, E)$:

$\text{Connected} \equiv \forall \mathbf{V}_1 \, \forall \mathbf{V}_2 \; (\mathbf{V}_1 \neq \emptyset \; \wedge \; \mathbf{V}_2 \neq \emptyset \; \wedge \; \mathbf{V}_1 \cup \mathbf{V}_2 = V) \Rightarrow \text{Adj}(\mathbf{V}_1, \mathbf{V}_2)$,

$\text{Adj}(\mathbf{V}_1, \mathbf{V}_2) \equiv \exists \mathbf{v}_1 \, \exists \mathbf{v}_2 \;\; \mathbf{v}_1 \in \mathbf{V}_1 \; \wedge \; \mathbf{v}_2 \in \mathbf{V}_2 \; \wedge \; \text{adj}(\mathbf{v}_1, \mathbf{v}_2)$,

$\text{adj}(\mathbf{v}_1, \mathbf{v}_2) \equiv \exists \mathbf{e} \;\; \mathbf{Inc}(\mathbf{e}, \mathbf{v}_1, \mathbf{v}_2)$,

where

$(\mathbf{V}_i \neq \emptyset) \equiv \exists \mathbf{v} \;\; \mathbf{p}_v(\mathbf{v}) \; \wedge \; \mathbf{v} \in \mathbf{V}_i \;\; (i = 1, 2)$

and

$(\mathbf{V}_1 \cup \mathbf{V}_2 = V) \equiv \forall \mathbf{v} \;\; \mathbf{p}_v(\mathbf{v}) \Rightarrow (\mathbf{v} \in \mathbf{V}_1 \; \vee \; \mathbf{v} \in \mathbf{V}_2)$.

Example 2.2.11 We can also define in MS if a given set of edges C of a graph G forms a simple path linking two given vertices u and v.

Using the previous example, we can define the MS-formula $\theta(\mathbf{E}, \mathbf{u}, \mathbf{v})$ such that $\theta[C, u, v]$ is true iff the graph $G_C = (V_C, C)$ (where V_C is the set of ends of edges in C) is connected and $u, v \in V_C$. Then the required formula is defined as

$$\theta[C, u, v] \; \wedge \; \forall \mathbf{E} \, (\mathbf{E} \subseteq C \wedge \theta[\mathbf{E}, u, v] \Rightarrow \mathbf{E} = C).$$

(It should be obvious that the equality and inclusion of sets are MS-definable.)

Definition 2.2.12 A property P over a class $\mathcal{G}$ of graphs G is called *F-definable (MS-definable, or CMS-definable)* over a class $\mathcal{G}'$ of the corresponding coloured graphs

$$G^c = (G, \langle V_1, \ldots, V_{n_v} \rangle, \langle E_1, \ldots, E_{n_e} \rangle)$$

iff there is an F-formula (MS-formula, or CMS-formula) $\Phi(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e})$ such that for every graph $G \in \mathcal{G}$, G satisfies P iff Φ is true on the corresponding coloured graph G^c, i.e.,

$$G \models \Phi[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}].$$

Colouring a graph G "properly" imposes some additional structure on G, which is then used in defining a required formula Φ for a property P. A colouring that provides the desired structure on G will be called *admissible with respect to P*.

Definition 2.2.13 A property P of a graph G is called *F-colourable (MS-colourable, or CMS-colourable)* over a class $\mathcal{G}$ of graphs G iff it is possible to colour the graphs G (using the same constant number of colours for each) so that the property P is F-definable (MS-definable, or CMS-definable) over the class of thus coloured graphs.

Lemma 2.2.14 *Let P be an F-colourable (MS-colourable, or CMS-colourable) property over a class of coloured graphs $G^c = (G, \langle V_1, \ldots, V_{n_v} \rangle, \langle E_1, \ldots, E_{n_e} \rangle)$. If there is an F-formula (MS-formula, or CMS-formula) that checks the admissibility of a colouring with respect to P, then P is F-definable (MS-definable, or CMS-definable) over the class of underlying graphs G.*

Proof. Indeed, let $\Phi(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e})$ be the formula corresponding to P, and let $\Psi(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e})$ be the formula that checks the admissibility of a colouring with respect to P. The required formula for P over the class of underlying graphs is the following:

$$\exists \mathbf{X}_1 \ldots \exists \mathbf{X}_{n_v} \exists \mathbf{Y}_1 \ldots \exists \mathbf{Y}_{n_e} \Phi(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}) \wedge \Psi(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}).$$

□

The following definitions are similar to those given by Courcelle [11] for describing ***definable graph transductions***.

Let G be a graph (i.e., a $\{p_v, p_e, p_{\text{inc}}\}$-structure (D, π)), let

$$G^c = (G, \langle V_1, \ldots, V_{n_v} \rangle, \langle E_1, \ldots, E_{n_e} \rangle)$$

be a coloured graph, and let

$$\gamma_v(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}),$$

$$\gamma_e(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}),$$

and

$$\gamma_{\text{inc}}(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$$

be CMS-formulas defined over G.

Definition 2.2.15 A tuple $(G^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$ is called *admissible* iff

i. for any $d \in D$, at most one of the formulas

$$\gamma_v[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$$

and

$$\gamma_e[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$$

holds, and

ii. for any $d_1, d_2, d_3 \in D$, if

$$\gamma_{\text{inc}}[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d_1, d_2, d_3]$$

is true, then so are

$$\gamma_e[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d_1],$$

$$\gamma_v[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d_2],$$

and

$$\gamma_v[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d_3].$$

Definition 2.2.16 The *graph defined by an admissible tuple* $(G^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$, denoted by $\Delta(G^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$, is the $\{p_v, p_e, p_{\text{inc}}\}$-structure $(\hat{D}, \hat{\pi})$, where $\hat{D}$ is the set of those d from D for which either $\gamma_v[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$ or $\gamma_e[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$ holds, and for any $d \in \hat{D}$, $\hat{\pi}(p_v)$ is true on d iff so is $\gamma_v[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$, $\hat{\pi}(p_e)$ is true on d iff so is $\gamma_e[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d]$, and for any $d_1, d_2, d_3 \in \hat{D}$, $\hat{\pi}(p_{\text{inc}})$ is true on d_1, d_2, d_3 iff so is $\gamma_{\text{inc}}[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}, d_1, d_2, d_3]$.

Let G and G' be some graphs (i.e., relational structures (D, π) and (D', π'), respectively).

Definition 2.2.17 A graph G' is said to be *CMS-colourable in terms of* G iff there is an admissible tuple $(G^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$ such that the graph $G' = (D', \pi')$ is isomorphic to the graph $\Delta(G'^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$.

Definition 2.2.18 A graph G' is said to be *CMS-definable in terms of* G iff there exist CMS-formulas

$$\gamma(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}),$$

$$\gamma_v(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}),$$

$$\gamma_e(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}),$$

and

$$\gamma_{\text{inc}}(\mathbf{X}_1, \ldots, \mathbf{X}_{n_v}, \mathbf{Y}_1, \ldots, \mathbf{Y}_{n_e}, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$$

defined over G such that for any sequence of vertex subsets $V_1, \ldots, V_{n_v}$ and any sequence of edge subsets $E_1, \ldots, E_{n_e}$ of G, the validity of the formula $\gamma[V_1, \ldots, V_{n_v}, E_1, \ldots, E_{n_e}]$ implies that the tuple $(G^c = (G, \langle V_1, \ldots, V_{n_v} \rangle, \langle E_1, \ldots, E_{n_e} \rangle), \gamma_v, \gamma_e, \gamma_{\text{inc}})$ is admissible and $G' \cong \Delta(G^c, \gamma_v, \gamma_e, \gamma_{\text{inc}})$.

Theorem 2.2.19 (Courcelle [11]) *Let G be a graph and let ρ be an equivalence relation on its set of vertices V. If ρ is CMS-definable, then the quotient graph G/ρ is CMS-definable in terms of G.*

Proof. First we show that G/ρ is CMS-colourable in terms of G.

Consider a coloured graph $G^c = (G, \langle V_1 \rangle, \langle E_1 \rangle)$, where

i. $V_1 \subseteq V$ is such that it contains exactly one vertex from each ρ-equivalence class in V/ρ, and

ii. $E_1 \subseteq E$ is such that it contains exactly one edge from each ρ-equivalence class in E/ρ. (Two edges $e_1 = \{u_1, v_1\}$ and $e_2 = \{u_2, v_2\}$ are ρ-equivalent iff u_i and v_i are not ρ-equivalent ($i = 1, 2$), and $[u_1]_\rho = [u_2]_\rho$ and $[v_1]_\rho = [v_2]_\rho$.)

Then

$$\gamma_v(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{z}) \equiv \mathbf{z} \in \mathbf{X}_1,$$

$$\gamma_e(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{z}) \equiv \mathbf{z} \in \mathbf{Y}_1,$$

and

$$\gamma_{\mathrm{inc}}(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3) \equiv \exists \mathbf{u}\, \exists \mathbf{v}\ \ \rho(\mathbf{u}, \mathbf{z}_2)\ \wedge\ \rho(\mathbf{v}, \mathbf{z}_3)\ \wedge\ \mathbf{p}_{\mathrm{inc}}(\mathbf{z}_1, \mathbf{u}, \mathbf{v}).$$

It is clear that conditions (i) and (ii) can be checked in MS, which ends the proof. □

We have the following lemmas for our earlier definitions.

Lemma 2.2.20 *For any partial k-path G with a nice decomposition B, the digraph G^d induced by B is MS-colourable in terms of G.*

Proof. The set of vertices of G^d is the same as that of G.

By definition, each vertex of G^d has at most k outgoing arrows. We colour the edges and vertices of G with some new $k+1$ colours as follows:

The vertices in B_1 are coloured with $1, \ldots, k+1$ so that no two vertices get the same colour. If for some vertices u and v in B_1 such that u is coloured with i and v is coloured with j $(1 \leq i, j \leq k+1)$, $(u, v) \in E$, then the edge $\{u, v\}$ of G is coloured with j.

For any other vertex w of G^d with outgoing arrows to some vertices $u_1, \ldots, u_d$ $(d \leq k)$ coloured with $j_1, \ldots, j_d$, respectively, we colour w with an arbitrary colour from the set $\{1, \ldots, k+1\} \setminus \{j_1, \ldots, j_d\}$, and we colour every edge $\{w, u_i\}$ of G with j_i $(1 \leq i \leq d)$.

Then an edge $e = \{u, v\} \in E$ is an arc (u, v) in G^d iff e and v are coloured with the same colour in the coloured graph defined above. This is easily expressible in MS.

To encode the sets of double arrows, vertices with loop arrows, and vertices in the first bag of B, one just should use three new colours (one for each set). □

Lemma 2.2.21 *The precedence relation $\preceq$ induced by a digraph G^d of G is MS-colourable over G.*

Proof. The graph G^d is MS-colourable in terms of G, and $\preceq$ is readily definable in MS over G^d. □

2.3 Automata and Recognizability

2.3.1 Basic Definitions

Here we recall some standard terminology (see, e.g., [16]).

A *deterministic finite automaton* is a 5-tuple $A = (\Sigma, Q, \delta, q_0, F)$, where Σ is a finite set of input symbols, Q is a finite state of states, δ is a transition function (i.e., a map from $Q \times \Sigma$ to Q), $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is a set of final states.

A word w over an alphabet Σ (i.e., $w \in \Sigma^*$) is *accepted* by an automaton A iff $\delta^*(q_0, w) \in F$, where $\delta^* : Q \times \Sigma^* \rightarrow Q$ is the *extended transition function* of A defined as follows: $\delta^*(q, \sigma) = \delta(q, \sigma)$ for any $\sigma \in \Sigma$, $\delta^*(q, \sigma s) = \delta^*(\delta(q, \sigma), s)$ for any $\sigma \in \Sigma$ and any word s over Σ.

The set of words (language) accepted by an automaton A is denoted by $L(A)$, i.e., $L(A) = \{w \in \Sigma^* | \delta^*(q_0, w) \in F\}$.

2.3.2 Recognizability of Partial k-Paths

Let $G = (V, E)$ be a partial k-path with an extended k-decomposition $B = \langle B_1, \ldots, B_m \rangle$. Let $\beta : V \rightarrow \{1, \ldots, k+1\}$ be a labeling function on the set of vertices of G such that any two distinct vertices in the same bag or in two consecutive bags have different labels. We shall call such labeling functions *admissible* by B. (It is not difficult to see that $k+1$ colours always suffice in the case of *extended* decompositions.)

Notation 2.3.1 For the labeling function β and any set of vertices $W \subseteq V$, $\beta(W) = \cup_{w \in W} \beta(w)$.

For B and β given above, we define a string $\sigma_\beta(B)$ of coloured undirected graphs (on at most $k+1$ vertices) as follows: $\sigma_\beta(B) = \langle \sigma_\beta(B_1), \ldots, \sigma_\beta(B_m) \rangle$, where for a bag B_i $(1 \leq i \leq m)$, $\sigma_\beta(B_i) = (V_\beta(B_i), E_\beta(B_i))$ such that

i. $V_\beta(B_i) = \beta(B_i)$,

ii. for every $u, u' \in B_i$, $\{\beta(u), \beta(u')\} \in E_\beta(B_i)$ iff $\{u, u'\} \in E$.

Let Σ_g be the set of all coloured (with colours $1, \ldots, k+1$) undirected graphs on at most $k+1$ vertices. (Clearly, the cardinality of Σ_g is bounded by a function of k.)

Definition 2.3.2 A family $\mathcal{G}$ of partial k-paths G is called *recognizable* iff there is an automaton A with the set of input symbols Σ_g such that for any partial k-path G with some extended k-decomposition B and some labeling function β (admissible by B), $G \in \mathcal{G}$ iff $\sigma_\beta(B) \in L(A)$.

Remark 2.3.3 In the above definition, the choice of an extended k-decomposition B and a labeling function β is not important, i.e., for any other extended k-decomposition B' of G and a labeling function β' admissible by B', we must have $\sigma_\beta(B) \in L(A)$ iff $\sigma_{\beta'}(B') \in L(A)$.

2.4 Definability vs. Recognizability for Words

Here we consider the solution for the case of words (since decompositions of partial k-paths are also words over a special alphabet of coloured graphs Σ_g).

Let $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ be a finite alphabet. A word $\omega = \sigma_{i_1} \ldots \sigma_{i_n}$ over the alphabet Σ can be considered as a labeled digraph (path) $G_\omega = (V_\omega, E_\omega)$, where $V_\omega = \{1, \ldots, n\}$, for every $j \in \{1, \ldots, n-1\}$, $(j, j+1) \in E_\omega$, and each vertex j is labeled with σ_{i_j}.

In our terminology, ω is a coloured graph $G^c = (G, \langle V_1, \ldots, V_s \rangle, \langle\rangle)$ such that G is a directed path on n vertices, $V_1, \ldots, V_n$ form a partitioning of V (i.e., they are pairwise disjoint and their union is V), and $v \in V_i$ ($v \in V$ and $1 \leq i \leq s$) means that a vertex v is labeled with σ_i, which will be denoted by $\alpha(v) = \sigma_i$. For G^c, $\alpha(G^c) = \omega$ denotes the word over Σ "encoded" by the graph G^c.

Let $\mathcal{G}^c$ be some family of coloured graphs G^c corresponding to some words over Σ. That family is *recognizable* by a DFA $A = (\Sigma, Q, \delta, q_0, F)$ iff $L(A) = \{\alpha(G^c) | G^c \in \mathcal{G}^c\}$.

Theorem 2.4.1 (Büchi [8]) *A language over Σ is recognizable iff it is MS-definable.*

We only give a proof of the first part (i.e., that of recognizability implying MS-definability), since that is what we want to prove for the case of partial k-paths.

Proof. ($\Rightarrow$) Let $A = (\Sigma, Q, \delta, q_0, F)$ be a DFA with $\Sigma = \{\sigma_1, \ldots, \sigma_s\}$ and $Q = \{q_0, \ldots, q_t\}$. First we assume that $L(A)$ does not contain the empty word ϵ.

A coloured graph $G^c = (G, \langle V_1, \ldots, V_s \rangle, \langle\rangle)$ (where G is a path $v_0 \rightarrow \cdots \rightarrow v_n$) is recognized by A iff there is a labeling $q : V \rightarrow Q$ such that

i. $\delta(q_0, \alpha(v_0)) = q(v_0)$,

ii. for every $i \in \{0, \ldots, n-1\}$, $\delta(q(v_i), \alpha(v_i)) = q(v_{i+1})$, and

iii. $q(v_n) \in F$.

This can be written in MS as follows:
$\exists \mathbf{X}_0 \ldots \exists \mathbf{X}_t$ "$\mathbf{X}_0, \ldots, \mathbf{X}_t$ form a partitioning of V" $\wedge$
$(\forall \mathbf{v}\ \mathrm{fst}(\mathbf{v}) \Rightarrow \bigvee_{(k,j):\, \delta(q_0,\sigma_k)=q_j} (\mathbf{v} \in V_k\ \wedge\ \mathbf{v} \in \mathbf{X}_j)) \wedge$
$(\forall \mathbf{u}\, \forall \mathbf{v}\, \forall \mathbf{e}\ \mathbf{Inc}(\mathbf{e}, \mathbf{u}, \mathbf{v}) \Rightarrow \bigvee_{(k,i,j):\, \delta(q_i,\sigma_k)=q_j} (\mathbf{u} \in \mathbf{X}_i\ \wedge\ \mathbf{v} \in \mathbf{X}_j\ \wedge\ \mathbf{v} \in V_k)) \wedge$
$(\forall \mathbf{v}\ \mathrm{lst}(\mathbf{v}) \Rightarrow \bigvee_{j:\, q_j \in F} \mathbf{v} \in \mathbf{X}_j)$,
where $\mathbf{v} \in \mathbf{X}_j$ $(1 \leq j \leq t)$ has the intended meaning of $q(\mathbf{v}) = q_j$, $\mathrm{fst}(\mathbf{v})$ is true for v_0 only, and $\mathrm{lst}(\mathbf{v})$ is true for v_n only. Clearly, both $\mathrm{fst}(\mathbf{v})$ and $\mathrm{lst}(\mathbf{v})$ are easy to define in MS.

The case of $L(A)$ containing ϵ can also be described in MS. The empty word is represented by the empty graph, which is accepted by A iff $q_0 \in F$. □

Chapter 3

The Case of Connected $(k, 1)$-Paths

All the graphs considered in this chapter are connected.

We will define certain coloured $(k, 1)$-paths G^c and show that recognizability is CMS-definable for families of these coloured $(k, 1)$-paths. To convert the corresponding CMS-formula into a formula for the underlying $(k, 1)$-paths G, we give the MS-definable admissibility conditions that check if a colouring of G induces the required structure on G (see Lemma 2.2.14).

To show that a recognizable family of (coloured) $(k, 1)$-paths G is CMS-definable, it suffices to define in CMS some extended decomposition of G. (Then one can proceed by analogy with the case of words (see Theorem 2.4.1).)

A decomposition of G can be defined if some linear order on V is known. Let $\leq$ be an arbitrary linear order on V, and let $\langle v_1, \ldots, v_n \rangle$ be the sequence of vertices in V ordered according to $\leq$. We define the following sequence of sets $B' = \langle B_1, \ldots, B_n \rangle$, where

$$B_i = \{v_i\} \cup \{v_j | j < i \text{ and there is } j' \geq i \text{ such that } \{v_j, v_{j'}\} \in E\}.$$

Clearly, B' is a decomposition of G. We will denote the decomposition B' by $B_{\leq}$.

For a partial k-path G, a linear order $\leq$ on V is called *k-generative* if the path-width of the decomposition $B_{\leq}$ is at most k.

Given a $(k, 1)$-decomposition B of G, one can define the following partial order on V: For any $u, v \in V$, u is less than v iff $\text{first}(u) < \text{first}(v)$. Ordering the vertices in B_1 arbitrarily gives us a linear order on V that is obviously a k-generative linear order on G.

We will also need the following definition: For a partial k-path G, a partial order on V is called *k-generative* if every completion to a linear order on V is k-generative.

We will show that k-generative linear orders are MS-definable over suitably coloured graphs G^c. (More precisely, we define k-generative linear orders on certain quotient graphs of G^c. Then we show how to get the CMS-formula for recognizability of G^c by using these linear orders on the quotient graphs of G^c.)

3.1 Extracting a Decomposition

Here we describe how, using a suitable colouring of a $(k, 1)$-path G, one can uniquely reconstruct the string of coloured graphs $\sigma_\beta(B')$ for some decomposition B' of G and some labeling function β admissible by B'.

3.1.1 A k-Generative Partial Order

Let G be a $(k, 1)$-path, let $B = \langle B_1, \ldots, B_m \rangle$ be an arbitrary $(k, 1)$-decomposition of G, and let G^d be the digraph induced by B. In this subsection, we show that a k-generative partial order on G is MS-definable over G^d_B.

Remark 3.1.1 By the definition of a $(k, 1)$-decomposition, each vertex of G^d (except for those forming the bag B_1) has an outgoing double arrow or a loop arrow.

Remark 3.1.2 Since each bag B_i $(1 < i \leq m)$ of the $(k, 1)$-decomposition B contains exactly one new vertex, for any vertices $u, v \in \bar{V}$, $u \prec v$ means that $\text{first}(u) < \text{first}(v)$.

Lemma 3.1.3 *Let G, G^d, and B be the same as above. Let $S = B_r$ for some $1 \leq r < m$ and let $v \in V \setminus S$ be any vertex minimal with respect to $\preceq$ (i.e., for every $u \in V$, $u \prec v$ implies that $u \in S$). Then there exists a $(k, 1)$-decomposition $B' = \langle B_1, \ldots, B_r, B'_{r+1}, \ldots, B'_m \rangle$ of G such that $B'_{r+1} = \text{old}(B_{r+1}) \cup \{v\}$.*

Proof. By Remark 3.1.1 and by the definition of $\preceq$, the bag B'_{r+1} contains at least one drop vertex. Thus we can continue the sequence $\langle B_1, \ldots, B_r, B'_{r+1} \rangle$ to some $(k, 1)$-decomposition B' by adding the non-added vertices of G in the order they were added to the original decomposition B. □

Here is a *nondeterministic* algorithm $\mathcal{A}$ suggested by Lemma 3.1.3 for constructing some (possibly different from B) $(k, 1)$-decomposition $B' = \langle B'_1, \ldots, B'_m \rangle$ of G given the digraph G^d_B:

Algorithm $\mathcal{A}$

Step 1. $B'_1 = B_1$.

Step i *($1 < i \leq m$).* Let $\langle B'_1, \ldots, B'_{i-1}\rangle$ be an already constructed prefix of the $(k,1)$-decomposition B'. Take $B'_i = \text{non-drop}(B'_{i-1}) \cup \{v\}$, where v is an arbitrary minimal (with respect to $\preceq$) non-added vertex (i.e., v is any minimal vertex in the set $V \setminus \mathcal{B}'_{i-1}$).

We extend $\preceq$ so that for any two vertices $u \in B_1$ and $v \notin B_1$ incomparable with respect to $\preceq$, u is less than v. Let $\preceq^1$ denote the transitive closure of that extension.

Now we can state the following theorem.

Theorem 3.1.4 *The relation $\preceq^1$ is a k-generative partial order on G.*

As shown in Lemma 2.2.21, the precedence $\preceq$ (and therefore, the k-generative partial order $\preceq^1$ on G) is MS-definable over the digraph G^d.

We end this subsection with the following lemma.

Lemma 3.1.5 *For the $(k,1)$-decomposition B' constructed by algorithm $\mathcal{A}$, $G^d_{B'}$ is isomorphic to G^d_B.*

Proof. For any vertex $v \in \bar{V}$, the vertices adjacent to it can be divided (using the structure of G^d_B) into two sets: the set of those with incoming arrows from v, denoted by $V_{\text{in}}(v)$, and the set of those with outgoing arrows to v, denoted by $V_{\text{out}}(v)$.

By the construction of B', for every B'_i ($1 < i \leq m$) such that $\text{new}(B'_i) = \{v\}$ ($v \in V$), $V_{\text{in}}(v) \subseteq \text{old}(B'_i)$ and $V_{\text{out}}(v) \cap \text{old}(B'_i) = \emptyset$. It is also not difficult to see that the drop vertices of B'_i are exactly those that had incoming double arrows from v or had loop arrows in the digraph G^d_B. Thus, B' induces the same single arrows, double arrows, and loop arrows as B. □

3.1.2 A k-Generative Linear Order

Algorithm $\mathcal{A}$ from the previous subsection is nondeterministic because there can be more than one minimal vertex in the set of vertices yet to be added (any of which can be chosen). In this subsection, we show how one can linearly order those alternatives (or, more precisely, certain equivalence classes of those alternatives) by dividing the set of vertices V into a sequence of (pairwise disjoint) $k+1$ sets $\langle P_1, \ldots, P_k, L\rangle$.

We colour the digraph G^d_B so that the precedence relation $\preceq$ is completed to a linear order on the set non-drop(B_1). (We do so by colouring the non-drop vertices of B_1 with colours $1,\ldots,k$ so that no two vertices are coloured the same).

Notation 3.1.6 We will denote this new coloured digraph G^d_B by G^{d1}_B (or simply G^{d1}).

Using G^{d1} enables us to define the following k sets $P_1,\ldots,P_k$:

For any $v \in V$, $v \in P_i$ $(1 \le i \le k)$ iff i is the minimum over the labels of the vertices $u \in$ non-drop(B_1) such that there is a path of double arrows in the digraph G^{d1} from v to u.

Example 3.1.7 The digraphs G^d_2 and G'^d_2 from Example 2.1.30 can be considered as G^{d1}_2 and G'^{d1}_2, respectively. We have the following two sets for them: $P_1 = \{1,3,6\}$ and $P_2 = \{2\}$.

Remark 3.1.8 Since no vertex in G^d can have more than one *incoming* double arrow, the induced graph $G^{d1}[P_i]$ is a path of double arrows for each $i \in \{1,\ldots,k\}$.

Definition 3.1.9 The vertices in $\cup_{i=1}^k P_i$ are called *nodes*. The set of all nodes of G will be denoted by N.

Definition 3.1.10 The set of *leaves* is defined as $\bar{V} \setminus N$ and is denoted by L.

Example 3.1.11 For the digraphs G^d_2 and G'^d_2 from Example 2.1.30, $L = \{4,5\}$.

Remark 3.1.12 Leaves are exactly those vertices of G^{d1} that have loop arrows.

Remark 3.1.13 By the definition of a $(k,1)$-decomposition, each leaf $w \in L$ has at most k outgoing single arrows and no incoming arrows (except for the loop arrow). Thus, all its arrows point to at most k nodes from *different* sets $P_1,\ldots,P_k$.

Below we define a partial order on the set V that will be a linear order on the set of nodes N. We add new vertices $v_\perp$ and $v_\top$ to V such that for any $v \in V$, $v_\perp \preceq v \preceq v_\top$. We assume that $v_\perp$ and $v_\top$ belong to the set P_1.

Since each P_i $(1 \le i \le k)$ is linearly ordered by $\preceq$ (see Remark 3.1.8), we can write it as a corresponding sequence of vertices. Let $P_1 = \langle v^1_1,\ldots,v^1_{l_1}\rangle,\ldots,P_k = \langle v^k_1,\ldots,v^k_{l_k}\rangle$.

Notation 3.1.14 For any $u, v \in V$, we denote by $(u, v]$ the set $\{w \in V | \, w \preceq v \text{ and } w \not\preceq u\}$. And for any $u_1, \ldots, u_t, v_1, \ldots, v_t \in V$ $(t \geq 1)$, we denote by $\langle (u_1, v_1], \ldots, (u_t, v_t] \rangle$ the set $(u_1, v_1] \cap \ldots \cap (u_t, v_t]$.

We partition $N \cup L$ into the sets $\langle (v_i^1, v_{i+1}^1] \rangle$, $1 \leq i < l_1$. This partitioning will be called the *partitioning of level 1*. The sets $\langle (v_i^1, v_{i+1}^1] \rangle$, $1 \leq i < l_1$ can be linearly ordered according to the sequence P_1.

Let $\langle (v_i^1, v_{i+1}^1] \rangle \cap P_2$ $(1 \leq i < l_1)$ correspond to the subsequence $\langle v_{j_1}^2, \ldots, v_{j_i}^2 \rangle$ of P_2. We define a new sequence $s_i^2 = \langle v_i^1, v_{j_1}^2, \ldots, v_{j_i}^2, v_{i+1}^1 \rangle$. Then we can divide $\langle (v_i^1, v_{i+1}^1] \rangle$ further into the sets $\langle (v_i^1, v_{i+1}^1], (u, u'] \rangle$ for every two consecutive elements u and u' of s_i^2. These sets are linearly ordered according to s_i^2. Thus we get another partitioning of $N \cup L$, called the *partitioning of level 2*.

Continuing in this manner gives us k partitionings (refinements of each other) and the corresponding k linear orders (each of which is consistent with $\preceq$).

Remark 3.1.15 Every set of vertices in the partitioning of level k contains exactly one node of G.

In view of this remark, the k linear orders defined above induce a partial order on V which is a linear order on the set of nodes N. We will denote this partial order by $\preceq^n$.

We have the following two lemmas.

Lemma 3.1.16 *Let u and v be two nodes such that $u \preceq^n v$ and there is no other node v' for which $u \preceq^n v' \preceq^n v$. Then any two leaves w_1 and w_2 such that $u \preceq^n w_1, w_2 \preceq^n v$ are incomparable with respect to $\preceq$.*

Proof. Let us suppose that $w_1 \prec w_2$. Then there must exist some node v' such that $w_1 \prec v' \prec w_2$ (by the definition of $\preceq$). But this means that $u \preceq^n v' \preceq^n v$, which contradicts the condition of the lemma. □

Lemma 3.1.17 *Any two leaves incomparable with respect to $\preceq^n$ are also incomparable with respect to $\preceq$.*

Proof. For any two leaves w_1 and w_2 incomparable with respect to $\preceq^n$, there will be two nodes u and v such that $u \preceq^n w_1, w_2 \preceq^n v$, with no node v' coming between u and v.

(Otherwise w_1 and w_2 would be comparable with respect to $\preceq^n$.) Our claim now follows from Lemma 3.1.16. □

We can order some leaves in L (incomparable with respect to $\preceq^n$) according to the set of nodes they point to (by defining a certain lexicographical ordering), but there can be leaves that are "structurally indistinguishable" (consider leaves 4 and 5 of the digraph $G_2'^d$, for example). We will see, however, that it is not necessary to order such indistinguishable leaves to solve the problem of recognizability implying definability for partial $(k,1)$-paths. We will just linearly order certain equivalence classes of leaves (i.e., sets of indistinguishable leaves).

Notation 3.1.18 For a leaf $w \in L$, $P(w)$ denotes the set of nodes to which there are arrows from w, i.e., $P(w) = \{v \in N | (w,v) \in E\}$.

We associate with each leaf $w \in L$ its *characteristic vector* $\chi(w) = (\chi_1(w), \ldots, \chi_k(w))$, where for each $i \in \{1, \ldots, k\}$,

$$\chi_i(w) = \begin{cases} 1, & \text{if } P(w) \cap P_i \neq \emptyset \\ 0 & \text{otherwise} \end{cases}.$$

Now we extend $\preceq^n$ to a new partial order, denoted by $\preceq^{nl}$, so that for any two leaves w_1 and w_2 incomparable with respect to $\preceq^n$, $w_1 \preceq^{nl} w_2$ iff $\chi(w_1)$ is lexicographically less than $\chi(w_2)$. (It should be clear that thus defined relation $\preceq^{nl}$ is indeed a partial order on V.)

Lemma 3.1.19 *If leaves w_1 and w_2 are incomparable with respect to $\preceq$, then $\chi_i(w_1) = \chi_i(w_2)$ iff $P(w_1) \cap P_i = P(w_2) \cap P_i$ for any $1 \leq i \leq k$.*

Proof. If some leaves w_1 and w_2 have arrows to distinct nodes v_1 and v_2, respectively, such that $v_1, v_2 \in P_i$ $(1 \leq i \leq k)$ and $v_1 \prec v_2$, then $w_1 \prec w_2$. □

Definition 3.1.20 We define the following relation of *p-equivalence*, denoted by $\overset{p}{\sim}$, on the set of vertices V of G:

For any two vertices $w_1, w_2 \in V$, $w_1 \overset{p}{\sim} w_2$ iff $w_1, w_2 \in L$ and $P(w_1) = P(w_2)$.

Let us consider the quotient graph $G_p = G/\overset{p}{\sim} = (V_p, E_p)$.

Notation 3.1.21 We denote by $\bar{V}_p$ the quotient set $\bar{V}/\overset{p}{\sim}$.

We extend $\preceq^{nl}$ to the set V_p as follows: For any $[u]_{\overset{p}{\sim}}, [v]_{\overset{p}{\sim}} \in V_p$, $[u]_{\overset{p}{\sim}} \preceq [v]_{\overset{p}{\sim}}$ iff there exist $u' \in [u]_{\overset{p}{\sim}}$ and $v' \in [v]_{\overset{p}{\sim}}$ such that $u' \preceq^{nl} v'$.

By Lemmas 3.1.17 and 3.1.19, any two incomparable (with respect to $\preceq^n$) leaves u and v for which $\chi(u) = \chi(v)$ are p-equivalent. Thus $\preceq^{nl}$ is a linear order on the set $(N \cup L)/\overset{p}{\sim}$. By arbitrarily ordering the drop vertices of B_1, we will get the linear order on V_p, denoted by $\leq_p$.

Notation 3.1.22 We will denote the digraph G_B^{d1} with ordered drop vertices of B_1 by $G_B^{d1'}$ (or just $G^{d1'}$).

Example 3.1.23 For the partial 2-path G_2', the order $\preceq^{nl}$ considered over $\bar{V}_p$ gives the following sequence of p-equivalence classes: $\langle\{3\},\{4,5\},\{6\}\rangle$.

The linear order $\leq_p$ allows us to construct a $(k,1)$-decomposition B_p' of the graph G_p as follows: Take $B_1' = B_1/\overset{p}{\sim}$. For a constructed prefix $\langle B_1', \ldots, B_{i-1}'\rangle$ $(i > 1)$ of B_p', take $B_i' =$ non-drop$(B_{i-1}') \cup \{v\}$, where v is the *minimum* (with respect to $\leq_p$) vertex in the set $V_p \setminus B_{i-1}'$. (It should be clear that thus constructed sequence B_p' is indeed a $(k,1)$-decomposition of G_p.)

Example 3.1.24 For the partial 2-path G_2',

$$B_p' = \langle\{[1],[1'],[2]\},\{[1],[2],[3]\},\{[2],[3],[4]\},\{[2],[3],[6]\}\rangle,$$

where $[u]$ denotes the set of vertices p-equivalent to u $(u \in V)$.

Thus we have the lemma.

Lemma 3.1.25 *For G_p and $\leq_p$ as above, $\leq_p$ is a k-generative linear order on G_p.*

We conclude this subsection by showing that the quotient graph G_p is MS-definable in terms of $G^{d1'}$ and that the linear order $\leq_p$ on G_p is MS-definable over $G^{d1'}$.

The following lemma must be obvious.

Lemma 3.1.26 *The p-equivalence $\overset{p}{\sim}$ is MS-definable over $G^{d1'}$.*

Corollary 3.1.27 *The graph G_p is MS-definable in terms of $G^{d1'}$.*

Lemma 3.1.28 *The partial order $\preceq^{nl}$ on the set V_p is MS-definable over $G^{d1'}$.*

Proof. Clearly, the relations of "being a node from the set P_i" $(1 \le i \le k)$ are expressible in MS on $G^{d1'}$.

The k equivalences on V corresponding to the partitionings of V of level i, $1 \le i \le k$, are MS-definable, since for each level i, an equivalence class is determined uniquely by some vector of i pairs of vertices.

Checking the lexicographical order of characteristic vectors for any two leaves can also be done in MS. □

Thus we have the following lemma.

Lemma 3.1.29 *The linear order $\le_p$ on the set V_p is MS-definable over $G^{d1'}$.*

3.1.3 Using $\le_p$ to Construct a Decomposition of G

In the previous subsection, we defined $\le_p$ and proved it to be a k-generative linear order on G_p. In this subsection, we show how to reconstruct a decomposition of the original graph G using the decomposition of the quotient graph G_p.

Let $\langle v_1, \ldots, v_l \rangle$ be a sequence of vertices in $\bar{V}_p$ ordered with respect to $\le_p$. The decomposition B'_p can then be written out as $B'_p = \langle B'_1, B'(v_1), \ldots, B'(v_l) \rangle$.

Remark 3.1.30 For any $w \in \bar{V}_p$ and any $u \in V_p$, $u \in B'(w)$ iff

i. $u = w$, or

ii. $u \le_p w$ and there is some $w' \in V_p$ such that $w \le_p w'$ and $(w', u) \in E_p$.

Notation 3.1.31 We identify B'_1 with $B'(v)$ for any $v \in B'_1$.

Remark 3.1.32 For every $i \in \{1, \ldots, l\}$, $\text{new}(B'(v_i)) = \{v_i\}$ and every old vertex of $B'(v_i)$ is a $\overset{p}{\sim}$-class containing exactly one vertex of G.

We can construct a $(k.1)$-decomposition of the graph G as follows: In the sequence B'_p, replace B'_1 with B_1. For every $i \in \{1, \ldots, l\}$ such that v_i is a $\overset{p}{\sim}$-class containing exactly one vertex w of G, replace $B'(v_i) = \{[u_1]_{\overset{p}{\sim}}, \ldots, [u_{s_i}]_{\overset{p}{\sim}}, [w]_{\overset{p}{\sim}}\}$ with the bag $B(w) = \{u_1, \ldots, u_{s_i}, w\}$. For every $i \in \{1, \ldots, l\}$ such that v_i is a $\overset{p}{\sim}$-class containing $t_i > 1$

vertices $w_1, \ldots, w_{t_i}$ of G (all of which should be $\overset{p}{\sim}$*-leaves* in this case), replace $B'(v_i) = \{[u_1]_{\overset{p}{\sim}}, \ldots, [u_{s_i}]_{\overset{p}{\sim}}, [w_1]_{\overset{p}{\sim}}\}$ with the sequence of bags $B(w_1) = \{u_1, \ldots, u_{s_i}, w_1\}, \ldots, B(w_{t_i}) = \{u_1, \ldots, u_{s_i}, w_{t_i}\}$. Let B' denote thus constructed decomposition of G.

Example 3.1.33 For the partial 2-path G'_2, the following two decompositions are possible:

$$B' = \langle\{1,1',2\},\{1,2,3\},\{2,3,4\},\{2,3,5\},\{2,3,6\}\rangle$$

or

$$B' = \langle\{1,1',2\},\{1,2,3\},\{2,3,5\},\{2,3,4\},\{2,3,6\}\rangle.$$

Notation 3.1.34 We identify B_1 with $B(u)$ for any $u \in B_1$.

Notation 3.1.35 For any $w \in V$, we denote by $B^-(w)$ the set non-drop$(B(w))$.

Remark 3.1.36 The above-mentioned procedure for reconstructing a decomposition B' of G does not produce a unique result since $\overset{p}{\sim}$-leaves are ordered arbitrarily. However, the bags associated with any two $\overset{p}{\sim}$-leaves correspond to two isomorphic subgraphs of G.

Below we show the relationship between the string of coloured graphs associated with B'_p and that associated with B'.

Let us convert B'_p into the extended decomposition $\bar{B}'_p$ and colour G_p with some labeling function $\beta_p : V_p \to \{1, \ldots, k+1\}$ admissible by $\bar{B}'_p$ (see Section 2.3.2). Let us also convert the decomposition B' of G into the extended decomposition $\bar{B}'$ and colour the graph G with the labeling function $\beta : V \to \{1, \ldots, k+1\}$ such that, for every $v \in V$, $\beta(v) = \beta_p([v]_{\overset{p}{\sim}})$. (The labeling function β is admissible by $\bar{B}'$ since no leaf appears in two consecutive bags of $\bar{B}'$.)

The string $\sigma_\beta(\bar{B}')$ is no longer dependent on the order in which $\overset{p}{\sim}$-leaves were added to the decomposition B', because the symbols in the alphabet Σ_g that correspond to the bags $\bar{B}'(w_1)$ and $\bar{B}'(w_2)$, for any two $\overset{p}{\sim}$-leaves w_1 and w_2, are identical.

Remark 3.1.37 It is noteworthy that although the string $\sigma_\beta(\bar{B})$ is determined uniquely on a suitably coloured graph, the colouring itself depends on some arbitrary choices.

Let $\sigma_{\beta_p}(\bar{B}'_p) = \langle \sigma_0, \sigma_{0'}, \sigma_1, \sigma_{1'}, \ldots, \sigma_l, \sigma_{l'} \rangle$. (It corresponds to the extended decomposition $\bar{B}'_p = \langle B'_1, B'^-_1, B'(v_1), B'^-(v_1), \ldots, B'(v_l), B'^-(v_l) \rangle$.) Then $\sigma_\beta(\bar{B}')$ can be obtained from $\sigma_{\beta_p}(\bar{B}'_p)$ by replacing the subsequence $\langle \sigma_i, \sigma_{i'} \rangle$, for each $i \in \{1, \ldots, l\}$ such that v_i is a $\overset{p}{\sim}$-class containing $t_i > 1$ $\overset{p}{\sim}$-leaves, with the subsequence $\langle \sigma_i, \sigma_{i'} \rangle$ repeated t_i times, i.e., with $\langle \sigma_i, \sigma_{i'} \rangle^{t_i}$.

3.2 A CMS-Formula

In this section, we define a CMS-formula that is true on a graph G iff some decomposition of G is accepted by the automaton A. First we show that for the coloured graph $G^{d1'}$, and then give the corresponding admissibility conditions for a colouring of G.

3.2.1 CMS-Definability of Recognizability for Coloured $(k,1)$-Paths

As a consequence of Remark 3.1.30, we have the following lemma.

Lemma 3.2.1 *For any vertices $u, w \in V_p$, the relations "$u \in B'(w)$" and "$u \in B'^{-}(w)$" are MS-definable over $G^{d1'}$.*

Let β be a labeling function admissible by $\bar{B'}_p$. Its admissibility means the following: No two vertices appearing in the same bag $B'(v)$ (for any $v \in V_p$) are labeled the same, and no two vertices appearing in two bags $B'(u)$ and $B'(v)$ such that u immediately precedes v with respect to $\leq_p$ are labeled the same. This is clearly expressible in MS.

For $\bar{B'}_p$ and β defined above, we have the following statements.

Lemma 3.2.2 *For any vertex $w \in V_p$ and any coloured graph $\sigma \in \Sigma_g$, there are an MS-formula $\phi_{w\sigma}$ which is true iff $\sigma_\beta(B'(w)) \cong \sigma$ and an MS-formula $\phi^{-}_{w\sigma}$ which is true iff $\sigma_\beta(B'^{-}(w)) \cong \sigma$.*

Proof. This follows from Lemma 3.2.1 and the fact that each $\sigma \in \Sigma_g$ is of size at most $k+1$. □

Theorem 3.2.3 *The string $\sigma_\beta(\bar{B'}_p)$ is MS-definable in terms of $G^{d1'}$.*

Proof. We will show that a path G_σ of length $|\sigma_\beta(\bar{B'}_p)|/2$ each vertex of which is labeled with a pair of symbols from Σ_g so that the string of these pairs corresponds to the word $\sigma_\beta(\bar{B'}_p)$ is MS-definable in terms of $G^{d1'}$. (We consider G_σ as a $\{p_{\text{inc}}\}$-relational structure whose domain D is the set of vertices of G_σ and p_{inc} is the binary incidence relation over D^2.)

We take the set $\bar{V}_p \cup \{v_0\}$, where v_0 is an arbitrary vertex in B'_1, as the domain of G_σ. (Clearly, this set of vertices is MS-definable in terms of $G^{d1'}$.)

The incidence relation of G_σ is defined in an obvious way according to the linear order $\leq_p$ on V_p. (Again, this is MS-definable.)

To check that G_σ is labeled properly, i.e., that its set vertices is partitioned into the two families of sets $\{V_\sigma\}_{\sigma\in\Sigma_g}$ and $\{V_\sigma^-\}_{\sigma\in\Sigma_g}$ so that a vertex u of G_σ is in V_σ (V_σ^-), $\sigma\in\Sigma_g$, iff $\sigma_\beta(B'(u))\cong\sigma$ ($\sigma_\beta(B'^-(u))\cong\sigma$), we just use the corresponding MS-formulas from Lemma 3.2.2. □

Recall that one can obtain from $\sigma_\beta(\bar{B}'_p)$ the string over the alphabet Σ_g corresponding to some decomposition of the original graph G by repeating the subsequences

$$\langle\sigma_\beta(B'(u))\sigma_\beta(B'^-(u))\rangle,$$

such that $u\in V_p$ is a p-equivalence class of cardinality t, t times.

Let $A=(\Sigma_g,Q,\delta,q_0,F)$ be the automaton recognizing a family $\mathcal{G}$ of $(k,1)$-paths G (coloured $(k,1)$-paths $G^{d1'}$). To define the required CMS-formula for G_σ, we can proceed similarly to the case of words (see Theorem 2.4.1). That is, we first "guess" a colouring of vertices of G_σ such that every vertex u is associated with the state $q(u)\in Q$ which the automaton A enters after having read the string $\langle\sigma_\beta(B'(u))\sigma_\beta(B'^-(u))\rangle^t$, where t is the cardinality of the p-equivalence class $u\in V_p$. Then we check the admissibility of our guess.

By finiteness of A, it suffices to know $t \bmod a$ (for some number a dependent on a state q and the symbols $\sigma_1=\sigma_\beta(B'(u))$ and $\sigma_2=\sigma_\beta(B'^-(u))$) to determine the state $q'=\delta^*(q,\langle\sigma_1\sigma_2\rangle^t)$. Therefore, the required admissibility check can be expressed in CMS.

Remark 3.2.4 Note that it is not important which particular $(k,1)$-decomposition B of a $(k,1)$-path G is used for defining the digraph $G_B^{d1'}$. Our reasoning will hold for any other $(k,1)$-decomposition of G.

Thus we have proved the following result.

Theorem 3.2.5 *Every recognizable family of coloured connected $(k,1)$-paths $G^{d1'}$ is CMS-definable.*

3.2.2 Admissibility Conditions

Here we state the conditions on a colouring of a $(k,1)$-path $\tilde{G}$ that check the admissibility of that colouring with respect to recognizability of $(k,1)$-paths. That is, they verify that the colouring induces a digraph $\tilde{G}_{B''}^{d1'}$ for some $(k,1)$-decomposition B'' of $\tilde{G}$.

Clearly, the digraph $G^{d1'}$ is MS-colourable in terms of a $(k,1)$-path G (see Lemma 2.2.20 and the definition of $G^{d1'}$). Let $G^{c_{d1'}}$ denote the coloured graph that induces $G^{d1'}$.

Let $\tilde{G}^c$ be a coloured graph having the same format as $G^{c_{d1'}}$. Consider the following conditions on $\tilde{G}^c$:

i. The set denoting the bag B_1 contains at most $k+1$ vertices, of which at most k are non-drop vertices of B_1 (i.e., at most k vertices in B_1 are adjacent to some vertices not in B_1).

ii. $\tilde{G}^c$ defines some directed graph $\tilde{G}'$ over $\tilde{G}$. That is the $k+1$ vertex-sets and $k+1$ edge-sets used for directing edges of $\tilde{G}$ are such that they form partitionings of the sets V and E, respectively, and for any edge $e = \{u, v\} \in E$, u and v belong to different vertex sets (i.e., are coloured differently) and e is coloured with the same colour as u or v (cf. the proof of Lemma 2.2.20).

iii. The labeled digraph $\tilde{G}' = (V, E')$ (labeled with double and loop arrows) is such that

   (a) there is no arc $(u, v) \in E'$ such that $u \in B_1$ and $v \notin B_1$,

   (b) for every $v \in V \setminus \mathrm{drop}(B_1)$ with incoming arrow (or arrows), there is exactly one incoming double arrow,

   (c) every vertex with a loop arrow has no outgoing double arrows and no incoming arrows of any kind,

   (d) every vertex in $\tilde{V}$ has either an outgoing double arrow or a loop arrow.

iv. The relation $\preceq'$ induced by the labeled digraph $\tilde{G}'$ (as in Definition 2.1.33) is a partial order on V (i.e., it is reflexive, transitive, and antisymmetric).

v. The non-drop and drop vertices of B_1 are ordered. We assume that two sequences of k vertex-sets are used, the first for ordering the non-drop vertices of B_1, the second for ordering the drop vertices of B_1. They should form the partitions of the sets non-drop(B_1) and drop(B_1), respectively.

It should be obvious that conditions (i)-(v) are expressible in MS.

Lemma 3.2.6 *If $\tilde{G}^c$ satisfies conditions (i)-(iv) stated above, then the relation $\preceq'^1$ induced by that coloured graph is a k-generative partial order on $\tilde{G}$.*

Proof. By condition (iv), $\preceq'$ is a partial order on V. Let $\langle v_1, \ldots, v_l \rangle$ be a sequence of vertices in $\tilde{V}$ ordered according to some extension of $\preceq'$ to a linear order on V.

The vertex v_1 has at most k outgoing arrows to non-drop vertices of B_1 (by conditions (i) and (iii.a)). By conditions (iii.b), (iii.c), (iii.d), and by the definition of $\preceq'$, the set $B_2' = \text{non-drop}(B_1) \cup \{v_1\}$ contains at least one drop vertex.

We can apply the same arguments to the vertex v_2 and the set non-drop(B_2'), and so on. □

Corollary 3.2.7 *For an arbitrary graph $\hat{G}$, if there exists a coloured graph $\hat{G}^c$ satisfying conditions (i)-(iv), then $\hat{G}$ is a $(k,1)$-path.*

Remark 3.2.8 Corollary 3.2.7 means that there is an MS-formula that checks if a given graph is a $(k,1)$-path. This formula can be constructed explicitly.

Let B'' be some $(k,1)$-decomposition induced by the partial order $\preceq'^1$ from Lemma 3.2.6. It follows from Lemma 3.1.5 that the digraph $\tilde{G}'$ defined above is isomorphic to $\tilde{G}^d_{B''}$. Condition (v) ensures then that $\tilde{G}^c$ defines the digraph $\tilde{G}^{d1'}_{B''}$.

Remark 3.2.9 For a $(k,1)$-path G, the coloured graph $G^{c_{d1'}}$ satisfies conditions (i)-(v). Therefore, there alway exists a colouring of a $(k,1)$-path that satisfies conditions (i)-(v).

Now we can state the principal result of this chapter.

Theorem 3.2.10 *A recognizable family of connected $(k,1)$-paths is CMS-definable.*

Proof. It follows from Theorem 3.2.5, Lemma 2.2.14, and Remark 3.2.9. □

Chapter 4

The General Case

Here we consider the case of partial k-paths, i.e., (k,p)-paths for any $p \in \{1, \ldots, k\}$. First, we deal with connected partial k-paths. The solution for possibly disconnected partial k-paths will be given in Section 4.3.5.

As in the case of $(k,1)$-paths, we define coloured partial k-paths and show that recognizability implies definability for these coloured graphs. Then we give the corresponding admissibility conditions on colourings of partial k-paths.

Let $B = \langle B_1, \ldots, B_m \rangle$ be a nice k-decomposition of a partial k-path G. The family of sets $\text{new}(B_i)$, $1 \leq i \leq m$, forms a partitioning of the vertex-set V of G. We will call the equivalence on V that is induced by that partitioning the *1-equivalence* (denoted by $\stackrel{1}{\sim}$). The decomposition B also induces a linear order on the quotient set $V/\stackrel{1}{\sim}$ (denoted by $\leq_1$). Clearly, given the pair $(\stackrel{1}{\sim}, \leq_1)$, one can reconstruct the decomposition B of G.

We will show that the 1-equivalence is MS-colourable, but it does not seem possible to MS-colour the linear order $\leq_1$. (Recall that we were unable to linearly order in MS the leaves of a $(k,1)$-path.)

A nice decomposition B can also be viewed as a sequence of monotonic pieces

$$\langle M_1, \ldots, M_d \rangle,$$

where $M_s = \langle B_{i_s}, \ldots, B_{j_s} \rangle$ for each $1 \leq s \leq d$. We define the sets

$$\text{new}(M_s) = \cup_{i_s \leq r \leq j_s} \text{new}(B_r), \;\; 1 \leq s \leq d,$$

the family of which forms another partitioning of V. The corresponding equivalence on V will be called the *2-equivalence* (denoted by $\stackrel{2}{\sim}$). (Obviously, $\stackrel{1}{\sim}$ is a refinement of $\stackrel{2}{\sim}$.) This sequence of monotonic pieces induces a linear order on the quotient set $V/\overset{2}{\sim}$ (denoted by $\leq_2$).

We denote by $\text{new}^1(M_s)$ the set $\text{new}(B_{i_s})$, $1 \leq s \leq d$. Note that $\text{new}^1(M_s)$ is the only $\overset{1}{\sim}$-class inside $\text{new}(M_s)$ of cardinality greater than one.

Remark 4.0.1 By the definition of a nice decomposition, every vertex in the set $\text{new}(M_s) \setminus \text{new}^1(M_s)$ has an outgoing double arrow or a loop arrow in the digraph G^d.

We have the following lemma.

Lemma 4.0.2 *For a partial k-path G with a nice k-decomposition $B = \langle M_1, \ldots, M_d \rangle$, given the relations $\overset{1}{\sim}$, $\overset{2}{\sim}$, and $\leq_2$ defined above, one can define a k-generative partial order on G.*

Proof. We extend the precedence $\preceq$ so that for any two incomparable (with respect to $\preceq$) vertices u and v from different sets $\text{new}(M_i)$ and $\text{new}(M_j)$, $i \neq j$, respectively, u is less than v iff $i < j$, and for any two incomparable (with respect to $\preceq$) vertices v and w from the same $\text{new}(M_i)$ such that $v \in \text{new}^1(M_i)$ and $w \notin \text{new}^1(M_i)$, v is less than w. Then the transitive closure of this extension, denoted by $\preceq^p$, is a k-generative partial order on G.

Indeed, it is obvious that $\text{new}^1(M_1) = B_1$. By Remark 4.0.1 and the definition of $\preceq^p$, any vertex in $\text{new}(M_1) \setminus \text{new}^1(M_1)$ minimal with respect to $\preceq^p$ can be chosen to form the next bag B'_2 (see the proof of Lemma 3.1.3).

Let B'_l be the last bag constructed for $\text{new}(M_1)$. It should be clear that then

$$\text{non-drop}(B'_l) = \text{non-drop}(B_{j_1}).$$

By the definition of the linear order $\leq_2$, the set $\text{non-drop}(B'_l) \cup \text{new}^1(M_2)$ equals B_{i_2}, thus we can continue with $\text{new}(M_2) \setminus \text{new}^1(M_2)$ as above, and so on. □

Definition 4.0.3 For a partial k-path G, a triple $(\overset{1}{\sim}', \overset{2}{\sim}', \leq'_2)$, where $\overset{1}{\sim}'$ and $\overset{2}{\sim}'$ are equivalences on V and $\leq'_2$ is a linear order on $V/\overset{2}{\sim}'$, is called a *linear k-generative structure on G* iff there exists some nice k-decomposition B of G such that $\overset{1}{\sim}'$ and $\overset{2}{\sim}'$ are the 1-equivalence and 2-equivalence, respectively, induced by B, and $\leq'_2$ is the linear order on 2-equivalence classes induced by B.

Definition 4.0.4 For a partial k-path G, a triple $(\overset{1}{\sim}', \overset{2}{\sim}', \preceq'_2)$, where $\overset{1}{\sim}'$ and $\overset{2}{\sim}'$ are equivalences on V and $\preceq'_2$ is a partial order on $V/\overset{2}{\sim}'$, is called a *partial k-generative structure on G* iff any completion of $\preceq'_2$ to a linear order yields a linear k-generative structure on G.

Again, the 2-equivalence is MS-colourable, but not the linear order $\leq_2$. Thus, we cannot MS-colour a linear k-generative structure on G. We will be unable to MS-colour a partial k-generative structure on G either.

Let us extend the relation of strong precedence to $\overset{2}{\sim}$-classes (it will be called the *strong 2-precedence* and denoted by $\overset{s2}{\preceq}$):

For any two distinct $\overset{2}{\sim}$-classes $[u]_{\overset{2}{\sim}}$ and $[v]_{\overset{2}{\sim}}$ $(u, v \in V)$, $[u]_{\overset{2}{\sim}} \overset{s2}{\prec} [v]_{\overset{2}{\sim}}$ iff there are $u' \in [u]_{\overset{2}{\sim}}$ and $v' \in [v]_{\overset{2}{\sim}}$ such that $u' \overset{s}{\prec} v'$. Then the *2-precedence* is defined as the reflexive and transitive closure of strong 2-precedence and denoted by $\overset{2}{\preceq}$.

The 2-precedence is a partial order on the set of $\overset{2}{\sim}$-classes such that for all $u, v \in V$, $[u]_{\overset{2}{\sim}} \overset{2}{\prec} [v]_{\overset{2}{\sim}}$ implies that the vertices in $[u]_{\overset{2}{\sim}}$ come before those in $[v]_{\overset{2}{\sim}}$ in our decomposition B, i.e., for any $u' \in [u]_{\overset{2}{\sim}}$ and $v' \in [v]_{\overset{2}{\sim}}$, first$(u') <$ first(v').

However, $(\overset{1}{\sim}, \overset{2}{\sim}, \overset{2}{\preceq})$ is not necessarily a partial k-generative system on G (i.e., we cannot take just any $\overset{2}{\sim}$-class minimal with respect to $\overset{2}{\preceq}$ to continue the decomposition of G constructed so far). One reason is that each $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ starts with a set of more than one vertex all of which must be put into the same bag. The other reason is that $[u]_{\overset{2}{\sim}}$ can contribute more non-drop vertices than drop vertices. (We did not have the latter problem in the case of $(k, 1)$-paths, because there adding a new vertex always produced at least one drop vertex.)

In the following subsections, we define certain (MS-colourable) sets of 2-equivalence classes on which a desired partial order is MS-colourable. We partition each such set of 2-equivalence classes into subsets and define a new partial order on these subsets. We continue in this manner until each set contains exactly one 2-equivalence class. (We will show that there can be at most k such partitionings.)

We need the following definitions.

Definition 4.0.5 For a partial k-path G, a triple $(\overset{1'}{\sim}, \{\sim^i\}_{i=0}^r, \{\leq^i\}_{i=0}^r)$ (for some constant r), where

i. $\overset{1'}{\sim}$ and $\sim^i$, $1 \leq i \leq r$, are equivalences on V such that every two vertices of V are $\sim^0$-equivalent and $\sim^j$ is a refinement of $\sim^i$ for every $j > i$, and

ii. $\leq^i$ is a linear order on $\sim^i$-classes, $0 \leq i \leq r$, such that $\leq^j$ is a refinement of $\leq^i$ for every $j > i$ (i.e., the restriction of $\leq^j$ to $V/\sim^i$ coincides with $\leq^i$)

is called a *linear k^r-generative structure on* G iff the triple $(\stackrel{1'}{\sim}, \sim^r, \leq^r)$ is a linear k-generative structure on G.

Definition 4.0.6 For a partial k-path G, a triple $(\stackrel{1'}{\sim}, \{\sim^i\}_{i=0}^r, \{\preceq^i\}_{i=0}^r)$, where $\stackrel{1'}{\sim}$ and $\sim^i$, $1 \leq i \leq r$, are equivalences on V satisfying the same conditions as in Definition 4.0.5 and $\preceq^i$ are partial orders on $\sim^i$-classes, $0 \leq i \leq r$, is called a *partial k^r-generative structure on* G iff for any completions of $\preceq^i$ to linear orders $\leq'^i$, $0 \leq i \leq r$, such that $\leq'^j$ is a refinement of $\leq'^i$ for every $j > i$, the triple $(\stackrel{1'}{\sim}, \{\sim^i\}_{i=0}^r, \{\leq'^i\}_{i=0}^r)$ is a linear k^r-generative structure on G.

We will show that a certain partial k^k-generative structure on G is MS-definable over a suitably coloured graph G^c. This will allow us to reconstruct some decomposition of G and to check in CMS if it is accepted by the corresponding automaton A.

4.1 A Partial k^k-Generative Structure

In this section, we define a partial k^k-generative structure on a partial k-path G that will be MS-colourable in terms of G. First we define $k+1$ so-called 3_i-*equivalences* on V, $0 \leq i \leq k$, such that every two vertices in V are 3_0-equivalent, $\stackrel{3_{i+1}}{\sim}$ is a refinement of $\stackrel{3_i}{\sim}$ for each $0 \leq i < k$, and $\stackrel{3_k}{\sim}$ coincides with $\stackrel{2}{\sim}$. Then for each 3_i-equivalence class C, $0 \leq i < k$, we partially order the 3_{i+1}-equivalence classes in $C/\stackrel{3_{i+1}}{\sim}$.

4.1.1 3-Equivalences

First define the *balance* of a sequence of $\stackrel{2}{\sim}$-classes so that for each sequence with non-positive balance, the number of non-drop vertices produced by it is at most that of drop vertices.

Let G be a partial k-path, let B be an arbitrary nice k-decomposition of G, and let G^d be the digraph induced by B.

Notation 4.1.1 For $u \in V$, the monotonic subsequence M of B such that $\text{new}(M) = [u]_{\stackrel{2}{\sim}}$ will be denoted by $M_u = \langle B_{i_u}, \ldots, B_{j_u} \rangle$.

Definition 4.1.2 For $u \in V$ and the corresponding monotonic subsequence M_u of B, the cardinality of the set $\text{new}(B_{i_u})$ is called the *width* of the $\stackrel{2}{\sim}$-class $[u]_{\stackrel{2}{\sim}}$ and is denoted by $\text{width}([u]_{\stackrel{2}{\sim}})$, i.e., $\text{width}([u]_{\stackrel{2}{\sim}}) = |\text{new}(B_{i_u})|$.

Remark 4.1.3 Informally, the width of a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$ is a measure of the "jump" that occurred in the decomposition B at the beginning of the monotonic piece M_u.

Definition 4.1.4 A vertex v in a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ (for some $u \in V$) is called a *drop vertex* of that $\overset{2}{\sim}$-class iff every vertex $w \in V$ such that $(w, v) \in E$ is in the same $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$.

Remark 4.1.5 For M_u corresponding to $[u]_{\overset{2}{\sim}}$ $(u \in V)$, a vertex $v \in [u]_{\overset{2}{\sim}}$ is a drop vertex of $[u]_{\overset{2}{\sim}}$ iff there exists $r \in \{i_u, \ldots, j_u\}$ such that $v \in \text{drop}(B_r)$.

Notation 4.1.6 The set of all drop vertices of a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$ will be denoted by $\text{drop}([u]_{\overset{2}{\sim}})$.

Definition 4.1.7 A vertex v' in a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$ that is not a drop vertex of $[u]_{\overset{2}{\sim}}$ is called a *non-drop vertex* of that $\overset{2}{\sim}$-class.

Notation 4.1.8 The set of all non-drop vertices of a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$ will be denoted by $\text{non-drop}([u]_{\overset{2}{\sim}})$.

Definition 4.1.9 A vertex $w \notin [u]_{\overset{2}{\sim}}$ $(u \in V)$ is said to be *removable* by the $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ iff there is $v \in [u]_{\overset{2}{\sim}}$ such that $(v, w) \in E_{\Rightarrow}$.

Remark 4.1.10 For M_u corresponding to $[u]_{\overset{2}{\sim}}$ $(u \in V)$, a vertex $w \in V$ is removable by $[u]_{\overset{2}{\sim}}$ iff $w \in \text{old}(B_{i_u})$ and there exists $r \in \{i_u, \ldots, j_u\}$ such that $w \in \text{drop}(B_r)$.

Notation 4.1.11 The set of all vertices removable by a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$ will be denoted by $\text{remov}([u]_{\overset{2}{\sim}})$.

Definition 4.1.12 The *balance* of a $\overset{2}{\sim}$-class $[u]_{\overset{2}{\sim}}$ $(u \in V)$, denoted by $\text{bal}([u]_{\overset{2}{\sim}})$, is defined by the formula:

$$\text{bal}([u]_{\overset{2}{\sim}}) = |\text{non-drop}([u]_{\overset{2}{\sim}})| - |\text{remov}([u]_{\overset{2}{\sim}})|.$$

Remark 4.1.13 For M_u corresponding to $[u]_{\overset{2}{\sim}}$ $(u \in V)$,

$$\text{bal}([u]_{\overset{2}{\sim}}) = |\text{non-drop}(B_{j_u})| - |\text{old}(B_{i_u})|.$$

Remark 4.1.14 If we apply our definition of balance to single vertices in a $(k, 1)$-path, we will see that the balance of each new vertex of the $(k, 1)$-decomposition (except for those in the first bag) is *non-positive* (because, by definition, each new vertex either "removes" at least one old vertex or is a drop vertex itself).

Let $S = \langle S_1, \ldots, S_d \rangle$ be the sequence of $\stackrel{2}{\sim}$-classes such that $S_i = \text{new}(M_i)$, $1 \leq i \leq d$, for our decomposition $B = \langle M_1, \ldots, M_d \rangle$.

Definition 4.1.15 A contiguous subsequence of the sequence S will be called a *2-block*. The *cumulative balance* of a 2-block is defined as the sum of the balances of its components. A vertex *belongs* to a 2-block iff it belongs to the union of its components.

Definition 4.1.16 A non-empty 2-block $T = \langle T_1, \ldots, T_l \rangle$ is called *balanced* iff $\text{bal}(T) \leq 0$ and no proper non-empty prefix of T has a non-positive balance, i.e., $\text{bal}(\langle T_1, \ldots, T_i \rangle) > 0$ for every $i \in \{1, \ldots, l-1\}$.

We extend the definition of width to 2-blocks as follows.

Definition 4.1.17 For a 2-block $T = \langle T_1, \ldots, T_l \rangle$, the *width* of T, denoted by width(T), is defined by the formula:

$$\text{width}(T) = \max_{1 \leq i \leq l} \{\text{bal}(\langle T_1, \ldots, T_{i-1} \rangle) + \text{width}(T_i)\}.$$

Remark 4.1.18 If $\langle B_{i_T}, \ldots, B_{j_T} \rangle$ is the subsequence of B that corresponds to a 2-block T, then

$$\text{width}(T) = \max_{i_T \leq r \leq j_T} \{|B_r|\} - |\text{old}(B_{i_T})|.$$

Guided by the analogy with the case of $(k,1)$-paths, we will split the sequence S into disjoint 2-blocks $T_1, \ldots, T_m$, so that $S = T_1 \ldots T_m$, where $T_1 = \langle S_1 \rangle$ and each 2-block T_j $(1 < j \leq m)$ is balanced. We do the same for each subsequence T_j, $1 < j \leq m$, of length greater than one.

Formally, we define $k+1$ sequences $\mathcal{T}_i$ $(0 \leq i \leq k)$ of 2-blocks by the following algorithm:

Algorithm $\mathcal{P}$

Step 0. $\mathcal{T}_0 = \langle T_1^0 \rangle$, with $T_1^0 = S$.

Step i $(1 \leq i \leq k)$. Let $\mathcal{T}_{i-1} = \langle T_1^{i-1}, \ldots, T_{t_{i-1}}^{i-1} \rangle$. The partitioning $\mathcal{T}_i$ is obtained from $\mathcal{T}_{i-1}$ by keeping each T_j^{i-1} of length one and replacing each $T_j^{i-1} = \langle S_{j1}^{i-1}, \ldots, S_{jd_{j,i-1}}^{i-1} \rangle$ $(1 \leq j \leq t_{i-1})$ of length greater than one with a sequence of 2-blocks $T_1, \ldots, T_m$ so that $T_j^{i-1} = T_1 \ldots T_m$, where $T_1 = \langle S_{j1}^{i-1} \rangle$ and each 2-block T_j $(1 < j \leq m)$ is balanced.

Lemma 4.1.19 *Every 2-block in the sequence $\mathcal{T}_k$ is of length one.*

Proof. Let T be a balanced 2-block obtained at step $i-1$ $(1 \leq i < k)$ which was replaced with a sequence of 2-blocks $T_1, \ldots, T_s$ $(s > 1)$ at step i. Assume that T_j (for some $1 < j \leq s$) was replaced with a sequence of 2-blocks $T_{j_1}, \ldots, T_{j_t}$ $(j_t > 1)$ at step $i+1$.

Let T correspond to the sequence of bags

$$\langle B_{i_{T_1}}, \ldots, B_{j_{T_1}}, \ldots, B_{i_{T_j}}, \ldots, B_{j_{T_j}}, \ldots, B_{i_{T_s}}, \ldots, B_{j_{T_s}} \rangle$$

in the decomposition B, and let T_j correspond to the sequence of bags

$$\langle B_{i_{T_{j_1}}}, \ldots, B_{j_{T_{j_1}}}, \ldots, B_{i_{T_{j_t}}}, \ldots, B_{j_{T_{j_t}}} \rangle.$$

(It is clear that $B_{i_{T_{j_1}}} = B_{i_{T_j}}$.)

By the definition of a balanced 2-block, $\text{bal}(T_1 \ldots T_{s'}) > 0$ for any $1 \leq s' < s$, and $\text{bal}(T_{j_1} \ldots T_{j_{t'}}) > 0$ for any $1 \leq t' < t$. Therefore, $|\text{old}(B_{i_{T_{s'}}})| > |\text{old}(B_{i_{T_1}})|$ for any $1 < s' \leq s$, and $|\text{old}(B_{i_{T_{j_{t'}}}})| > |\text{old}(B_{i_{T_{j_1}}})|$ for any $1 < t' \leq t$.

Thus, we have the inequalities $|\text{old}(B_{i_{T_{j_{t'}}}})| > |\text{old}(B_{i_{T_j}})| > |\text{old}(B_{i_{T_1}})|$. Since there are at most k old vertices in any given bag, our claim follows. $\square$

Definition 4.1.20 Each sequence $\mathcal{T}_i = \langle T_1^i, \ldots, T_{t_i}^i \rangle$ $(0 \leq i \leq k)$ induces the following equivalence relation on V, called 3_i-*equivalence*:

For any $u, v \in V$, $u \overset{3_i}{\sim} v$ iff there is some $j \in \{1, \ldots, t_i\}$ such that u and v belong to T_j^i.

Remark 4.1.21 Obviously, every two vertices of G are 3_0-equivalent.

The definitions of drop, non-drop, and removable vertices for 3_i-equivalence classes $(0 \leq i \leq k)$, as well as the balance of a 3_i-equivalence class, are analogous to those for 2-equivalence classes. The notation is also similar.

Remark 4.1.22 Every $\overset{3_i}{\sim}$-class C $(0 \leq i \leq k)$ can be uniquely associated with the 2-block T_C in the sequence $\mathcal{T}_i$ such that the vertices in C are exactly those that belong to T_C. It is easy to see that for these C and T_C, $\text{bal}(C) = \text{bal}(T_C)$.

Definition 4.1.23 For each $\overset{3_i}{\sim}$-class C $(0 \leq i \leq k)$, we define $\text{width}(C) = \text{width}(T_C)$, where T_C is as in Remark 4.1.22.

4.1.2 Ordering 3_i-Equivalence Classes

We consider the case of $\overset{3_1}{\sim}$-classes first.

Let $\hat{C} = \langle C_1, \ldots, C_s \rangle$ be the sequence of $\overset{3_1}{\sim}$-classes taken in the order their vertices appear in the decomposition B. Let R denote the set $\{C_1, \ldots, C_s\}$, and let $\bar{R} = R \setminus \{C_1\}$.

We associate with each C_i, $1 \leq i \leq s$, the number $b(C_i) = \mathrm{bal}(C_1) + \cdots + \mathrm{bal}(C_i)$. By the definition of 3_1-equivalence, $\mathrm{bal}(C_j) \leq 0$ for each $j \in \{2, \ldots, s\}$, therefore the sequence $\langle b_1, \ldots, b_s \rangle$ is monotonically non-increasing. Thus, for any two $\overset{3_1}{\sim}$-classes $C, C' \in \bar{R}$, if $b(C) > b(C')$ or if $b(C) = b(C')$ and $\mathrm{bal}(C) < 0$, then C precedes C' in the sequence $\hat{C}$.

Remark 4.1.24 We can extend the definition of balance to any subset U of V (by extending the corresponding definitions of drop, non-drop, and removable vertices for 2-equivalence classes). Then we will have that $b(C_i) = \mathrm{bal}(C_1 \cup \ldots \cup C_i)$.

Note that the following holds for any $C \in R$:

$$b(C) = \sum_{C' : b(C') \geq b(C)} \mathrm{bal}(C') = \mathrm{bal}(\cup_{C' : b(C') \geq b(C)} C').$$

Lemma 4.1.25 *Let $C, C' \in \bar{R}$ be such that C immediately precedes C' in $\hat{C}$, and C and C' are incomparable with respect to the corresponding extension of $\preceq$. If $\mathrm{bal}(C) = \mathrm{bal}(C') = 0$, then C and C' can be interchanged in $\hat{C}$ with the resulting sequence $\bar{C}$ corresponding to some nice k-decomposition of G.*

Proof. Let us add the vertices in C' (in the same order as before) instead of those in C, and the vertices of C instead of those in C'. Since $\mathrm{bal}(C) = \mathrm{bal}(C') = 0$, we will not increase the path-width of the resulting decomposition. □

Remark 4.1.26 It should also be noted that for the $\hat{C}$ and $\bar{C}$ as in Lemma 4.1.25,

$$\mathrm{width}(\hat{C}) = \mathrm{width}(\bar{C}).$$

Definition 4.1.27 We extend the relation of strong precedence to the set R, which will be called *strong 3_1-precedence* and denoted by $\overset{s3_1}{\preceq}$, as follows: For any two distinct $\overset{3_1}{\sim}$-classes $C, C' \in R$, $C \overset{s3_1}{\prec} C'$ iff

i. $u \overset{s}{\prec} v$ for some $u \in C$ and some $v \in C'$,

ii. $b(C) > b(C')$, or

iii. $b(C) = b(C')$ and $\mathrm{bal}(C) \neq 0$.

The 3_1*-precedence*, denoted by $\stackrel{3_1}{\preceq}$, is defined as the reflexive and transitive closure of $\stackrel{s3_1}{\preceq}$.

Remark 4.1.28 If some $\stackrel{3_1}{\sim}$-classes C and C' in R are incomparable with respect to $\stackrel{3_1}{\preceq}$, then $b(C) = b(C')$ and $\mathrm{bal}(C) = \mathrm{bal}(C') = 0$.

We apply similar reasoning to each $\stackrel{3_1}{\sim}$-class C_j $(1 < j \leq s)$ considered as a sequence of $\stackrel{3_2}{\sim}$-classes, and so on. This will give us $k+1$ partial orders $\stackrel{3_i}{\preceq}$, $0 \leq i \leq k$, on $\stackrel{3_i}{\sim}$-classes. (Note that analogues of Lemma 4.1.25, Remark 4.1.26, and Remark 4.1.28 hold for each $\stackrel{3_i}{\sim}$-class, $1 \leq i \leq k$.)

Lemma 4.1.29 *The triple* $(\stackrel{1}{\sim}, \{\stackrel{3_i}{\sim}\}_{i=0}, \{\stackrel{3_i}{\preceq}\}_{i=0}^{k})$ *is a partial k^k-generative structure on G.*

Proof. Let $\{\leq^i\}_{i=0}^k$ be an arbitrary family of completions of $\stackrel{3_i}{\preceq}$, $0 \leq i \leq k$, to linear orders such that $\leq^j$ is a refinement of $\leq^i$ for every $0 \leq i < j \leq k$.

Consider a sequence of $\stackrel{3_k}{\sim}$-classes (within some $\stackrel{3_{k-1}}{\sim}$-class C^{k-1}) linearly ordered by $\leq^k$. That sequence can be obtained from the original sequence of these $\stackrel{3_k}{\sim}$-classes (i.e., the one induced by the decomposition B) after a finite number of interchanges of consecutive $\stackrel{3_k}{\sim}$-classes incomparable with respect to $\stackrel{3_k}{\preceq}$. By Remark 4.1.28 (formulated for $\stackrel{3_k}{\sim}$), any two such $\stackrel{3_k}{\sim}$-classes C and C' have balance zero, and therefore, interchanging them will not change the width of C^{k-1}.

Now let us consider the sequence of $\stackrel{3_{k-1}}{\sim}$-classes (within some $\stackrel{3_{k-2}}{\sim}$-class C^{k-2}) induced by B. If we order the $\stackrel{3_k}{\sim}$-classes within each $\stackrel{3_{k-1}}{\sim}$-class of C^{k-2} according to $\leq^k$, the width of C^{k-2} will remain the same, because, as shown above, the width of each such $\stackrel{3_{k-1}}{\sim}$-class is not changed. Repeating the above arguments, one can show that C^{k-2} ordered by $\leq^{k-1}$ with each $\stackrel{3_{k-1}}{\sim}$-class in C^{k-2} ordered by $\leq^k$ (i.e., C^{k-2} ordered by $\leq^k$) has the same width as originally.

Continuing in this manner, we can show that the width of the $\stackrel{3_0}{\sim}$-class C^0 ordered by $\leq^k$ has the same width as that for the ordering induced by B, which means that $\leq^k$ yields a linear k-generative structure on G. ■

4.2 MS-Colouring a Partial k^k-Generative Structure

Here we prove that the partial k^k-generative structure defined in the previous section is MS-colourable in terms of G. We will show that there always exists some nice k-decomposition of a (k,p)-path G for which $\stackrel{1}{\sim}$, $\stackrel{2}{\sim}$, and $\stackrel{3_i}{\sim}$ and $\stackrel{3_i}{\preceq}$ $(0 \leq i \leq k)$ are MS-definable over a suitably coloured graph G^c.

4.2.1 MS-Colouring $\stackrel{1}{\sim}$ and $\stackrel{2}{\sim}$

In this subsection, we prove that for any nice k-decomposition $B = \langle B_1, \ldots, B_m \rangle$ of a partial k-path G, the equivalences $\stackrel{1}{\sim}$ and $\stackrel{2}{\sim}$ induced by B are MS-colourable. (As before, we also view B as a sequence of monotonic pieces $M_1, \ldots, M_d$.)

Theorem 4.2.1 *The 1-equivalence is MS-colourable over G.*

Proof. Since B_1 is MS-colourable, so is $\text{new}(B_1) = B_1$. For every bag B_r, $1 < r \leq m$, the following is true:

i. $\text{drop}(B_r) \cap \text{old}(B_r) \neq \emptyset$, or

ii. $\text{drop}(B_r) \cap \text{old}(B_r) = \emptyset$, and therefore, $\text{drop}(B_r) \cap \text{new}(B_r) \neq \emptyset$.

By Remark 2.1.28, every new vertex of B_r has an outgoing arrow to each vertex in $\text{drop}(B_r) \cap \text{old}(B_r)$ (case (i)), or there is some $v \in \text{drop}(B_r) \cap \text{new}(B_r)$ such that every vertex in $\text{new}(B_r) \setminus \{v\}$ has an outgoing arrow to v (case (ii)).

In case (i), we chose some vertex $u \in \text{drop}(B_r) \cap \text{old}(B_r)$ and colour each arc $w \to u$, $w \in \text{new}(B_r)$, with some colour c_1. In case (ii), we colour each arc $w \to v$, $w \in \text{new}(B_r) \setminus \{v\}$, with some colour c_2.

We do such colouring for each B_r, $1 < r \leq m$. (The same colours c_1 and c_2 can be used for the corresponding arcs in all the B_r, $1 < r \leq m$, since these coloured arcs go into drop vertices, and therefore, no two vertices from different sets $\text{new}(B_r)$ and $\text{new}(B_{r'})$, $r \neq r'$, can have the coloured arcs going into the same vertex of V.)

We say that two vertices v and v' in V satisfy a relation R_1 iff either there is some $u \in V$ such that $(v,u), (v',u) \in E_{c_1}$ (case i) or $(v,v') \in E_{c_2}$ (case ii).

By its definition, the relation R_1 is MS-colourable. Therefore, the 1-equivalence (which is the reflexive and transitive closure of R_1) is also MS-colourable (see Fact 2.2.9). □

To show the MS-colourability of the 2-equivalence, we need the following two lemmas.

Lemma 4.2.2 *For every new vertex v of a monotonic piece $M_s = \langle B_{i_s}, \ldots, B_{j_s} \rangle$ $(1 \leq s \leq d)$, there is some vertex $u \in new(B_{i_s})$ such that $u \preceq v$.*

Proof. If $v \in \text{new}(B_{i_s})$, then $v \preceq v$. Otherwise, let $v \in \text{new}(M_s)$ be the first vertex that is not preceded by any vertex in $\text{new}(B_{i_s})$, i.e., v is such that $\text{first}(v) = \min_{v' \in \text{new}(M_s)} \{\text{first}(v') | u \not\preceq v'$ for any vertex $u \in \text{new}(B_{i_s})\}$.

By the definition of a nice decomposition, v should have been added to the decomposition B before the vertices in $\text{new}(B_{i_s})$ (see Remark 4.0.1 and condition (iv.a) of Definition 2.1.19). This contradiction proves the claim. □

Lemma 4.2.3 *For any $u, v \in new(M_s)$ $(1 \leq s \leq d)$, $u \preceq v$ iff there are $u_1, \ldots, u_l \in new(M_s)$ such that $u \overset{s}{\prec} u_1 \overset{s}{\prec} \cdots \overset{s}{\prec} u_l \overset{s}{\prec} v$.*

Proof. The only thing to be proved here is that $u_1, \ldots, u_l \in \text{new}(M_s)$, but this easily follows from the fact that $u, v \in \text{new}(M_s)$. □

These two lemmas imply the following.

Theorem 4.2.4 *The 2-equivalence is MS-colourable over G.*

Proof. We have by Lemmas 4.2.2 and 4.2.3 that for every $v \in \text{new}(M_s)$ $(1 \leq s \leq d)$, there is $u \in \text{new}(B_{i_s})$ and there are $u_1, \ldots, u_l \in \text{new}(M_s)$ such that $u = u_0 \overset{s}{\prec} u_1 \overset{s}{\prec} \cdots \overset{s}{\prec} u_l \overset{s}{\prec} u_{l+1} = v$.

By the definition of the strong precedence $\overset{s}{\prec}$, the following holds for each u_i and u_{i+1} $(0 \leq i \leq l)$:

i. either the arc $u_{i+1} - u_i$ is in E^d, or

ii. there is some $w \in V$ such that the arcs $u_i \rightarrow w$ and $u_{i+1} \Rightarrow w$ are in E^d.

In case (i), we colour the arc $u_{i+1} - u_i$ with some colour c_3. In case (ii), we colour both $u_i \rightarrow w$ and $u_{i+1} \Rightarrow w$ with some colour c_4.

We do such colouring for each M_s, $1 \leq s \leq d$. (It is not difficult to see that we can use the same colours c_3 and c_4 for all the M_s, $1 \leq s \leq d$.)

We say that two vertices v and v' satisfy a relation R_2 iff $v \overset{1}{\sim} v'$, $(v, v') \in E_{c_3}$ (case i), or there is some $w \in V$ such that $(v, w), (v', w) \in E_{c_4}$ (case ii).

Then the 2-equivalence is the reflexive and transitive closure of the MS-colourable relation R_2, and therefore is also MS-colourable. □

4.2.2 MS-Colouring $\overset{3_i}{\sim}$ and $\overset{3_i}{\preceq}$

Here we show that for each partial k-path G, there always exists a nice k-decomposition which induces MS-colourable $\overset{3_i}{\sim}$ and $\overset{3_i}{\preceq}$, $0 \leq i \leq k$.

Definition 4.2.5 A 2-block T is called *locally connected* iff for every $\overset{2}{\sim}$-classes t and t' in T, there are $\overset{2}{\sim}$-classes $t_1, \ldots, t_l$ in T such that every two consecutive elements in the sequence $\langle t = t_0, t_1, \ldots, t_l, t_{l+1} = t' \rangle$ are comparable with respect to $\overset{s2}{\prec}$, i.e., for every $i \in \{0, \ldots, l\}$, $t_i \overset{s2}{\prec} t_{i+1}$ or $t_{i+1} \overset{s2}{\prec} t_i$.

Lemma 4.2.6 *The equivalence relation on V induced by a partitioning of V into sequence of locally connected 2-blocks is MS-colourable over G.*

Proof. The reasoning is similar to that in the proof of Theorem 4.2.4. □

To achieve the local connectedness of 2-blocks in the sequences $\mathcal{T}_i$, $0 \leq i \leq k$, generated by algorithm $\mathcal{P}$, we re-order the sequence S of $\overset{2}{\sim}$-classes induced by the decomposition B of G. The new sequence S', however, will correspond to some nice k-decomposition B' of G such that $G^d_{B'}$ is isomorphic to G^d_B.

Lemma 4.2.7 *Let $S = \langle S_1, \ldots, S_i, S_{i+1}, \ldots, S_d \rangle$ (for some $1 \leq i < d$) be the sequence of $\overset{2}{\sim}$-classes induced by the k-decomposition B of G. If $\mathrm{bal}(S_i) \geq 0$, $\mathrm{bal}(S_{i+1}) < 0$, and $S_i \overset{2}{\not\prec} S_{i+1}$, then the sequence $S' = \langle S_1, \ldots, S_{i+1}, S_i, \ldots, S_d \rangle$ is induced by some nice k-decomposition B' of G such that $G^d_{B'} \cong G^d_B$.*

Proof. Since $S_i \overset{2}{\not\prec} S_{i+1}$, we can add the class S_{i+1} to the decomposition B before S_i. (We do so by adding the vertices of S_{i+1} in the same order as they were added to B before.) Let B'^{i+1} denote the sequence of bags corresponding to the sequence $\langle S_1, \ldots, S_{i-1}, S_{i+1} \rangle$. The path-width of B'^{i+1} is at most k because $\mathrm{bal}(S_i) \geq 0$.

Since $\mathrm{bal}(S_{i+1}) < 0$, the class S_i can "fit in" after S_{i+1}, and thus B'^{i+1} can be completed to some k-decomposition B' of G.

For any $\overset{2}{\sim}$-class S_i with non-positive balance that immediately follows S_1, $S_1 \overset{2}{\prec} S_i$. (Indeed, that S_i is of non-positive balance means that either all of its vertices will be dropped before the next $\overset{2}{\sim}$-class is added (i.e., $\mathrm{drop}(S_i) = S_i$) or some of the old vertices are removed by S_i (i.e., $\mathrm{remov}(S_i) \neq \emptyset$). In the first case, since the graph G is connected, there must be a vertex in S_i that is adjacent to some vertex in S_1. In the second case, there is a vertex in S_1 adjacent to some vertex in S_i by the definition of vertices removable by a $\overset{2}{\sim}$-class.)

Thus, the first bag of B' is B_1. It is easy to see that B' induces the digraph isomorphic to G_B^d, since the vertices within each S_i, $1 \leq i \leq d$, are added to the decomposition B' in the same order as they were added to B. □

Consider the following transformation of the sequence S:

Transformation $\mathcal{S}$

Step 1. If length(S) = 1, stop. Otherwise interchange every two consecutive elements S_i and S_{i+1} of S such that bal$(S_i) \geq 0$, bal$(S_{i+1}) < 0$, and $S_i \overset{2}{\not\prec} S_{i+1}$.

Step 2. If two consecutive elements S_i and S_{i+1} of S $(1 \leq i < d)$ are such that bal$(S_i) \geq 0$ and bal$(S_{i+1}) < 0$, but $S_i \overset{2}{\prec} S_{i+1}$, "merge" them into one so-called S-*block*. (The balance of an S-block is defined as the sum of the balances of its components, and two S-blocks are comparable with respect to $\overset{2}{\prec}$ iff so are some of their components.) Replace S with the sequence of S-blocks and go to Step 1.

In view of Lemma 4.2.7, it is easy to see that the sequence of $\overset{2}{\sim}$-classes $S' = \langle S'_1, \ldots, S'_d \rangle$ resulting from transformation $\mathcal{S}$ is induced by some nice k-decomposition B' of G such that $G_{B'}^d$ is isomorphic to G_B^d.

Remark 4.2.8 By definition, each S-block constructed in the course of transformation $\mathcal{S}$ is a locally connected 2-block.

Remark 4.2.9 Since G is connected and bal$(S) = 0$, one can prove that transformation $\mathcal{S}$ merges the sequence S into a single S-block.

Now we apply algorithm $\mathcal{P}$ to the sequence S' to get a new family of 3_i-equivalences, $0 \leq i \leq k$. As the following lemma shows, these 3_i-equivalences are MS-colourable.

Lemma 4.2.10 *Let $\{T_i\}_{i=0}^k$ be the family of sequences of 2-blocks generated by algorithm $\mathcal{P}$ for the sequence S'. For any sequence $T_i = \langle T_1^i, \ldots, T_{t_i}^i \rangle$ $(0 \leq i \leq k)$, each 2-block T_j^i $(1 < j \leq t_i)$ is locally connected.*

Proof. By the definition of our transformation $\mathcal{S}$ of the sequence S, every balanced 2-block T_j^i $(1 < j \leq t_i)$ is a single S-block at some step of transformation $\mathcal{S}$.

Indeed, by the definition of a balanced 2-block, each such T_j^i is a sequence with an S-block of positive balance at the beginning and with an S-block of negative balance at the end for each step of transformation $\mathcal{S}$. (Otherwise, we would have a proper prefix with non-positive balance.) Therefore, $\mathcal{S}$ merges all the S-blocks inside T_j^i into a single S-block. Our claim now follows from Remark 4.2.8. □

Now we can state the following lemma.

Lemma 4.2.11 *There always exists a nice k-decomposition B of a partial k-path G such that 3_i-equivalences, $0 \le i \le k$, induced by B are MS-colourable over G.*

Lemma 4.2.12 *Each partial order $\overset{3_i}{\preceq}$, $0 \le i \le k$, is MS-colourable over G.*

Proof. One just needs to colour each 3_i-equivalence class C with $b(C)$ and $\mathrm{bal}(C)$ (which are numbers bounded by k). □

Thus we have proved the following.

Theorem 4.2.13 *The partial k^k-generative structure $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i}{\preceq}\}_{i=0}^{k})$ is MS-colourable over G.*

4.3 A CMS-Formula

Here we will prove that recognizability implies definability for partial k-paths G. First, we define another MS-colourable partial k^k-generative structure on G. Using this structure we will be able to define the required CMS-formula for suitably coloured connected partial k-paths. Then we formulate the MS-definable admissibility conditions on colourings of partial k-paths. Finally, we solve the problem of recognizability implying definability for disconnected partial k-paths.

4.3.1 Another Partial k^k-Generative Structure

In this subsection, we divide the set of 3_i-equivalence classes (for each $i \in \{1, \dots, k\}$) into k sets of nodes and one set of leaves by analogy with the case of $(k, 1)$-paths.

We consider the set of $\overset{3_1}{\sim}$-equivalence classes first. Let $\langle C_1, \dots, C_s\rangle$ be the sequence of $\overset{3_1}{\sim}$-equivalence classes induced by the decomposition B of G. (As before, we denote by R the set $\{C_1, \dots, C_s\}$, and by $\bar{R}$ the set $R \setminus \{C_1\}$.)

Notation 4.3.1 For a $\overset{3_1}{\sim}$-class C_l $(1 \leq l \leq s)$, we denote by $B(C_l) = \langle B_{i_{C_l}}, \ldots, B_{j_{C_l}} \rangle$ the subsequence of B such that new$(B(C_l)) = C_l$ (where new$(B(C_l))$ = new$(B_{i_{C_l}}) \cup \ldots \cup$ new$(B_{j_{C_l}})$).

Since every $\overset{3_1}{\sim}$-class C_l $(1 < l \leq s)$ has a non-positive balance, for every t and t' such that $1 \leq t < t' \leq s$, we have

$$|\text{non-drop}(B_{j_{C_{t'}}})| \leq |\text{non-drop}(B_{j_{C_t}})| \leq |\text{non-drop}(B_{j_{C_1}})|.$$

Let us define a labeling $\lambda_1 : V \to \{1, \ldots, k\}$ such that no two vertices in non-drop$(B_{j_{C_l}})$ $(1 \leq l \leq s)$ have the same label and for any $1 < t \leq s$, $\lambda_1(B_{j_{C_t}}) \subseteq \lambda_1(B_{j_{C_{t-1}}})$ (for the corresponding extension of λ_1 to sets of vertices).

Because non-drop$(C_l) \subseteq$ non-drop$(B_{j_{C_l}})$ $(1 \leq l \leq s)$, each non-drop vertex v of C_t, $1 < t \leq s$, can be thought of as the one "replacing" the non-drop vertex u of $B_{j_{C_{t-1}}}$ such that $\lambda_1(v) = \lambda_1(u)$.

We associate with each $\overset{3_1}{\sim}$-class C_l $(1 < l \leq s)$ the sets in$(C_l) = \lambda_1(\text{remov}(C_l))$ and out$(C_l) = \lambda_1(\text{non-drop}(C_l))$.

Remark 4.3.2 By definition, out$(C_l) \subseteq$ in(C_l) for every $l \in \{2, \ldots, s\}$.

We define the following k sets $P_1^R, \ldots, P_k^R$ of $\overset{3_1}{\sim}$-classes in R: For any $C \in \bar{R}$, $C \in P_j^R$ $(1 \leq j \leq k)$ iff $j = \min\{j' | j' \in \text{in}(C)\}$.

Remark 4.3.3 It is not difficult to see that each set P_j^R, $1 \leq j \leq k$, is linearly ordered by $\overset{3_1}{\preceq}$.

Definition 4.3.4 The $\overset{3_1}{\sim}$-classes in $\cup_{j=1}^{k} P_j^T$ are called the *nodes of* R. The set of all nodes of R is denoted by N_R.

Definition 4.3.5 The set $\bar{R} \setminus N_R$ is called the set of *leaves* of R and is denoted by L_R.

Remark 4.3.6 For each leaf C of R, remov$(C) = \emptyset$ and non-drop$(C) = \emptyset$.

Now we can extend the partial order $\overset{3_1}{\preceq}$ as in the case of $(k, 1)$-paths so that only certain leaves are incomparable. We will denote this new partial order by $\overset{3_1{}^n}{\preceq}$.

We do the same for each 3_i-equivalence class, $1 \leq i \leq k$, which will give us k partial orders $\overset{3_i{}^n}{\preceq}$.

Clearly, the triple $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i^n}{\preceq}\}_{i=0}^{k})$ is a partial k^k-generative structure on G. It is not difficult to see that this structure is MS-colourable over G. Let $G^{dk'}$ denote the corresponding coloured digraph on which this structure is MS-definable.

4.3.2 Constructing a Decomposition of G

Here we describe an algorithm that constructs some nice k-decomposition of G and checks if this decomposition is accepted by the automaton A.

Let $\hat{B}$ be some nice k-decomposition of G generated by the structure $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i^n}{\preceq}\}_{i=0}^{k})$ defined in the previous subsection.

Notation 4.3.7 For a $\overset{3_i}{\sim}$-class C $(0 \leq i \leq k)$, we denote by $\hat{B}(C)$ the subsequence of bags in $\hat{B}$ such that $\text{new}(\hat{B}(C)) = C$.

For any $\overset{3_1}{\sim}$-class C, the old vertices of the first bag in $\hat{B}(C)$ (the set of such vertices will be denoted by $\text{old}(\hat{B}(C))$) are some non-drop vertices of the $\overset{3_1}{\sim}$-classes coming before C in $\hat{B}$. The $\overset{3_1}{\sim}$-classes having non-drop vertices are nodes (by definition), and since $\overset{3_1^n}{\preceq}$ is a linear order on the set of $\overset{3_1}{\sim}$-classes that are nodes, we have that

$$\text{old}(\hat{B}(C)) \subseteq \bigcup_{C':C' \overset{3_1^n}{\preceq} C} \text{non-drop}(C').$$

More specifically,

$$\text{old}(\hat{B}(C)) = \text{non-drop}(\bigcup_{C':C' \overset{3_1^n}{\preceq} C} C') = \text{non-drop}(\{u \in V | [u]_{\overset{3_1}{\sim}} \overset{3_1^n}{\preceq} C\}).$$

Let C be an arbitrary $\overset{3_i}{\sim}$-class $(1 \leq i \leq k)$ and let v be an arbitrary vertex in C (i.e., $C = [v]_{\overset{3_i}{\sim}}$). Then

$$\text{old}(\hat{B}(C)) = \text{non-drop}(\{u \in V | [u]_{\overset{3_i}{\sim}} \overset{3_i^n}{\preceq} [v]_{\overset{3_i}{\sim}} \vee \ldots \vee [u]_{\overset{3_1}{\sim}} \overset{3_1^n}{\preceq} [v]_{\overset{3_1}{\sim}}).$$

In other words, the set of vertices appearing in $\hat{B}(C)$ (for any $\overset{3_i}{\sim}$-class C, $1 \leq i \leq k$) is determined uniquely by the structure $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i^n}{\preceq}\}_{i=0}^{k})$.

Notation 4.3.8 Since for any other decomposition $\hat{B}'$ generated by $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i^n}{\preceq}\}_{i=0}^{k})$, $\text{old}(\hat{B}'(C)) = \text{old}(\hat{B}(C))$ (for every $\overset{3_i}{\sim}$-class C, $1 \leq i \leq k$), we will denote this set of vertices simply by $\text{old}(C)$.

Remark 4.3.9 For any two $\overset{3_i}{\sim}$-classes C and C' $(1 \leq i \leq k)$ that are incomparable (with respect to $\overset{3_i^n}{\preceq}$) leaves in some $\overset{3_{i-1}}{\sim}$-class, $\hat{B}(C)$ and $\hat{B}(C')$ can be interchanged in $\hat{B}$, with the resulting sequence still being a decomposition of G.

Our discussion suggests the following non-deterministic algorithm for constructing some decomposition of G generated by $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^{k}, \{\overset{3_i^n}{\preceq}\}_{i=0}^{k})$:

Algorithm $\mathcal{R}$

For each $\overset{3_k}{\sim}$-class C (which is a monotonic piece), construct the sequence of bags as in the case of $(k,1)$-paths, taking the set $\text{old}(C) \cup \text{new}^1(C)$ as the first bag of that sequence. (Here $\text{new}^1(C)$ denotes the unique $\overset{1}{\sim}$-class in C of cardinality greater than one.)

For each $\overset{3_{k-1}}{\sim}$-class C', order the sequences of bags constructed for its $\overset{3_k}{\sim}$-classes according to an arbitrary completion of $\overset{3_k^n}{\preceq}$ to a linear order on $C'/\overset{3_k}{\sim}$.

Continuing in this way, we will get a sequence of bags for the $\overset{3_0}{\sim}$-class, which is a decomposition of G.

Note that the non-determinism of algorithm $\mathcal{R}$ stems from the fact that there can be $\overset{3_i}{\sim}$-classes $(1 \leq i \leq k)$ contained in the same $\overset{3_{i-1}}{\sim}$-class which are incomparable with respect to $\overset{3_i^n}{\preceq}$. (Obviously, they are leaves in this case.) To check in CMS whether an automaton $A = (\Sigma_g, Q, \delta, q_0, F)$ recognizes the partial k-path G, it would suffice to define a certain linear order on such incomparable leaves that is expressible in CMS.

Since there can be an arbitrary large number of incomparable leaves, we should define a certain equivalence relation on leaves so that the number of equivalence classes is bounded by a constant, and the behaviour of our automaton is the same on every two equivalent leaves. (Note that we should define such an equivalence for every $\overset{3_i}{\sim}$, $0 < i \leq k$, thus we will have k equivalence relations.)

In the case of $(k,1)$-paths we defined p-equivalence on leaves and then ordered the $\overset{p}{\sim}$-classes. This was sufficient for defining the required CMS formula, because any two p-equivalent leaves w' and w'' corresponded to the bags B' and B'' such that $\sigma_\beta(B') = \sigma_\beta(B'')$ (for a suitable labeling function β).

Since leaves were single vertices in the case of $(k,1)$-paths, we could easily determine if two leaves corresponded to identical symbols in Σ_g by just looking at the set of vertices (nodes) to which these leaves had outgoing arrows. In the general case, however, leaves are not necessarily single vertices, and therefore, they can correspond to (arbitrary long) *sequences* of bags (strings of symbols in the alphabet Σ_g).

Calling two incomparable leaves equivalent if the corresponding strings over Σ_g are identical will give us an unbounded number of such equivalence classes. Instead, we should call two such leaves δ-equivalent if the corresponding strings over Σ_g are equivalent with respect to the transition function of the automaton A, i.e., if these strings ω_1 and ω_2 are such that for each $q \in Q$, $\delta^*(q, \omega_1) = \delta^*(q, \omega_2)$. (Clearly, the number of thus defined δ-equivalence classes is bounded by a function of $|Q|$.)

To determine if two leaves are δ-equivalent, one needs to construct the sequences of bags corresponding to those leaves (more precisely, to know the behaviour of the automaton A on those sequences), which is a problem similar to the original one of constructing a decomposition of G.

As indicated by algorithm $\mathcal{R}$, we can construct sequences of bags corresponding to each $\overset{3_k}{\sim}$-class. This will give us all the bags in the decomposition of G generated by $\mathcal{R}$. (Then we just need to put those bags in a proper order.) Thus, we can check if a labeling function $\beta : V \to \{1, \ldots, k+1\}$ is admissible by the decomposition constructed by algorithm $\mathcal{R}$.

Consider the following extension of algorithm $\mathcal{R}$. (We assume that $Q = \{q_1, \ldots, q_z\}$.)

Algorithm $\mathcal{R}'$

For each $\overset{3_k}{\sim}$-class C, construct the corresponding sequence of bags as before. Convert that sequence to a word over Σ_g, using the labeling function β. (Let ω denote that word.) Associate with C the vector $q(C) = (q'_1, \ldots, q'_z)$, where for each $i \in \{1, \ldots, z\}$, $q'_i = \delta^*(q_i, \omega)$. (We also consider this vector as a map from Q to Q.)

For each $\overset{3_{k-1}}{\sim}$-class C', order the sequences of bags constructed for its $\overset{3_k}{\sim}$-classes according to the completion of $\overset{3_k{}^n}{\preceq}$ to a linear order on $C'/\overset{3_k}{\sim}$ such that for any two incomparable (with respect to $\overset{3_k{}^n}{\preceq}$) leaves C and C', C is less than C' iff $q(C)$ is lexicographically less than $q(C')$. Let $\langle C_1, \ldots, C_r \rangle$ denote this sequence. Associate with C' the vector $q(C') = q(C_1) \circ \cdots \circ q(C_r)$, where $q(C_i)$, $1 \le i \le r$, are considered as maps, and $\circ$ denotes the composition.

Continuing in this way, we will get a decomposition of G as well as a map from Q to Q that defines the behaviour of A on that decomposition. To see if G is recognized by A, we just have to check whether q_0 is taken to some final state by the constructed map.

4.3.3 CMS-Definability of Recognizability for Coloured Connected Partial k-Paths

Here we show that it can be checked in CMS whether the labelings of 3_i-equivalence classes C $(0 \leq i \leq k)$ with some vectors $q(C)$ are the ones that would be produced by algorithm $\mathcal{R}$.

It is not difficult to see that the sets old(C) and new$^1(C)$ (defined in the previous subsection) are MS-definable over the coloured digraph $G^{dk'}$ for every $\overset{3_k}{\sim}$-class C. Thus, we can proceed as in the case of $(k,1)$-paths by MS-defining (over $G^{dk'}$) the p-equivalence on $C \setminus \text{new}^1(C)$, the linear order on the corresponding quotient set, and the bags induced by that linear ordering.

We can also verify in MS that a labeling β of our graph G is admissible by the sequence of bags constructed for each $\overset{3_k}{\sim}$-class C. (Let us denote this sequence of bags by $B(C)$.) This will ensure the admissibility of β by the decomposition generated with algorithm $\mathcal{R}'$.

We MS-define in terms of $G^{dk'}$ the word $\omega(C) = \sigma_\beta(B(C))$. Then we check the correctness of the vector $q(C) = (q'_1, \ldots, q'_z)$ associated with every C by guessing z colourings of the symbols in $\omega(C)$ with the states of A such that q_i is considered the initial state and q'_i the final state of A, $1 \leq i \leq z$, and then verifying for every two consecutive symbols of $\omega(C)$ that their labels agree with the transition function of A (see the case of $(k,1)$-paths).

For every $\overset{3_{k-1}}{\sim}$-class C', we define the δ_k-equivalence relation on the set of its $\overset{3_k}{\sim}$-classes by saying that two leaves C_1 and C_2 are δ_k-equivalent iff $q(C_1) = q(C_2)$. This enables us to MS-define the completion of $\overset{3_k}{\preceq}^{n}$ to the linear order on the set $\tilde{C}' = (C'/\overset{3_k}{\sim})/\overset{\delta_k}{\sim}$ as described in algorithm $\mathcal{R}'$.

To check the correctness of $q(C')$, we guess a colouring of the elements C of $\tilde{C}'$ with vectors $q'(C)$ from Q^z such that the first (with respect to the above-mentioned linear order) element C_1 of $\tilde{C}'$ is labeled with $q(C_1)$ and the last element C_r with $q(C')$, and then verify for every two consecutive elements C_j and C_{j+1} of $\tilde{C}'$ $(1 \leq j < r)$ that

$$q'(C_{j+1}) = q'(C_j) \circ \underbrace{q(C_{j+1}) \circ \cdots \circ q(C_{j+1})}_{t},$$

where t is the cardinality of the $\overset{\delta_k}{\sim}$-class C_{j+1} and $\circ$ is the composition of the maps from Q to Q. It should be clear that this is expressible in CMS.

We continue in this manner until we verify the correctness of the vector $q(C^0)$ associated with the $\overset{3_0}{\sim}$-class C^0. The graph G is recognized by A iff the state q_0 is mapped by $q(C^0)$

to some final state.

Thus we have proved the following statement.

Theorem 4.3.10 *Every recognizable family of coloured connected partial k-paths $G^{dk'}$ is CMS-definable.*

4.3.4 Admissibility Conditions

Here we state the MS-definable conditions on colourings of partial connected k-paths G such that any colouring satisfying these conditions induces the partial k^k-generative structure on G required by algorithm $\mathcal{R}'$.

Consider the following conditions on a colouring of G:

i. This colouring induces a labeled digraph $G' = (V, E')$ (which has the format of G^d).

ii. The relation $\preceq$ (precedence) induced by G' is a partial order on V.

iii. The relations $\overset{1}{\sim}$, $\overset{2}{\sim}$, and $\overset{3_i}{\sim}$ $(0 \leq i \leq k)$ induced by that colouring are equivalences on V such that

  (a) $\overset{1}{\sim}$ is a refinement of $\overset{2}{\sim}$,

  (b) every two vertices of V are 3_0-equivalent,

  (c) $\overset{3_{i+1}}{\sim}$ is a refinement of $\overset{3_i}{\sim}$, $0 \leq i < k$, and

  (d) $\overset{3_k}{\sim}$ coincides with $\overset{2}{\sim}$.

iv. For any $u \in V$, $[u]_{\overset{2}{\sim}}$ contains exactly one $\overset{1}{\sim}$-class of cardinality greater than one. (Below we denote such $[u]_{\overset{2}{\sim}}$ and $\overset{1}{\sim}$-class of cardinality greater than one contained in $[u]_{\overset{2}{\sim}}$ by $\mathrm{new}(M_u)$ and $\mathrm{new}^1(M_u)$, respectively.) The sets $\mathrm{new}(M_u)$ and $\mathrm{new}^1(M_u)$ satisfy the following conditions:

  (a) every $\mathrm{new}^1(M_u)$ either contains a drop vertex or has a unique vertex with an outgoing double arrow to some vertex not in $\mathrm{new}^1(M_u)$,

  (b) there is no arc $(w, w') \in E'$ such that $w \in \mathrm{new}^1(M_u)$ and $w' \in \mathrm{new}(M_u) \setminus \mathrm{new}^1(M_u)$,

  (c) for every $w \in \mathrm{new}(M_u) \setminus \mathrm{drop}(\mathrm{new}^1(M_u))$ with incoming arrow (or arrows), there is exactly one incoming double arrow,

(d) every vertex with a loop arrow has no outgoing double arrows and no incoming arrows of any kind,

(e) every vertex in $\mathrm{new}(M_u) \setminus \mathrm{new}^1(M_u)$ has either an outgoing double arrow or a loop arrow.

(Compare the above conditions with condition (iii) for the case of $(k,1)$-paths on page 35.)

v. The balances of 3_i-equivalence classes $(0 \le i \le k)$ are encoded correctly by the colouring (i.e., $\mathrm{bal}(C) = |\text{non-drop}(C)| - |\mathrm{remov}(C)|$ for each $\overset{3_i}{\sim}$-class C).

vi. Each $\overset{3_i}{\sim}$-class $(0 \le i < k)$ contains at most one $\overset{3_{i+1}}{\sim}$-class with positive balance.

vii. For each $\overset{3_i}{\sim}$-class $\hat{C}$, $0 \le i < k$, the following two conditions hold (we denote by $\tilde{C}$ the quotient set $\hat{C}/\overset{3_{i+1}}{\sim}$):

(a) For any two $\overset{3_{i+1}}{\sim}$-classes C and C' in $\tilde{C}$ such that $\mathrm{bal}(C) \ne 0$ and $\mathrm{bal}(C') \ne 0$, $b(C) \ne b(C')$.

(b) For each $\overset{3_{i+1}}{\sim}$-class C in $\tilde{C}$,

$$b(C) = \mathrm{bal}(\cup_{C' \in \tilde{C} : b(C') \ge b(C)} C').$$

viii. Each relation $\overset{3_i}{\preceq}$, $1 \le i \le k$, induced by the colouring is a partial order on $C/\overset{3_i}{\sim}$ for every $\overset{3_{i-1}}{\sim}$-class C such that no two $\overset{3_i}{\sim}$-classes contained in different $\overset{3_{i-1}}{\sim}$-classes are comparable w.r.t. $\overset{3_i}{\preceq}$, and the partial order on V induced by $\overset{3_i}{\preceq}$, $1 \le i \le k$, is consistent with the precedence $\preceq$ (i.e., they both can be completed to the same linear order on V).

ix. For the $\overset{3_0}{\sim}$-class C^0, $\mathrm{width}_0(C^0) \le k+1$, where the functions width_i, $0 \le i \le k$, mapping V to the set of natural numbers are defined recursively as follows: For any $u \in V$,

$\mathrm{width}_k([u]_{\underset{\sim}{3_k}}) = |\mathrm{new}^1(M_u)|$,

for $0 \le i < k$, $\mathrm{width}_i([u]_{\underset{\sim}{3_i}}) = \max_{v \in [u]_{\underset{\sim}{3_i}}} \{\bar{b}([v]_{\underset{\sim}{3_{i+1}}}) + \mathrm{width}_{i+1}([v]_{\underset{\sim}{3_{i+1}}})\}$, where for a $\overset{3_{i+1}}{\sim}$-class C contained in a $\overset{3_i}{\sim}$-class $\hat{C}$, $\bar{b}(C) = b(C) - \mathrm{bal}(C)$.

x. For a $\overset{3_i-1}{\sim}$-class C $(0 < i \leq k)$, let $P_1^C, \ldots, P_k^C$ be the k sets of nodes induced by the labeling λ_i. The partial order $\overset{3_i}{\preceq}$ induced by the colouring of G must be a linear order on each set P_j^C, $1 \leq j \leq k$.

It should be clear that conditions (i)-(x) are expressible in MS.

Lemma 4.3.11 *If a colouring of G satisfies conditions (i)-(ix) stated above, then the triple* $(\overset{1}{\sim}, \{\overset{3_i}{\sim}\}_{i=0}^k, \{\overset{3_i}{\preceq}\}_{i=0}^k)$ *induced by that colouring is a partial* k^k*-generative structure on* G.

Proof. It is not difficult to show that the path-width of any decomposition of G generated by this triple is equal to $\mathrm{width}_0(C^0) - 1$ which is less than k by condition (ix). □

As in the case of $(k, 1)$-paths, we have the following statement.

Corollary 4.3.12 *For an arbitrary graph* $\hat{G}$, *if there exists a colouring of G satisfying conditions (i)-(ix), then* $\hat{G}$ *is a partial k-path.*

Remark 4.3.13 Thus, we have shown how to construct the MS-formula for each $k > 0$ that will define the family of all partial k-paths.

Condition (x) ensures that the partial orders $\overset{3_i n}{\preceq}$, $1 \leq i \leq k$, induced by the colouring of G are linear orders on the set of nodes of each $\overset{3_i-1}{\sim}$-class. By conditions (vi) and (v), each leaf induced by the colouring of G does not contain any non-drop vertices, therefore algorithm $\mathcal{R}'$ will work correctly on this coloured graph.

So we can formulate the principal result of our thesis.

Theorem 4.3.14 *Every recognizable family of connected partial k-paths is CMS-definable.*

4.3.5 The Case of Disconnected Partial k-Paths

Here we extend Theorem 4.3.14 to disconnected partial k-paths.

Let a partial k-path G have $t \geq 1$ components $G_1, \ldots, G_t$. Obviously, each G_j, $1 \leq j \leq t$, is a connected partial k-path. We apply algorithm $\mathcal{R}'$ to each G_j, which will give us t vectors $q(G_j)$ describing the behaviour of the corresponding automaton A on each of the components of G.

If A recognizes G, then the decomposition of G obtained by arbitrarily ordering the decompositions of its components should be accepted by A. That is, the composition of the maps $q(G_j)$, $1 \leq j \leq t$, chosen in any order should define the map that takes the initial state of A to some of its final states.

Since connectedness of a graph is MS-definable, we can simulate in CMS applying algorithm $\mathcal{R}'$ to each component of G (e.g., we can define the 3_0-equivalence as the relation of being connected). Then we define the δ-equivalence (by saying that two components G' and G'' of G are δ-equivalent iff $q(G') = q(G'')$), order these $\overset{\delta}{\sim}$-classes lexicographically, and compute the corresponding composition. Clearly, this is expressible in CMS.

Thus we have the theorem.

Theorem 4.3.15 *Every recognizable family of (possibly disconnected) partial k-paths is CMS-definable.*

Combining this with the Courcelle's result on definability implying recognizability for partial k-trees yields the following.

Corollary 4.3.16 *Definability equals recognizability for partial k-paths.*

Chapter 5

Conclusion

We showed that every recognizable family of partial k-paths is CMS-definable, thereby proving a particular case of Courcelle's conjecture. Thus, we can now say that a problem on partial k-paths is solvable in linear time using a finite automaton iff it is CMS-definable.

As a byproduct of our solution, we obtained the MS-formula defining the class of partial k-paths for every given k. This implies that the obstruction sets for the classes of partial k-paths are computable.

Our results rely upon the possibility to MS-define a certain partial ordering on the set of vertices of a given partial k-path. In this respect, it is interesting to note some similarities between our approach and that of Courcelle in [9]. (Our results were obtained completely independently.) In his paper, Courcelle was able to MS-define a linear ordering on the vertex-set of a k-connected partial k-path G that corresponds to the order in which those vertices are added to some k-decomposition of G (i.e., a k-generative linear order on G).

A graph G is k-connected if $|V| \geq k+2$ and there is no subset $U \subseteq V$ of cardinality less than k such that the removal of U disconnects the graph G. A partial k-path (according to Courcelle) is a graph that allows a k-decomposition $B = \langle B_1, \ldots, B_m \rangle$ satisfying the following conditions:

i. $|B_i| = k + 1$ for each $i \in \{1, \ldots, m\}$,

ii. $|B_i \cap B_{i+1}| = k$ for each $i \in \{1, \ldots, m - 1\}$, and

iii. $B_i \cap B_{i+1} \neq B_{i+1} \cap B_{i+2}$ for each $i \in \{1, \ldots, m - 2\}$.

We will prove that a k-connected partial k-path (according to the above definition) is a particular case of a connected $(k,1)$-path (see Chapter 3) for which a k-generative linear order is MS-definable.

By the condition of k-connectedness, non-drop$(B_i) = k$ for each $i \in \{1,\ldots,m-1\}$. Combining this with condition (ii) yields that old(B_{i+1}) = non-drop(B_i) for each $i \in \{1,\ldots,m-1\}$. Thus, conditions (i) and (ii) and the k-connectedness of G imply that B is a $(k,1)$-decomposition.

Condition (iii) suggests that drop$(B_i) \cap$ new$(B_i) = \emptyset$ for each $i \in \{2,\ldots,m-1\}$. Therefore, the digraph G^d_B has no vertices with loop arrows, which means that there are no leaves. So, the partial order $\preceq^n$ is a linear order on the set $V \setminus$ drop(B_1) that can be completed (by making the drop vertex of B_1 the minimum) to a k-generative linear order on G.

Our results show that it is not necessary to MS-define the algebraic structure (path-decomposition, in our case) of a given graph in order to prove that recognizability implies CMS-definability.

Actually, it is impossible to MS-define k-decompositions for arbitrary partial k-paths. Consider the graphs $G_n = (\{0,1,\ldots,n\}, E_n)$, where $E_n = \{\{0,j\} | 1 \leq j \leq n\}$. Clearly, these graphs are 1-connected 1-paths (according to the general definition of k-paths). If we had an MS-formula defining a 1-decomposition for each G_n, $n \geq 1$, we could MS-define linear orderings on the vertex-sets V_{G_n}. But no linear orders can be MS-defined on G_n, since these graphs have nontrivial automorphisms, and the size of G_n can be arbitrary large.

The main open problem is to prove (or disprove) the Courcelle's conjecture for partial k-trees. It seems promising to apply the ideas from our solution for partial k-paths to this general case. A tree-decomposition induces a partial order on the vertex-set of a partial k-tree. We will need to define certain "nice" tree-decompositions such that this order can be MS-coloured.

We already know how to MS-colour the parts of a partial k-tree G that are partial k-paths. We can also define in CMS the behaviour of the corresponding tree-automaton on such "linear" parts. Thus, it would suffice to be able to MS-define a suitable partial order on these fragments of G.

Bibliography

[1] Arnborg, S., Lagergren, J., and Seese, D. Easy problems for tree decomposable graphs. *J. Algorithms 12* (1991), 308–340.

[2] Arnborg, S., and Proskurowski, A. Characterization and recognition of partial 3-trees. *SIAM J. Alg. Disc. Meth. 7* (1986), 305–314.

[3] Arnborg, S., Proskurowski, A., and Corneil, D. Forbidden minors characterization of partial 3-trees. *Disc. Math. 80* (1990), 1–19.

[4] Bern, M., Lawler, E., and Wong, A. Linear-time computation of optimal subgraphs of decomposable graphs. *J. Algorithms 8* (1987), 216–235.

[5] Bodlaender, H. A linear time algorithm for finding tree-decompositions of small treewidth. In *Proc. 25^{th} STOC* (1993), pp. 226–234.

[6] Bondy, J., and Murty, U. *Graph Theory with Applications.* North Holland, 1976.

[7] Borie, R., Parker, R., and Tovey, C. Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. *Algorithmica 7* (1992), 555–581.

[8] Büchi, J. Weak second-order arithmetic and finite automata. *Zeitschr. j. math. Logik und Grundlagen d. Math. 6* (1960), 66–92.

[9] Courcelle, B. The monadic second-order logic of graphs. X. Linear orderings. Unpublished manuscript.

[10] Courcelle, B. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation 85* (1990), 12–75.

[11] COURCELLE, B. The monadic second-order logic of graphs. V. On closing the gap between definability and recognizability. *Theoret. Comput. Sci. 80* (1991), 153–202.

[12] COURCELLE, B. The monadic second-order logic of graphs. III. Tree-decompositions, minors and complexity issues. *Informatique théorique et Applications 26* (1992), 257–286.

[13] DONER, J. Tree acceptors and some of their applications. *J. Computer and System Sciences 4* (1970), 406–451.

[14] FELLOWS, M., AND LANGSTON, M. An analogue of the Myhill-Nerode theorem and its use in computing finite-basis characterization. In *Proc. 30^{th} FOCS* (1989), pp. 520–525.

[15] HARARY, F. *Graph Theory.* Addison-Wesley, 1969.

[16] HOPCROFT, J., AND ULLMAN, J. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[17] KALLER, D. Definability equals recognizability of partial 3-trees, 1996. Workshop on Graph-Theoretic Concepts in Computer Science (WG '96).

[18] ROBERTSON, N., AND SEYMOUR, P. Graph minors.XV. Wagner's conjecture. Revised version, March 1988.

[19] ROBERTSON, N., AND SEYMOUR, P. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms 7* (1986), 309–322.

[20] SEESE, D. The structure of the models of decidable monadic theories of graphs. *Ann. Pure Appl. Logic 53* (1991), 169–195.

[21] WALD, J., AND COLBOURN, C. Steiner trees, partial 2-trees, and IFI networks. *Networks 13* (1983), 159–167.