Problem Formulations

TO COMMUNICATE problems in mathematics one makes use of logic. In this section we discuss two languages that can be used to express 'most' problems in algorithmic graph theory.

3.1 Graph Algebras

Many sets of graphs can be described via a 'gluing' or a 'bridging' procedure.

Given two graphs — say G_1 and G_2 — in a bridging operation one creates a new graph by adding edges between certain subsets of vertices of G_1 and G_2. Examples of classes of graphs that fall into this category are trees, cographs, and distance-hereditary graphs.[1]

[1] A graph is distance hereditary if for any two vertices in the graphs, all chordless paths that connect them, have the same length.

In a gluing operation one builds a larger graph from G_1 and G_2 by identifying certain vertices of G_1 and G_2. An example of a class of graphs that falls in this category is the class of chordal graphs.

The languages that we will discuss next allow us to formulate both the membership of graphs in these classes and most problems that are in NP.

T. Kloks, M. Xiao, *A Guide to Graph Algorithms*, https://doi.org/10.1007/978-981-16-6350-5_3

3.2 Monadic Second – Order Logic

Monadic second – order logic is a logical language — that can be used to express properties of graphs.

— For example — according to Definition 1.5 — we can express that a graph G is connected as follows.

$$\forall_{A \subseteq V} \ \forall_{B \subseteq V} \ (A \neq \varnothing \wedge B \neq \varnothing \wedge A \cap B = \varnothing \wedge V = A \cup B) \Rightarrow \exists_{a \in V} \exists_{b \in V} \ (a \in A \wedge b \in B \wedge \{ a, b \} \in E) . \quad (3.1)$$

— As you can see — in this language we use quantifiers — that is — the symbols

$$\forall \quad \text{and} \quad \exists .$$

The quantities — that we quantify over — are subsets of V and elements of V. [2] The rest of Formula 3.1 is called the expression of the sentence.

[2] That's why the language is called 'monadic.'

3.2.1 Sentences and Expressions

A sentence consists of

1. Logical symbols:
   (i) Parentheses (and)
   (ii) Connective symbols $\Rightarrow$, $\wedge$, $\vee$, and $\neg$
   (iii) Variables, say x, y, A, any natural number of them
2. and Parameters:
   (a) Quantifier symbols $\forall$ and $\exists$
   (b) Equality symbol $=$
   (c) Constants symbols, eg, $\varnothing$, and any natural number

(d) Predicate symbols — that is — any number of functions

$$f : V^k \to \{ \text{true}, \text{false} \} \quad \text{for} \ k \in \mathbb{N}$$

— for example — $(\{a, b\} \in E(G))$.[3]

[3] The number of arguments of a predicate is called its arity.

For simplicity we assume that sentences are well – formed — that is — the parentheses are placed so that they break down the formulas into their atomic parts. These parts have no connective nor quantifier symbols.

3.2.2 Quantification over Subsets of Edges

A version of Monadic Second – Order Logic — where quantification over subsets of edges is allowed — turns out to be important for algorithms on graphs of bounded treewidth (see Section 4.2).

Definition 3.1. A property is expressible in MS_1 if it can be expressed in a monadic second – order formula with quantification over vertices and subsets of vertices.
A property is expressible in MS_2 if it can be expressed in a monadic second – order formula with quantification over vertices, edges and subsets of vertices and edges.

Exercise 3.1

Show that having a Hamiltonian Cycle is a problem that can be expressed in MS_2. Can you also express it in MS_1 ?
Hint: A Hamiltonian cycle is a of edges that form a cycle — and — that covers all vertices.

Remark 3.2. Courcelle proved the following two theorems — which we mention here for completeness' sake.

Theorem 3.3. *All* MS_1 *– sentences can be evaluated in* $O(n^3)$ *time on graphs bounded rankwidth.*[4]

[4] The class of graphs of bounded rankwidth is a parametrization of the class of distance-hereditary graphs.

These algorithms are fixed-parameter algorithms, parametrized by the rankwidth of the graphs.

Historically the next theorem came first.

Theorem 3.4. *All* MS_2 *–sentences can be evaluated in linear time on graphs of bounded treewidth.*

These are fixed–parameter algorithms, where the parameter is the treewidth of the graph.

Exercise 3.2

The language MS_2 is more powerful than MS_1. Therefore, if we let MS_i stand for the class of graphs, for which all MS_i-–problems can be solved in universal–constant polynomial time, then we have

$$\mathsf{MS}_2 \subseteq \mathsf{MS}_1 .$$

— Find other problems — that are expressible in MS_2 but not in MS_1.