MONADIC LOGICS OF GRAPHS DEFINED BY OPERATIONS

KAMAL LODAYA

Fix a finite alphabet A. We will consider graphs which are finite and loop-free, with vertices labelled from A. Let $Graph(A)$ be the set of all such graphs.

Logically seen, a graph is a relational structure $G = (V, E, \{P_a \mid a \in A\})$ where the graph signature $\gamma(A)$ has a binary irreflexive relation E (symmetric for an undirected graph) and unary relations P_a which partition V (i.e. they form a labelling function).

$FO(A)$, $MSO(A)$ and $CMSO(A)$ are defined using the usual logical syntax, including equality, over the signature $\gamma(A)$. $CMSO$ has in addition to MSO, the atomic predicates $|X| \equiv r \mod q$, for $q > 1$ and $0 \leq r < q$. The FO theory of all graphs is undecidable.

Another view of graphs is as two-sorted structures (vertices and edges), with an incidence relation between them. This allows quantification over the edges and the MSO theory (called MSO_2 or GSO) is more powerful.

We also define, given $F \supseteq A$, the tree signature $\tau(F)$ with two binary irreflexive successor relations S_1 and S_2 and unary labelling relations P_f for $f \in F$. The $CMSO_2$ theory of $Tree(F)$, the set of all such trees, is decidable.

1. Semantic interpretations

Definition 1 (Hodges). *An **interpretation** of a set $\mathcal{G}$ of graphs (γ-structures) in a set $\mathcal{T}$ of trees (τ-structures), alternatively of a γ-theory Γ in a τ-theory Θ, consists of:*

(1)
- *A* domain formula *(in τ) Dom with one free variable;*
- Translation formulas *(in Θ): $\mathsf{Tran}(E)$ and $\mathsf{Tran}(=)$ each with two free variables, and $\mathsf{Tran}(P_a), a \in A$, each with one free variable. The relation defined by $\mathsf{Tran}(=)$ must be a congruence.*

For every τ-structure T, these defining formulas *yield (upto isomorphism) a γ-structure $G = (\mathsf{Dom}, \mathsf{Tran}(E), \{\mathsf{Tran}(P_a), a \in A\})$ "inside" T.*

(2) *An onto* coordinate map $\mathsf{c} : \mathsf{Dom} \to G$ *such that for all $u, v \in \mathsf{Dom}$,*
- $G \models E(\mathsf{c}(u), \mathsf{c}(v)) \iff T \models \mathsf{Tran}(E)(u, v)$;
- $G \models \mathsf{c}(u) = \mathsf{c}(v) \iff T \models \mathsf{Tran}(=)(u, v)$;
- $G \models P_a(\mathsf{c}(u)) \iff T \models \mathsf{Tran}(P_a)(u)$.

(3) *A* value map Val, *defined to map T to the quotient of G by $\mathsf{Tran}(=)$, is required to be an onto function from $\mathcal{T}$ to $\mathcal{G}$.*

From the lectures at the Tata Institute of Fundamental Research, Mumbai, March 02. I thank Paritosh Pandya for inviting me to give these lectures.

This is an old technique going back to the beginning of model theory, which was effectively used for showing decidability of theories by Rabin around 1970 after he proved that the MSO theory of the infinite tree is decidable. Seese and Courcelle sought to use this technique in the late 1980s for proving decidability of MSO theories of sets of graphs.

Val is also known as a *(definable) transduction* or *reduction*. (Note that it goes in the opposite direction to the interpretation!) If you think of τ-structures as representing terms, Val gives the graph corresponding to a term.

Tran induces a translation $\mathsf{Tran}^{\#}$ from γ-formulas to τ-formulas:

$\mathsf{Tran}^{\#}(E(x_1, x_2)) = \mathsf{Dom}(x_1) \wedge \mathsf{Dom}(x_2) \wedge \mathsf{Tran}(E)(x_1, x_2)$, *etc.*
$\mathsf{Tran}^{\#}(|X| \equiv r \mod q) = |X| \equiv r \mod q$
$\mathsf{Tran}^{\#}(\neg\alpha) = \neg\mathsf{Tran}^{\#}(\alpha)$
$\mathsf{Tran}^{\#}(\alpha \vee \beta) = \mathsf{Tran}^{\#}(\alpha) \vee \mathsf{Tran}^{\#}(\beta)$
$\mathsf{Tran}^{\#}(\exists x\alpha) = \exists x(\mathsf{Dom}(x) \wedge \mathsf{Tran}^{\#}(\alpha))$
$\mathsf{Tran}^{\#}(\exists X\alpha) = \exists X(\forall y(X(y) \supset \mathsf{Dom}(y)) \wedge \mathsf{Tran}^{\#}(\alpha))$

Theorem 2 (Reduction). $T \models \mathit{Tran}^{\#}\alpha \iff \mathit{Val}(T) \models \alpha.$

1.1. Deciding satisfiability using interpretations.

Corollary 3. *If there is an interpretation of $\mathcal{G}$ in $\mathcal{T}$ (with computable Tran), then $Th(\mathcal{T})$ is decidable implies $Th(\mathcal{G})$ is decidable.*

Proof. To decide satisfiability of γ-formula α, compute $\mathsf{Tran}^{\#}\alpha$ and return its satisfiability.

If the algorithm returns yes, $\exists T \in \mathcal{T} : T \models \mathsf{Tran}^{\#}\alpha$. By reduction, $\mathcal{T}$ has $\mathsf{Val}(T) \models \alpha$.

If the algorithm returns no and supposing $\exists G \in \mathcal{G} : G \models \alpha$, since Val is onto, there is $T \in \mathcal{T}$ such that $T \models \mathsf{Tran}^{\#}\alpha$, a contradiction. $\square$

More generally, a *k-dimensional* interpretation maps a γ-variable x to a tuple of τ-variables $(y_1, \ldots, y_k)$. (Two-dimensional interpretations of the set of complex numbers in the set of reals are well known.) But now a set variable X gets interpreted as a k-ary relation, so the induced translation $\mathsf{Tran}^{\#}$ will not preserve MSO.

Definition 4 (Courcelle). *A k-*copying *interpretation maps a γ-set variable X into a partition of the domain by τ-set variables $X_1, \ldots, X_k$.*

Now $\mathsf{Tran}^{\#}$ does preserve MSO and $CMSO$. The counting formula translates to

$$\mathsf{Tran}^{\#}(|X| \equiv r \mod q) = \bigvee_{r_i, i \leq n} \left(\sum_{r_i \equiv r \mod q} |X_i| \equiv r_i \mod q \right).$$

1.2. Evaluating definable problems.

Evaluating a formula on a given graph (this includes "model checking") can also be transferred. We say an interpretation is *evaluation preserving polytime* if Val has a polytime converse function Parse which identifies a representative term (parse tree) for each graph, and $f(G)(\mathsf{Val}([a])) = \sum_{b \in [a]} f(\mathsf{Parse}(G))(b)$.

Theorem 5 (Arnborg, Lagergren, Seese 81)**.** *If $\mathcal{G}$ has an evaluation preserving polytime interpretation in $\mathcal{T}$, every γ-definable evaluation problem π on $\mathcal{G}$ has a polytime algorithm.*

Proof. The algorithm computes $\mathsf{Parse}(G)$ and $\mathsf{Tran}^{\#}\pi$. Evaluating a $CMSO$-definable problem on a tree is linear in the size of the tree. $\square$

1.3. **An example.** A poset is a transitive directed acyclic graph. Let $SP = \{\circ, +\}$, where $\circ$ is a binary associative "series" operation and $+$ a binary commutative associative "parallel" operation. $SP(A) = A \cup SP$ generates the *series-parallel posets*, which can be defined by:

$\mathsf{Val}(a)$ = singleton graph labelled by $a \in A$
$\mathsf{Val}(G + H)$ = disjoint union of graphs G and H
$\mathsf{Val}(G \circ H)$ = disjoint union and edges from all of G to all of H.

We equate sp-terms to their parse trees over $SP(A)$, where the interior nodes are labelled by $+$ or $\circ$, and the leaf nodes by letters from A. The vertex labelling by A is sometimes left implicit, so $Tree(SP)$ stands for $Tree(SP(A))$ and $Graph(SP)$ for the range of Val on $Tree(SP(A))$.

To show $\mathsf{Val} : Tree(SP) \to Graph(SP)$ is an MSO-reduction, we must produce the (backwards) interpretation of sp-posets in sp-trees. The defining τ-formulas are:

$$\mathsf{Dom}(x) = (leaf(x) \supset \bigvee_{a \in A} P_a(x)) \wedge (\neg leaf(x) \supset P_+(x) \vee P_\circ(x)),$$
$\mathsf{Tran}(P_a)(x) = P_a(x)$, $\mathsf{Tran}(=)(x, y) = (x = y)$,
$\mathsf{Tran}(E)(x, y)$ = the least common ancestor of x and y is a $\circ$-labelled node.

Every sp-poset has one or more term representations, so Val is onto; and one of them can be found in polytime. We can conclude that the MSO theory of sp-posets is decidable and MSO-definable evaluation problems on sp-posets have polytime algorithms.

2. Interpretations for graphs built up from operations

2.1. **Cliquewidth.** Our example of sp-posets made use of a particular set of operations SP, and the proof that Val is an MSO-transduction extends to the following set of operations VR, which appear in vertex replacement (aka C-edNCE) graph grammars. The set $Graph(VR)$ generated is also known as the graphs of **cliquewidth** at most $|A|$ [Courcelle, Engelfriet, Rozenberg 93].

$\mathsf{Val}(a)$ = singleton graph labelled by $a \in A$
$\mathsf{Val}(G + H)$ = disjoint union of graphs G and H
$\mathsf{Val}(addedge(a, b)(G))$ = edges from all a-labelled nodes to b-labelled ones;
[Translation: $\mathsf{Tran}(E)(x, y) = E(x, y) \vee (P_a(x) \wedge P_b(y))$]
$\mathsf{Val}(ren(a, b)(G))$ = rename all a-labelled nodes to b
[Translation: $\mathsf{Tran}(P_b)(x) = P_b(x) \vee P_a(x)$, $\mathsf{Tran}(P_a)(x) = false$].

The renaming operation is very powerful: any A-labelled graph can be represented using these operations, perhaps by using a larger alphabet B including A. Our interest is in sets over a fixed alphabet A. For instance, the set of cliques has cliquewidth 2, that of the set of trees is 3. At best an NP algorithm is possible for $\mathsf{Parse} : Graph(VR) \to Tree(VR)$ (for $|A| > 3$) [Fellows, Rosamond, Rotics, Szeider 09].

More general translation formulas are needed for the interpretation, since a node label depends on the renaming operations performed above it. [Courcelle, Makowsky 03] inductively build up for each formula α referring to m leaves a translation $\mathsf{Tran}^{\#}\alpha$ with $m+1$ free variables, such that for a tree T with subtree T_u rooted at node u,

$$T \models \mathsf{Tran}^{\#}\alpha(u, x_1, \ldots, x_m) \iff \mathsf{Val}(T_u) \models \alpha(x_1, \ldots, x_m).$$

Suppose the node u of the parse tree is labelled by an n-ary operation f, with subtrees rooted at $u1, \ldots, un$. We have by induction

$$T \models \mathsf{Tran}^{\#}\alpha_i(ui, x_1, \ldots, x_m) \iff Val(T_{ui}) \models \alpha_i(x_1, \ldots, x_m).$$

Let $\theta_i(ui)$ abbreviate $\mathsf{Tran}^{\#}\alpha_i(ui, x_1, \ldots, x_m)$. So the translation needs to have for each f an MSO formula $\mathsf{Tran}^{Op}(f)$ satisfying $T \models \theta(u) \iff \mathsf{Tran}^{Op}(f)(\theta_1(u1), \ldots, \theta_n(un))$.

Definition 6 (Seese)**.** *Call a set of graphs $\mathcal{G}$ **tree-interpretable** if it is the range of an MSO-transduction $\mathsf{Val} : Tree(F) \to Graph(A)$. Another way of saying this is that there is a parsing function $\mathsf{Parse} : \mathcal{G} \to Tree(F)$ together with an MSO translation (restricted to $\mathcal{G}$) Tran: $\gamma(A)$-formulas $\to \tau(F)$-formulas.*

Conjecture (Seese)**.** *If the MSO theory of $\mathcal{G}$ is decidable, then $\mathcal{G}$ is tree-interpretable.*

2.2. **Rankwidth.** The power of renaming extends to sets of graphs. [Engelfriet, van Oostrom 97] showed that if $\mathcal{G}$ is tree-interpretable, then it can be represented using the VR operations, perhaps using a larger $B \supseteq A$. A result close to Seese's conjecture uses another graph parameter called **rankwidth**, the representation of graphs using **matroids** and operations on them, and decomposition ideas [Geelen, Gerards, Whittle 02].

Theorem 7 (Courcelle, Oum 07)**.** *If the $CMSO$ theory of $\mathcal{G}$ is decidable, then $\mathcal{G}$ has bounded cliquewidth.*

2.3. **Treewidth.** The graphs of **treewidth** k were defined using a notion of decomposition [Halin 76; Robertson, Seymour 83]. From hyperedge replacement graph grammars, they can also be defined as the graphs $Graph(HR(A))$ with $|A| = k + 1$ generated by the HR operations below:

$\mathsf{Val}(ab)$ = graph with a single edge from an a node to a b node
$\mathsf{Val}(G + H)$ = disjoint union of graphs G and H
$\mathsf{Val}(ren(a, b)(G))$ = rename all a-labelled nodes to b
$\mathsf{Val}(fuse(a)(G))$ = all a-labelled nodes are collapsed to one
[Translation: $\mathsf{Tran}(=)(x, y) = (x = y) \vee (P_a(x) \wedge P_a(y))$].

The set of all trees has treewidth 0. Cliques are cliquewidth 2 but not bounded treewidth. The bounded treewidth graphs are exactly the bounded cliquewidth graphs which for some m, do not have $K_{m,m}$ as a subgraph. The set of square $n \times n$ grids for every n (planar, hence $K_{3,3}$-free) neither has bounded treewidth nor bounded cliquewidth.

To prove that Val remains a $CMSO_2$-transduction when the two sets of operations are combined, the Courcelle-Makowsky scheme has to inductively maintain an $CMSO_2$-definable equivalence relation on a subtree corresponding to the fusion operations within it.

Using the Robertson-Seymour theory of graph minors, Seese proved a version of his conjecture for the stronger MSO_2 logic.

Theorem 8 (Seese 91)**.** *If the MSO_2 theory of a set of graphs $\mathcal{G}$ is decidable, it has bounded treewidth.*

The advantage of using treewidth is efficient parsing, which yields polytime algorithms for MSO_2- (and $CMSO_2$-)definable evaluation problems.

Theorem 9 (Courcelle 89; Arnborg, Lagergren, Seese 91; Borie, Parker, Tovey 92)**.** *Val : $Tree(HR) \to Graph(HR)$ is an evaluation preserving polytime MSO_2-reduction (which is linear time in $|\mathsf{Parse}(G)|$).*

Theorem 10 (Bodlaender, Kloks 91; Bodlaender 96)**.** *Given a graph G, determining whether it has treewidth k, and computing Parse if it does, can be done in time linear in $|G|$.*

2.4. **Characterizing definability.** Courcelle initiated a search for converse transductions by providing a translation scheme for sp-trees in sp-graphs. This shows that for sp-graphs, the $CMSO_2$-definable sets are exactly those whose parse trees form regular tree languages. After several incomplete attempts the result was extended to bounded treewidth graphs.

Theorem 11 (Courcelle 89)**.** *There is a $CMSO_2$-reduction $\mathsf{Parse} : Graph(SP) \to Tree(SP)$.*

Theorem 12 (Bojańczyk, Pilipczuk 16)**.** *There is a $CMSO_2$-reduction $\mathsf{Parse} : Graph(HR) \to Tree(HR)$.*

The Institute of Mathematical Sciences, C.I.T. Campus, Chennai 600 113, India.