2.4 SOL-characterizations of graph-properties

In Chapter 1, we discussed FO-definable graph properties and their complexity. We continue this discussion and present examples of SOL-formulas for graph-properties that have no formalization in FOL. Later we will study the complexity of graph-properties that are definable within certain fragments of SOL.

Example 2.4.1 (Reachability is SO-definable). In Theorem 1.6.10 on page 89 and Theorem 1.6.44 on page 113, we saw that reachability in directed graphs is not FO-definable, even not when ranging over finite graphs only. However, there is a simple SOL-formula over the graph-vocabulary $\mathit{Voc}_{\mathrm{graph}}$ that characterizes reachability:

$$\phi_{\mathrm{reach}}(x,y) \ \stackrel{\mathrm{def}}{=}\ \forall Z.\big(Z(x)\wedge\psi(Z)\ \rightarrow\ Z(y)\big)$$

where Z is a set variable (i.e., unary predicate variable) and $\psi(Z)$ asserts that the node-sets represented by Z are "closed" under the edge relation E in the sense that whenever $u\in Z$ and $(u,v)\in E$ then $v\in Z$:

$$\psi(Z) \ \stackrel{\mathrm{def}}{=}\ \forall u\forall v.\big(Z(u)\wedge E(u,v)\ \rightarrow\ Z(v)\big)$$

Thus, $\phi_{\mathrm{reach}}(x,y)$ asserts that whenever Z is closed under the edge-relation E and contains x then it also contains y. We now show that $\phi_{\mathrm{reach}}(x,y)$ yields an SOL-characterization of reachability:

$$(\mathcal{G},a,b)\models\phi_{\mathrm{reach}}(x,y) \quad \text{iff} \quad b \text{ is reachable from } a \text{ in } \mathcal{G}$$

where $\mathcal{G}=(A,E)$ is a directed graph and a and b are distinguished nodes that serve as meaning for x and y, respectively. In what follows, let $\mathit{Reach}(a)$ denote the set of states that are reachable from a.

Proof of "$\Longrightarrow$": Suppose first that $(\mathcal{G},a,b)\models\phi_{\mathrm{reach}}(x,y)$. We now consider the interpretation where Z is interpreted by $\mathit{Reach}(a)$. Clearly, we have $a\in\mathit{Reach}(a)$ and with $u\in\mathit{Reach}(a)$ and $(u,v)\in E$ we also have $v\in\mathit{Reach}(a)$. Thus:

$$(\mathcal{G},a,b,\mathit{Reach}(a))\ \models\ Z(x)\wedge\psi(Z)$$

Since $(\mathcal{G},a,b)$ is a model for $\phi_{\mathrm{reach}}(x,y)$, the formula $Z(y)$ evaluates to true, when Z is interpreted by $\mathit{Reach}(a)$, i.e.:

$$(\mathcal{G},a,b,\mathit{Reach}(a))\ \models\ Z(y)$$

But this yields that $b\in\mathit{Reach}(a)$, i.e., b is reachable from a.

Proof of "$\Longleftarrow$": Let $\mathcal{G}=(A,E)$ be a graph and a, b nodes in $\mathcal{G}$ such that b is reachable from a. Let $C\subseteq A$ be an arbitrary node-set. The goal is to show that:

$$(\mathcal{G},a,b,C)\ \models\ Z(x)\wedge\psi(Z)\ \rightarrow\ Z(y) \qquad (*)$$

This is obvious if $(\mathcal{G},a,b,C)\ \not\models\ Z(x)\wedge\psi(Z)$. Let us suppose now that $(\mathcal{G},a,b,C)\models Z(x)$ and $(\mathcal{G},a,b,C)\models\psi(Z)$. Then, $a\in C$ and C is closed under the edge-relation. By induction on the length of a shortest path from a to an element $c\in\mathit{Reach}(a)$ we get that $\mathit{Reach}(a)\subseteq C$. By assumption $b \in \textit{Reach}(a)$. In particular, C also contains node b. This yields that $(\mathcal{G}, a, b, C) \models Z(y)$. Hence, we get (*). ⌋

We now provide an alternative formalization of reachability in directed graphs by means of another SOL-formula. For this, we switch to the unreachability problem and provide a formula $\phi_{\mathrm{unr}}(x,y)$ that holds for a graph $\mathcal{G} = (A, E)$ with distinguished nodes a and b if and only if b is not reachable from a in $\mathcal{G}$. The idea of the formula $\phi_{\mathrm{unr}}(x,y)$ is the following observation. Node b is not reachable from a iff $(a,b) \notin E^*$ where E^* denotes the reflexive, transitive closure of the edge relation E. The condition $(a,b) \notin E^*$ is equivalent to the existence of a relation $E' \subseteq A \times A$ such that the following three conditions hold:

(1) $E \subseteq E'$
(2) E' is transitive and reflexive.
(3) $(a,b) \notin E'$

Note that (1) and (2) imply that $E^* \subseteq E'$. Hence, (1), (2) and (3) imply that $(a,b) \notin E^*$. Vice versa, if $(a,b) \notin E^*$ then $E' = E^*$ satisfies (1), (2) and (3). It remains to provide a SOL-formula for the above characterization of unreachability. Here it is:

$$\phi_{\mathrm{unr}}(x,y) \stackrel{\mathrm{def}}{=} \exists Z. \big(\psi_1 \wedge \psi_2 \wedge \psi_3 \big)$$

where Z is a binary predicate variable representing relation E'. Subformulas ψ_1, ψ_2 and ψ_3 formalize the above three conditions.

$$\begin{aligned}
\psi_1 &\stackrel{\mathrm{def}}{=} \forall u \forall v. \big(E(u,v) \rightarrow Z(u,v) \big) \\
\psi_2 &\stackrel{\mathrm{def}}{=} \forall u. Z(u,u) \wedge \forall u \forall v \forall w. \big(Z(u,v) \wedge Z(v,w) \rightarrow Z(u,w)) \big) \\
\psi_3 &\stackrel{\mathrm{def}}{=} \neg Z(x,y)
\end{aligned}$$

We then have $(\mathcal{G}, a, b) \models \phi_{\mathrm{unr}}(x,y)$ if and only if b is not reachable from a in $\mathcal{G}$. ■

Example 2.4.2 (Hamilton path problem). By the results of Section 1.6, *NP*-hard graph problems cannot be described in FOL, unless $P = NP$. On the other hand, all graph-problems in *NP* have characterizations by SOL-formulas, even by formulas of the existential fragment of SOL. This result is known as Fagin's theorem. We will not provide a proof for this result, but illustrate it by means of the *Hamilton-path problem* that is known to be *NP*-complete. The Hamilton-path problem takes as input a finite directed graph $\mathcal{G}$. The task is to check whether there exists a path which visits all nodes of $\mathcal{G}$ exactly once. In SOL, we describe the property of a Hamilton path by means of a linear order on the nodes of $\mathcal{G}$. For this purpose, we deal with the following SOL-sentence:

$$\phi_{\mathrm{Hamilton}} \stackrel{\mathrm{def}}{=} \exists \sqsubseteq . \big(\psi_{\mathrm{lin\_order}}(\sqsubseteq) \wedge \psi_{\mathrm{edges}}(\sqsubseteq) \big)$$

where $\sqsubseteq$ is a binary predicate variable. Formula $\psi_{\mathrm{lin\_order}}(\sqsubseteq)$ is a FOL-formula stating that $\sqsubseteq$ is a linear order (see Definition 1.4.13 on page 53). Formula $\psi_{\mathrm{edges}}(\sqsubseteq)$ asserts that any consecutive nodes with respect to $\sqsubseteq$ are connected by an edge in $\mathcal{G}$:

$$\psi_{\mathrm{edges}}(X) \stackrel{\mathrm{def}}{=} \forall x \forall y. \big(x \lhd y \rightarrow E(x,y) \big)$$

where $x \lhd y$ stands short for $x \sqsubset y \wedge \neg \exists z. (x \sqsubset z \wedge z \sqsubset y)$. As usual, $z \sqsubset z'$ is an abbreviation for $z \sqsubseteq z' \wedge z \neq z'$. Thus, $x \lhd y$ states that y is direct successor of x with respect to $\sqsubseteq$. Let us check that for each finite graph $\mathcal{G}$:

$$\mathcal{G} \models \phi_{\text{Hamilton}} \quad \text{iff} \quad \mathcal{G} \text{ has a Hamilton path.}$$

If $\mathcal{G}$ has a Hamilton path $a_0\, a_1 \ldots a_n$ then we put

$$R \stackrel{\text{def}}{=} \big\{ (a_i, a_j) : 0 \leqslant i \leqslant j \leqslant n \big\}.$$

Relation R is a linear order and the nodes a_i, a_{i+1} are consecutive in R. Let R serve as interpretation for the predicate variable $\sqsubseteq$. We obtain:

$$(\mathcal{G}, R) \models \psi_{\text{lin\_order}}(\sqsubseteq) \wedge \psi_{\text{edges}}(\sqsubseteq).$$

This yields $\mathcal{G} \models \phi_{\text{Hamilton}}$. Vice versa, if $\mathcal{G}$ is a model for ϕ_{Hamilton} then there exists a binary relation R on A such that $\psi_{\text{lin\_order}}(\sqsubseteq) \wedge \psi_{\text{edges}}(\sqsubseteq)$ evaluates to true when $\sqsubseteq$ is interpreted by R. But then the sequence of consecutive nodes in R constitutes a Hamilton path in $\mathcal{G}$. ■

Recall that given a formula ϕ over *Voc*$_{\text{graph}}$, the problem ϕ-GRAPH takes as input (1) a finite directed graph $\mathcal{G} = (A, E)$ and (2) a variable valuation $\mathcal{V}$ for the free variables of ϕ, and asks whether ϕ holds for $(G, \mathcal{V})$. The above example shows that we cannot expect to have a result stating the computational efficiency of the problems ϕ-GRAPH for all SOL-formulas ϕ, as we did for FOL in Theorem 1.6.8 on page 88. (Recall that the Hamilton path problem is known to be *NP*-complete.) However, we can establish membership of ϕ-GRAPH to *NP* for SOL-formulas that start with a prefix of existential SO-quantifiers followed by a FOL-formula. More precisely:

Definition 2.4.3 (Existential SOL-formula). An *existential SOL-formula*, simply called ESOL-formula, is a SOL-formula over some purely relational vocabulary of the form

$$\exists X_1 \ldots \exists X_n . \psi$$

where the X_i's are predicate variables of arbitrary arity and ψ a FOL-formula (possibly with equality) when $X_1, \ldots, X_n$ are viewed as predicate symbols. Formula ψ does not contain any other second-order variable. I.e., there are no function variables in ψ and no predicate variables other than $X_1, \ldots, X_n$. In particular, when ψ is viewed as a SOL-formula then $\mathit{PFree}(\psi) \subseteq \{X_1, \ldots, X_n\}$. ■

Thus, ESOL-formulas do not permit universal second-order quantification and do not use function variables at all. For instance, ϕ_{Hamilton} and $\phi_{\text{unr}}(x, y)$ are existential SOL-formulas, while $\phi_{\text{reach}}(x, y)$ is not.

Theorem 2.4.4 (ESO-definable graph properties are in NP). *For each ESOL formula ϕ over Voc$_{\text{graph}}$, the problem ϕ-GRAPH belongs to NP. (Formula ϕ is viewed to be fixed.)*

Proof. Let $\phi = \exists X_1 \ldots \exists X_n . \psi(X_1, \ldots, X_n, y_1, \ldots, y_k)$ be an existential SOL-formula where X_i is a m_i-ary predicate variable and $y_1, \ldots, y_k$ are the first-order variables that appear free in ψ. Thus, $\mathit{Free}(\phi) = \{y_1, \ldots, y_k\}$.

A *nondeterministic poly-time algorithm* that solves ϕ-GRAPH can be designed as follows. Given a finite directed graph $\mathcal{G} = (A, E)$ and elements $b_1, \ldots, b_k \in A$ as interpretation for the free first-order variables $y_1, \ldots, y_k$ of ϕ (and ψ), the first step is to guess nondeterministically subsets R_i of A^{m_i} (e.g., by guessing one bit per m_i-tuple of nodes) for $i = 1, \ldots, n$. We then use the *deterministic poly-time algorithm* sketched in Theorem 1.6.9 on page 89 to compute the truth value of ψ when interpreted over $\mathcal{G}$ with the valuation that interpretes the SO-variables X_i by R_i and the first-order variables y_j by the given elements $b_j \in A$. If

$$(\mathcal{G}, [X_1 := R_1, \ldots, X_n := R_n, y_1 := b_1, \ldots, y_k := b_k]) \models \psi$$

then we return "yes", otherwise we return "no". The soundness of this nondeterministic *guess & check* algorithm follows from the fact that the above algorithm has a computation that returns "yes" if and only if ϕ holds for the given graph $\mathcal{G}$ with distinguished nodes $b_1, \ldots, b_k$. It remains to verify that the time complexity of this *guess & check* method is, in fact, polynomial in the size of the input structure $\mathcal{G}$. The guessing phase requires $\mathcal{O}(M)$ steps where

$$M \;=\; |A|^{m_1} + \ldots + |A|^{m_n}$$

steps. Remind that we guess relation $R_i \subseteq A^{m_i}$ by guessing one bit per tuple $(a_1, \ldots a_{m_i}) \in A^{m_i}$. The costs for the checking phase are as follows. It is no restriction to suppose that ψ is in prenex form, say

$$\psi \;=\; Q_1 z_1 \ldots Q_\ell z_\ell . \theta$$

where the Q_i's are quantifiers, $z_1, \ldots, z_\ell$ are FO-variables and θ is quantifier-free. To evaluate ψ over $(\mathcal{G}, [X_1 := R_1, \ldots, X_n := R_n, y_1 := b_1, \ldots, y_k := b_k])$ we regard all ℓ-tuples $(c_1, \ldots, c_\ell) \in A^\ell$ and evaluate θ over

$$(\mathcal{G}, [X_1 := R_1, \ldots, X_n := R_n, y_1 := b_1, \ldots, y_k := b_k, z_1 := c_1, \ldots, z_\ell := c_\ell])$$

Altogether these are $|A|^\ell$ interpretations. For each of these interpretations, we apply the deterministic poly-time model checking algorithm for FOL-formulas. This deterministic model checking algorithm has one recursive call per (occurrence of a) subformula $\theta' \in \mathit{subf}(\theta)$ of length $|\theta'| \geqslant 1$. Recall that θ is quantifier-free. The atoms in θ have the form $x_1 = x_2$, $E(x_1, x_2)$ or $X_i(x_1, \ldots, x_{m_i})$ where $1 \leqslant i \leqslant n$ and the x_j's are FO-variables in $\{y_1, \ldots, y_k, z_1, \ldots, z_\ell\}$. The cost for the treatment of atoms in $x_1 = x_2$ is in $\mathcal{O}(1)$, while checking the truth of an atom $E(x_1, x_2)$ or $X_i(x_1, \ldots, x_{m_i})$ can be done in time $\mathcal{O}(M + |E|)$, even when we assume a naive list representation for the edge relation E and the relations $R_1, \ldots, R_n$. In summary, the total cost are bounded by:

$$\mathcal{O}\big(\, M \;+\; |A|^\ell \cdot (M + |E|) \cdot |\mathit{subf}(\theta)| \,\big) \;=\; \mathcal{O}(\mathit{poly}(|A|))$$

since formula ϕ (and therefore also ℓ and the m_i's) are treated as constants. Note that M is polynomial in $|A|$ as the m_i's are constant and $|E|$ is bounded by $|A|^2$. □

The above theorem also applies to the SOL-formula $\phi_{\text{unr}}(x, y)$ for the unreachability problem in Example 2.4.1. However, the existence of a nondeterministic polynomially time-bounded algorithm for the unreachability problem is not of interest, as (un)reachability questions can be solved deterministically (e.g., by a depth- or breath-first search) in time linear in the size of the input graph. We now show that the existential SOL-formula $\phi_{\text{unr}}(x, y)$ belongs to a subclass of ESOL-formulas where the induced graph problem can shown to be solvable in deterministic poly-time.

Definition 2.4.5 (ESOL Horn expression). An existential SOL-formula $\exists X_1 \ldots \exists X_n . \psi$ is called a *Horn expression* if the FOL-formula ψ has the form

$$\psi \;=\; \forall x_1 \ldots \forall x_k . \, \varphi$$

where φ is quantifier-free and a conjunction of clauses such that each clause of φ contains at most one positive literal of the form $X_i(t_1,\dots,t_{m_i})$.

Recall that a *clause* is a disjunction of literals, i.e., a disjunction of atomic formulas $P(t_1,\dots,t_m)$, $X(t_1,\dots,t_m)$ and $t_1 = t_2$ or their negations, where P is a m-ary predicate symbol, X a m-ary predicate variable and $t_1,\dots,t_k$ are terms. As ESOL relies on a purely relational vocabulary, the terms are just variables. ■

There is no restriction on the number of positive literals $P(y_1,\dots,y_k)$ or $y_1 = y_2$ where P is a predicate symbol and $y_1,\dots,y_k$ are first-order variables. In case of the vocabulary Voc_{graph}, this means that the clauses of a Horn expression $\exists X_1 \dots \exists X_n.\psi$ are built by an arbitrary number of literals

$$y_1 = y_2,\;\; y_1 \neq y_2,\;\; E(y_1,y_2),\;\; \neg E(y_1,y_2) \;\;\text{and}\;\; \neg X_i(y_1,\dots,y_{m_i}),$$

but *at most* one positive literal $X_i(y_1,\dots,y_{m_i})$. For instance, if X is a binary predicate variable then

$$\exists X \forall y \forall z \forall z'.\big((\neg X(y,z) \vee \neg X(y,z') \vee z = z') \wedge (X(y,y) \vee E(y,z) \vee E(z,y))\big)$$

is a Horn expression, while $\exists X \forall y \exists z. X(y,z)$ and $\exists X \forall y \forall z. \big(X(y,z) \vee X(z,y)\big)$ are not.

The formula $\phi_{\text{unr}}(x,y)$ of Example 2.4.1 on page 150 is not a Horn expression (since it is not in prenex form), but it can easily be rewritten into an equivalent Horn expression:

$$\phi_{\text{unr}}(x,y) \;\; \equiv \;\; \exists X \forall u \forall v \forall w.\, \varphi$$

where φ is built by four clauses as follows:

$$\varphi \;=\; (\neg E(u,v) \vee X(u,v)) \wedge X(u,u) \wedge (\neg X(u,v) \vee \neg X(v,w) \vee X(u,w)) \wedge \neg X(x,y).$$

In fact, each clause in φ contains at most one unnegated atomic formula that involves the second-order variable X.

Theorem 2.4.6 (ESO-Horn-definable graph properties are in PTIME). *For any existential SOL-formula ϕ which is a ESOL Horn expression, the problem ϕ-GRAPH is in P. (The formula ϕ is assumed to be fixed.)*

Proof. For simplicity, we only consider the case of closed ESOL-formula with a single second-order quantifier, i.e., an ESOL-sentence of the form $\phi = \exists X.\psi$ where X is a m-ary predicate variable and

$$\psi = \forall x_1 \dots \forall x_k.\varphi$$

and φ is a conjunction of C clauses as above. The argument for several second-order quantifiers and ESOL Horn expressions with free first-order variables is analogous. The goal is to provide a deterministic polynomially time-bounded algorithm that solves ϕ-GRAPH and relies on a reduction to HORN-SAT, the satisfiability problem for propositional Horn formulas. Recall that a *propositional Horn formula* is a propositional formula in CNF (conjunctive normal form) $\varphi = \kappa_1 \wedge \dots \wedge \kappa_h$ such that each clause κ_j contains at most one positive literal. The clauses of Horn formulas are often written in arrow notation:

- $true \rightarrow p$ for the unit clause consisting of the positive literal p,
- $q_1 \wedge \ldots \wedge q_r \rightarrow p$ for the Horn clause $\neg q_1 \vee \ldots \vee \neg q_r \vee p$,
- $q_1 \wedge \ldots \wedge q_r \rightarrow false$ for the Horn clause $\neg q_1 \vee \ldots \vee \neg q_r$.

Horn clauses without any positive literal are called *goal clauses*, while unit clauses consisting of a positive literal are called *facts*.

The satisfiability problem for propositional Horn formulas is solvable in polynomial time by a *marking algorithm* that works as follows. In the initialization phase, the algorithm marks all facts, i.e., atomic propositions that appear in a unit clause consisting of a positive literal. Then, it marks iteratively any atomic proposition q where for some clause

$$\neg q_1 \vee \ldots \vee \neg q_r \vee p \;\equiv\; q_1 \wedge \ldots \wedge q_r \rightarrow p$$

the atomic propositions $q_1, \ldots, q_r$ have been marked before. This step is repeated step until either no further marking is possible or a goal clause

$$\neg q_1 \vee \ldots \vee \neg q_r \;\equiv\; q_1 \wedge \ldots \wedge q_r \rightarrow false$$

is obtained where the atoms $q_1, \ldots, q_r$ have been marked. In the latter case, i.e., where a goal clause $q_1 \wedge \ldots \wedge q_r \rightarrow false$ is obtained such that the atoms $q_1, \ldots, q_r$ are marked, the formula is not satisfiable. If no further marking is possible and there is no clause consisting of negative literals with marked atoms, then the given Horn formula evaluates to *true* when assigning *true* to all marked atomic propositions and *false* to the non-marked atomic propositions.

Let $\mathcal{G} = (A, E)$ be the input graph where $|A| = n$. (Recall that ϕ is supposed to be closed.) The task is to check whether there exists a relation $R \subseteq A^m$ such that $\psi = \forall x_1 \ldots \forall x_k . \varphi$ is satisfied when interpreted over $\mathcal{G}$ and R as semantics for X. Since all quantifications in ψ are universal, $(\mathcal{G}, [X := R]) \models \psi$ iff the formula φ holds for all n^k combinations of values for the variables $x_1, \ldots, x_k$. Thus, $\mathcal{G} \models \phi$ if and only if there is some $R \subseteq A^m$ such that the conjunction

$$\bigwedge_{(a_1,\ldots,a_k) \in A^k} \varphi[x_1/a_1, \ldots, x_k/a_k]$$

holds for $(\mathcal{G}, [X := R])$, where the nodes a_i are viewed as constant symbols with fixed semantics. The atomic formulas in the clauses of $\varphi[x_1/a_1, \ldots, x_k/a_k]$ have the form

$$a_i = a_j, \quad E(a_i, a_j) \quad \text{or} \quad X(a_{i_1}, \ldots, a_{i_m}).$$

The atoms $a_i = a_j$ and $E(a_i, a_j)$ can be replaced by *true* or *false*, depending on whether nodes a_i and a_j agree and whether there is an edge from a_i to a_j, respectively. If in this way, a literal becomes *false* it can be removed. If a literal becomes *true*, the clause containing this literal can be deleted. Similarly, if an empty clause has been generated then we may abort with the result that $\mathcal{G}$ is not a model for $\phi = \exists X . \psi$. Otherwise, the resulting formula is a conjunction of at most Cn^k clauses (recall that C is the number of clauses in φ) and each of these clauses is a disjunction of literals of the form $X(a_{i_1}, \ldots, a_{i_m})$ or $\neg X(a_{i_1}, \ldots, a_{i_m})$. Moreover, each clause contains at most one positive literal.

Since we are free to choose a relation R for X, each of these atomic formulas $X(a_{i_1}, \ldots, a_{i_m})$ can be regarded as a boolean variable (atomic proposition). The resulting formula φ' is a *propositional Horn formula* with at most Cn^k clauses and at most n^m boolean variables, where $n = |A|$ is the number of nodes. And we have:

$$
\begin{array}{ll}
& \mathcal{G} \models \phi \; = \; \exists X \forall x_1 \ldots \forall x_k . \varphi \\
\text{iff} & \text{there exists } R \subseteq A^m \text{ such that } (\mathcal{G}, [X := R]) \models \forall x_1 \ldots \forall x_k . \varphi \\
\text{iff} & \text{there exists } R \subseteq A^m \text{ such that} \\
& \quad (\mathcal{G}, [X := R]) \models \bigwedge_{(a_1,\ldots,a_k) \in A^k} \varphi[x_1/a_1, \ldots, x_k/a_k] \; \equiv \; \varphi' \\
\text{iff} & \text{there is an assignment for the boolean variables } X(a_{i_1}, \ldots, a_{i_m}) \\
& \text{that is satisfying for } \varphi' \\
\text{iff} & \text{the propositional Horn formula } \varphi' \text{ is satisfiable}
\end{array}
$$

Thus, we finally apply a polynomially time-bounded algorithm to check satisfiability of the obtained propositional Horn formula φ'. The described algorithm runs in time polynomial in the size of the input graph, since the formula ϕ is assumed to be fixed and thus, k (the number of first-order quantifiers in ϕ) and m (the arity of the second-order variable X) are treated as constants. □

The proof of Theorem 2.4.6 does not use any property that is specific for the vocabulary of graphs. Indeed, Theorem 2.4.6 can be rephrased for any finite relational vocabulary.