1.6 FO-definability

The purpose of this section is to study which properties of structures can be expressed by FOL-formulas. We start with a series of examples for mathematical properties that cannot be specified in FOL and then provide a game-theoretic method that supports proving that a certain property is not FO-definable.

Definition 1.6.1 (FO-definability of a class of structures). Given a vocabulary *Voc*, a class $\mathcal{C}$ of structures for *Voc* is said to be *FO-definable* if there exists a FOL-sentence ϕ over the vocabulary *Voc*, such that exactly the structures $\mathcal{A} \in \mathcal{C}$ are models for ϕ, i.e.,

$$\mathcal{C} \;=\; \big\{\mathcal{A} : \mathcal{A} \text{ is a structure over } \mathit{Voc} \text{ such that } \mathcal{A} \models \phi\big\}$$

In this case, we say that ϕ *defines* $\mathcal{C}$. ■

A weaker form of FO-definability requires the existence of a (possibly infinite) formula-set $\mathfrak{F}$ such that $\mathcal{C}$ agrees with all structures that are models for $\mathfrak{F}$. Although we will use the notion FO-definability in the stronger sense (where a single formula is required to define $\mathcal{C}$), at several places we will show that a certain class is not FO-definable in the weaker sense.

Obviously, if $\mathcal{C}$ is FO-definable then $\mathcal{C}$ must be closed under isomorphism as isomorphic structures yield the same truth value for all FOL-sentences.

Example 1.6.2 (FO-definability of groups). Several algebraic structures like groups, commutative groups, rings, or vector spaces are FO-definable. Let us here consider the class of groups. To specify the axioms of a group by FOL-formulas with equality we use the vocabulary $\mathit{Voc}_{\text{group}}$ which contains a binary function symbol $\circ$ (representing the group-operation) and a constant symbol e for the group's neutral element, but no other predicate or function symbols. Then, the group axioms are given by the formula

$$\phi_{\text{group}} \stackrel{\text{def}}{=} \phi_{\text{ass}} \wedge \phi_{\text{inv}} \wedge \phi_{\text{id}}$$

where

ϕ_{ass}	$\stackrel{\text{def}}{=}$	$\forall x \forall y \forall z. (x \circ y) \circ z = x \circ (y \circ z)$	(associativity)
ϕ_{inv}	$\stackrel{\text{def}}{=}$	$\forall x \exists y. x \circ y = e$	(existence of right inverses)
ϕ_{id}	$\stackrel{\text{def}}{=}$	$\forall x. x \circ e = x$	(unity, neutral element from the right)

As ϕ_{group} formalizes exactly the conditions that are required for a group, for any structure $\mathcal{A} = (A, \circ, e)$ for $\mathit{Voc}_{\text{group}}$ we have:

$$\mathcal{A} \models \phi_{\text{group}} \quad \text{iff} \quad \mathcal{A} \text{ is a group}$$

Thus, the class of groups is FO-definable and ϕ_{group} defines it. In particular, each property that is specified by some formula ψ over $\mathit{Voc}_{\text{group}}$ with $\phi_{\text{group}} \Vdash \psi$ holds for any group. E.g.,

$$\forall x \exists y. (y \circ x = e) \wedge \forall x. (e \circ x = x)$$

is a sentence in the FO-theory of groups (i.e., the FO-theory $Cl(\phi_{group}) = \{\phi : \phi_{group} \Vdash \phi\}$), stating the existence of left inverses and the left-neutrality of e. We even have

$$\phi_{group} \Vdash \forall x \forall y. \big(y \circ x = e \leftrightarrow x \circ y = e \big) \wedge \exists! y. (x \circ y = e)$$

stating that the existence and uniqueness of inverse elements. However,

$$\phi_{group} \nVdash \forall x \forall y. (y \circ x = y \circ x),$$

since there are non-commutative groups. For example, if B is a set with at least three elements and A the set of bijections $f : B \to B$ then $(A, \circ, id)$ is a non-commutative group, where $\circ$ means standard function composition and $id : B \to B$ the identity function. ■

Remark 1.6.3 (FO-definability and complete theories). Let us discuss the connection between FO-definability and finite axiomatizations of theories induced by structures, see Section 1.4 on page 45 ff. Let $\mathcal{A}$ be a structure and $Isom(\mathcal{A})$ the class of all structures that are isomorphic to $\mathcal{A}$ (for some fixed vocabulary). Recall that $Th(\mathcal{A})$ is $\{\phi : \mathcal{A} \models \phi\}$. Then:

> If $Isom(\mathcal{A})$ is FO-definable then $\mathcal{A}$ is finite and $Th(\mathcal{A})$ is finitely axiomatizable.

Let us see why this holds. Let ϕ be a FOL-sentence that defines $Isom(\mathcal{A})$. In particular, $\mathcal{A} \models \phi$ as we have $\mathcal{A} \in Isom(\mathcal{A})$. We now show that:

$$Th(\mathcal{A}) \;=\; \big\{ \psi : \phi \Vdash \psi \big\}$$

"$\subseteq$:" Let $\psi \in Th(\mathcal{A})$. The goal is to show that $\phi \Vdash \psi$. So, we pick a structure $\mathcal{B}$ such that $\mathcal{B} \models \phi$. Then, $\mathcal{B} \in Isom(\mathcal{A})$, i.e., $\mathcal{B}$ is isomorphic to $\mathcal{A}$. But then $\mathcal{B} \models \psi$, as $\mathcal{A} \models \psi$. Hence, ψ is a consequence of ϕ.

"$\supseteq$:" If ψ is a logical consequence of ϕ then $\psi \in Th(\mathcal{A})$ as we have $\mathcal{A} \models \phi$, and hence, $\mathcal{A} \models \psi$.

Furthermore, structure $\mathcal{A}$ must be finite because of the upward Löwenheim-Skolem theorem (Theorem 1.5.4 on page 73). Note that if $\mathcal{A}$ would be infinite with domain A then formula ϕ that defines $Isom(\mathcal{A})$ also has a model $\mathcal{B}$ where $\mathcal{B}$'s domain subsumes 2^A. (As before, 2^A denotes the powerset of A.) Structure $\mathcal{B}$ is a model for $Th(\mathcal{A})$ and therefore $Th(\mathcal{B}) = Th(\mathcal{A})$ (see Theorem 1.4.6 on page 48). But for cardinality reasons, $\mathcal{A}$ and $\mathcal{B}$ cannot be isomorphic.

This shows that if $\mathcal{A}$ is infinite then the class $Isom(\mathcal{A})$ of structures that are isomorphic to $\mathcal{A}$ is *not* FO-definable. However, given a finite vocabulary then for any *finite* structure $\mathcal{A}$ the class $Isom(\mathcal{A})$ is FO-definable and $Th(\mathcal{A})$ is finitely axiomatizable. (See Exercises.) ■

1.6.1 Properties that are not FO-definable

In the following we provide several examples that illustrate the limitations of FOL to characterize mathematical structures. We start with the observation that the class of all finite structures is not FO-definable, no matter which vocabulary is used.

Theorem 1.6.4 (The class of finite structures is not FO-definable). *There is no vocabulary Voc and set $\mathfrak{F}$ consisting of FOL-formulas over Voc such that for all structures $\mathcal{A}$ for Voc:*

$\mathcal{A}$ is a model for $\mathfrak{F}$ iff $\mathcal{A}$ is finite.

Proof. Theorem 1.6.4 can be derived from the upward Löwenheim-Skolem theorem ("from finite to infinite models", see Theorem 1.5.2 on page 72). For amusement, let us recall the argument. Assume such a formula-set $\mathfrak{F}$ exists. As in the proof of Theorem 1.5.2 we use the FOL-sentences

$$\psi_n = \exists x_1 \ldots \exists x_n. \bigwedge_{1 \leqslant i < j \leqslant n} (x_i \neq x_j)$$

to encode structures with n or more elements. Then, $\mathfrak{F}^+ = \mathfrak{F} \cup \{\psi_n : n \geqslant 2\}$ is finitely satisfiable, and hence, satisfiable. Let $\mathcal{A}$ be a model for $\mathfrak{F}^+$. Then, $\mathcal{A}$ is infinite (since $\mathcal{A} \models \psi_n$ for all $n \geqslant 2$) and $\mathcal{A}$ is a model for $\mathfrak{F}$. Contradiction. □

In the next theorem we consider FOL-formulas with equality over the empty vocabulary. Thus, terms are just variables and formulas are built by the atoms $x = y$ where x and y are variables, the boolean connectives and quantification. Structures for the empty vocabulary are just nonempty sets.

Theorem 1.6.5 (FOL over the empty vocabulary fails to characterize even structures). *There is* no *FOL-sentence ϕ over the empty vocabulary such that for each finite structure, i.e., nonempty finite set A:*

$$A \models \phi \quad \textit{iff} \quad |A| \textit{ is even.}$$

Proof. Assume that ϕ is a FOL-sentence over the empty vocabulary such that, for all finite structures (i.e., nonempty finite sets) A we have: $A \models \phi$ iff $|A|$ is even. Let ψ_n be a FOL-sentence that characterizes sets with n or more elements, see Theorem 1.6.4. Let

$$\mathfrak{F}_1 \stackrel{\text{def}}{=} \{\phi\} \cup \{\psi_n : n \geqslant 2\},$$
$$\mathfrak{F}_2 \stackrel{\text{def}}{=} \{\neg\phi\} \cup \{\psi_n : n \geqslant 2\}.$$

Both $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are satisfiable. Let us see why this holds. We apply the compactness theorem (see page 71) and show that all finite subsets of $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are satisfiable.

- If $\mathfrak{G}$ is a finite subset of $\mathfrak{F}_1$ then we pick some $m \geqslant 1$ such that none of the formulas ψ_n, $n > m$, belongs to $\mathfrak{G}$. Hence, $\mathfrak{G} \subseteq \{\phi\} \cup \{\psi_2, \psi_3, \ldots, \psi_m\}$. But then any finite set A where $|A| = 2k$ for some integer k with $k \geqslant m/2$ is a model for $\mathfrak{G}$.
- Similarly, for any finite subset $\mathfrak{G}$ of $\mathfrak{F}_2$ there is some m such that all finite sets of odd size $\geqslant m$ are models for $\mathfrak{G}$.

Hence, $\mathfrak{F}_1$ and $\mathfrak{F}_2$ are finitely satisfiable, and therefore satisfiable (compactness property).

The downward Löwenheim-Skolem Theorem (Theorem 1.5.5 on page 75) yields the existence of countable models A_1 and A_2 for $\mathfrak{F}_1$ and $\mathfrak{F}_2$, respectively. Clearly, since $\{\psi_n : n \geqslant 2\} \subseteq \mathfrak{F}_i$, each model for $\mathfrak{F}_i$ is infinite. Thus, A_1 and A_2 are infinite and countable sets. (Recall that we deal with the empty vocabulary. Thus, structures are sets.) But then A_1 and A_2 are isomorphic. Hence, they fulfill the same FOL-formulas over the empty vocabulary. As $A_1 \models \phi$, we also have $A_2 \models \phi$. This is impossible since A_2 is a model for $\mathfrak{F}_2$, and hence, $A_2 \models \neg\phi$. □

FOL also fails to characterize the natural numbers. This is a consequence of the upward Löwenheim-Skolem theorem (Theorem 1.5.4 on page 73) which yields that any formula ϕ which is satisfiable over a structure with the domain $\mathbb{N}$ also has uncountable models. In the following theorem we will provide a direct proof and show that $\mathbb{N}$ together with its natural successor function $n \mapsto n+1$ cannot be characterized in FOL, even when we stick to countable structures. To prove this, we assume a vocabulary Voc_{nat} consisting of a constant symbol 0 (for the natural number 0) and an unary function symbol *succ* (for the successor function $n \mapsto n+1$).

Theorem 1.6.6 (The natural numbers cannot be characterized in FOL). *There is* no *satisfiable set* $\mathfrak{F}$ *of FOL-sentences over* Voc_{nat} *such that* $(\mathbb{N}, succ, 0) \models \mathfrak{F}$ *and all countable models for* $\mathfrak{F}$ *are isomorphic to* $(\mathbb{N}, succ, 0)$.

Proof. The statement of Theorem 1.6.6 is a consequence of the existence of non-standard countable models for the arithmetic (see Theorem 1.5.7 on page 81), rephrased for the function symbol *succ* and the constant symbol 0. For the sake of completeness, we provide the proof which uses roughly the same arguments as the proof for Theorem 1.5.7.

Let $\mathcal{N}$ be the structure $(\mathbb{N}, succ, 0)$. Assume there is a satisfiable formula-set $\mathfrak{F} \subseteq Th(\mathcal{N})$ such that all countable models for $\mathfrak{F}$ are isomorphic to $\mathcal{N}$. Let x be a variable and let

$$\mathfrak{G} \stackrel{\text{def}}{=} \mathfrak{F} \cup \{ x \neq succ^n(0) : n \geqslant 1 \}.$$

Every finite subset $\mathfrak{G}'$ of $\mathfrak{G}$ is satisfiable, since there is some formula "$x \neq succ^n(0)$" which is not contained in $\mathfrak{G}'$. Then, $(\mathcal{N}, [x := n]) \models \mathfrak{G}'$. Hence, $\mathfrak{G}$ is finitely satisfiable. By the compactness theorem (see page 71), $\mathfrak{G}$ is satisfiable. The downward Löwenheim-Skolem theorem then yields the existence of a countable model $(\mathcal{A}, \mathcal{V})$ for $\mathfrak{G}$.

However, there is no (countable) model $(\mathcal{A}, \mathcal{V})$ for $\mathfrak{G}$ such that structure $\mathcal{A}$ is isomorphic to $\mathcal{N}$. The reason is that $\mathcal{A}$ contains an element, namely $a = \mathcal{V}(x)$, that is not obtained by finite applications of $succ^{\mathcal{A}}$ to $0^{\mathcal{A}}$, while all elements of $\mathbb{N}$ have this form. Therefore, the structure of any model for $\mathfrak{G}$ induces a model for $\mathfrak{F}$ that is not isomorphic to $\mathcal{N}$. Contradiction. $\Box$

A further example that illustrates the limitations of FOL to formalize mathematical structures is the impossibility to provide a FOL-characterization of periodic groups. A group $\mathcal{A} = (A, \circ, e)$ is called *periodic* iff for each element $a \in A$ there exists a positive integer $n \geqslant 1$ such that $a^n = e$. Here, e denotes the neutral element of $\mathcal{A}$ and

$$a^n = \underbrace{a \circ \ldots \circ a}_{n \text{ times}},$$

i.e., $a^0 = e$ and $a^{n+1} = a \circ a^n$. The smallest positive number n such that $a^n = e$ is called the *order* of a. The order of a perodic group is the maximal order of its elements. For instance, the modulo-groups $(\{1, \ldots, p-1\}, *)$ where p is a prime number $\geqslant 3$ and $*$ denotes multiplication modulo p are periodic as for all integers a with $1 \leqslant a < p$ we have

$$a^{p-1} \bmod p = 1,$$

and there is an element a of order $p-1$. Another example are the groups with the elements $0, 1, \ldots, m-1$ where $m \in \mathbb{N}$, $m \geqslant 2$, and where addition modulo m serves as group operator.

I.e., we deal here with the operation $i \circ j = (i+j) \bmod m$. There are also infinite periodic groups. For instance,

$$\{ e^{2\pi i q} : q \text{ is rational} \}$$

with standard multiplication of complex numbers constitutes a group which is periodic, but not cyclic. Here, $e^{2\pi i q}$ is the complex number with real part $\cos(2\pi q)$ and imaginary part $\sin(2\pi q)$, i.e.:

$$e^{2\pi i q} = \cos(2\pi q) + i \cdot \sin(2\pi q)$$

where i is the imaginary unity (complex number with $i^2 = -1$). The elements $e^{2\pi i x}$ with $x \in \mathbb{R}$ are exactly the elements of the unit circle, and thus, the elements $e^{2\pi i q}$ with $q \in \mathbb{Q}$ are dense on the unit circle. Multiplication of $e^{2\pi i q}$ and $e^{2\pi i p}$ yields $e^{2\pi i (q+p)}$. As $e^{2\pi i k} = 1$ iff k is an integer, we get:

- $\left(e^{2\pi i \frac{k}{n}}\right)^n = 1$ for all integers k and positive natural numbers n,
- the order of $e^{2\pi i \frac{1}{n}}$ (an n-th root of unity) is n for each positive integer n.

Thus, all elements are of finite order (which yields the periodicity) and there is no lower bound for the order.

Theorem 1.6.7 (The class of periodic groups is not FO-definable). *There is no set $\mathfrak{F}$ of FOL-sentences over the vocabulary Voc$_{group}$ such that the models for $\mathfrak{F}$ are exactly the periodic groups.*

Recall that the vocabulary *Voc*$_{\text{group}}$ has a binary function symbol $\circ$ for the group operation and a constant symbol e for the neutral element (cf. Example 1.6.2 on page 83).

Proof. Assume by contradiction that there exists such a set $\mathfrak{F}$ of FOL-sentences. Let

$$\mathfrak{G} \stackrel{\text{def}}{=} \mathfrak{F} \cup \{ x^n \neq e : n \in \mathbb{N}, n \geqslant 1 \}.$$

We show that $\mathfrak{G}$ is finitely satisfiable. Each finite subset of $\mathfrak{G}$ is contained in $\mathfrak{G}_m \stackrel{\text{def}}{=} \mathfrak{F} \cup \{ x^n \neq e : 1 \leqslant n < m \}$ for some $m \geqslant 1$. A model for $\mathfrak{G}_m$ is obtained by $(G, [x := a])$ where G is a periodic group of order at least m and a one of its elements with maximal order. Thus, $\mathfrak{G}$ is finitely satisfiable, and hence, satisfiable (compactness property).

On the other hand, for each model $(\mathcal{A}, [x := a])$ for $\mathfrak{G}$: structure $\mathcal{A} = (A, \circ, e)$ is a periodic group and a an element in A such that $a^n \neq e$ for all $n \geqslant 1$. This, however, is impossible by the definition of periodic groups. □

1.6.2 FO-definable graph properties

We now discuss the FO-definability of graph properties. To express properties of graphs in FOL, we use the vocabulary *Voc*$_{\text{graph}}$ which consists of a binary predicate symbol E for the edge-relation. The structures of *Voc*$_{\text{graph}}$ are (possibly infinite) directed graphs $\mathcal{G} = (A, E)$ consisting of a node-set A (the domain) and the edge-relation $E \subseteq A \times A$ (the interpretation of the edge predicate symbol). Here and in the sequel, we often use the same symbol for predicate symbols and their semantics in a given structure. I.e., we identify the predicate symbol E of Voc_{graph} with the edge-relation E of the given graph $\mathcal{G}$. For instance,

$$\phi_1 = \forall x \exists ! y . E(x,y)$$

characterizes all graphs where each node has outdegree 1. (Recall the short hand notation $\exists ! y$ for "there exists exactly one y" from page 14.) The *outdegree* of a node a is the number of edges (a,b) emanating from a, while the *indegree* is the number of edges (b,a) leading to a. Thus, the condition stating that each node has outdegree 1 means that each node has exactly one successor. The formulas

$$\phi_2 = \forall x \forall y . (E(x,y) \to E(y,x)) \text{ and } \phi_3 = \forall x \forall y \forall z . (E(x,y) \wedge E(y,z) \to E(x,z))$$

state the symmetry and transitivity of the edge relation, respectively. The three graph-properties induced by ϕ_1, ϕ_2 and ϕ_3 can be checked in polynomial time for finite graphs. In fact, FOL can only express such simple graph properties, as we will show now.

Given a formula ϕ over Voc_{graph} the problem ϕ-GRAPH takes as input a finite interpretation (i.e., a finite directed graph $\mathcal{G} = (A, E)$ and a variable valuation for the free variables of ϕ) and asks whether ϕ holds for this interpretation. In the sequel, if $\phi = \phi(x_1, \ldots, x_k)$ then we write $(\mathcal{G}, a_1, \ldots, a_k)$ for the interpretation with structure $\mathcal{G}$ and the variable valuation that assigns node a_i to variable x_i. Thus, the task of ϕ-GRAPH is to check whether

$$(\mathcal{G}, a_1, \ldots, a_k) \models \phi(x_1, \ldots, x_k)$$

for a given formula ϕ over Voc_{graph} and a finite directed graph $\mathcal{G}$ and distinguished nodes $a_1, \ldots, a_k$ in $\mathcal{G}$. This problem is known as the *model checking problem* as the task is to check whether the given interpretation is a model for ϕ. It should be noticed that the formula ϕ is viewed to be fixed, and not part of the input. Thus, the cost function for the time complexity of an algorithm for ϕ-GRAPH just depends on the size of the input graph, while the length of ϕ can be treated as a constant.

Theorem 1.6.8 (FO-definable graph properties are in PTIME). *For each formula ϕ over Voc_{graph} the problem ϕ-GRAPH is solvable in time polynomial in the size of the input-graph. (Formula ϕ is viewed to be fixed and the size of a finite graph is the total number of nodes and edges.)*

Proof. It suffices to provide a (deterministic) polynomially time-bounded algorithm that solves ϕ-GRAPH. Let $Free(\phi) = \{x_1, \ldots, x_k\}$ and let $\mathcal{I} = (\mathcal{G}, a_1, \ldots, a_k)$ be an interpretation consisting of a finite graph $\mathcal{G} = (A, E)$ and a variable valuation for the free variables of ϕ. The goal is to check whether $\mathcal{I} \models \phi$. For this, we use a recursive approach.

- If ϕ is an atomic formula $E(x_i, x_j)$ or $x_i = x_j$ then we simply check whether (a_i, a_j) is an edge in $\mathcal{G}$ or whether $a_i = a_j$. Recall that Voc_{graph} does not have any function symbols. Hence, all terms are variables.
- If $\phi = \neg \psi$ then we recursively check whether $\mathcal{I} \models \psi$ and return "no" if $\mathcal{I} \models \psi$ and "yes" if $\mathcal{I} \not\models \psi$.

- The treatment of $\phi = \phi_1 \wedge \phi_2$ is analogous. We call recursively the subroutines for checking whether $\mathcal{I} \models \phi_1$ and $\mathcal{I} \models \phi_2$.
- Let us now assume that $\phi = \forall x.\psi$. Then, we consider all nodes $a \in A$ and apply the algorithm recursively to check whether $\mathcal{I}[x := a]$ is a model for ψ. If some node a with $\mathcal{I}[x := a] \not\models \psi$ has been found then we return "no". Otherwise, i.e., if $\mathcal{I}[x := a] \models \psi$ for all nodes a, then we return "yes".

Let $n = |A|$ be the number of nodes in $\mathcal{G}$ and $m = |E|$ the number of edges. For atomic subformulas there are no recursive calls. For subformulas where the outmost operator is negation or conjunction, there are at most two recursive calls, while for subformulas with a universal quantifier at the top level, there are n recursive calls. Thus, assuming $n \geqslant 2$, then for each occurrence of a subformula there are at most n recursive calls. This yields the asymptotic upper bound $\mathcal{O}\big(n^{2|\phi|+1}\big)$ for the total number of recursive calls as we have $|\mathit{subf}(\phi)| \leqslant 2|\phi|+1$. Here, $\mathit{subf}(\phi)$ is viewed as a multiset that "counts" the number of occurrences of subformulas in ϕ.

Atomic formulas of the form $x = y$ can be treated in constant time. Supposing a matrix-representation of the edge-relation E, we can deal with the atomic formulas $E(x,y)$ in constant time as well. For a representation of E by adjacency lists or by a list of node-pairs, checking atomic formulas $E(x,y)$ can be done in time $\mathcal{O}(m)$. Hence, we can establish $\mathcal{O}\big((n+m)^{2|\phi|+1}\big)$ as an upper bound for the time complexity of the procedure. Since ϕ is viewed to be fixed and $n+m$ is the size of $\mathcal{G}$, this gives a polynomial bound for the time complexity. $\square$

The model checking procedure sketched in the proof of Theorem 1.6.8 works for any finite structure for a relational vocabulary, provided that membership of a tuple $(a_1,\ldots,a_k)$ to the relation $P^{\mathcal{A}}$ assigned to an k-ary predicate symbol P is decidable in polynomial time, which is the case, e.g., if $P^{\mathcal{A}}$ is represented as a list of k-tuples. Hence:

Theorem 1.6.9 (Model checking for FOL-formulas over relational vocabularies). *The model checking problem for FOL-formulas over relational vocabularies and finite structures is in the complexity class P, provided that the formula is viewed to be fixed.*

Let us return to the problem ϕ-GRAPH. By Theorem 1.6.8, for all FOL-formulas over $\mathit{Voc}_{\text{graph}}$ the problem of checking an FOL-definable property for a given finite graph is solvable by a polynomially time-bounded algorithm. Hence, FOL fails to provide characterizations for computationally hard graph problems, such as the Hamilton-path problem (which asks for a path that visits each node exactly once and is known to be *NP*-complete), unless $P = NP$. First-order logic is even not expressive enough to formalize all graph-properties that are checkable in polynomial time, as it fails to express reachability, and related problems such as strong connectivity. The following theorem shows that the class of directed graphs with a distinguished source node a and a distinguished target node b such that b is reachable from a is not FO-definable.

Theorem 1.6.10 (Reachability is not FO-definable). *There is no FOL-formula $\phi(x,y)$ over $\mathit{Voc}_{\text{graph}}$ such that for all graphs $\mathcal{G}$ and nodes a, b in $\mathcal{G}$ we have:*

$$(\mathcal{G},a,b) \models \phi(x,y) \quad \textit{iff} \quad b \textit{ is reachable from } a \textit{ in } \mathcal{G}$$

Recall that the notation $\phi(x,y)$ indicates that x and y are distinct variables and $\mathit{Free}(\phi) \subseteq \{x,y\}$ and the notation $(\mathcal{G},a,b)$ means the interpretation with structure $\mathcal{G}$ and a variable valuation that assigns node a to variable x and node b to y.

Proof. Let us assume by contradiction that such a formula $\phi(x,y)$ exists. Then,

$$\psi_0 \stackrel{\text{def}}{=} \forall x \forall y . \phi(x,y)$$

states that the given graph is *strongly connected*, i.e., all nodes are reachable from each other. Let

$$\psi_1 \stackrel{\text{def}}{=} \forall x \exists ! y . E(x,y) \quad \text{and} \quad \psi_2 \stackrel{\text{def}}{=} \forall x \exists ! y . E(y,x)$$

Then, ψ_1 states that all nodes have outdegree 1, while ψ_2 states that all nodes have indegree 1. Thus, $\psi_1 \wedge \psi_2$ characterizes all graphs where each node has exactly one successor and exactly one predecessor. But then, the sentence

$$\psi \stackrel{\text{def}}{=} \psi_0 \wedge \psi_1 \wedge \psi_2$$

characterizes all *simple cycles*, i.e., graphs consisting of a cycle that visits each node exactly once. This even holds for all strongly connected graphs where the outdegree of all nodes is 1 or, alternatively, where all nodes have indegree 1. Thus, one of the formulas ψ_1 or ψ_2 could be dropped. Let us see why. Let G be a graph.

- If each node has outdegree 1, then for each node a in $\mathcal{G}$ there is essentially one path starting in a. By "essentially one path" we mean that for each two paths π_1 and π_2 starting in a, either π_1 is a prefix of π_2, or vice versa.
- If $\mathcal{G}$ is strongly connected, then for each pair (a,b) of nodes in $\mathcal{G}$ there are (finite) paths from a to b and from b to a. Thus, there is a cycle that runs through a and b.

Putting things together, we get that each strongly connected graph $\mathcal{G}$ where each node has outdegree 1 consists of a simple cycle. To see why, pick some nodes a in $\mathcal{G}$ and a simple cycle $a_0\, a_1 \ldots a_k$ where $a_0 = a_k = a$. Given that each node has precisely one successor, all finite paths emanating from a run through that simple cycle several times followed by a finite prefix $a_0\, a_1 \ldots a_i$ of that cycle. Since $\mathcal{G}$ is strongly connected, all nodes in $\mathcal{G}$ must be visited in that cycle. Thus, $\{a_1, \ldots, a_k\}$ is the node set of $\mathcal{G}$ and $\{(a_{i-1}, a_i) : 1 \leqslant i \leqslant k\}$ its edge relation.

Since there are simple cycles with an arbitrary (finite) number of nodes, ψ has finite models of each size. The upward Löwenheim-Skolem theorem (Theorem 1.5.2 on page 72) yields that ψ has an infinite model too. But there are no infinite simple cycles. Contradiction. $\square$

1.6.3 Ehrenfeucht-Fraïssé games

Ehrenfeucht-Fraïssé games yield a powerful and elegant tool to study the expressiveness of logics, in particular when reasoning over finite structures. The purpose of such games is to provide a characterization of when two structures over the same vocabulary fulfill the same formulas of a certain type.

The general concept of such games is rather simple: we have two players, called *spoiler* and *duplicator*. The board of the game are two structures $\mathcal{A}$ and $\mathcal{B}$ for some fixed vocabulary with domains A and B, respectively. The goal of the spoiler is to show that the structures are different, while the duplicator aims at proving the equivalence of $\mathcal{A}$ and $\mathcal{B}$. In the classical approach for FOL the game proceeds in *rounds*, where each round consists of the following steps:

1. The spoiler makes the first move by selecting one of the structure $\mathcal{A}$ or $\mathcal{B}$ and an element in the chosen structure, i.e., an element $a \in A$ or an element $b \in B$.

2. The duplicator responds by picking an element in the other structure.

The *outcome* of a k-round play is given by a pair $\langle \overline{a}, \overline{b} \rangle$ where $\overline{a} = (a_1, \ldots, a_k) \in A^k$ and $\overline{b} = (b_1, \ldots, b_k) \in B^k$ and where the i-th components a_i and b_i of $\overline{a}$ and $\overline{b}$, respectively, represent the choices of the two players in round i. To define who wins the game after k rounds (for some fixed k), we need some notations.

For technical reasons, we assume here that the underlying vocabulary is *finite* and *relational*, i.e., contains no function symbols of arity 1 or more. Let

$$Const \;=\; Func_0 \;=\; \{c_1, \ldots, c_\ell\}$$

be the set of constant symbols in *Voc*, where $c_1, \ldots, c_\ell$ are supposed to be pairwise distinct. In the following, we often use the tuple notation for the meanings of the constant symbols:

$$\overline{c}^{\mathcal{A}} \;\stackrel{\text{def}}{=}\; (c_1^{\mathcal{A}}, \ldots, c_\ell^{\mathcal{A}}) \quad \text{and} \quad \overline{c}^{\mathcal{B}} \;\stackrel{\text{def}}{=}\; (c_1^{\mathcal{B}}, \ldots, c_\ell^{\mathcal{B}})$$

Definition 1.6.11 (Partial isomorphism). Let $\mathcal{A}$ and $\mathcal{B}$ be structures for the same finite and relational vocabulary *Voc* with domains $A = Dom^{\mathcal{A}}$ and $B = Dom^{\mathcal{B}}$, and let $\overline{a} = (a_1, \ldots, a_k) \in A^k$, $\overline{b} = (b_1, \ldots, b_k) \in B^k$. The pair $\langle \overline{a}, \overline{b} \rangle$ is said to define a *partial isomorphism* between $\mathcal{A}$ and $\mathcal{B}$ iff the following conditions hold:

(1) $a_i = a_j$ iff $b_i = b_j$ for all $1 \leqslant i < j \leqslant k$

(2) for all constant symbols $c \in Const$ and all $i \in \{1, \ldots, k\}$: $a_i = c^{\mathcal{A}}$ iff $b_i = c^{\mathcal{B}}$

(3) for each n-ary predicate symbol P and each index-tuple $(i_1, \ldots, i_n) \in \{1, \ldots, k\}^n$:

$$(a_{i_1}, \ldots, a_{i_n}) \in P^{\mathcal{A}} \text{ iff } (b_{i_1}, \ldots, b_{i_n}) \in P^{\mathcal{B}}$$

■

Thus, $\langle \overline{a}, \overline{b} \rangle$ defines a partial isomorphism between $\mathcal{A}$ and $\mathcal{B}$ iff the function $\{a_1, \ldots, a_k\} \to \{b_1, \ldots, b_k\}$, $a_i \mapsto b_i$ for $1 \leqslant i \leqslant k$, is well-defined and bijective (first item) and yields an isomorphism between the substructures induced by $\{a_1, \ldots, a_k\}$ and $\{b_1, \ldots, b_k\}$ for the vocabulary consisting of the predicate symbols in *Voc* and a subset of *Const* (second and third item).

Example 1.6.12 (Partial isomorphism). Let us consider a purely relational vocabulary with a single binary predicate symbol and the structures $\mathcal{A} = (\mathbb{Q}, \leqslant)$ and $\mathcal{B} = (\mathbb{N}, \leqslant)$.

- The pair $\langle\, (\frac{7}{3}, -\frac{13}{3}, \frac{3}{4}), (4, 1, 0)\,\rangle$ does not define a partial isomorphism, since

$$a_2 = -\tfrac{13}{3} < \tfrac{3}{4} = a_3, \quad \text{while} \quad b_2 = 1 > 0 = b_3.$$

- The pair $\langle\, (\frac{7}{3}, -\frac{13}{3}, \frac{3}{4}), (4, 0, 1)\,\rangle$ defines a partial isomorphism as

$$a_2 = -\tfrac{13}{3} < a_3 = \tfrac{3}{4} < a_1 = \tfrac{7}{3} \quad \text{and} \quad b_2 = 0 < b_3 = 1 < b_1 = 4.$$

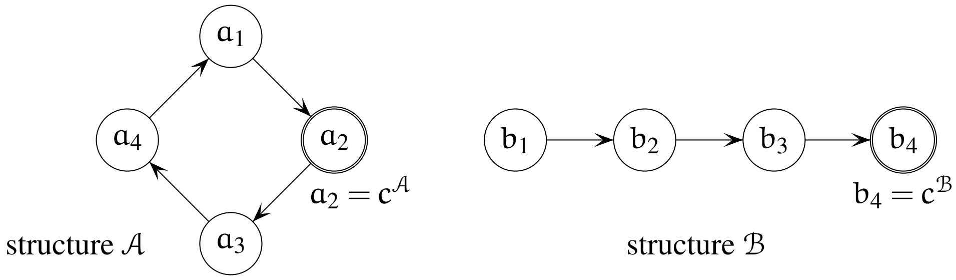

Figure 13: Two pointed graphs $\mathcal{A}$ and $\mathcal{B}$

If we add a constant symbol c to the vocabulary and deal with the structures $\mathcal{A} = (\mathbb{Q}, \leqslant, \frac{3}{4})$ and $\mathcal{B} = (\mathbb{N}, \leqslant, 0)$ then the pairs

$$\langle (\tfrac{7}{3}, -\tfrac{13}{3}, \tfrac{3}{4}), (4,0,1) \rangle \qquad \text{and} \qquad \langle (\tfrac{7}{3}, -\tfrac{13}{3}, \tfrac{3}{4}), (4,2,3) \rangle$$

do not define partial isomorphisms. In both cases, the third component in the $\mathcal{A}$-tuple is $a_3 = \frac{3}{4} = c^{\mathcal{A}}$, while the third component of the $\mathcal{B}$-tuple is different from $c^{\mathcal{B}} = 0$.

For another example, we regard the vocabulary of pointed graphs (i.e., a vocabulary with a single binary predicate symbol E and a constant symbol c) and the pointed graphs $\mathcal{A}$ and $\mathcal{B}$ shown in Figure 13. That is, $\mathcal{A} = (\{a_1, a_2, a_3, a_4\}, E^{\mathcal{A}}, a_2)$ where the edges constitute a simple cycle and $c^{\mathcal{A}} = a_2$. Structure $\mathcal{B}$ consists of the graph with the node-set $\{b_1, b_2, b_3, b_4\}$, the edge relation $E^{\mathcal{B}} = \{(b_i, b_{i+1}) : 1 \leqslant i < 4\}$ and interpretes the constant symbol c by node b_4.

- The pair $\langle (a_4, a_2), (b_1, b_4) \rangle$ defines a partial isomorphism since a_4, a_2 and b_1, b_4 are not connected via an edge and the second component in both tuples is the meaning of the constant symbol c as we have $a_2 = c^{\mathcal{A}}$ and $b_4 = c^{\mathcal{B}}$.
- The pair $\langle (a_4, a_1), (b_1, b_2) \rangle$ also defines a partial isomorphism, since both tuples (a_4, a_1) and (b_1, b_2) are edges and $c^{\mathcal{A}} \notin \{a_4, a_1\}$, $c^{\mathcal{B}} \notin \{b_1, b_2\}$.
- The pair $\langle (a_3, a_4), (b_3, b_4) \rangle$ does not define a partial isomorphism as we have $a_4 \neq c^{\mathcal{A}}$, while $b_4 = c^{\mathcal{B}}$. ■