Another important application of Corollary 2.6.20 is that it provides the basis for proving the impossibility to express certain properties by MSO-formulas. In the following theorem we consider the *Hamilton cycle problem* for finite directed graphs and consider MSO-formulas over the vocabulary $\mathit{Voc}_{\text{graph}}$. Remind that $\mathit{Voc}_{\text{graph}}$ consists of a single binary predicate symbol E for the edge relation. Thus, MSO-formulas over $\mathit{Voc}_{\text{graph}}$ are built by the atoms $x = y$, $x \in X$, $E(x,y)$, conjunction, negation, first-order quantification and second-order quantification over set variables. E.g., the formula $\phi_{\text{reach}}(x,y)$ considered in Example 2.4.1 on page 150 is a MSO-formula over $\mathit{Voc}_{\text{graph}}$.

Recall that a *Hamilton cycle* in a finite directed graph $\mathcal{G}$ denotes a cycle $a_0\, a_1 \ldots a_n$ in $\mathcal{G}$ that visits each node of $\mathcal{G}$ exactly once; except for the first and last node $a_0 = a_n$, which appears twice. The Hamilton cycle problem asks whether a given graph has a Hamilton cycle.

Theorem 2.6.22 (Hamilton cycle property not MSO-definable). *There is no MSO-sentence ϕ over the vocabulary Voc_{graph} that characterizes the Hamilton cycle property, i.e., such that for each finite directed graph $\mathcal{G}$:*

$$\mathcal{G} \models \phi \quad \textit{iff} \quad \mathcal{G} \textit{ has a Hamilton cycle.}$$

Proof. Suppose by contradiction that there exists such a MSO-sentence ϕ over $\mathit{Voc}_{\text{graph}}$. We show that under this assumption, it is possible to define a MSO-sentence θ over (some subvocabulary of) $\mathit{Voc}_{\Sigma,\text{graph}}$ such that $\mathfrak{L}(\theta)$ agrees with the non-regular language $\{a^n b^n : n \in \mathbb{N}\}$. This contradicts the statement of Corollary 2.6.20.

We consider the bipartite graph $K_{n,m}$ with node set

$$A_{n,m} \;=\; \{\ell_1, \ldots, \ell_n, r_1, \ldots, r_m\}$$

and the edges (ℓ_i, r_j), (r_j, ℓ_i) for all $1 \leqslant i \leqslant n$ and $1 \leqslant j \leqslant m$. We then have:

$$K_{n,m} \text{ has a Hamilton cycle} \quad \text{iff} \quad n = m \tag{*}$$

Proof of (*). "$\Longleftarrow$": For $n = m$, the path $\ell_1\, r_1\, \ell_2\, r_2 \ldots \ell_n\, r_n\, \ell_1$ is a Hamilton cycle.

"$\Longrightarrow$": suppose that $v_0\, v_1\, v_2 \ldots v_k$ is a Hamilton cycle in $K_{n,m}$. Then, $k = n+m$. W.l.o.g., we can assume that $v_0 = \ell_1$. Since each path in $K_{n,m}$ is an alternating sequence of ℓ-nodes and r-nodes, we get that k is even (otherwise $v_k = v_{n+m} = \ell_1$ could not hold) and

$$\begin{aligned} \{v_0, v_2, v_4, \ldots, v_{k-2}\} &= \{\ell_1, \ell_2, \ldots, \ell_n\} \\ \{v_1, v_3, v_5, \ldots, v_{k-1}\} &= \{r_1, r_2, \ldots, r_m\} \end{aligned}$$

Since the nodes v_i for $i = 0, 1, \ldots, k-1$, are pairwise distinct, we get $n = m$. This completes the proof of (*). $\rfloor$

Let $\Sigma = \{a, b\}$ and let ϕ' be the MSO-formula over $\mathit{Voc}_{\Sigma,\text{graph}}$ that results from ϕ by replacing the atoms $E(x,y)$ with $P_a(x) \leftrightarrow P_b(y)$. Then, ϕ' is a MSO-formula that uses as atoms only the formulas $x = y$, $x \in X$ and $P_c(x)$ for $c \in \{a, b\}$, but does not use the atoms $E(x,y)$. That is, ϕ' is a MSO formula over the vocabulary Voc_Σ consisting of the monadic predicate symbols P_a and P_b. For example, if

$$\phi \;=\; \forall x \forall x' \forall z.\, \big(\, E(x,z) \wedge E(x',z) \;\rightarrow\; \neg E(x,x') \,\big)$$

then

$$\phi' \;=\; \forall x \forall x' \forall z.\, \Big(\underbrace{\big(P_a(x) \leftrightarrow P_b(z)\big)}_{E(x,z)} \wedge \underbrace{\big(P_a(x') \leftrightarrow P_b(z)\big)}_{E(x',z)} \rightarrow \neg \underbrace{\big(P_a(x) \leftrightarrow P_b(x')\big)}_{E(x,x')} \Big)$$

We now show that:

$$K_{n,m} \models \phi \quad \text{iff} \quad \text{Graph}(a^n b^m) \models \phi' \tag{**}$$

To establish (**), we show by structural induction on all MSO-formulas ψ over $\mathit{Voc}_{\text{graph}}$ (with possibly free variables) built by the atoms $x = y$, $x \in X$ and $E(x,y)$ that

$$(K_{n,m}, \mathcal{V}) \models \psi \quad \text{iff} \quad (\text{Graph}(a^n b^m), \mathcal{V}') \models \psi' \tag{***}$$

where ψ' arises from ψ by replacing any atom $E(x,y)$ with $P_a(x) \leftrightarrow P_b(y)$ and where $\mathcal{V}$ and $\mathcal{V}'$ are variable valuations that are related in the following way:

- for $1 \leqslant i \leqslant n$: $\quad \mathcal{V}(x) = \ell_i \quad$ iff $\quad \mathcal{V}'(x) = i$
- for $1 \leqslant j \leqslant m$: $\quad \mathcal{V}(x) = r_j \quad$ iff $\quad \mathcal{V}'(x) = n{+}j$
- $\mathcal{V}'(X) \;=\; \big\{ i \in \{1,\ldots,n\} : \ell_i \in \mathcal{V}(X) \big\} \cup \big\{ n+j \in \{n{+}1,\ldots,n{+}m\} : r_j \in \mathcal{V}(X) \big\}$

Thus, $\mathcal{V}' = h \circ \mathcal{V}$ where h is the bijection from the node-set of $K_{n,m}$ to the word positions of $a^n b^m$ which is given by:

$$h(\ell_i) \stackrel{\text{def}}{=} i, \quad h(r_j) \stackrel{\text{def}}{=} n{+}j$$

Proof of (***). The basis of induction is obvious for the formula *true*. For the atomic formula $x \in X$ the relation between $\mathcal{V}$ and $\mathcal{V}'$ yields:

$$\begin{aligned} (K_{n,m}, \mathcal{V}) \models\; x \in X \quad &\text{iff} \quad \mathcal{V}(x) \in \mathcal{V}(X) \\ &\text{iff} \quad \mathcal{V}'(x) \in \mathcal{V}'(X) \\ &\text{iff} \quad (\text{Graph}(a^n b^m), \mathcal{V}') \models\; x \in X \end{aligned}$$

The treatment of atomic formulas of the form $x = y$ is similar. For the atoms $E(x,y)$ we have:

$$\begin{aligned} & (K_{n,m}, \mathcal{V}) \models E(x,y) \\ \text{iff} \quad & (\mathcal{V}(x), \mathcal{V}(y)) \in \big\{ (\ell_i, r_j), (r_j, \ell_i) : 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m \big\} \\ \text{iff} \quad & (\mathcal{V}'(x), \mathcal{V}'(y)) \in \big\{ (i, n{+}j), (n{+}j, i) : 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m \big\} \\ \text{iff} \quad & (\text{Graph}(a^n b^m), \mathcal{V}') \models \big(P_a(x) \wedge P_b(y) \big) \vee \big(\neg P_a(x) \wedge \neg P_b(y) \big) \\ \text{iff} \quad & (\text{Graph}(a^n b^m), \mathcal{V}') \models P_a(x) \leftrightarrow P_b(y) \end{aligned}$$

The step of induction is straightforward for $\psi = \psi_1 \wedge \psi_2$ and $\psi = \neg\psi_0$. For first- or second-order quantification, i.e., $\psi = \forall x.\psi_0$ or $\psi = \forall X.\psi_0$, we use the fact that h is a bijection. Thus, if a ranges over all nodes in $K_{n,m}$ then $h(a)$ ranges over all word positions $i \in \{1,\ldots,n{+}m\}$, and vice versa. The argument for first-order quantification (i.e., for $\psi = \forall x.\psi_0$ where x is a FO-variable) is now as follows. The induction hypothesis yields:

$$\begin{aligned} & (\mathrm{Graph}(a^n b^m), \mathcal{V}'[x := i]) \models \psi_0' \quad \text{for all word positions } i \in \{1,\ldots,n+m\} \\ \text{iff} \quad & (K_{n,m}, \mathcal{V}[x := a]) \models \psi_0 \quad \text{for all nodes } a \text{ in } K_{n,m} \end{aligned}$$

Thus:

$$(\mathrm{Graph}(a^n b^m), \mathcal{V}') \models \forall x.\psi_0' \quad \text{iff} \quad (K_{n,m}, \mathcal{V}) \models \forall x.\psi_0.$$

For second-order quantification over the set variable X we apply an analogous argument. If B ranges over all node-sets of $K_{n,m}$ then $h(B)$ ranges over all sets I of word positions in $\{1,\ldots,n+m\}$. Thus:

$$\begin{aligned} & (\mathrm{Graph}(a^n b^m), \mathcal{V}'[X := I]) \models \psi_0' \quad \text{for all subsets } I \text{ of } \{1,\ldots,n+m\} \\ \text{iff} \quad & (K_{n,m}, \mathcal{V}[X := B]) \models \psi_0 \quad \text{for all subsets } B \text{ of the node-set in } K_{n,m}. \end{aligned}$$

We conclude that $(\mathrm{Graph}(a^n b^m), \mathcal{V}') \models \forall X.\psi_0'$ iff $(K_{n,m}, \mathcal{V}) \models \forall X.\psi_0$. This completes the proof of (***). ⌋

Assertion (**) follows from (***) by considering the MSO-sentence $\psi = \phi$. Recall that ϕ denotes the (fictive) MSO-sentence over $\mathit{Voc}_{\mathrm{graph}}$ that defines the Hamilton cycle property for finite graphs. We now switch from ϕ to the sentence ϕ' over Voc_Σ where $\Sigma = \{a, b\}$. The combination of (*) and (**) yields that:

$$\begin{aligned} \mathrm{Graph}(a^n b^m) \models \phi' \quad & \text{iff} \quad K_{n,m} \models \phi \\ & \text{iff} \quad K_{n,m} \text{ has a Hamilton cycle} \\ & \text{iff} \quad n = m \end{aligned}$$

The goal is now to derive from ϕ' a MSO-formula θ over the vocabulary $\mathit{Voc}_{\Sigma,\mathrm{graph}}$ that defines the non-regular language $\{a^n b^n : n \geqslant 1\}$. Let

$$\theta \stackrel{\text{def}}{=} \phi' \wedge \varphi$$

where

$$\varphi \stackrel{\text{def}}{=} \exists x.\, P_a(x) \wedge \exists y.\, P_b(y) \wedge \forall x \forall y.\, \big(P_a(x) \wedge P_b(y) \rightarrow x < y \big)$$

Then, φ defines the language $a^+ b^+$, since for each word $w \in \Sigma^*$ we have:

$$\begin{aligned} \mathrm{Graph}(w) \models \exists x.P_a(x) \quad & \text{iff} \quad w \text{ contains the symbol } a \text{ at least once} \\ \mathrm{Graph}(w) \models \exists y.P_b(y) \quad & \text{iff} \quad w \text{ contains the symbol } b \text{ at least once} \end{aligned}$$

and $\mathrm{Graph}(w) \models \forall x \forall y.\big(P_a(x) \wedge P_b(y) \rightarrow x < y\big)$ iff $w \in \mathfrak{L}(a^* b^*)$. Hence, for each word $w \in \Sigma^*$:

$$\begin{aligned} \mathrm{Graph}(w) \models \theta \quad & \text{iff} \quad \text{there exists } n, m \geqslant 1 \text{ such that } w = a^n b^m \text{ and } \mathrm{Graph}(w) \models \phi' \\ & \text{iff} \quad w = a^n b^n \text{ for some } n \geqslant 1 \end{aligned}$$

Thus, the MSO-sentence θ defines the language

$$L \stackrel{\text{def}}{=} \mathfrak{L}(\theta) = \big\{ a^n b^n : n \geqslant 1 \big\}.$$

This is impossible since L is not regular, which can be shown by applying the pumping lemma for regular languages. On the other hand, the languages defined by MSO-formulas over $\mathit{Voc}_{\Sigma,\mathrm{graph}}$ are regular, see Theorem 2.6.18 on page 178. □

FOL over finite words

We finally discuss the relation between MSO over words and its first-order fragment. As reachability is not FO-definable in the vocabulary of graphs, the "less than or equal" predicate $\leqslant$ for the word positions cannot be derived in FOL over the vocabulary $\mathit{Voc}_{\Sigma,\mathrm{graph}}$. For this reason we switch to the vocabulary $\mathit{Voc}_{\Sigma,\leqslant}$ consisting of:

- the unary predicate symbols P_a for all symbols $a \in \Sigma$
- a binary predicate symbol $\leqslant$

By FOL over finite words, we mean the set of FOL-formulas over $\mathit{Voc}_{\Sigma,\leqslant}$, i.e., FOL-formulas built by the atomic formulas $P_a(x)$, $x \leqslant y$ and $x = y$ where $x, y \in \mathit{Var}$ and $a \in \Sigma$.

Given a finite nonempty word $w = a_1 a_2 \ldots a_n \in \Sigma^+$, the associated structure over $\mathit{Voc}_{\Sigma,\leqslant}$ is defined as the structure $\mathrm{Graph}(w)$ over $\mathit{Voc}_{\Sigma,\mathrm{graph}}$ except that the edge relation E^w is replaced with the natural order $\leqslant$ on $\{1,\ldots,n\}$. With abuse of notations, we use the same notation $\mathrm{Graph}(w)$ which, in the context of FOL over finite words, stands for the structure:

$$\mathrm{Graph}(w) \;=\; \big(\{1,\ldots,n\}, \leqslant, (P_a^w)_{a\in\Sigma}\big)$$

where $P_a^w = \big\{ i \in \{1,\ldots,n\} : a_i = a \big\}$. For the empty word, $\mathrm{Graph}(\varepsilon) = \big(\{0\},\{(0,0)\},(P_a^\varepsilon)_{a\in\Sigma}\big)$ where $P_a^\varepsilon = \varnothing$. Although the edge predicate E has been dropped from the vocabulary, it can be derived as the "direct successor relation" of the linear order $\leqslant$:

$$E(x,y) \;\stackrel{\text{def}}{=}\; x < y \;\wedge\; \neg\exists z.\,(x < z \wedge z < y)$$

where $x < y$ stands (as usual) for $x \leqslant y \wedge x \neq y$. Thus, each MSO-formula over $\mathit{Voc}_{\Sigma,\mathrm{graph}}$ without any occurrences of set variables can be viewed as a FOL-formula over $\mathit{Voc}_{\Sigma,\leqslant}$. Vice versa, each FOL-formula over $\mathit{Voc}_{\Sigma,\leqslant}$ can be viewed as a MSO-formula over $\mathit{Voc}_{\Sigma,\mathrm{graph}}$. In this sense, FOL over finite words is a sublogic of MSO over finite words.

Of course, we could have used the vocabulary $\mathit{Voc}_{\Sigma,\leqslant}$ for MSO as well. As E can be derived from $\leqslant$ as shown above and $\leqslant$ from E by using a MSO-formula for reachability, the vocabularies $\mathit{Voc}_{\Sigma,\leqslant}$ and $\mathit{Voc}_{\Sigma,\mathrm{graph}}$ are "equivalent" for MSO over finite words.

Definition 2.6.23 (FO-definable languages). Let $L \subseteq \Sigma^*$, i.e., L is a language of finite words over Σ. L is said to be *FO-definable* if there exists a FOL-sentence ϕ over $\mathit{Voc}_{\Sigma,\leqslant}$ such that

$$L \;=\; \big\{\, w \in \Sigma^* : \mathrm{Graph}(w) \models \phi \,\big\}$$

In this case we say that ϕ *defines* L. ■

Example 2.6.24 (FOL-sentences for regular languages). The regular language given by the regular expression a^*b^* is defined by the FOL-sentence

$$\forall x \forall y.\,\big(P_a(x) \wedge P_b(y) \rightarrow x < y\big).$$

The FOL-sentence

$$\forall x.\big(P_a(x) \rightarrow \exists y.(E(x,y) \wedge P_b(y)\big)$$

defines the regular language $b^*(ab^+)^*$ consisting of all finite words where any a is followed by the letter b. ■

We now show that the class of FO-definable languages is a proper subclass of the regular languages, namely the so-called star-free languages.

Definition 2.6.25 (Star-free languages). The set of star-free languages over Σ is the smallest set of languages $L \subseteq \Sigma^*$ that contains $\varnothing$, the singleton language $\{\varepsilon\}$ and the singleton languages $\{\mathfrak{a}\}$ for $\mathfrak{a} \in \Sigma$ and is closed under union, concatenation and complementation. ■

Each star-free language is regular. This follows from the fact that also the class of regular languages contains $\varnothing$, the singleton languages $\{\varepsilon\}$ and $\{\mathfrak{a}\}$ for $\mathfrak{a} \in \Sigma$ and is closed under union, concatenation and complementation.

Star-free languages are often described by *star-free expressions* built by the symbols $\emptyset$, ε and $\mathfrak{a} \in \Sigma$ and the operators union, concatenation and complementation. The abstract syntax of star-free expressions over Σ is given by:

$$\alpha \;::=\; \emptyset \;\mid\; \varepsilon \;\mid\; \mathfrak{a} \;\mid\; \alpha_1 + \alpha_2 \;\mid\; \alpha_1\,\alpha_2 \;\mid\; \overline{\alpha}$$

where $\mathfrak{a}$ ranges over all symbols in Σ. The associated (star-free) language $\mathfrak{L}(\alpha) \subseteq \Sigma^*$ of a star-free expression is defined in the obvious way, see Figure 22 on page 188.

$$\begin{aligned}
\mathfrak{L}(\emptyset) &\stackrel{\text{def}}{=} \varnothing \\
\mathfrak{L}(\varepsilon) &\stackrel{\text{def}}{=} \{\varepsilon\} \\
\mathfrak{L}(\mathfrak{a}) &\stackrel{\text{def}}{=} \{\mathfrak{a}\} \\
\mathfrak{L}(\,\alpha_1 + \alpha_2\,) &\stackrel{\text{def}}{=} \mathfrak{L}(\alpha_1) \cup \mathfrak{L}(\alpha_2) \\
\mathfrak{L}(\,\alpha_1\,\alpha_2\,) &\stackrel{\text{def}}{=} \mathfrak{L}(\alpha_1)\,\mathfrak{L}(\alpha_2) &&= \big\{ w_1 w_2 : w_i \in \mathfrak{L}(\alpha_i), i = 1,2 \big\} \\
\mathfrak{L}(\,\overline{\alpha}\,) &\stackrel{\text{def}}{=} \overline{\mathfrak{L}(\alpha)} &&= \Sigma^* \setminus \mathfrak{L}(\alpha)
\end{aligned}$$

Figure 22: Semantics of star-free expressions

Thus, the syntax of star-free expressions is as for ordinary regular expressions with the only difference that the Kleene-star operator α^* is not permitted for star-free languages. Instead they use the complementation operator. (Recall that the complementation operator is not included in the syntax of regular expressions. Nevertheless regular languages are closed under complementation.) Given two languages $L_1, L_2 \subseteq \Sigma^*$, we have

$$L_1 \cap L_2 \;=\; \overline{\overline{L_1} \cup \overline{L_2}}.$$

Thus, the class of star-free languages is also closed under intersection. Hence, the intersection operator $\cap$ could be used in star-free expressions as well.

Example 2.6.26 (Star-free languages). If $\Sigma = \{\mathfrak{a}, \mathfrak{b}\}$ then Σ^* is star-free as it is given by the complement $\overline{\emptyset}$ of $\emptyset$. Likewise, the language given by the regular expression

$$(\mathfrak{a} + \mathfrak{b})^* \mathfrak{b} (\mathfrak{a} + \mathfrak{b})^*,$$

i.e., the language consisting of all finite words in $\{\mathfrak{a}, \mathfrak{b}\}^*$ that contain at least one $\mathfrak{b}$, is star-free as it is given by the concatenation $L_1 L_2 L_3$ where $L_1 = L_3 = \{\mathfrak{a}, \mathfrak{b}\}^*$ and $L_2 = \{\mathfrak{b}\}$. Hence,

$$\overline{\emptyset}\;\mathfrak{b}\;\overline{\emptyset}$$

is a star-free expression for the language given by $(\mathfrak{a}+\mathfrak{b})^*\mathfrak{b}(\mathfrak{a}+\mathfrak{b})^*$.

The language $\{\mathfrak{a}^n : n \in \mathbb{N}\}$, given by the regular expression $\mathfrak{a}^*$, is star-free too, as it is the complement of the above language, and given by the star-free expression

$$\overline{\overline{\emptyset}\,\mathfrak{b}\,\overline{\emptyset}}.$$

Also the language given by the regular expression $\mathfrak{a}^*\mathfrak{b}^*$ is star-free, as it is obtained by the star-free expression:

$$\overline{\overline{\emptyset}\,\mathfrak{b}\,\overline{\emptyset}}\quad\overline{\overline{\emptyset}\,\mathfrak{a}\,\overline{\emptyset}}$$

Alternatively, the language given by $\mathfrak{a}^*\mathfrak{b}^*$ is characterized by the property that no $\mathfrak{b}$ is preceeding an $\mathfrak{a}$, which corresponds to the star-free expression:

$$\overline{\overline{\emptyset}\,\mathfrak{b}\,\mathfrak{a}\,\overline{\emptyset}}$$

■

We now show that the class of FO-definable languages agrees with the class of star-free languages.

Theorem 2.6.27 (FO-definable languages). *Let* $L \subseteq \Sigma^*$. *Then:*

$$L \textit{ is FO-definable} \quad \textit{iff} \quad L \textit{ is star-free.}$$

The proof of Theorem 2.6.27 splits into two parts: the FO-definability of all star-free languages (Lemma 2.6.28, see below) and the star-freedom of all FO-definable languages (Lemma 2.6.29 on page 192).

Lemma 2.6.28 (Star-free languages are FO-definable). *Each star-free language is FO-definable.*

Proof. The proof is by induction on the length of a shortest star-free expression for L. The basis of induction is obvious:

- For the empty language (i.e., $L = \emptyset$), we may deal with the FOL-sentence *false*.
- For $L = \{\varepsilon\}$ we take the FOL-sentence $\neg\forall x.\ \bigvee_{\mathfrak{a}\in\Sigma} P_\mathfrak{a}(x)$.
- The singleton language $L = \{\mathfrak{a}\}$ for some symbol $\mathfrak{a}$ in Σ is defined by the FOL-formula

  $$\exists x.(\mathit{first}(x) \wedge \mathit{last}(x) \wedge P_\mathfrak{a}(x))$$

  where $\mathit{first}(x)$ and $\mathit{last}(x)$ are defined as on page 170. Instead of $\mathit{first}(x)$ we could also deal here with $\neg\exists y.(y < x)$ and with $\neg\exists y.(x < y)$ rather than $\mathit{last}(x)$.

Step of induction: The argument for the cases where the outermost operator is complementation or union is straightforward. If L is defined by the FOL-sentence ϕ then $\neg\phi$ defines $\overline{L}$. If ϕ_1 and ϕ_2 are FOL-sentences that define the star-free languages L_1 and L_2, respectively, then $\phi_1 \vee \phi_2$ defines the language $L_1 \cup L_2$.

The most interesting case is the treatment of concatenation. Let us consider the language

$$L_1 L_2 \;=\; \big\{ w_1 w_2 : w_1 \in L_1, w_2 \in L_2 \big\}$$

where L_1, L_2 are star-free languages that are defined by the FOL-sentences ϕ_1 and ϕ_2, respectively. We define

$$\psi_1 \stackrel{\text{def}}{=} \begin{cases} \phi_2 & : \text{ if } \varepsilon \in L_1 \\ \mathit{false} & : \text{ if } \varepsilon \notin L_1 \end{cases} \qquad \psi_2 \stackrel{\text{def}}{=} \begin{cases} \phi_1 & : \text{ if } \varepsilon \in L_2 \\ \mathit{false} & : \text{ if } \varepsilon \notin L_2 \end{cases}$$

Then:

$$\mathrm{Graph}(w) \models \psi_1 \vee \psi_2 \quad \text{iff} \quad w \in L_1 \,\&\, \varepsilon \in L_2 \;\text{ or }\; w \in L_2 \,\&\, \varepsilon \in L_1 \qquad (*)$$

Formula $\psi_1 \vee \psi_2$ serves as characterization of the words in $L_1 L_2$ that arise by the concatenation of the empty word in L_1 with some word in L_2, or vice versa. We now define a formula ϕ' that defines the set of words $w_1 w_2$ where $w_1 \in L_1$ and $w_2 \in L_2$ are nonempty. Let x be a fresh first-order variable that does not appear in ϕ_1 and ϕ_2. Let $\phi_1'(x)$ be the formula that results from ϕ_1 when

- each subformula $\exists y.\psi$ of ϕ_1 is replaced with $\exists y.(\, y \leqslant x \wedge \psi\,)$,
- each subformula $\forall y.\psi$ of ϕ_1 is replaced with $\forall y.(\, y \leqslant x \to \psi\,)$.

Similarly, formula $\phi_2'(x)$ arises from ϕ_2 by relativizing each quantifier to the positions $y > x$. That is, $\phi_2'(x)$ results from ϕ_2 when

- each subformula $\exists y.\psi$ of ϕ_2 is replaced with $\exists y.(\, y > x \wedge \psi\,)$,
- each subformula $\forall y.\psi$ of ϕ_2 is replaced with $\forall y.(\, y > x \to \psi\,)$.

Then, for each word $w = a_1 a_2 \ldots a_n$ of length at least 2 and position $m \in \{1,\ldots,n-1\}$ we have:

$$\begin{aligned} (\mathrm{Graph}(a_1 \ldots a_m \ldots a_n), [x := m]) \models \phi_1'(x) \quad &\text{iff} \quad \mathrm{Graph}(a_1 \ldots a_m) \models \phi_1 \\ &\text{iff} \quad a_1 \ldots a_m \in L_1 \\ (\mathrm{Graph}(a_1 \ldots a_m \ldots a_n), [x := m]) \models \phi_2'(x) \quad &\text{iff} \quad \mathrm{Graph}(a_{m+1} \ldots a_n) \models \phi_2 \\ &\text{iff} \quad a_{m+1} \ldots a_n \in L_2 \end{aligned}$$

Hence, for each word $w \in \Sigma^*$:

$$\begin{aligned} &\mathrm{Graph}(w) \models\; \exists x.\,(\, x < \mathit{last} \wedge \phi_1'(x) \wedge \phi_2'(x)\,) \\ \text{iff} \quad &w = w_1 w_2 \text{ for nonempty words } w_1 \in L_1 \text{ and } w_2 \in L_2 \end{aligned}$$

where "$x < \mathit{last}$" stands shortly for the formula $\exists y.\, x < y$. It should be noticed that

$$(\mathrm{Graph}(w), [x := m]) \models x < \mathit{last} \quad \text{implies} \quad |w| \geqslant 2.$$

We now consider the formula

$$\theta \stackrel{\text{def}}{=} \exists x.\, (\, x < \mathit{last} \wedge \phi_1'(x) \wedge \phi_2'(x)\,) \vee \psi_1 \vee \psi_2$$

Then, for each word $w \in \Sigma^*$:

$$\begin{array}{ll} & w \in L_1 L_2 \\ \text{iff} & w \in L_1 \wedge \varepsilon \in L_2 \text{ or } w \in L_2 \wedge \varepsilon \in L_1 \text{ or } w = w_1 w_2 \text{ where } w_i \in L_i \setminus \{\varepsilon\},\ i = 1,2 \\ \text{iff} & \mathrm{Graph}(w) \models \psi_1 \vee \psi_2 \quad \text{or} \quad \mathrm{Graph}(w) \models \exists x.(\, x < \mathit{last} \wedge \phi_1'(x) \wedge \phi_2'(x)\,) \\ \text{iff} & \mathrm{Graph}(w) \models \theta \end{array}$$

Hence, θ defines $L_1 L_2$. □