We now show that also the converse of Theorem 2.6.10 holds. More precisely, we show that for any MSO-sentence ϕ over $Voc_{\Sigma,\text{graph}}$ the finite word models of ϕ constitute a regular language. That is,

$$\mathfrak{L}(\phi) \stackrel{\text{def}}{=} \{ w \in \Sigma^* \,:\, \text{Graph}(w) \models \phi \}$$

is regular. Our argument will use an inductive construction of NFA for MSO-formulas with possibly free first-order and set variables. For this, we first have to extend the above definition of the associated language $\mathfrak{L}(\phi)$ for formulas with possibly free variables. We start with the consideration of formulas with free set variables, but no free first-order variables, and extend this definition for the general case later (see Notation 2.6.13).

Notation 2.6.11 (The language of a MSO-formula (first part)). Let $Y_1,\ldots,Y_r$ be pairwise distinct set variables and $r \geqslant 1$. Let $\phi(Y_1,\ldots,Y_r)$ be a MSO-formula over the vocabulary $Voc_{\Sigma,\text{graph}}$ which has no free first-order variables, but might contain the free set variables $Y_1,\ldots,Y_r$. We then define $\mathfrak{L}(\phi,Y_1,\ldots,Y_r)$ to be the set of finite nonempty words

$$w \;=\; (a_1,\overline{\xi}_1)\,(a_2,\overline{\xi}_2)\ldots(a_n,\overline{\xi}_n) \;\in\; \big(\Sigma\times\{0,1\}^r\big)^+$$

over the extended alphabet $\Sigma_r \stackrel{\text{def}}{=} \Sigma\times\{0,1\}^r$ such that

$$(\text{Graph}(a_1\ldots a_n),\mathcal{V}_w) \models \phi$$

where $\mathcal{V}_w$ arises from the bit-tuples $\overline{\xi}_i = (\xi_{i,1},\ldots,\xi_{i,r})$ as follows:

$$\mathcal{V}_w(Y_j) \;=\; \big\{ i \in \{1,\ldots,n\} : \xi_{i,j} = 1 \big\}.$$

That is, $\overline{\xi}_i$ specifies the sets Y_j where word position i belongs to. We then put:

$$\mathfrak{L}(\phi,Y_1,\ldots,Y_r) \stackrel{\text{def}}{=} \{ w \in \Sigma_r^+ \,:\, (\text{Graph}(w),\mathcal{V}_w) \models \phi \}$$

where $\text{Graph}(w) \stackrel{\text{def}}{=} \text{Graph}(a_1\,a_2\ldots a_n)$ if $w \;=\; (a_1,\overline{\xi}_1)\,(a_2,\overline{\xi}_2)\ldots(a_n,\overline{\xi}_n)$. ■

Example 2.6.12 (Regular language for MSO-formula). For the MSO-formula

$$\begin{aligned}\phi(Y) \stackrel{\text{def}}{=} \forall x\forall y.\big(\; & (\,(P_a(x) \wedge x\in Y \wedge E(x,y)\,) \rightarrow P_b(y)\,) \wedge \\ & (\,(P_b(x) \wedge x\in Y \wedge E(x,y)\,) \rightarrow P_a(y)\,) \quad \big)\end{aligned}$$

the language $\mathfrak{L}(\phi,Y)$ is a set of words over the alphabet $\Sigma_1 = \Sigma\times\{0,1\}$. The word

$$w_1 \;=\; (a,0)\,(a,1)\,(b,1)\,(a,0)$$

belongs to $\mathfrak{L}(\phi,Y)$ and specifies the set $Y = \{2,3\}$. Note that at position $x = 2 \in Y$ there is letter a and in the next position $y = 3$ there is letter b, as required by the subformula

$$(\,P_a(x) \wedge x\in Y \wedge E(x,y)\,) \rightarrow P_b(y)$$

of $\phi(Y)$. Similarly, letter b appears in the word w_1 at position $x = 3 \in Y$, while the letter in the next position $y = 4$ is a. This meets the constraint imposed by the subformula

$$(\,P_b(x) \wedge x\in Y \wedge E(x,y)\,) \rightarrow P_a(y)$$

of $\phi(Y)$. On the other hand,

$$w_2 \;=\; (\mathfrak{a},1)\,(\mathfrak{a},1)\,(\mathfrak{b},1)\,(\mathfrak{a},0)$$

does not belong to $\mathfrak{L}(\phi,Y)$. Note that w_2 specifies the set $Y=\{1,2,3\}$, but letter $\mathfrak{a}$ in position $1\in Y$ is not followed by letter $\mathfrak{b}$. More general, assuming $\Sigma=\{\mathfrak{a},\mathfrak{b}\}$, the language $\mathfrak{L}(\phi,Y)$ consists of all words

$$(\mathfrak{a}_1,\xi_1)\,(\mathfrak{a}_2,\xi_2)\,\ldots\,(\mathfrak{a}_n,\xi_n)\in(\{\mathfrak{a},\mathfrak{b}\}\times\{0,1\})^+$$

such that each occurrence of letter $(\mathfrak{a},1)$ at some position $i<n$ is followed by $(\mathfrak{b},0)$ or $(\mathfrak{b},1)$ at position $i{+}1$, and similarly, if $(\mathfrak{b},1)$ is the letter at some word position $i<n$ then the symbol at position $i{+}1$ is either $(\mathfrak{a},0)$ or $(\mathfrak{a},1)$. ■

Notation 2.6.13 (The language of a MSO-formula (second part)). Treating first-order variables as singleton sets, any MSO-formula $\phi(z_1,\ldots,z_m,Y_1,\ldots,Y_r)$ with free first-order variables in $\{z_1,\ldots,z_m\}$ and free set variables in $\{Y_1,\ldots,Y_r\}$ can be interpreted as a language consisting of finite nonempty words

$$w \;=\; (\mathfrak{a}_1,\overline{\zeta}_1,\overline{\xi}_1)\,(\mathfrak{a}_2,\overline{\zeta}_2,\overline{\xi}_2)\,\ldots\,(\mathfrak{a}_n,\overline{\zeta}_n,\overline{\xi}_n)$$

over the alphabet $\Sigma_{m+r}=\Sigma\times\{0,1\}^m\times\{0,1\}^r$ where $\overline{\zeta}_i=(\zeta_{i,1},\ldots,\zeta_{i,m})$ and $\overline{\xi}_i=(\xi_{i,1},\ldots,\xi_{i,r})$ are bit tuples of length m and r, respectively. While the bits $\xi_{i,j}$ for the set-variables are arbitrary, for each index $j\in\{1,\ldots,m\}$ the bit sequence $\zeta_{1,j},\ldots,\zeta_{n,j}$ must contain exactly one "1". The $\zeta_{i,j}$'s serve to specify the meaning of the first-order variables $z_1,\ldots,z_m$:

$$\mathcal{V}_w(z_j)=i \quad \text{for the unique index } i\in\{1,\ldots,n\} \text{ where } \zeta_{i,j}=1,$$

while the $\xi_{i,j}$'s describe the meaning of the set variables $Y_1,\ldots,Y_r$ as before. That is,

$$\mathcal{V}_w(Y_j) \;=\; \big\{\, i\in\{1,\ldots,n\} : \xi_{i,j}=1 \,\big\}, \quad j=1,\ldots,r.$$

We then define $\mathfrak{L}(\phi,z_1,\ldots,z_m,Y_1,\ldots,Y_r)$ as the set of all finite nonempty words $w\in\Sigma_{m+r}^+$ such that $(\mathrm{Graph}(w),\mathcal{V}_w)$ yields a model for ϕ.

$$\mathfrak{L}(\phi,z_1,\ldots,z_m,Y_1,\ldots,Y_r) \;\stackrel{\text{def}}{=}\; \big\{\, w\in(\Sigma\times\{0,1\}^{m+r})^+ : (\mathrm{Graph}(w),\mathcal{V}_w)\models\phi \,\big\}$$

Recall that if w is as above then $\mathrm{Graph}(w)=\mathrm{Graph}(\mathfrak{a}_1\ldots\mathfrak{a}_n)$. ■

For technical reasons, the languages $\mathfrak{L}(\phi,z_1,\ldots,z_m,Y_1,\ldots,Y_r)$ for $m{+}r\geqslant 1$ do not contain the empty word (as it does not specify a variable valuation in a unique way), while $\mathfrak{L}(\phi)$ for a MSO-sentence ϕ might contain the empty word. This, however, is not problematic for our purposes. Given an MSO-sentence ϕ, we will construct an NFA $\mathcal{M}$ such that:

$$\mathfrak{L}(\mathcal{M})\cap\Sigma_{m+r}^+ \;=\; \mathfrak{L}(\phi)\cap\Sigma_{m+r}^+$$

As before, $\Sigma_{m+r}=\Sigma\times\{0,1\}^{m+r}$. But then $\mathfrak{L}(\phi)$ must be regular as it agrees with one of the regular languages $\mathfrak{L}(\mathcal{M})$ or $\mathfrak{L}(\mathcal{M})\cup\{\varepsilon\}$ or $\mathfrak{L}(\mathcal{M})\setminus\{\varepsilon\}$.

Example 2.6.14 (Regular languages associated with MSO-formulas). The MSO-formula

$$\phi(z) \;=\; \forall x.\big(P_a(x) \to x < z\big)$$

specifies that letter a can only occur before the position of z. Thus, we have, e.g.,

$$w_1 \;=\; (a,0)\,(a,0)\,(b,1)\,(b,0) \;\in\; \mathfrak{L}(\phi,z)$$

since z stands for position 3 and there is no a at some position $\geqslant 3$. However:

$$\begin{aligned} w_2 &= (a,0)\,(a,0)\,(b,1)\,(a,0) \;\notin\; \mathfrak{L}(\phi,z)\\ w_3 &= (a,0)\,(a,0)\,(b,1)\,(b,1) \;\notin\; \mathfrak{L}(\phi,z) \end{aligned}$$

In w_2, letter a appears after position $z = 3$. The word w_3 does not provide a proper encoding of z's meaning since "1" appears at two positions (positions 3 and 4). ■

Example 2.6.15 (Regular languages associated with MSO-formulas). For another example, we regard the MSO-formula

$$\phi(z,Y) \;=\; P_a(z) \wedge z \notin Y$$

Then, $\mathfrak{L}(\phi,z,Y)$ denotes the set of all finite nonempty words $(a_1,\zeta_1,\xi_1)\ldots(a_n,\zeta_n,\xi_n)$ where $a_i \in \Sigma$, $\zeta_i,\xi_i \in \{0,1\}$, $i = 1,\ldots,n$, such that $\zeta_i = 1$ for precisely one word position $i \in \{1,\ldots,n\}$ and for this word position i we have $a_i = a$ and $\xi_i \neq 1$. Examples for words in $\mathfrak{L}(\phi,z,Y)$ are:

$$\begin{aligned} w_1 &= (a,0,0)\,(a,1,0)\,(b,0,1) \;\in\; \mathfrak{L}(\phi,z,Y)\\ w_2 &= (a,0,1)\,(b,0,1)\,(a,1,0) \;\in\; \mathfrak{L}(\phi,z,Y) \end{aligned}$$

An example for a word not in $\mathfrak{L}(\phi,z,Y)$ is:

$$w_3 \;=\; (a,0,0)\,(a,0,1)\,(b,0,1) \;\notin\; \mathfrak{L}(\phi,z,Y)$$

since there is no "1" for the bits for first-order variable z. We also have

$$w_4 \;=\; (a,0,0)\,(a,1,1)\,(b,0,1) \;\notin\; \mathfrak{L}(\phi,z,Y)$$

as the induced variable valuation is given by $\mathcal{V}_{w_4}(z) = 2$ and $\mathcal{V}_{w_4}(Y) = \{2,3\}$. Thus, $z \notin Y$ does not hold. ■

Notation 2.6.16 (Alphabet Σ_ϕ and language $\mathfrak{L}(\phi)$). Recall that the notation $\phi(z_1,\ldots,z_m, Y_1, \ldots,Y_r)$ indicates that at most the variables $z_1,\ldots,z_m$, Y_1, $\ldots,Y_r$ appear free in ϕ, but some of these variables might not have free occurrences in ϕ. In the sequel, the notation Σ_ϕ will be used to denote the alphabet

$$\Sigma_\phi \;\stackrel{\text{def}}{=}\; \Sigma \times \{0,1\}^{m+r}$$

where $m = |\mathit{Free}(\phi)|$ is the number of free first-order variables in ϕ and $r = |\mathit{PFree}(\phi)|$ the number of free set variables. If ϕ is a sentence then $m = r = 0$ and $\Sigma_\phi = \Sigma$. Furthermore, we write $\mathfrak{L}(\phi)$ to denote the associated language

$$\mathfrak{L}(\phi) \;\stackrel{\text{def}}{=}\; \mathfrak{L}(\phi,\underbrace{z_1,\ldots,z_m}_{\mathit{Free}(\phi)},\underbrace{Y_1,\ldots,Y_r}_{\mathit{PFree}(\phi)})$$

over Σ_ϕ where $\mathit{Free}(\phi) = \{z_1,\ldots,z_m\}$ and $\mathit{PFree}(\phi) = \{Y_1,\ldots,Y_r\}$ when some enumeration of the free first-order and set variables is fixed. ■

Remark 2.6.17 (On the choice of the variables). The language associated with an MSO-formula ϕ depends on the choice of the variable lists. We can deal with additional variables that have no free occurrences in ϕ. The underlying alphabet then has the form $\Sigma_\phi \times \{0,1\}^k$ where k is the number of extra variables. For the following considerations, it is only important that $\mathfrak{L}(\phi)$ is regular if and only if the language $\mathfrak{L}'(\phi)$ over the extended alphabet is. Let us briefly explain why. The induced language $\mathfrak{L}'(\phi)$ arises from $\mathfrak{L}(\phi)$ by adding appropriate bits for the auxiliary variables. For set variables that do not appear free in ϕ, we can add arbitrary bits. For the treatment of an additional first-order variable z, however, we have to take care about the requirement that in each word the sequence of bits for z contains exactly one "1". For example, assume that

$$Free(\phi) = \{z_1,\dots,z_m\} \quad \text{and} \quad PFree(\phi) = \{Y_1,\dots,Y_r\}$$

and that z is fresh first-order variable and Y a fresh set variable. The language

$$\mathfrak{L}(\phi) \;\subseteq\; \Sigma_\phi^* \;=\; \big(\Sigma \times \{0,1\}^{m+r}\big)^*$$

and the language

$$\mathfrak{L}'(\phi) \stackrel{\text{def}}{=} \mathfrak{L}(\phi, z_1,\dots,z_m, Y_1,\dots,Y_r, z, Y) \;\subseteq\; \big(\Sigma \times \{0,1\}^{m+r+2}\big)^* \;=\; \big(\Sigma_\phi \times \{0,1\}^2\big)^*$$

for ϕ viewed as a formula of the type $\phi(z_1,\dots,z_m,Y_1,\dots,Y_r,z,Y)$ can be transformed to each other by dropping and adding bits, respectively. For simplicity, we add both the free first-order variable z and the free set variable Y at the end of the variable list. The languages $\mathfrak{L}(\phi)$ and $\mathfrak{L}'(\phi)$ are related via the homomorphism

$$h\colon \big(\Sigma_\phi \times \{0,1\}^2\big)^* \to \Sigma_\phi^*$$

that erases the bits ρ and η for the extra variables z and Y in each letter $(a, \overline{\zeta}, \overline{\xi}, \rho, \eta)$ in the alphabet $\Sigma_\phi \times \{0,1\}^2$. Then:

$$\mathfrak{L}(\phi) \setminus \{\varepsilon\} \;=\; h\big(\mathfrak{L}'(\phi)\big) \qquad \mathfrak{L}'(\phi') \;=\; h^{-1}\big(\mathfrak{L}(\phi) \setminus \{\varepsilon\}\big)$$

Given that images and inverse images of regular languages under homoromorphisms are regular, $\mathfrak{L}(\phi)$ is regular if and only $\mathfrak{L}'(\phi)$ is regular.

In the proof of the following theorem we will make use of this observation by ignoring bits for variables that do not appear free in the formula under consideration. For the composition $\phi_1 \wedge \phi_2$ we suppose $\mathfrak{L}(\phi_1)$ and $\mathfrak{L}(\phi_2)$ to be languages over the same alphabet. In case that, e.g., ϕ_1 has free variables that do not appear free in ϕ_2, the language for ϕ_2 has to be extended by bits for them. ■

Theorem 2.6.18 (Regular languages for MSO-formulas). *For each MSO-formula ϕ over $Voc_{\Sigma,graph}$, the language $\mathfrak{L}(\phi) \subseteq \Sigma_\phi^*$ is regular.*

Thus, if ϕ is a MSO-sentence over $Voc_{\Sigma,\text{graph}}$ then the language $\mathfrak{L}(\phi) \subseteq \Sigma^*$ is regular.

Proof. We provide the proof by structural induction. In the basis of induction we have to consider *true* and atomic formulas.

- For the trivial formula $\phi = \mathit{true}$ we have $\mathfrak{L}(\mathit{true}) = \Sigma^*$, which is obviously regular.
- If ϕ is an atomic formula of the form $P_a(z)$ then $\mathfrak{L}(\phi)$ is regular as

$$\mathfrak{L}(\,P_a(z)\,) \;=\; (\Sigma\times\{0\})^*\,(a,1)\,(\Sigma\times\{0\})^*$$

- Suppose now that $\phi = E(z_1,z_2)$. The intuitive meaning of $E(z_1,z_2)$ is that z_2 is the next position of z_1. That is, $z_2 = z_1+1$. Thus, the language $\mathfrak{L}(\,E(z_1,z_2)\,) \subseteq (\Sigma\times\{0,1\}^2)^*$ consists of all words

$$(a_1,\zeta_{1,1},\zeta_{1,2})\,(a_2,\zeta_{2,1},\zeta_{2,2})\,\ldots\,(a_n,\zeta_{n,1},\zeta_{n,2})$$

where $n \geqslant 2$ and the word positions of z_1 and z_2 are consecutive, i.e., there exists an index $i \in \{1,\ldots,n-1\}$ such that $\zeta_{i,1} = 1$ and $\zeta_{i+1,2} = 1$ and $\zeta_{\ell,j} = 0$ in all other cases, i.e., for $(\ell,j) \in (\{1,\ldots,n\}\times\{1,2\})\setminus\{(i,1),(i+1,2)\}$. Thus, $\mathfrak{L}(\,E(z_1,z_2)\,)$ agrees with the regular language given by:

$$(\Sigma\times\{(0,0)\})^*\,\underbrace{(\Sigma\times\{(1,0)\})}_{\zeta_{i,1}=1}\,\underbrace{(\Sigma\times\{(0,1)\})}_{\zeta_{i+1,2}=1}\,(\Sigma\times\{(0,0)\})^*$$

Similarly, $\mathfrak{L}(\,E(z_2,z_1)\,)$ is given by

$$(\Sigma\times\{(0,0)\})^*\,(\Sigma\times\{(0,1)\})\,(\Sigma\times\{(1,0)\})\,(\Sigma\times\{(0,0)\})^*$$

while the language for $E(z,z)$ is empty.

- Let us now consider an atomic formula of the form $z \in Y$. The words in $\mathfrak{L}(\,z \in Y\,)$ are sequences $(a_1,\zeta_1,\xi_1)\ldots(a_n,\zeta_n,\xi_n)$ such that $a_i \in \Sigma$, $\zeta_i,\xi_i \in \{0,1\}$ and, for some index $k \in \{1,\ldots,n\}$, $\zeta_k = \xi_k = 1$ and $\zeta_i = 0$ for all $i \neq k$. Hence:

$$\mathfrak{L}(\,z \in Y\,) \;=\; \Gamma^*\,(\Sigma\times\{(1,1)\})\,\Gamma^*$$

where $\Gamma = \Sigma\times\{0\}\times\{0,1\}$. Thus, $\mathfrak{L}(z \in Y)$ is regular.

- If ϕ is $z_1 = z_2$ then the associated language over $\Sigma\times\{0,1\}^2$ requires that the bits for z_1 and z_2 agree. Thus,

$$\mathfrak{L}(z_1 = z_2) \;=\; (\Sigma\times\{(0,0)\})^*\,(\Sigma\times\{(1,1)\})\,(\Sigma\times\{(0,0)\})^*$$

is regular.

Induction step. For $\phi = \phi_1 \wedge \phi_2$ and $\phi = \neg\psi$, the claim follows from the induction hypothesis and the fact that regular languages are closed under intersection and complementation. Let us first address the case $\phi = \neg\psi$. The induction hypohesis yields that $\mathfrak{L}(\psi)$ is regular. But then $\mathfrak{L}(\,\neg\psi\,)$ is the complement of a regular language, and therefore regular. More precisely, if $\phi = \neg\psi$ is a sentence then so is ψ and

$$\mathfrak{L}(\,\neg\psi\,) \;=\; \overline{\mathfrak{L}(\psi)} \;=\; \Sigma^*\setminus\mathfrak{L}(\psi).$$

If ϕ and ψ have (exactly) k free (first- or second-order) variables where $k \geqslant 1$, then $\mathfrak{L}(\phi)$ and $\mathfrak{L}(\psi)$ are languages over $\Sigma_\phi = \Sigma_\psi = \Sigma \times \{0,1\}^k$ which do not contain the empty word. Thus:

$$\mathfrak{L}(\neg\psi) = \Sigma_\psi^+ \setminus \mathfrak{L}(\psi)$$

For the treatment of conjunction, i.e., $\phi = \phi_1 \wedge \phi_2$ we make use of the observation that if $\mathit{Free}(\phi) = \{z_1,\ldots,z_m\}$ and $\mathit{PFree}(\phi) = \{Y_1,\ldots,Y_r\}$ then $\phi = \phi(z_1,\ldots,z_m,Y_1,\ldots,Y_r)$ and both subformulas ϕ_1 and ϕ_2 can be viewed as formulas

$$\phi_i = \phi_i(z_1,\ldots,z_m,Y_1,\ldots,Y_r), \qquad i = 1,2.$$

Their languages

$$\begin{aligned} L_1 &\stackrel{\text{def}}{=} \mathfrak{L}(\phi_1, z_1,\ldots,z_m,Y_1,\ldots,Y_r) \\ L_2 &\stackrel{\text{def}}{=} \mathfrak{L}(\phi_2, z_1,\ldots,z_m,Y_1,\ldots,Y_r) \end{aligned}$$

over the alphabet $\Sigma_\phi = \Sigma \times \{0,1\}^{m+r}$ are regular by induction hypothesis (which yields that $\mathfrak{L}(\phi_1)$ and $\mathfrak{L}(\phi_2)$ are regular) and the observation made in Remark 2.6.17 (which yields that with $\mathfrak{L}(\phi_i)$ also $L_i = \mathfrak{L}(\phi_i,\ldots)$ is regular). Thus, $\mathfrak{L}(\phi_1 \wedge \phi_2) = L_1 \cap L_2$ is regular.

We now turn to the case where the outermost operator of ϕ is a quantifier. By duality of existential and universal quantification we may assume that ϕ has the form $\exists z.\psi$ or $\exists Y.\psi$.

Second-order quantification. Suppose $\phi = \exists Y.\psi$ where Y is a set variable. If Y has no free occurrences in ψ then $\phi \equiv \psi$ and $\mathfrak{L}(\phi) = \mathfrak{L}(\psi)$ is regular by induction hypothesis. Suppose now that Y appears free in ψ. Suppose that

$$\psi = \psi(z_1,\ldots,z_m,Y_1,\ldots,Y_r,Y) \text{ and } \phi = \phi(z_1,\ldots,z_m,Y_1,\ldots,Y_r)$$

where all variables z_i and Y_j appear free in ψ and ϕ. That is, for the alphabets of the languages $\mathfrak{L}(\psi) \subseteq \Sigma_\psi^*$ and $\mathfrak{L}(\phi) \subseteq \Sigma_\phi^+$ we suppose that

$$\begin{aligned} \Sigma_\psi &= \Sigma \times \{0,1\}^m \times \{0,1\}^{r+1} \\ \Sigma_\phi &= \Sigma \times \{0,1\}^m \times \{0,1\}^r \end{aligned}$$

Recall that $\mathfrak{L}(\phi)$ might contain the empty word, if ϕ is a sentence, while $\varepsilon \notin \mathfrak{L}(\psi)$ as ψ has free variables (namely at least Y). Thus,

$$\Sigma_\psi = \Sigma_\phi \times \{0,1\}$$

where the auxiliary bit serves to encode the interpretation of the set-variable Y. By induction hypothesis, $\mathfrak{L}(\psi)$ is regular. Moreover, we have:

$$\begin{aligned} \mathfrak{L}(\phi) \setminus \{\varepsilon\} = \Big\{ \sigma_1 \sigma_2 \ldots \sigma_n \in \Sigma_\phi^+ : \ & \text{there exists } (\xi_1,\xi_2,\ldots,\xi_n) \in \{0,1\}^n \text{ s.t.} \\ & (\sigma_1,\xi_1)(\sigma_2,\xi_2)\ldots(\sigma_n,\xi_n) \in \mathfrak{L}(\psi) \quad \Big\} \end{aligned}$$

Thus, $\mathfrak{L}(\phi) \setminus \{\varepsilon\}$ agrees with the image of $\mathfrak{L}(\psi)$ under the homomorphism $\Sigma_\psi \to \Sigma_\phi$ that drops the last bit in each symbol of Σ_ψ, i.e., maps each symbol $(\sigma,\xi) \in \Sigma_\psi$ to $\sigma \in \Sigma_\phi$. As regular languages are known to be closed under homomorphism, $\mathfrak{L}(\phi) \setminus \{\varepsilon\}$ is regular. Instead of using this result, we will explain how this *projection* can be realized on the automata-level. As $\mathfrak{L}(\psi)$ is regular, there is an NFA

$$\mathcal{M}_\psi = (Q, \Sigma_\psi, \delta, Q_0, Q_F)$$

such that $\mathfrak{L}(\mathcal{M}_\psi) = \mathfrak{L}(\psi)$. The goal is to construct an NFA $\mathcal{M}_\phi$ with the alphabet Σ_ϕ for $\mathfrak{L}(\phi) \setminus \{\varepsilon\}$ that results from $\mathcal{M}_\psi$ by *projection*. The idea is to *guess* the auxiliary bits in the symbols of Σ_ψ *nondeterministically*. Formally,

$$\mathcal{M}_\phi \stackrel{\text{def}}{=} (Q, \Sigma_\phi, \delta', Q_0, Q_F)$$

where

$$\delta'(q, \sigma) \stackrel{\text{def}}{=} \delta\big(q, (\sigma, 0)\big) \cup \delta\big(q, (\sigma, 1)\big)$$

for all states $q \in Q$ and symbols $\sigma \in \Sigma_\phi$. We now show that for each nonempty word $w \in \Sigma_\phi^+$ we have:

$$w \in \mathfrak{L}(\mathcal{M}_\phi) \quad \text{iff} \quad w \in \mathfrak{L}(\phi)$$

Let us check this. Suppose first that $w = \sigma_1 \ldots \sigma_n \in \mathfrak{L}(\phi)$. Then, $\sigma_i \in \Sigma_\phi$ for $1 \leqslant i \leqslant n$ and there exists a bit vector $(\xi_1, \ldots, \xi_n) \in \{0,1\}^n$ such that:

$$w' = (\sigma_1, \xi_1)(\sigma_2, \xi_2) \ldots (\sigma_n, \xi_n) \in \mathfrak{L}(\psi)$$

Let $q_0\, q_1 \ldots q_n$ be an accepting run for w' in $\mathcal{M}_\psi$. That is, $q_0 \in Q_0$, $q_n \in Q_F$ and

$$q_i \in \delta\big(q_{i-1}, (\sigma_i, \xi_i)\big)$$

for $i = 1, \ldots, n$. But then $q_i \in \delta'(q_{i-1}, \sigma_i)$ and $q_0\, q_1 \ldots q_n$ is an accepting run for w in $\mathcal{M}_\phi$. Therefore, $w \in \mathfrak{L}(\mathcal{M}_\phi)$.

Vice versa, suppose that $w = \sigma_1 \ldots \sigma_n \in \mathfrak{L}(\mathcal{M}_\phi)$. Let $q_0\, q_1 \ldots q_n$ be an accepting run for w in $\mathcal{M}_\phi$. Then, $q_0 \in Q_0$, $q_n \in Q_F$, and $q_i \in \delta'(q_{i-1}, \sigma_i)$ for $1 \leqslant i \leqslant n$. By definition of δ', for each $i \in \{1, \ldots, n\}$ there exists $\xi_i \in \{0,1\}$ such that

$$q_i \in \delta(q_{i-1}, (\sigma_i, \xi_i)).$$

Let

$$w' \stackrel{\text{def}}{=} (\sigma_1, \xi_1)(\sigma_2, \xi_2) \ldots (\sigma_n, \xi_n) \in \Sigma_\psi^*.$$

But then $q_0\, q_1 \ldots q_n$ is an accepting run for w' in $\mathcal{M}_\psi$. Therefore:

$$w' \in \mathfrak{L}(\mathcal{M}_\psi) = \mathfrak{L}(\psi)$$

As w arises from w' by erasing the ξ_i's, this yields $w \in \mathfrak{L}(\phi)$. $\rfloor$

We conclude that $\mathfrak{L}(\phi)$ either agrees with $\mathfrak{L}(\mathcal{M}_\phi)$ or with $\mathfrak{L}(\mathcal{M}_\phi) \cup \{\varepsilon\}$. (The latter case is possible if ϕ is a sentence, i.e., $m = r = 0$.) However, in both cases $\mathfrak{L}(\phi)$ is regular.

First-order quantification. For $\phi = \exists z.\psi$ where z is a first-order variable and

$$\psi = \psi(z_1, \ldots, z_m, z, Y_1, \ldots, Y_r),$$

we may apply exactly the same technique and generate an NFA $\mathcal{M}_\phi$ from an NFA $\mathcal{M}_\psi$ for ψ by projection. Roughly this means that we identify z with the singleton set $\{z\}$. Note that (up to some permutation of the bits for the free variables) we also have $\Sigma_\phi = \Sigma_\psi \times \{0,1\}$ and

$$\mathfrak{L}(\phi) \setminus \{\varepsilon\} = \Big\{ \sigma_1 \sigma_2 \ldots \sigma_n \in \Sigma_\phi^+ : \begin{array}{l} \text{there exists } (\zeta_1, \zeta_2, \ldots, \zeta_n) \in \{0,1\}^n \text{ s.t.} \\ (\sigma_1, \zeta_1)(\sigma_2, \zeta_2) \ldots (\sigma_n, \zeta_n) \in \mathfrak{L}(\psi) \end{array} \Big\}$$

The reason why we do not have to care about the number of 1's in the guessed bits $\zeta_1, \dots \zeta_n$ for z is that for each bit-tuple $(\zeta_1, \dots, \zeta_n) \in \{0,1\}^n$ with none or at least two 1's there is *no* word $\sigma_1 \dots \sigma_n \in (\Sigma \times \{0,1\}^m \times \{0,1\}^r)^*$ such that the insertion of the bit ζ_i in σ_i results in a word of $\mathfrak{L}(\psi)$. $\square$

Example 2.6.19 (Projection). Let $\psi = P_a(z) \wedge z \notin Y \wedge last(z)$ and $\phi = \exists Y.\psi$. Suppose that the underlying alphabet Σ consists of the letters a and b. Then, $\mathfrak{L}(\psi)$ is a language over the alphabet $\Sigma_\psi = \{a, b\} \times \{0,1\}^2$ where the first bit is used for FO-variable z and the second bit for the set variable Y. The alphabet for the language of $\phi = \exists Y.\psi$ is $\Sigma_\phi = \{a, b\} \times \{0,1\}$.

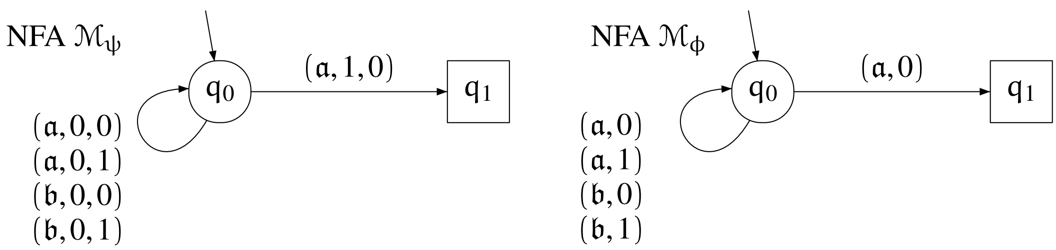

Figure 21: NFA $\mathcal{M}_\psi$ (left) and NFA $\mathcal{M}_\phi$ (right)

Figure 21 shows on the left an NFA $\mathcal{M}_\psi$ for the language $\mathfrak{L}(\psi)$. The NFA on the right results from $\mathcal{M}_\psi$ by projection and accepts the language $\mathfrak{L}(\phi)$. Intuitively, when reading the input symbol $(a, 0) \in \Sigma_\phi$ then $\mathcal{M}_\phi$ guesses nondeterministically whether this input symbol is the last one. $\blacksquare$

Putting Theorems 2.6.10 and 2.6.18 together we obtain that the class of MSO-definable languages over finite words agrees with the class of regular languages. Since Theorem 2.6.10 even yields the existence of an EMSO-formula for each regular language, we even get that EMSO is as expressive as MSO for finite word structures:

Corollary 2.6.20 (MSO, EMSO and regular languages). *Let* $L \subseteq \Sigma^*$ *be a language over finite words. Then, the following statements are equivalent:*

(1) L *is regular.*

(2) L *is EMSO-definable.*

(3) L *is MSO-definable.*

Proof. The implication "(1) $\Longrightarrow$ (2)" has been shown in Theorem 2.6.10, and "(3) $\Longrightarrow$ (1)" in Theorem 2.6.18. The implication "(2) $\Longrightarrow$ (3)" is obvious. $\square$

The MSO-characterization of regular languages has many important applications. One of the immediate consequences is the decidability of the question whether a given MSO-sentence ϕ over $Voc_{\Sigma,\text{graph}}$ is satisfiable over some word structure. For this we can apply the inductive construction of an NFA $\mathcal{M}_\phi$ for ϕ as in the proof of Theorem 2.6.18 and then check whether the accepted language of $\mathcal{M}_\phi$ is nonempty. Note that:

$$\begin{aligned} \text{Graph}(w) \models \phi \text{ for some word } w \in \Sigma^* \quad & \text{iff} \quad \mathfrak{L}(\phi) \neq \varnothing \\ & \text{iff} \quad \mathfrak{L}(\mathcal{M}_\phi) \neq \varnothing \end{aligned}$$

Checking nonemptiness of an NFA simply requires to check whether some final state is reachable from some initial state. This can be realized by means of a DFS-based backward search from the final states in time linear in the size of $\mathcal{M}_\phi$.

Corollary 2.6.21 (Decidability of MSO over finite word structures). *The question whether a given MSO-sentence has a finite word model is decidable.*

Thus, also the dual question that asks whether a given MSO-sentence ϕ over $Voc_{\Sigma,\text{graph}}$ holds for all word models $\text{Graph}(w)$ is decidable. One possibility is to construct an NFA $\mathcal{M}_\phi$ for ϕ and to address the universality problem for $\mathcal{M}_\phi$, i.e., to check whether $\mathfrak{L}(\mathcal{M}_\phi) = \Sigma^*$. The typical method to do so is to construct a finite automaton for the complement of $\mathfrak{L}(\mathcal{M}_\phi)$ and check emptiness for it. This complementation of $\mathcal{M}_\phi$ can be avoided by switching from ϕ to $\neg\phi$ and checking emptiness for $\mathfrak{L}(\mathcal{M}_{\neg\phi})$ as we have:

$$\begin{aligned} \text{Graph}(w) \models \phi \text{ for all words } w \in \Sigma^* \quad & \text{iff} \quad \text{Graph}(w) \models \neg\phi \text{ for no word } w \in \Sigma^* \\ & \text{iff} \quad \mathfrak{L}(\neg\phi) = \varnothing \\ & \text{iff} \quad \mathfrak{L}(\mathcal{M}_{\neg\phi}) = \varnothing \end{aligned}$$

Since complementation of nondeterministic finite automata (which is needed to treat negation in the construction of NFA from MSO-formulas) can lead to an exponential blow-up, the state space of the NFA constructed for a given MSO-formula can be very large. In fact, this is not a matter of our construction as the satisfiability problem for MSO-formulas over (finite) words is known to be *non-elementary*. Non-elementary means that the time complexity of any decision algorithm for the satisfiability problem of MSO over words grows faster than any function of the form

$$2^{2^{2^{\cdot^{\cdot^{\cdot^{2^n}}}}}}$$

This even holds for the first-order fragment of MSO over words.