# 3 Star-free languages

Goal 1: Show that FO[<] defines strict subclass of regular languages.

Goal 2: Find alternative characterisation
↳ Definable in FO[<]
↳ Represented by star-free regular expression

Recall: First-order formulas are WMSO-formulas without second-order variables/quantifiers.

Example:

$\varphi = \forall x: P_a(x) \to \exists y: x<y \wedge P_b(x)$ over $\Sigma = \{a,b,c\}$ defines

"Every a is followed by a b"

$L(\varphi) = \{a,b,c\}^* . b . \{b,c\}^* \cup \{b\}^*$

Note:

$first(x)$, $last(x)$, $x=y$ are still in FO[<]

Goal 1:

Known: FO[<]-languages are regular
(as FO[<] is syntactic restriction of WMSO).

Show: $(aa)^*$ is not FO[<]-definable (but regular).
↳ There is no $\psi$ in FO[<] with $L(\psi) = (aa)^*$

$\Rightarrow$ FO[<]-definable languages form strict subclass of regular languages.

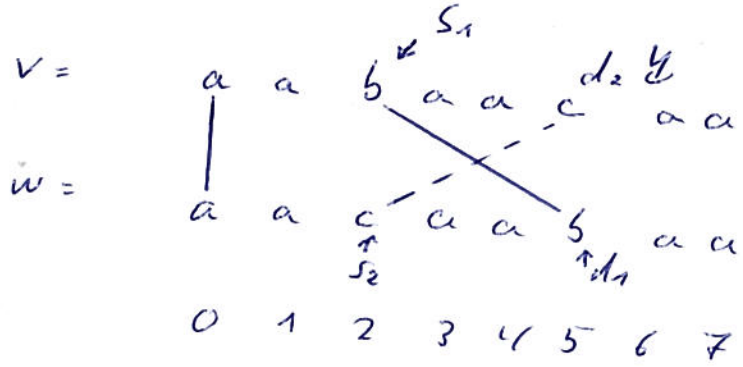
## 3.1 Ehrenfeucht-Fraïssé Games

Tool from finite model theory (logic) for proving inexpressibility results.

The game (informally):

- Two players: spoiler and duplicator
- Two words: $v$ and $w$ over $\Sigma$.
- Number of rounds: $k \in \mathbb{N}$,
- Potentially some existing edges

Let




Per round:

- Spoiler selects position in $v$ or $w$
- Duplicator selects position in other word and connects them by a line
  - ↳ positions have same labelling (preserve $P_a$)
  - ↳ new line does not cross existing lines (preserve $<$)

Next round.

Spoiler may reuse positions:

- then duplicator has to reply by corresponding position
- otherwise duplicator is forced to select new position (if spoiler does not reuse).

- Duplicator looses if cannot reply
- Duplicator wins if number of rounds passes without loss.

In the example:

- Duplicator wins 1 round
- Duplicator looses 2 rounds.

Definition (Partial isomorphism on word structures):

Let $S_v = (D_v, <_v, (P_{a_v})_{a \in \Sigma})$ and

$S_w = (D_w, <_w, (P_{a_w})_{a \in \Sigma})$.

A partial isomorphism (from $S_v$ to $S_w$) is a partial function $p : D_v \rightharpoonup D_w$ so that

(1) $p$ is injective

(2) $\forall x \in dom(p) : \forall a \in \Sigma : P_{a_v}(x)$ iff $P_{a_w}(p(x))$.
// identically labelled.

(3) $\forall x,y \in dom(p)$: $x <_v y$ iff $p(x) <_w p(y)$
// no crossing edges.

Let $\vec{s} = (s_1, \dots, s_n)$ and $\vec{t} = (t_1, \dots, t_n)$.
Typically write $\vec{s} \mapsto \vec{t}$ for $p = \{(s_1, t_1), \dots, (s_n, t_n)\}$

With this definition, interpret EF-game as follows:
- Let $S_v$, $S_w$ two structures with designated elements $\vec{s}, \vec{t}$.
- Duplicator tries to establish partial isomorphism
- Spoiler tries to avoid this.

Overall goal of EF games:
- Compare the structures
- If duplicator wins k-rounds, $S_v$, $S_w$ cannot be distinguished by formulas of quantifier-depth $\leq k$.

Definition (EF-game):
Let $S_v, S_w$ two word structures
and $\vec{s}, \vec{t}$ two vectors of positions in $S_v$ and $S_w$.
Let $k \in \mathbb{N}$.

The EF-game $G_k((S_v, \vec{s}), (S_w, \vec{t}))$ consists of the following rules and rules:
- $k$ rounds
- initial configuration $\vec{s} \mapsto \vec{t}$
- Given a configuration $r$, a round consists of the following moves
  - Spoiler chooses $s \in D_v$ or $t \in D_w$
  - Duplicator chooses $t \in D_w$ or $s \in D_v$.
  - The game continues with $r \cup \{(s,t)\}$ as new configuration.

Duplicator wins k-rounds, if last configuration is partial isomorphism.
Duplicator wins $G_k((S_v, \vec{s}), (S_w, \vec{t}))$ if has a winning strategy: whatever moves spoiler does, duplicator can win $k$-rounds.

Note:

Last configuration a partial isomorphism only if all intermediary configurations are partial isomorphisms.

How to check that duplicator wins $G_k((S_v, \vec{s}), (S_w, \vec{t}))$?

Lemma:

(1) Duplicator wins $G_0((S_v, \vec{s}), (S_w, \vec{t}))$
iff $\vec{s} \mapsto \vec{t}$ is partial isomorphism.

(2) Duplicator wins $G_{k+1}((S_v, \vec{s}), (S_w, \vec{t}))$ iff
(2a) $\forall s \in D_v : \exists t \in D_w :$ Duplicator wins $G_k((S_v, s.\vec{s}), (S_w, t.\vec{t}))$
and
(2b) $\forall t \in D_w : \exists s \in D_v :$ Duplicator wins $G_k((S_v, s.\vec{s}), (S_w, t.\vec{t}))$.

Intuition:

$G_k((S_v, s.\vec{s}), (S_w, t.\vec{t}))$ gives arbitrary first step in $G_{k+1}((S_v, \vec{s}), (S_w, \vec{t}))$.

Ehrenfeucht-Fraïssé Theorem (roughly) says that duplicator wins $G_k((S_v, \vec{s}), (S_w, \vec{t}))$ iff
v and w cannot be distinguished by FO[$\tau$]-formulas of quantifier depth $\leq k$.

Definition (Quantifier depth):

Let $\varphi$ an FO[$\tau$]-formula. The quantifier depth $qd(\varphi)$ is the nesting depth of quantifiers in $\varphi$.
It is defined inductively by

$qd(x < y) := 0$ $\quad$ $qd(P_a(x)) := 0$
$qd(\varphi_1 \vee \varphi_2) := \max\{qd(\varphi_1), qd(\varphi_2)\}$ $\quad$ $qd(\neg\varphi) := qd(\varphi)$
$qd(\exists x : \varphi) := 1 + qd(\varphi)$.

Example:

Let $\varphi = \forall x : P_a(x) \to \exists y : x < y \wedge P_b(x)$. Here, $qd(\varphi) = 2$.

**Definition (k-Equivalence):**

Let $v, w \in \Sigma^*$ and $\vec{s} = (s_1, \dots, s_n)$, $\vec{t} = (t_1, \dots, t_n)$ two vectors of positions.

Then $(S_v, \vec{s})$, $(S_w, \vec{t})$ are *k-equivalent*, $(S_v, \vec{s}) \equiv_{k,n} (S_w, \vec{t})$, if for all $\ell(x_1, \dots, x_n)$ with $qd(\ell) \leq k$ we have

$$S_v, I[\vec{s}/\vec{x}] \models \ell \quad \text{iff} \quad S_w, I[\vec{t}/\vec{x}] \models \ell.$$

Here, $\vec{x} = (x_1, \dots, x_n)$.

For the case of empty sequences, this means the two structures satisfy the same sentences of quantifier depth up to $k$.

**Theorem (Ehrenfeucht-Fraïssé):**

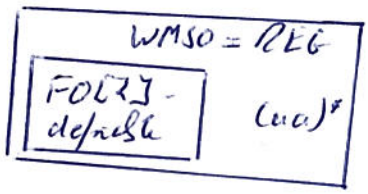
Duplicator wins $G_k((S_v, \vec{s}), (S_w, \vec{t}))$ iff $(S_v, \vec{s}) \equiv_{k,n} (S_w, \vec{t})$.

Why is this cool?

Gives a "pumping argument"

**Proposition:**

Language $L = (aa)^*$ is not FOL[<]-definable.




**Proof:**

Towards a contradiction, assume there was $\ell$ with $L = L(\ell)$.

Then $\ell$ has some quantifier-depth $k \in \mathbb{N}$.

One can check that duplicator wins

$$G_k(a^{2^k}, a^{2^k+1})$$

Hence, the two words cannot be distinguished by formulas of qd at most $k$ (by Ehrenfeucht-Fraïssé Theorem)

Hence, $a^{2^k}$ and $a^{2^k+1}$ cannot be distinguished by $\mathcal{C}$.
If contradiction to $a^{2^k} \in L(\mathcal{C})$ but $a^{2^k+1} \notin L(\mathcal{C})$.
Formula $\mathcal{C}$ cannot exist. □