In propositional logic, we have seen that every property can be expressed by a formula where the negation sign $\neg$ is only used in front of propositional atoms ($\top$,$\bot$,p). A similar observation holds true for first-order logic. This time we're not arguing semantically (we don't have an overview over the expressible properties yet!) but syntactically. A similar syntactical argument would have also been possible for propositional logic.

Negation Normal Form

Lemma.
$\mathcal{A}, \eta \models \neg\forall x\varphi$ iff $\mathcal{A}, \eta \models \exists x\neg\varphi$
$\mathcal{A}, \eta \models \neg\exists x\varphi$ iff $\mathcal{A}, \eta \models \forall x\neg\varphi$

Recall from propositional logic.
$\mathcal{A}, \eta \models \neg(\varphi \wedge \psi)$ iff $\mathcal{A}, \eta \models (\neg\varphi) \vee (\neg\psi)$
$\mathcal{A}, \eta \models \neg(\varphi \vee \psi)$ iff $\mathcal{A}, \eta \models (\neg\varphi) \wedge (\neg\psi)$
$\mathcal{A}, \eta \models \neg\neg\varphi$ iff $\mathcal{A}, \eta \models \varphi$

So we can push all the $\neg$ symbols inwards, till they're only in front of atomic formulae. In fact, inspecting this transformation in more detail, we note that the size of the formula at most doubles; in the worst case, every atomic formula gets a negation in front.

The new concept in first-order logic are quantifiers. Also for them, we can find a canonical place in the formula: the beginning. For our argument, we can already assume the formula to be in negation normal form.

Prenex Normal Form

Lemma. Assume $x \notin \mathsf{fv}(\psi)$.
$\mathcal{A}, \eta \models (\forall x \varphi) \land \psi$ iff $\mathcal{A}, \eta \models \forall x(\varphi \land \psi)$
$\mathcal{A}, \eta \models (\forall x \varphi) \lor \psi$ iff $\mathcal{A}, \eta \models \forall x(\varphi \lor \psi)$
$\mathcal{A}, \eta \models (\exists x \varphi) \land \psi$ iff $\mathcal{A}, \eta \models \exists x(\varphi \land \psi)$
$\mathcal{A}, \eta \models (\exists x \varphi) \lor \psi$ iff $\mathcal{A}, \eta \models \exists x(\varphi \lor \psi)$

For the proof we use that, by the coincidence lemma, the truth value of ψ is not affected by the assignment to x.

The lemma allows us to "pull out" all quantifiers to the front. Not that we may always rename the bound variables in such a way that no conflict arises. Also note, that bringing a formula to prenex normal form does not increase its size.

2.4 Ehrenfeucht-Fraïssé Games

For LTL the simple observation that we can describe the model-checking problem by a finite automaton immediately lead to a result that certain properties are not expressible by LTL-formulae. To show get concrete examples of properties not expressible in first-order logic, we need slightly more elaborate techniques.

So, what does "inexpressibility" mean? To show that a certain property (like "the universe has even size") *cannot* be expressed by a certain class $\mathcal{L}$ of formulae, it certainly suffices to exhibit two structures $\mathcal{A}$ and $\mathcal{B}$ such that

> $\mathcal{A}$ has the property, $\mathcal{B}$ doesn't
>
> For all $\varphi \in \mathcal{L}$
>
> $$\mathcal{A} \models \varphi \quad \text{iff} \quad \mathcal{B} \models \varphi$$

So, the problem we're looking at is how to construct different structures that are hard to distinguish. Of course, the smaller $\mathcal{L}$ is the easier it is to come up with examples. Since we will need the last property several times, we introduce a notation for it.

> **Definition.** $\mathcal{A}, \vec{a} \equiv_{\mathcal{L}} \mathcal{B}, \vec{b}$ iff for all $\varphi \in \mathcal{L}$, either both $\mathcal{A}, \vec{a} \models \varphi$ and $\mathcal{B}, \vec{b} \models \varphi$ or neither.
>
> Here $\vec{a} = a_1, \ldots, a_\ell$ and $\vec{b} = b_1, \ldots, b_\ell$.

So, how can first order structures look different? Let's look at some examples.

Words—Spot the difference!

`a a b a a` *vs* `a a a a a`

`a a b a a` *vs* `a a a a b`

`a b a c a` *vs* `a a b c a`

In the first example, the left-hand structure has a letter `b`, which the right-hand structure doesn't.

In the second example, both structures have a letter `b`, but on the left-hand structure we can find an to the right of it.

In the third example both structures have a `b` and the same set of letters (`a`, `c`) to the right of it. But only in the left-hand structure we can find a `b` and a `c` with an `a` between them.

Generalising this sequence of more and more complicated differences between structures, we define a game that mimics this step by step discovery of a difference in the structure.

The game, called Ehrenfeucht-Fraïssé game, is a game for two players.

- **Spoiler** tries to prove that the two structures are different.
- **Duplicator** tries to pretend as long as possible, that the structures are the same.

The game is organised in rounds. In each round, Spoiler tries to highlight a difference between the structures by putting an additional pebble in one of the structure. Duplicator tries to match this move in the other structure. The game ends when either the fixed number of rounds is over, or the substructure given by the pebbles differ.

Ehrenfeucht-Fraïssé Games

The game is played on configurations

$$\underbrace{(A, R_1^{\mathcal{A}}, R_2^{\mathcal{A}}, \ldots)}_{\mathcal{A}}, a_1, \ldots, a_k \mid \underbrace{(B, R_1^{\mathcal{B}}, R_2^{\mathcal{B}}, \ldots)}_{\mathcal{B}}, b_1, \ldots, b_k$$

where $a_1, \ldots, a_k \in A$ and $b_1, \ldots, b_k \in B$.

In each round, Spoiler picks either $a_{k+1} \in A$ or $b_{k+1} \in B$. Then Duplicator picks the other.

Duplicator needs to keep the invariants

- $a_i = a_j$ iff $b_i = b_j$
- $R_i^{\mathcal{A}}(a_{i_1}, \ldots, a_{i_\ell})$ iff $R_i^{\mathcal{B}}(b_{i_1}, \ldots, b_{i_\ell})$

Back to our examples, we see that in the first structure, Duplicator cannot survive a single round. In the second structure, Duplicator can survive the first round, but not the second. And in the third structure, Duplicator can survive two rounds, but not the third.