#Remember

1 Things to remember

- FO - First order logic is powerful and expressive
- Model-checking is decidable (in PSPACE) when the universe is finite.
- Satisfiability,Validity,logical equivalence are all undecidable (proven a by the Turing machine to Domino-Problem reduction)
- For infinite universes one ca fix a model and study its FO-Theory. Some FO-theories are decidable some not.
- Some FO-theories can be compared to other FO-theories using Logical reduction

2 Games in Logic

Goal: check which properties/ languages are *definable* in logic (for instance FO - First order logic)

Examples:

- Is the property "Universe has even Cardinality" definable in FO - First order logic?
- Is the class of "Strongly connected graphs" definable in FO - First order logic using edges? (Not possible in FO - First order logic)
- Is the language $L = (AA)*$ definable in FO - First order logic using $\leq$ (Not possible in FO - First order logic)

  > meaning: $L = (AA)*$ is the set of all finite strings that have even length

Problem: If we want to prove one of this properties it is very easy to proof if it is possible to prove. We write down the formula, check if it says what we want it says and it is done.
It is different though if we can not find a prove. It is very difficult to show that something is not possible.

2.1 The evaluation game also known as model checking game

GOAL: check whether $s \models \phi$

Example:
The evaluation game for QBF
input: ϕ
e.g. : $\forall p \exists q \forall r (p \vee \neg q) \wedge (r \vee q)$

The Structure $S = \emptyset$

Question: $S \overset{?}{\models} \phi$

Now we use a Tableaux tree structure back from Verification 3 to see if the formula is valid. We add $\forall$ and $\exists$.

This is the first picture of the Tableaux. We start with a $\forall$. If there is a $\forall$ every branch has to be positive. This is every node that has red text color. The nodes with $\exists$ have green text color and there only one of the branches needs to be $True$.

After the branches with the $\exists$ and $\forall$ we have to append the tree bottom at every position that has a $'*'$ in the tree top .

Tree top:

Tree bottom:

Question: How do we evaluate the tree? What happens if there is a contradiction like in this tree.

This is very similar to a game (2-player game)

2.1.1 2-player game

Player:

- $\text{Adam } (\forall, \wedge)$ in literature called **Verifyer**
- $\text{Eve } (\exists, \vee)$ in literature called **Falsifier**

One game can have multiple rounds. One **round** of the game consists of Adam and Eve choosing branches of the node (either go left or right) taking turns. $Adam$ chooses whenever there is a $\forall$ or a $\wedge$, Eve chooses when there is a $\exists$ or a $\vee$. The round ends when they have reached the end of a branch i.e. a leaf of the tree.

What is the goal ?

> Eve wants to find a Strategy that allows her to reach a $True$ leaf (of the board) no matter what Adam does!

Board / Arena:
Is a tree or a graph might be finite or infinite. In our case it is the tree that we drew above.

What is the point? What is behind this?
We want to check if a formula holds on a Structure! Here we have a Lemma!

Lemma 8:
$S \models \phi \iff$ Eve has a Strategy where she always wins the game over the board $G_{\phi,S}$

Example:

$$\phi = \forall p \exists q (p \wedge \neg q) \vee (\neg p \wedge q)$$

Our structure S is empty because we do not have free variables. $S = \emptyset$

What is the board:
Upper tree:

Lower tree:

Now we evaluate the turns Eve can do to counter $Adam$.

Due to our way to paint the graph the $Adam$ chooses at the $\forall$ and Eve chooses the way at the $\exists$. The choice of $Adam$ at the $\wedge$ is symbolized by the fact that a leave has to have all entries *True* to count as *True*.

First what happens if $Adam$ chooses the left branch and sets $p = 0$

Eve has two options. First we look what happens when she chooses the left branch setting $q = 0$. As already noted before Eve controls the $\lor$ meaning she can choose at the crossings in the lower tree where to go. If she finds a way to a *True* leaf the entire evaluation is counted as *True*.

> $p = 0, q = 0$
>
> - left branch: $\{p = 0, \neg q = 1\} \rightarrow \text{False}$
> - right branch: $\{\neg p = 1, q = 0\} \rightarrow \text{False}$
>   As both leafs are *False* the Eve does not have a choice and this branch is counted as *False*.

Now we look at the other option of $Eve's$ the right branch setting $q = 1$

> $p = 0, q = 1$
>
> - left branch: $\{p = 0, \neg q = 0\} \rightarrow \text{False}$
> - right branch: $\{\neg p = 1, q = 1\} \rightarrow \text{True}$
>   As the second leave is True Eve always can choose to go to the *True* leave hereby this branch counts as *True*

Concluding if $Adam$ chooses the left branch and sets $p = 0$, Eve can respond by choosing first the right branch setting $q = 1$ and then again the right branch at the first crossing in the lower tree, hereby finding a way to set the formula *True* and win.

What happens if $Adam$ chooses the right branch and sets $p = 1$?

> $p = 1, q = 0$
>
> - left branch: $\{p = 1, \neg q = 1\} \rightarrow \text{True}$
> - right branch: $\{\neg p = 0, q = 0\} \rightarrow \text{False}$
>   As the first leaf is True Eve always can choose to go to the *True* leaf, hereby this branch counts as *True*.

Now we look at the other option of Eve choosing the right branch setting $q = 1$.

> $p = 1, q = 1$
>
> - $\{p = 1, \neg q = 0\} \rightarrow \text{False}$
> - $\{\neg p = 0, q = 1\} \rightarrow \text{False}$
>   As both leaves after the crossing are *False* the Eve does not have a choice and this branch is counted as *False*.

To sum up if $Adam$ chooses the right branch Eve can always counter by choosing the left branch in the upper tree setting $q = 0$ and then the left branch to reach a *True* leaf

---

Summarizing if $Adam$ chooses the left branch Eve can counter by choosing the right branch and then the right branch and win.
If $Adam$ chooses the right branch Eve can counter by choosing the left branch and then the left branch again.

This means that $\textit{Eve}$ always has a strategy to win over the board $G_{\phi,S_\emptyset}$! Hereby showing that $S \models \phi$ after Lemma 8.

This was an example for QBF. We could also do The evaluation game also for FO - First order logic the only difference is that the universe does not only contain two elements but n elements meaning that $\textit{Adam}$ and $\textit{Eve}$ have n possibilities to choose from instead of only two.

2.1.2 Formal Definitions for the 2-player game

What do we need:

- We need the formula ϕ in Negation Normal Form (due to a negation swapping the roles of Adam and Eve)

Arena:

> Nodes: are named α and λ
>
> - α: is a sub-formula of ϕ
> - λ is a binding interpreting the Free variables α into the universe U_s

Rules:
First lets look at the easy case: α is a leaf and does not contain $\vee$ or $\wedge$.

- if $\alpha = R(x_1, \dots, x_k)$ then the game ends $\textit{Eve}$ wins if $(\lambda(x_1), \dots \lambda(x_k)) \in R^s$, otherwise $\textit{Adam}$ wins.
  Meaning:
  Before we have been in QBF the we did not have proper relationship symbols. Now the Relationship symbols R are in leafs and need to be applied to our variables x i.e. $R(x_1, \dots, x_k)$. Afterwards we interpret the resulting variables x and see if it is part of the interpretation of the relation. For instance the in the QBF example the interpretation of p i.e. $\lambda(p)$ needed to be equal to one.
- if $\alpha = \neg R(x_1, \dots, x_k)$ then game ends $\textit{Adam}$ wins if $(\lambda(x_1), \dots \lambda(x_k)) \in R^s$, otherwise $\textit{Eve}$ wins.
  Meaning: the same as above only that the result is negated by the $\neg$ following that $\textit{Adam}$ wins
- if $\alpha = \alpha_1 \vee \alpha_2$: then eve can choose which branch to take down the tree. i.e. $\alpha' = \{\alpha_1, \alpha_2\}$ hoping that the chosen branch is *True*. The game continues at (α', λ). The tree looks like this and eve can choose which branch to follow down:
- if $\alpha = \alpha_1 \wedge \alpha_2$: then Adam can choose which branch to take down the tree. i.e. $\alpha' = \{\alpha_1, \alpha_2\}$ hoping that the chosen branch is *False*. The game continues at (α', λ).
- if $\alpha = \exists x \alpha'(x)$ $\textit{Eve}$ can choose any element u that is part of the universe U^s $(u \in U^S)$ to be assigned (bound) to x, the game continues with at the position (α', λ') where $\lambda' = \lambda[x := u]$ i.e. a variable x gets assigned and element u of the universe U^S
- if $\alpha = \forall x \alpha'(x)$ $\textit{Adam}$ can choose any element u that is part of the universe U^s $(u \in U^S)$ to be assigned (bound) to x, the game continues with at the position (α', λ') where $\lambda' = \lambda[x := u]$ i.e. a variable x gets assigned and element u of the universe U^S

Break

2.2 Definability vs elementary equivalence vs n-equivalence

2.2.1 Definability

Sometimes the question is if a certain property P (often expressed in natural language) can be expressed in a certain logic.

Example:
We have a Definability problem and we express a property in Monadic second order logic, a more complex logic than FO. Now we want to know can we also express this property in FO - First order logic?

Notation:

$$P: \text{ property (i.e. a set of structures)}$$
$$S: Structure$$
$$\phi: \text{FO formula}$$

1. P **is defined by** ϕ if for every S $S \in P \iff S \models \phi$

2.2.2 elementary equivalence

1. S, S' elementary equivalent if for every ϕ: $S \models \phi \iff S' \models \phi$

elementary equivalence is probably called elementary equivalence because FO - First order logic was previously called elementary logic.

Lemma 9

If there are two structure S, S' such that:
$S \in P$ and $S \notin P$ and S and S' are elementary equivalent then $\text{P is not definable in First order logic}$

Proof through contradiction:
We have S, S' that are elementary equivalent and $S \in P$ and $S \notin P$.

Lets assume we have a property P that is defined by ϕ.

defined by means that every Structure S holds:

1. $S \models \phi$
2. $S \in P$

Now we know that S and S' are logically equivalent i.e. for every ϕ $S \models \phi \iff S' \models \phi$

So that P is definable in FO - First order logic the following statement needs to hold:
$S \in P \iff S \models \phi \iff S' \models \phi \iff S' \in P$
But by definition S' is not part of P i.e. $S' \notin P$ that means this statement cant hold and we found a contradiction resulting in Lemma 9!

2.2.2.1 What is the use-case of Lemma 9?

If we want to prove that a property is not definable than we find two structures S and S' that are elementary equivalent and one fulfills the property and one not. Then we can point at Lemma 9 and have proven that the property is not definable in FO - First order logic.

2.2.3 Quantifier rank

Note

ϕ has Quantifier rank n if it has at most n nested quantifiers

The number of Quantifiers is normally much smaller than the Quantifier rank.

The Quantifier rank is smaller or equal the number of Quantifiers.

Example:

$$\phi = \forall x \forall y (\neg R(x,y) \vee (\exists z R(x,z)) \vee (\exists t R(t,y)))$$

ϕ has Quantifier rank 3.

2 Ranks are because of the two Quantifiers $\forall x$ and $\forall y$ which cover the entire formula.

It is only one Rank for $\exists z$ and $\exists t$ as they are on the same level in the formula when looking at the formula in tree form.

One could resolve the formula to the following making the rank clearer:

$$\phi = (\forall x \forall y \quad \neg R(x,y) \vee (\forall x \forall y \exists z \quad R(x,z)) \vee (\forall x \forall y \exists t \quad R(t,y)))$$

2.2.4 n-equivalence

Note

S, S' are n-equivalent if for every ϕ with Quantifier rank n the following holds true: $S \models \phi \iff S' \models \phi$

elementary equivalence is more strict as n-equivalence.

Lemma 10

If there are two structure S, S' such that:
$S \in P$ and $S \notin P$ and S and S' are n-equivalent then $\text{P is not definable in First order logic}$

NEW GOAL:
We want to construct a game $G_{S,S'}$ whose winner determines weather S and S' are n-equivalent.

2.3 The Ehrenfeucht-Fraise Games

Players:

- Duplicator :

  > Goal: Wants to prove that S' and S are n-equivalent.

- Spoiler :

  > Goal: Want to disprove that S' and S are n-equivalent

Arena:
The arena is a graph. The nodes of the graph are tuples $(u_1, \ldots, u_i, v_1, \ldots, v_i)$ that are chosen from the universe $u_x \in U^S$ and $v_x \in U^{S'}$

One plays **n** rounds.
At the start there are two empty sets $S|\{\}$ and $S'|\{\}$ who will contain chosen elements of the Spoiler and the Duplicator.

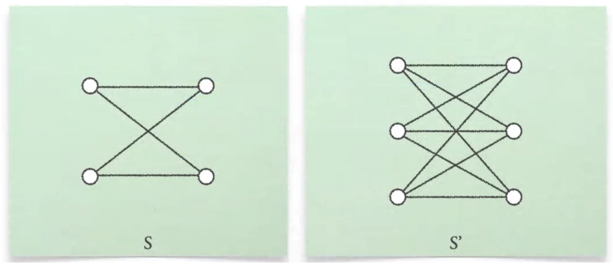
one round:
Spoiler begins: and chooses an element u_i from the universe U^S or a element v_i from the universe $U^{S'}$ and adds it either to $S|\{\}$ or $S'|\{\}$.

Duplicator responds: and chooses and element v_i from $U^{S'}$ if Spoiler has chosen a element u_i from U^S or by choosing a element u_i from U^S if Spoiler has chosen a v_i from $U^{S'}$ and adds it to the sets.

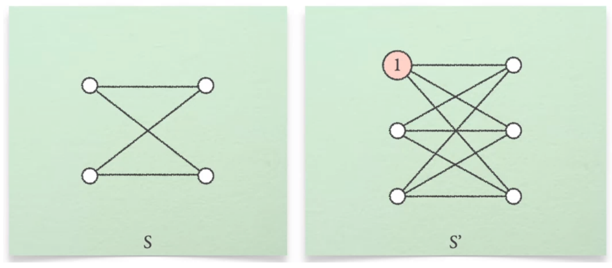
Who wins?
The duplicator wins if after n turns the two sets (now containing the chosen elements of u_x and v_x) $S|\{u_1, \ldots, u_n\}$ and $S'|\{v_1, \ldots, v_i\}$ are isomorphic

2.3.1 Example 1

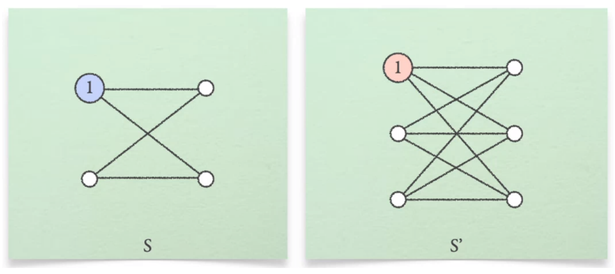
Board:
Two Graphs S and S':



First turn:
The Spoiler chooses a node in S'



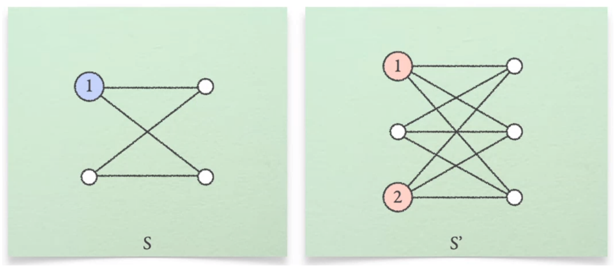
And the duplicator responds. In this stage of the game he can choose any node. He can't loose. But in this example he chooses the one in the upper left corner node in S because it is similar to the choice of the Spoiler.



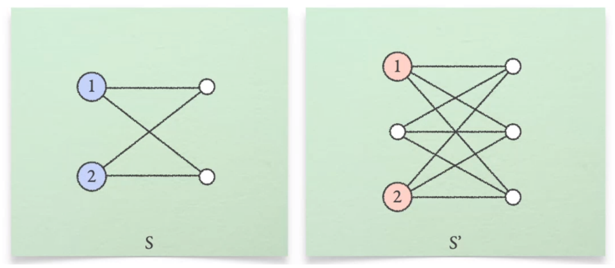
The result of this round is that both sets have a single isolated node in them. They are still isomorphic.

Second turn

Now the duplicator chooses another node that is isolated. There are still no nodes that have no connections (edge).

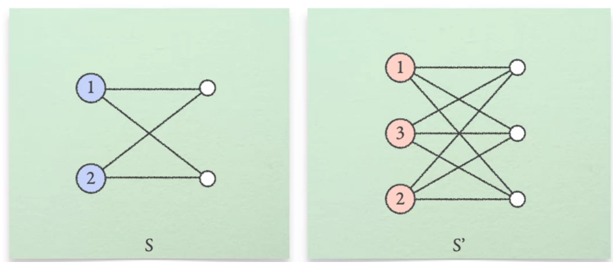


The challenge for the duplicator is now to find a node that is also isolated. And he finds one by the node in the lower left corner.



Third turn:

The Spoiler chooses now a third node that is isolated!



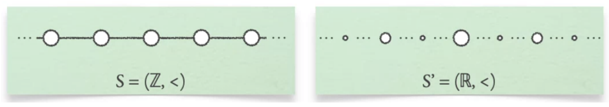
But here the duplicator can not respond anymore independent of the choice he does he can not choose a third node that is isolated! He loses after two turns!

Spoiler has a strategy to kill the duplicator in the third rounds! They are only 2-equivalent

Note: One can also choose the same node twice, but in the example the structure is still different!

2.3.2 Example 2

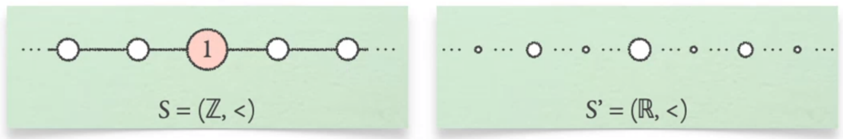
Board:
Two Graphs S and S':



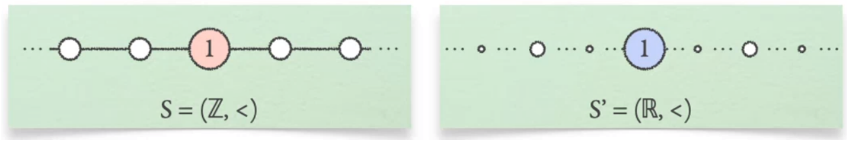
Questions to ask:

- Can the Duplicator survive n rounds e.g. 3 rounds?
- Can we survive n rounds for every n? -> a finite number of rounds
- Can the Duplicator survive for ever?

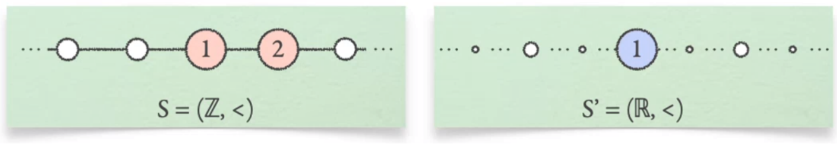
Turn 1 =
The spoiler chooses one element in S:



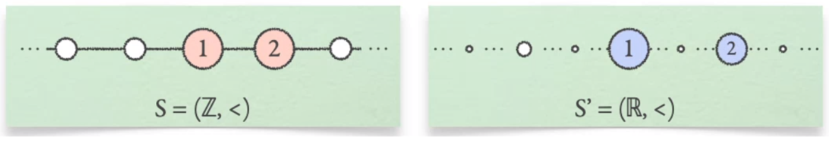
And the Duplicator has plenty of choice and responds:



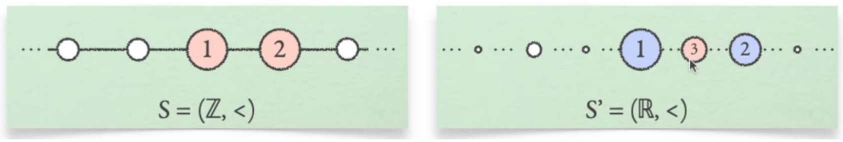
Turn 2
The spoiler chooses to adjacent elements in S



The duplicator chooses another element in S'.



Turn 3
Now the spoiler chooses a node not in S but in S' he can also do that between the two elements that the Duplicator choose. This is possible due to the nature of $\mathbb{R}$ where there are always an infinite amount of elements between two elements. And with this he caught the Duplicator in a trap.



The duplicator can not choose a element between the previous chosen two elements of the Spoiler and hereby he loses.

Spoiler has a strategy to kill the duplicator in the third rounds! They are only 2-equivalent

2.3.2.1 How does this example look in mathematical writing?

$$(\mathbb{R}, <) \models \phi = \forall x \forall y \exists (x < y \to (x < z \land z < y))$$

$$(\mathbb{Z} \not\models \phi)$$

Question: How does this translate to n-equivalence Does it mean that the above formula can be at most 3-equivalent because it has only three nested Quantifiers?

2.4 When does the Spoiler always win?

On non-isomorpic finite structures, Spoiler wins the Ehrenfeucht Fraise game eventually.

Why? ==
Due to the **finite** amount of elements in the structure at some point all elements of one of the two structures will be selected. Then, due to its non-isomorphic nature a ==element that is not present in one of the two structures will be chosen. In that moment the Spoiler wins as the Duplicator can not chose the element as it is not present.

A measure of similarity remoteness

How long the duplicator survies in the game is a measure of how simmilar two structures are. The longer he survives the more simmilar the structures are.