Finite model theory

2. Games

Games: general issues

in finite model theory, games are typically used to show that certain properties are **not** expressible is certain logics. These are typically referred to as 'undefinability results'. Let us briefly discuss informally what this all is about—we shall define all the notions in this slide in detail later on formally, beginning from the next slide.

Typically, a logic $\mathcal{L}$ is associated with a class $\mathcal{G}$ of games, and the individual games in $G \in \mathcal{G}$ can then be used to show that some relevant properties of interest cannot be expressed in $\mathcal{L}$ with sentences satisfying some syntactic restrictions dictated by G. For example, the class $\mathcal{G}_{\mathrm{EF}}$ of so-called Ehrenfeucht-Fraïssé games can be used to study first-order logic FO. So-called 'Ehrenfeucht-Fraïssé games $\mathrm{EF}_k \in \mathcal{G}_{\mathrm{EF}}$ of rank k' can be used to identify properties that cannot be defined by sentences of FO with 'quantifier rank k or less'.

Ehrenfeucht-Fraïssé Game

The Ehrenfeucht-Fraïssé game is a type of game that can be used to measure *similarity* of models of the same vocabulary. The game is played by two players, the **spoiler** and **duplicator**. Sometimes the players are also called **Abelard** and **Eloise**, or **player I** and **player II**.

Let τ be a purely relational vocabulary. Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-models, and let $k \in \mathbb{N}$. The **rank k Ehrenfeucht-Fraïssé game** $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ between $\mathfrak{A}$ and $\mathfrak{B}$ is played by the spoiler and duplicator for k-rounds, each round consisting of the following actions:

1. The spoiler picks an element v of one of the models.
2. The duplicator responds by picking an element u of the other model.

Ehrenfeucht-Fraïssé Game

The moves of the players are encoded in **stages**. The stage 0 (or zeroeth stage) is the pair

$$((\mathfrak{A}, \emptyset), (\mathfrak{B}, \emptyset)).$$

The ith stage (or stage i) is a pair of type

$$((\mathfrak{A}, (a_1, \ldots, a_i)), (\mathfrak{B}, (b_1, \ldots, b_i)))$$

where $a_j \in A$ and $b_j \in B$ are the jth elements picked by the players.

Note: it is of course to leave some of the brackets unwritten as long as it is clear what is meant.

Ehrenfeucht-Fraïssé Game

The **duplicator loses** (and spoiler wins) at the jth stage

$$\left((\mathfrak{A}, (a_1, \ldots, a_j)), (\mathfrak{B}, (b_1, \ldots, b_j)) \right)$$

if $\{(a_1, b_1), \ldots, (a_j, b_j)\}$ is not a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. No further moves are made in this case, and the jth stage is called an **ending position** or a **final stage**.

Ehrenfeucht-Fraïssé Game

The last possible stage in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ is

$$\big((\mathfrak{A}, (a_1, \ldots, a_k)), (\mathfrak{B}, (b_1, \ldots, b_k)) \big).$$

The **spoiler loses** (and duplicator wins) if $\{(a_1, b_1), \ldots, (a_k, b_k)\}$ is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. Otherwise the duplicator loses and spoiler wins.

Thereby the task of the duplicator is to *maintain a partial isomorphism* between the models (intuitively duplicating the choices of the spoiler). The spoiler's task is to break the partial isomorphism (spoil it).

Ehrenfeucht-Fraïssé Game

Thereby the duplicator's goal is sometimes referred to as a *safety goal* while the spoiler's goal is a *reachability goal*. Indeed, the duplicator is trying to maintain a 'safe' stage and the spoiler—on the other hand—is trying to reach a stage where the partial isomorphism no longer holds.

Note that precisely one of the players will succeed and win, and the other player will lose.

Reachability and safety are more generally paradigmatic concepts in algorithmic game theory and automata theory...

Ehrenfeucht-Fraïssé Game

The sequence

0. $\big(\ (\mathfrak{A}, \emptyset), (\mathfrak{B}, \emptyset)\ \big)$
1. $\big(\ (\mathfrak{A}, a_1), (\mathfrak{B}, b_1)\ \big)$
2. $\big(\ (\mathfrak{A}, (a_1, a_2)), (\mathfrak{B}, (b_1, b_2))\ \big)$

$\vdots$

i. $\big(\ (\mathfrak{A}, (a_1, \ldots, a_i)), (\mathfrak{B}, (b_1, \ldots, b_i))\ \big),$

where stage $i \leq k$ is the final stage, is called a **play** of the game $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$. Note that thus we differentiate between the *game* and a *play* of that game. Indeed, there are generally several plays associated with the game $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$.

Ehrenfeucht-Fraïssé Game

A **strategy of the spoiler** in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ is function that outputs, with the input of a stage, a move (i.e., an element in one of the domains of the two models) for the spoiler. Strictly speaking, the output is a pair $(d, \mathfrak{M})$ where d is the element chosen and $\mathfrak{M} \in \{\mathfrak{A}, \mathfrak{B}\}$ the model from which d was chosen; the only reason we need to write $\mathfrak{M}$ into the output specification is that the models $\mathfrak{A}$ and $\mathfrak{B}$ could have overlapping domains.

A **strategy of the duplicator** is a function that outputs, with the input of a stage and the spoiler's most recent move, a response by the duplicator, i.e., an element in the model where the spoiler did not choose. For example, the duplicator's strategy could output b_{i+1} with the input

$$\big((\mathfrak{A}, (a_1, \ldots, a_i, a_{i+1})), (\mathfrak{B}, (b_1, \ldots, b_i)) \big),$$

leading to the stage

$$\big((\mathfrak{A}, (a_1, \ldots, a_i, a_{i+1})), (\mathfrak{B}, (b_1, \ldots, b_i, b_{i+1})) \big).$$

Note that the input to the strategy function is not strictly speaking a stage, because we have the extra move a_{i+1} of the spoiler marked, and thus the tuples $(a_1, \ldots, a_i, a_{i+1})$ and $(b_1, \ldots, b_i)$ are of different lengths.

Ehrenfeucht-Fraïssé Game

However, the output tuple

$$\left((\mathfrak{A}, (a_1, \ldots, a_i, a_{i+1})), (\mathfrak{B}, (b_1, \ldots, b_i, b_{i+1})) \right)$$

is a legitimate stage.

Note carefully that the spoiler can always choose from either of the models $\mathfrak{A}$ and $\mathfrak{B}$. In our example above, the spoiler chose from $\mathfrak{A}$ and the duplicator responded by choosing from $\mathfrak{B}$, but this can be also the other way around.

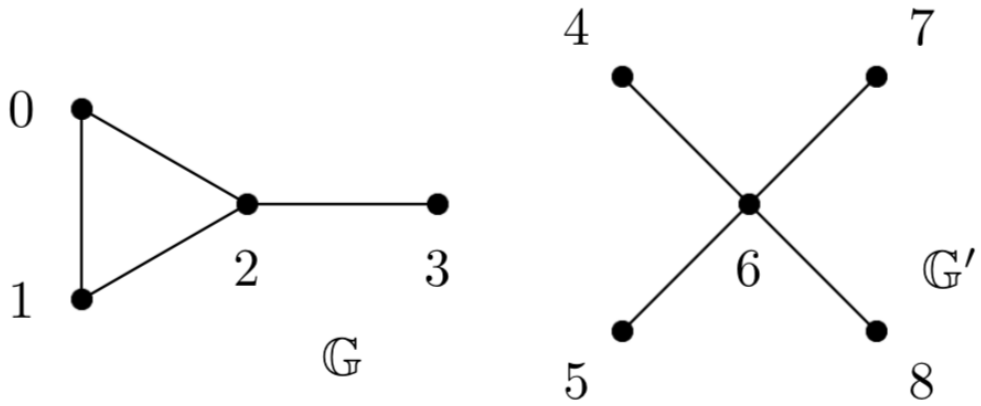
Ehrenfeucht-Fraïssé Game

Definition 2.1

A strategy for the spoiler (respectively, the duplicator) is a **winning strategy** in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ if the spoiler (respectively, duplicator) wins every play of the game $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ where the choices are made according to the strategy.

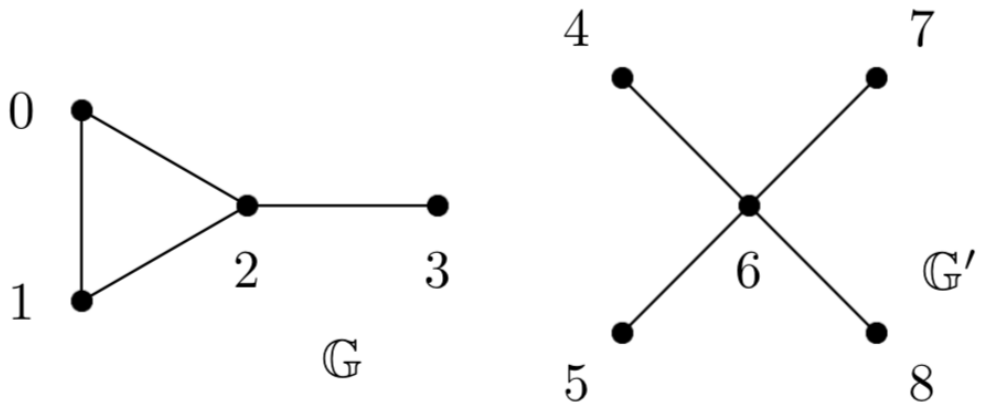
Rather than writing explicitly the function corresponding to a winning strategy, we typically describe winning strategies informally (but nevertheless rigorously).

Ehrenfeucht-Fraïssé Game



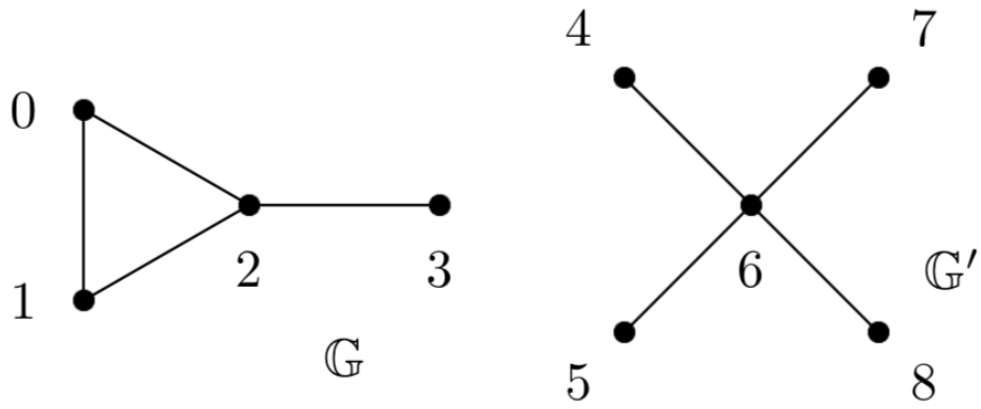
Consider the game $\mathrm{EF}_2(\mathbb{G}, \mathbb{G}')$. We will show that the duplicator has a winning strategy in this game. Indeed, suppose the spoiler chooses a point $s_1 \in G \cup G'$ in the first round. Let us first assume that $s_1 \notin \{2, 6\}$. The duplicator responds by a choice $d_1 \notin \{2, 6\}$ from the other model. Now suppose the spoiler chooses s_2. There are several cases.

Ehrenfeucht-Fraïssé Game



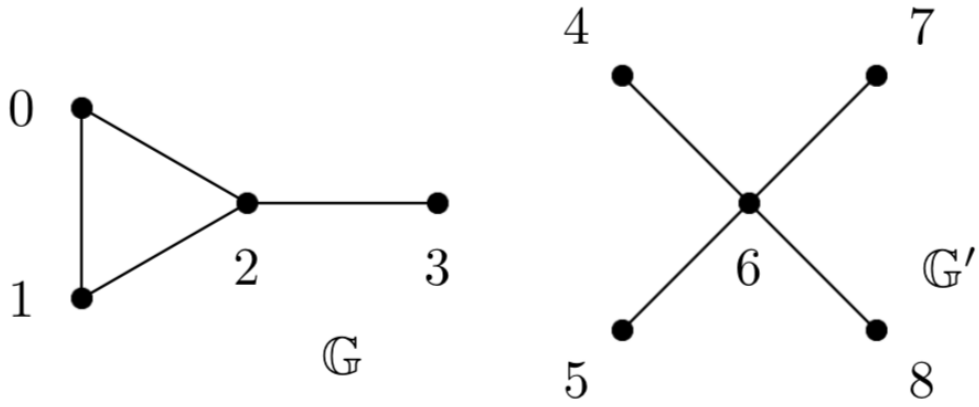
1. If s_2 is a node that was already chosen in round 1, then the duplicator can clearly also choose d_2 from the other model to be a node that was already chosen in round 1. Thus the duplicator wins the game. Thereby we now assume that s_2 does not repeat an already chosen node. Note that the duplicator can always respond to such repeated choices also in longer games and in all models, so we do not really have to in general take such repetition moves into account.

Ehrenfeucht-Fraïssé Game



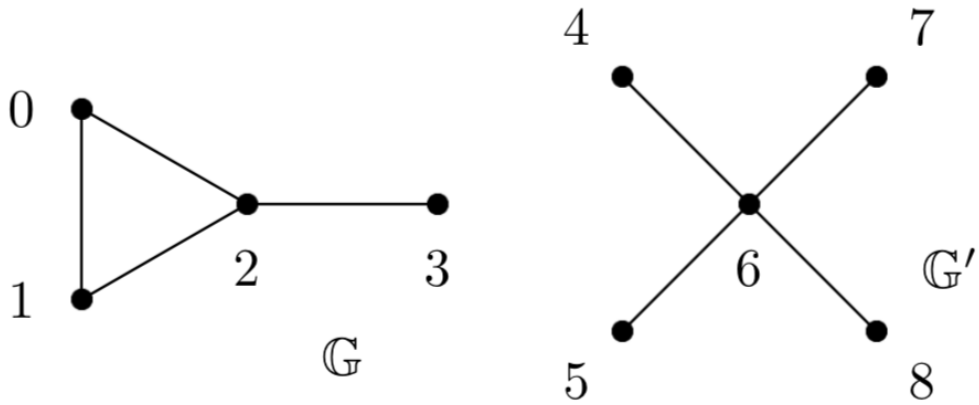
2. If s_1 and s_2 are in the same model $\mathfrak{M} \in \{\mathbb{G}, \mathbb{G}'\}$ and it holds that $(s_1, s_2) \in E^{\mathfrak{M}}$, then the duplicator can clearly choose a node d_2 from the other model $\mathfrak{N}$ so that $(d_1, d_2) \in E^{\mathfrak{N}}$. Thus the duplicator wins.
3. If s_1 and s_2 are in the same model $\mathfrak{M} \in \{\mathbb{G}, \mathbb{G}'\}$ and it holds that $(s_1, s_2) \notin E^{\mathfrak{M}}$, then the duplicator can clearly choose a node d_2 from the other model $\mathfrak{N}$ so that $(d_1, d_2) \notin E^{\mathfrak{N}}$. Thus the duplicator wins.

Ehrenfeucht-Fraïssé Game



2. If s_1 and s_2 are in different models and it holds that $(d_1, s_2) \in E^{\mathfrak{M}}$, then the duplicator can clearly choose a node d_2 from the other model $\mathfrak{N}$ so that $(s_1, d_2) \in E^{\mathfrak{N}}$. Thus the duplicator wins.
3. If s_1 and s_2 are in different models and it holds that $(d_1, s_2) \notin E^{\mathfrak{M}}$, then the duplicator can clearly choose a node d_2 from the other model $\mathfrak{N}$ so that $(s_1, d_2) \notin E^{\mathfrak{N}}$. Thus the duplicator wins.

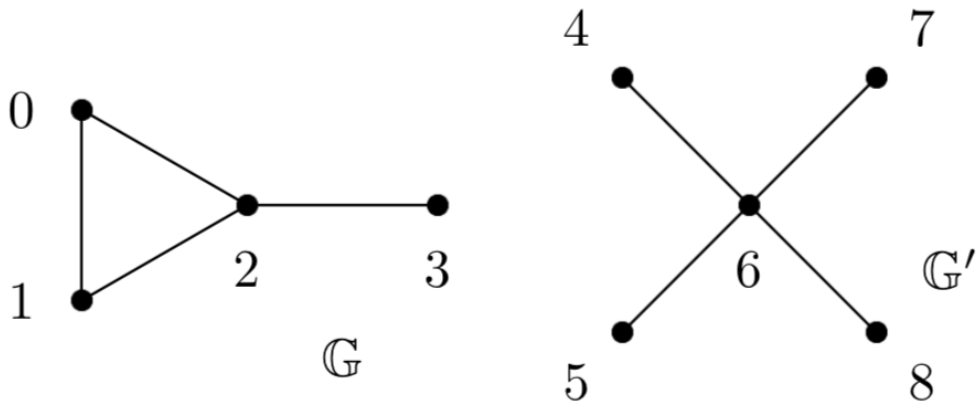
Ehrenfeucht-Fraïssé Game



We still need to discuss the case where the spoiler begins by choosing $s_1 \in \{2, 6\}$. Then the duplicator responds by choosing $d_1 = \{2, 6\} \setminus \{s_1\}$. The rest of the game is easy for the duplicator, as it is necessary that the spoiler chooses s_2 so that it links via E directly to the earlier move in that model.[1] It is then trivial for the duplicator to choose a likewise linking node from the other model.

[1] Here we are assuming the spoiler does not choose an already chosen node.

Ehrenfeucht-Fraïssé Game



We showed that the duplicator has a winning strategy in the game $\mathrm{EF}_2(\mathbb{G}, \mathbb{G}')$. To contrast this result, it is easy to show that the duplicator does not have a winning strategy in $\mathrm{EF}_3(\mathbb{G}, \mathbb{G}')$. This is easy, the spoiler simply first chooses 0, then 1 and finally 2. There is no cycle like this in $\mathbb{G}'$, so the duplicator cannot win, no matter what the response moves are.

Ehrenfeucht-Fraïssé Game

Definition 2.2

We write $\mathfrak{A} \cong_k \mathfrak{B}$ if the duplicator has a winning strategy in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$.

Ehrenfeucht-Fraïssé Game

We used stages of type

$$((\mathfrak{A}, (a_1, \ldots, a_i)), (\mathfrak{B}, (b_1, \ldots, b_i)))$$

for defining the Ehrenfeucht-Fraïssé game. We can alternatively define this stage as the triple

$$(\mathfrak{A}, \ \mathfrak{B}, \ \{(a_1, b_1), \ldots, (a_i, b_i)\})$$

where $\{(a_1, b_1), \ldots, (a_i, b_i)\}$ is now explicitly the partial isomorphism at the ith stage. This is simply an alternative representation of stages, and it is clear that it makes no difference which representation we use. This new representation is especially handy when we discuss *pebble games* below.

Note, the zeroeth stage in this new representation is of course $(\mathfrak{A}, \mathfrak{B}, \emptyset)$, where $\emptyset$ now represents the empty partial isomorphism.

Pebble Game

We will next discuss the pebble game. It is similar to the Ehrenfeucht-Fraïssé game but can be played for an infinite number of rounds. Rather than relating to first-order logic like the Ehrenfeucht-Fraïssé game, the pebble game characterizes some infinitary logics—as we shall show later on in the course.

Pebble Game

LIke the Ehrenfeucht-Fraïssé game, the pebble game is played by two players, again called the **spoiler** and **duplicator** (also **Abelard** and **Eloise**, or **player I** and **player II**).

Let τ be a purely relational vocabulary. Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-models, and let $k \in \mathbb{N}$. The k-**pebble game** $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ between $\mathfrak{A}$ and $\mathfrak{B}$ is played by the spoiler and duplicator for indefinitely many rounds, possibly infinitely long. A **play** of the k-pebble begins from the **zeroeth stage**

$$(\mathfrak{A}, \mathfrak{B}, \emptyset)$$

and proceeds exactly as the Ehrenfeucht-Fraïssé game up to **stage** k

$$(\mathfrak{A}, \mathfrak{B}, \{(a_1, b_1), \ldots, (a_k, b_k)\}).$$

Pebble Game

From the kth stage onwards, the game evolution is as follows:

Let $i \geq k$ and suppose the ith stage is

$$(\mathfrak{A}, \mathfrak{B}, \{(a'_1, b'_1), \ldots, (a'_k, b'_k)\}).$$

where $a'_1, \ldots, a'_k \in A$ and $b'_1, \ldots, b'_k \in B$. (Note indeed that we have exactly k nodes picked from both models.) Then the $(i+1)$st stage is determined such that

1. The spoiler erases one pair (a'_j, b'_j) from the partial isomorphism $\{(a'_1, b'_1), \ldots, (a'_k, b'_k)\}$ and picks either
   - and element $a \in A$ or
   - an element $b \in B$.
2. In the first case (the spoiler picking $a \in A$), the duplicator picks an element $b' \in B$ and the $(i+1)$st stage is $(\mathfrak{A}, \mathfrak{B}, p)$ where p is obtained from $\{(a'_1, b'_1), \ldots, (a'_k, b'_k)\}$ by replacing (a'_j, b'_j) by (a, b'). In the second case (spoiler picking $b \in B$), the duplicator picks some $a' \in A$ and the $(i+1)$st stage is $(\mathfrak{A}, \mathfrak{B}, q)$ where q is obtained from $\{(a'_1, b'_1), \ldots, (a'_k, b'_k)\}$ by replacing (a'_j, b'_j) by (a', b).

Pebble Game

So intuitively, the stage $(\mathfrak{A}, \mathfrak{B}, f_i)$ (with $i \geq k$) becomes replaced by $(\mathfrak{A}, \mathfrak{B}, f_{i+1})$ where f_{i+1} is obtained from f_i such that

1. the spoiler removes one pair from f_i
2. chooses an element from one model
3. the duplicator chooses a corresponding element from the other model
4. the resulting new pair is inserted into the partial isomorphism and thereby f_{i+1} is obtained.

Note that both f_i and f_{i+1} have precisely k pairs of nodes. The stages $0, \ldots, k-1$ involve smaller partial isomorphisms, but from stage k onwards, they will indeed always have k pairs. Note also that there can be up to $\mathbb{N}$ stages, so the play of the game can indeed last for infinitely many rounds.

Pebble Game

For each $j \in \mathbb{N}$, the **duplicator loses** (and spoiler wins) at the jth stage

$$(\mathfrak{A}, \mathfrak{B}, f_j)$$

if f_j is not a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. No further moves are made in this case, and the jth stage is called an **ending position** or a **final stage**.

Pebble Game

The possibly infinite sequence

0: $(\mathfrak{A}, \mathfrak{B}, \emptyset)$

1: $(\mathfrak{A}, \mathfrak{B}, \{(a_1, b_1)\})$

2: $(\mathfrak{A}, \mathfrak{B}, \{(a_1, b_1), (a_2, b_2)\})$

$\vdots$

i $\geq$ k: $(\mathfrak{A}, \mathfrak{B}, \{(a'_1, b'_1), \ldots (a'_k, b'_k)\})$

$\vdots$

is called a **play** of the game $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$. The **duplicator wins** the play if the game goes on for infinitely long and thereby the spoiler never succeeds to break the partial isomorphism.

Pebble Game

A **strategy of the spoiler** in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ is function that takes as in input an ith stage $(\mathfrak{A}, \mathfrak{B}, p)$ and outputs a tuple $(t, m, \mathfrak{M})$ specified as follows.

- If $i \geq k$, then the $t \in p$ is a tuple to be removed from the partial isomorphism p. If $i < k$, then $t = \emptyset$.
- $\mathfrak{M} \in \{\mathfrak{A}, \mathfrak{B}\}$ and m is an element of $\mathfrak{M}$. The element m is the new choice of the spoiler.

The reason we include $\mathfrak{M}$ in the output is that the domains of $\mathfrak{A}$ and $\mathfrak{B}$ could overlap, and then it would not necessarily be clear which model m belongs to.

Pebble Game

A **strategy of the duplicator** in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ is function that takes as in input a pair $((\mathfrak{A}, \mathfrak{B}, p), (t, m, \mathfrak{M}))$ where $(\mathfrak{A}, \mathfrak{B}, p)$ is an ith stage and $(t, m, \mathfrak{M})$ is a possible answer to that stage given by some strategy of the spoiler. Given such an input, the output of the strategy of the duplicator is an element m' belonging to the model $\mathfrak{M}' \in \{\mathfrak{A}, \mathfrak{B}\} \setminus \{\mathfrak{M}\}$.

This results in an $(i+1)$st stage $(\mathfrak{A}, \mathfrak{B}, q)$ where q is the partial isomorphism specified as follows.

1. If $i \leq k$, then
   - $q = p \cup \{(m, m')\}$ if $\mathfrak{M} = \mathfrak{A}$,
   - $q = p \cup \{(m', m)\}$ if $\mathfrak{M} = \mathfrak{B}$.
2. If $i > k$, then
   - $q = (p \setminus \{t\}) \cup \{(m, m')\}$ if $\mathfrak{M} = \mathfrak{A}$,
   - $q = (p \setminus \{t\}) \cup \{(m', m)\}$ if $\mathfrak{M} = \mathfrak{B}$.

Pebble Game

Definition 2.3

A strategy for spoiler (respectively, the duplicator) is a **winning strategy** in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ if the spoiler (respectively, duplicator) wins every play of the game $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ where the choices are made according to the strategy.

As for EF-games, we typically describe winning strategies informally (but nevertheless rigorously).

Pebble Games

Having defined the type of functions that define strategies in the pebble game, we can now state and prove the following theorem.

Theorem 2.4

Let $\mathfrak{A}$ and $\mathfrak{B}$ be finite models. Every strategy for the duplicator as well as spoiler in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ is finite.

Proof.

The claim is immediate, as due to $\mathfrak{A}$ and $\mathfrak{B}$ being finite, there are only finitely many possible inputs to any strategy. □

The significance of the theorem lies in the fact that pebble game plays can be infinite. Still, strategies are always finite on finite models.

Pebble Games

Theorem 2.5

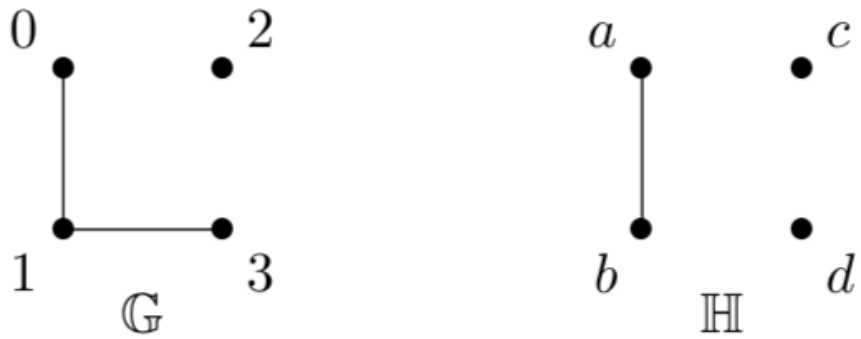
Let $\mathfrak{A}$ and $\mathfrak{B}$ be finite models. There exist finitely many strategies for the duplicator as well as for the spoiler in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$.

Proof.

As $\mathfrak{A}$ and $\mathfrak{B}$ are finite, and thereby there are at most finitely many inputs to any strategy, it is easy to see that there exist only finitely many strategies. □

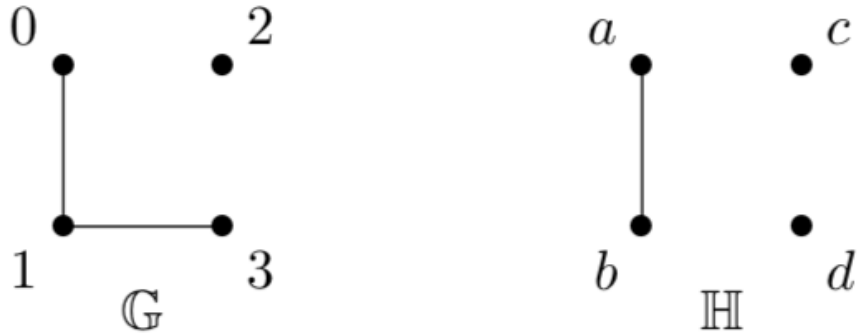
It is also worth noting that if we play $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ for enough many rounds, either the play will end with one player winning, or we shall end up repeating a stage.

Pebble Game



Consider the graphs $\mathbb{G}$ and $\mathbb{H}$. It is easy to show that the duplicator has a winning strategy in $\mathrm{EF}_2(\mathbb{G}, \mathbb{H})$. We will now show that, nevertheless, the duplicator does not have a winning strategy in $\mathrm{PG}_2(\mathbb{G}, \mathbb{H})$. In fact, we shall establish that the spoiler has a winning strategy in $\mathrm{PG}_2(\mathbb{G}, \mathbb{H})$.

Pebble Game



The spoiler first chooses the node 0 from $\mathbb{G}$. If the duplicator chooses c or d, then the spoiler will win in round two by choosing 1. Therefore we assume that the duplicator chooses a (the case for the choice b is the same due to symmetry, so we can ignore it). Thus the partial isomorphism after round one is $\{(0, a)\}$. Then, in round two, the spoiler chooses 3 and the duplicator d (the case for the choice c is the same due to symmetry). The partial isomorphism is $\{(0, a), (3, d)\}$. Now the spoiler erases $(0, a)$ and chooses 1. Now the duplicator does not have a suitable choice, because d does not have an edge anywhere. Thus the duplicator loses.

Constants in games

So far we considered the Ehrenfeucht-Fraïssé game and the pebble game over purely relational vocabularies. Adding constants to the vocabulary under consideration is easy. The idea is to think of constants as elements that have already been chosen in rounds before the play of the game even begins.

Constants in games

Let us first consider the Ehrenfeucht-Fraïssé game. Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-models over a relational vocabulary τ where $c_1, \ldots, c_r \in \tau$ are constant symbols. Recall that the initial stage (i.e., stage 0) in $\mathrm{EF}_k(\mathfrak{S}, \mathfrak{T})$ for models $\mathfrak{S}$ and $\mathfrak{T}$ over a purely relational vocabulary was

$$((\mathfrak{S}, \emptyset), (\mathfrak{T}, \emptyset))$$

and an ith position (i.e., ith stage) was of type

$$((\mathfrak{M}, (s_1, \ldots, s_i)), (\mathfrak{T}, (t_1, \ldots t_i))).$$

The stage 0 of $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ is

$$((\mathfrak{A}, (c_1^{\mathfrak{A}}, \ldots, c_r^{\mathfrak{A}})), (\mathfrak{B}, (c_1^{\mathfrak{B}}, \ldots, c_r^{\mathfrak{B}})))$$

and an i-th stage is of type

$$((\mathfrak{A}, (c_1^{\mathfrak{A}}, \ldots, c_r^{\mathfrak{A}}, a_1, \ldots, a_i)), (\mathfrak{B}, (c_1^{\mathfrak{B}}, \ldots, c_r^{\mathfrak{B}}, b_1, \ldots, b_i))).$$

Constants in games

The final kth stage is of type

$$((\mathfrak{A}, (c_1^{\mathfrak{A}}, \ldots, c_r^{\mathfrak{A}}, a_1, \ldots, a_k)), (\mathfrak{B}, (c_1^{\mathfrak{B}}, \ldots, c_r^{\mathfrak{B}}, b_1, \ldots, b_k))).$$

Altogether, a play of the game $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ is the same as for purely relational vocabularies, but the initial zeroeth stage partial isomorphism is $\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}$ while with purely relational vocabularies, the zeroeth stage partial isomorphism is the empty set. The ith stage partial isomorphims in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ is then of type

$$\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}}), (a_1, b_1), \ldots, (a_i, b_i)\}$$

rather than of type $\{(s_1, t_1), \ldots, (s_i, t_i)\}$ as in the game $\mathrm{EF}_k(\mathfrak{S}, \mathfrak{T})$ with models of a purely relational vocabulary.

Constants in games

The moves and winning conditions in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ are exactly the same as in the game without constant symbols, but now taking into account the full partial isomorphism $\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}}), (a_1, b_1), \ldots, (a_i, b_i)\}$ at each stage i.

Note that this implies that it is possible to win $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$ already at the zeroeth stage. Indeed, if $\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}$ is not a partial isomorphism, then the duplicator wins at the zeroeth stage, and the game play ends there.

Constants in games

We also used the representation $(\mathfrak{S}, \mathfrak{T}, p)$, where p is a partial isomorphism from $\mathfrak{S}$ to $\mathfrak{T}$ to represent games. Using constants in this framework is done in a relatively simple way. Letting $\mathfrak{A}$ and $\mathfrak{B}$ be as above, the zeroeth stage of the Ehrenfeucht-Fraïsse game $\mathrm{EK}_k(\mathfrak{A}, \mathfrak{B})$ is

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \emptyset),$$

and an ith stage is of type

$$(\ \mathfrak{A},\ \mathfrak{B},\ \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\},\ \{(a_1, b_1), \ldots, (a_i, b_i)\}\ ).$$

The task of the duplicator is to maintain the partial isomorphism

$$\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\} \cup \{(a_1, b_1), \ldots, (a_i, b_i)\}.$$

Constants in games

The only reason we are not writing a function

$$\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}}), (a_1, b_1), \ldots, (a_i, b_i)\}$$

to the technical specification of the ith stage

$$(\ \mathfrak{A},\ \mathfrak{B},\ \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\},\ \{(a_1, b_1), \ldots, (a_i, b_i)\}\ ),$$

but instead write two functions

$$\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\} \text{ and } \{(a_1, b_1), \ldots, (a_i, b_i)\},$$

relates to the pebble game. Indeed, as we shall see, the spoiler should not modify (remove tuples from) $\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}$ in the pebble game. Tuples should only be removed from $\{(a_1, b_1), \ldots, (a_i, b_i)\}$.

Constants in games

We begin the pebble game $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{B})$ from

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \emptyset)$$

and an ith stage for $i \leq k$ looks like

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \{(a_1, b_1), \ldots, (a_i, b_i)\}).$$

The ith stage for $i > k$ looks like

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \{(a'_1, b'_1), \ldots, (a'_k, b'_k)\}).$$

The rules of the game are as in the case for purely relational vocabularies: The players first construct the partial isomorphism $\{(a_1, b_1), \ldots, (a_i, b_i)\}$ in stages $i \leq k$ and then keep altering $\{(a'_1, b'_1), \ldots, (a'_i, b'_i)\}$ in the subsequent stages as before—the spoiler removing a tuple, introducing and element, and the duplicator responding by an element from the other model.

Constants in games

However, the task of the duplicator is to maintain the partial isomorphism

$$\{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}), \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\} \cup \{(a_1'', b_1''), \ldots, (a_j'', b_j'')\})$$

at every stage

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}) \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \{(a_1'', b_1''), \ldots, (a_j'', b_j'')\})$$

of the play of the game. Therefore it is—just like in EF-games with constants—possible for the duplicator to lose already at the zeroeth stage.

As already discussed above, we mention once more that the players only modify the second one of the two isomorphisms of a stage

$$(\mathfrak{A}, \mathfrak{B}, \{(c_1^{\mathfrak{A}}, c_1^{\mathfrak{B}}) \ldots, (c_r^{\mathfrak{A}}, c_r^{\mathfrak{B}})\}, \{(a_1'', b_1''), \ldots, (a_j'', b_j'')\}).$$

Constants in games

The following theorem is trivial.

Theorem 2.6

Let $\mathfrak{A}$, $\mathfrak{B}$ and $\mathfrak{C}$ be τ-models for a relational vocabulary τ. Suppose $\mathfrak{A}$ and $\mathfrak{B}$ are isomorphic.

1. *If the duplicator (respectively, spoiler) has a winning strategy in $\mathrm{EF}_k(\mathfrak{B}, \mathfrak{C})$, then the duplicator (respectively, spoiler) has a winning strategy in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{C})$.*
2. *If the duplicator (respectively, spoiler) has a winning strategy in $\mathrm{PG}_k(\mathfrak{B}, \mathfrak{C})$, then the duplicator (respectively, spoiler) has a winning strategy in $\mathrm{PG}_k(\mathfrak{A}, \mathfrak{C})$.*

As for Ehrenfeucht-Fraïssé games with purely relational vocabularies, also for models with additional constant symbols, we write $\mathfrak{A} \cong_k \mathfrak{B}$ to indicate that the duplicator has a winning strategy in $\mathrm{EF}_k(\mathfrak{A}, \mathfrak{B})$.

Further examples on playing games

Consider two models $\mathfrak{A}$ and $\mathfrak{B}$ over the vocabulary $\{R\}$, where R is a binary relation symbol. Suppose that $R^{\mathfrak{A}}$ is transitive and $R^{\mathfrak{B}}$ is not. Does it hold that $\mathfrak{A} \cong_4 \mathfrak{B}$, i.e., that the duplicator has a winning strategy in $\mathrm{EF}_4(\mathfrak{A}, \mathfrak{B})$? Justify your answer.

Further examples on playing games

We shall disprove the claim by showing that the spoiler has a winning strategy already in $\mathrm{EF}_3(\mathfrak{A}, \mathfrak{B})$, i.e., the spoiler can always win with three moves.

Now, $R^{\mathfrak{B}}$ is not transitive, so there exist points $b_1, b_2, b_3 \in B$ such that $(b_1, b_2) \in R^{\mathfrak{B}}$ and $(b_2, b_3) \in R^{\mathfrak{B}}$ while $(b_1, b_3) \notin R^{\mathfrak{B}}$. The spoiler chooses these points in rounds 1, 2 and 3. No matter how the duplicator responds in $\mathfrak{A}$, the spoiler's pattern in $\mathfrak{B}$ cannot be matched because $\mathfrak{A}$ is transitive.

Further examples on playing games

The next example is more difficult. We skecth the proof of the below theorem (with enough details for you to easily fill in the gaps).

Theorem 2.7

Let $\mathfrak{S} = (S, <^{\mathfrak{M}})$ and $\mathfrak{T} = (T, <^{\mathfrak{N}})$ be $\{<\}$-models interpreting $<$ as strict linear order. Let $k \in \mathbb{Z}_+$. If $|S| > 2^k$ and $|T| > 2^k$, then $\mathfrak{S} \cong_k \mathfrak{T}$.

Proof Sketch. We assume, without loss of generality, that $S = \{1, \ldots, |S|\}$ and $T = \{t_1, \ldots, |T|\}$. Consider an rth stage $(\mathfrak{S}, (s_1, \ldots s_r), \mathfrak{T}, (t_1, \ldots, t_r))$ where $r \leq k$. Suppose the elements here are listed in such an order that $s_i \leq s_{i+1}$ and $t_i \leq t_{i+1}$ for all i (so the subindex does not here have to mean the round when the element was picked; this assumption is made simply to simplify the notation below). Continues...

Further examples on playing games

Suppose that one of the following conditions holds for each i:

1. $s_{i+1} - s_i \geq 2^{k-r}$ and $t_{i+1} - t_i \geq 2^{k-r}$.
2. $s_{i+1} - s_i = t_{i+1} - t_i$.

Suppose also that one of the following holds

1. $s_1 - 1 \geq 2^{k-r}$ and $t_1 - 1 \geq 2^{k-r}$.
2. $s_1 - 1 = t_1 - 1$.

and that one of the following holds

1. $|S| - s_r \geq 2^{k-r}$ and $|T| - t_r \geq 2^{k-r}$.
2. $|S| - s_r = |T| - t_r$.

Intuitively, the choices have been made such that each 'neigbouring' pair (s_i, s_{i+1}) of pebbles either has the pebbles equally far from each other as the pebbles (t_i, t_{i+1}), or alternatively, both pairs (s_i, s_{i+1}) and (t_i, t_{i+1}) have the pebbles so far from each other (at least 2^{k-r} steps) that the actual distance does not even matter. A similar condition holds for the (first and last) pebbles near the endpoints of the linear orders.

Further examples on playing games

We will next show that if the spoiler chooses w from between s_i and s_{i+1}, then the duplicator can choose w' from between t_i and t_{i+1} so that the conditions on the previous slide hold for the updated tuples

$$(s_1, \ldots, s_i, w, s_{i+1}, \ldots, s_r) \text{ and } (t_1, \ldots, t_i, w', t_{i+1}, \ldots, t_r)$$

and for the updated (decreased) distance $2^{k-(r+1)}$.

Assume first that $s_i - s_{i+1} \geq 2^{k-r}$. We have three cases.

1. Suppose $w - s_i \geq 2^{k-(r+1)}$ and $s_{i+1} - w \geq 2^{k-(r+1)}$. Now, since $t_{i+1} - t_i \geq 2^{k+r}$ and $2 \cdot 2^{k-(r+1)} = 2^{k-r}$, the duplicator can choose w' from between these points so that we have $w' - t_i \geq 2^{k-(r+1)}$ and $t_{i+1} - w' \geq 2^{k-(r+1)}$.
2. Suppose $w - s_i < 2^{k-(r+1)}$. Now $s_{i+1} - w \geq 2^{k-(r+1)}$ and the duplicator can choose w' so that $w' - t_i = w - s_i$ and $t_{i+1} - w' \geq 2^{k-(r+1)}$.
3. The case $s_{i+1} - w < 2^{k-(r+1)}$ is similar to the above case.

Suppose then that $s_{i+1} - s_i < 2^{k-r}$. Then $t_{i+1} - t_i = s_{i+1} - s_i$, so the interval from s_i to s_{i+1} is isomorphic to the interval from t_i to t_{i+1}, and therefore the duplicator chooses w' so that it is as close to t_i and t_{i+1} as w is to s_i and s_{i+1}.

Further examples on playing games

We discussed the case where the spoiler chose from $\mathfrak{S}$. The case where the spoiler chooses from $\mathfrak{T}$ is of course symmetric to this, so we do not need to discuss that case separately. Also, we discussed the case where the spoiler chooses from between two already chosen nodes. The cases where the spoiler chooses an element smaller than s_1 or greater than s_r are also similar to the case we discussed. Finally, the cases where the spoiler chooses an already chosen node can be ignored, as we have observed earlier. Therefore we have shown that the duplicator can maintain the partial isomorphism (the respective ordering of all chosen nodes) all the way up to $r = k$ moves. This concludes the proof of the theorem.□