The Ehrenfeucht Game and First Order Logic

Asa Levi

August 13, 2006

1 Introduction

This paper is written at an introductory level. The reader should be familiar with some graph theory, but does not need to know anything about first order logic. The goal of the paper is to prove that graphs with the same k-Ehrenfeucht value also satisfy exactly the same first order sentences with quantifier depth k. This paper was written with two major references, David Marker's introduction to model theory[1] and Joel Spencer's book on random graphs[2]. If the reader is unfamilar with first order logic an introduction is provided, but it is hardly exhaustive, especially when discussing semantic implication.

1.1 Languages

First order logic is certainly similar to most readers. The three main types of logic are propositional logic, first order logic and second order logic. All of them share some common elements.

The definition of a language exists in all three forms.

Definition 1. *A language $\mathcal{L}$ is a family of sets of symbols. These symbols fall into three categories:*

1. *Function Symbols, $\mathcal{F}$: A function f with an associated $n_f \in \mathbb{N}$, where f is a function with arity n_f, for each function in the language.*
2. *Relation Symbols, $\mathcal{R}$: A relation R with an associated $n_R \in \mathbb{N}$, where R is a relation with arity n_R, for each relation in the language.*
3. *Constant Symbols, $\mathcal{C}$: A symbol, c_i, for each constant in the language.*

A language can be abbreviated as $\mathcal{L} = (\mathcal{F}, \mathcal{R}, \mathcal{C}) = (\{(f_1, n_1), ...\}, \{(R_1, n_1), ...\}, \{c_1, c_2, ...\})$. This is also sometimes written $\mathcal{L} = (f, R, c)$ when the arities are clear.

Whenever you want to talk about something in mathematical logic you have to begin by defining the language that you are working in.

Example 1 (Group Theory). In group theory you have the identity element, 0, and a binary operation, +. These are the only two symbols necessary in group theory, so the language of group theory is $\mathcal{L}_{Group}$ = (+,=,0).

Example 2 (Ring Theory). In ring theory we need the same symbols as group theory, but also two additional ones for the second binary operation, $\cdot$, and the identity for that operation, 1. So the language of ring theory is $\mathcal{L}_{Ring}(\{+,\cdot\},=,\{0,1\})$.

One last example which will be important in this paper is the case of graph theory.

Example 3 (Graph Theory). The two things that are universally required in graph theory are the ability to tell whether two verticies are neighbors or not and the ability to tell whether two verticies are the same or not. Thus the language of graph theory is $\mathcal{L}_{Graph} = (\emptyset, \{R, =\}, \emptyset)$.

An important thing to realize about all of these languages is that at this point they have no meaning. The meaning that we are associating with these language is only one of many possible meanings. For instance, in $\mathcal{L}_{Group}$ we simply have a binary operation symbol, binary relation symbol and a constant symbol, there is nothing forcing this constant symbol to be the identity for the binary operation. To do this you have to interpret the language in an $\mathcal{L}$-*structure*:

Definition 2. *An $\mathcal{L}$-structure, $\mathcal{M}$, is an interpretation of the language given by the following properties:*

1. *A universe, which is a set M.*
2. *A function $f^{\mathcal{M}} : M^{n_f} \longrightarrow M \quad \forall f \in \mathcal{F}$*
3. *A set $R^{\mathcal{M}} \subseteq M^{n_R} \quad \forall R \in \mathcal{R}$*
4. *An element $c^{\mathcal{M}} \quad \forall c \in \mathcal{C}$*

The $\mathcal{L}$-structure is usually written $\mathcal{M} = (M, f^{\mathcal{M}}, R^{\mathcal{M}}, c^{\mathcal{M}})$. Also from now on I will leave out the empty sets in my notation unless they provide clarity.

We can give the meanings that we intended for the languages to have by simply interpreting the symbols as what we would expect them to mean. For instance you could define an $\mathcal{L}_{Group}$-*structure* by $(\mathbb{Z}, +, 0)$ where $+$ is the conventional addition. Note also though that $(\mathbb{N}, +, 0)$ is an $\mathcal{L}_{Group}$-*structure*, but is not a group.

Next we can start saying things with our language. First it is necessary to introduce two more things; the logical connectives are $\wedge$ (and), $\vee$ (or) and $\neg$ (not), each has its regular logical meaning. These are all of the logical operators in propositional logic, but in first order logic there are also $\exists$ and $\forall$. The relational symbol $=$ is always included in our language and has its regular meaning. In addition to constants, there is also an infinite set of variables $v_1, v_2, \ldots$ (or any letter other than c.) These variables have no fixed value, unlike the constants, and represent any member of the universe in an $\mathcal{L}$-*structure*. The first class of things that we can say are the terms.

Definition 3. *The $\mathcal{L}$-terms are the smallest set, $\mathcal{T}$, satisfying these properties:*

1. $c \in \mathcal{T}$ $\forall c \in \mathcal{C}$

2. $v_i \in \mathcal{T}$ $\forall i \in \mathbb{N}$

3. $f(t_1, t_2, \ldots, t_{n_f}) \in \mathcal{T}$ $\forall f \in \mathcal{F}$

So for example, $v_1 = v_2$ is a term in propositional and first order logic, while $\forall v_1(\exists v_2(v_1 = v_2))$ is a term in first order logic, but not propositional. Note also the parenthesis enclosing what the quantifier, in this case $\forall$, is quantifying over. Now we move on to the next most elementary thing that we can say:

Definition 4. *The atomic formulas,* $\mathrm{Atom}(\mathcal{L})$ *are the smallest set satisfying the following properties:*

1. $t \in \mathrm{Atom}(\mathcal{L}) \forall t \in \mathcal{L}$*-terms*

2. $t_1 = t_2 \in \mathrm{Atom}(\mathcal{L})$ $\forall t_1, t_2 \in \mathcal{T}$

3. $R(t_1, t_2, \ldots, t_{n_R}) \in \mathrm{Atom}(\mathcal{L})$ $\forall t_1, \ldots, t_{n_R} \in \mathcal{T}$ $\forall R \in \mathcal{R}$

Finally we get to the most complex things that can be said in first order logic, the formulas. These will include all of the connectives and symbols that have been introduced so far.

Definition 5. *The formulas,* $\mathrm{Form}(\mathcal{L})$ *are the smallest set satisfying the following conditions:*

1. $\varphi \in \mathrm{Form}(\mathcal{L})$ $\forall \varphi \in \mathrm{Atom}(\mathcal{L})$

2. $\phi_1 \wedge \phi_2 \in \mathrm{Form}(\mathcal{L})$ $\forall \phi_1, \phi_2 \in \mathrm{Form}(\mathcal{L})$

3. $\phi_1 \vee \phi_2 \in \mathrm{Form}(\mathcal{L})$ $\forall \phi_1, \phi_2 \in \mathrm{Form}(\mathcal{L})$

4. $\neg\phi \in \mathrm{Form}(\mathcal{L})$ $\forall \phi \in \mathrm{Form}(\mathcal{L})$

5. $\forall v_i(\phi) \in \mathrm{Form}(\mathcal{L})$ $\forall \phi \in \mathrm{Form}(\mathcal{L})$

6. $\exists v_i(\phi) \in \mathrm{Form}(\mathcal{L})$ $\forall \phi \in \mathrm{Form}(\mathcal{L})$

An important thing to note about all of these definitions is that the sets defined are inductive sets. Because of this the most common method of proof on these sets is induction. And although I won't be doing any inductive proofs on the formulas here, they are an extremely important part of logic. Also it an extremely important fact that all formulas are of finite length (where the length is the number of symbols in the formula.)

Now we have one further distinction to make in the formulas that we have. Here are two examples of formulas in $\mathrm{Form}(\mathcal{L}_{Group})$: $\phi_1 : \forall v_1(\exists v_2(v_1 + v_2 = 0))$ $\phi_2 : v_1 + v_2 = 0$ Now, these might appear to be the same, but they might have different truth values in an $\mathcal{L}_{Group}$*-structure*. This is because in ϕ_1 we know what the possible values for v_1 and v_2 are, they are defined by the quantifiers, while in ϕ_2 nothing is known about them. When nothing is known about them any values are possible. So if we look at a particular $\mathcal{L}_{Group}$-*structure* like $(\mathbb{Z}, +, 0)$ ϕ_1 is always true, while ϕ_2 depends on how v_1 and v_2 are treated. In ϕ_1 v_1 and v_2 are called bound variables, while in ϕ_2 they are free. I will write formulas with free variables as $\phi_2(v_1, v_2)$ indicating that the two free variables in the formula are v_1 and v_2.

Definition 6. *Formulas that have no free variables are called sentences. I will use σ to represent a sentence.*

Definition 7. *A theory is a set of sentences.*

Now we can make some progress. Firstly we say $\mathcal{M} \models \phi$ if ϕ is always true in $\mathcal{M}$. In the previous example if we take ϕ_2(1,2) then $1 + 2 \neq 0$ so $\mathcal{M} \not\models \phi_2$. Also if $\Sigma = \{\sigma_1, \ldots, \sigma_n \mid \sigma_i \in \text{Sent}(\mathcal{L})\}$ we say $\mathcal{M} \models \Sigma$ if $\mathcal{M} \models \sigma_i \ \forall \sigma_i \in \Sigma$.

Finally we can define the $\mathcal{L}_{Group}$-*structures* that are actually groups. First we definte a theory Σ_{Group}, let:

$$\sigma_1 = \forall v_i(v_i + 0 = v_i)$$

$$\sigma_2 = \forall v_i(\exists v_j(v_i + v_j = 0))$$

$$\sigma_3 = \forall v_i(\forall v_j(\forall v_k((v_i + v_j) + v_k = v_i + (v_j + v_k))))$$

Now let $\Sigma_{Group} = \{\sigma_1, \sigma_2, \sigma_3\}$. We have finally that for any $\mathcal{L}_{Group}$-*structure*, $\mathcal{M}$, $\mathcal{M} \models \Sigma_{Group}$ only when $\mathcal{M}$ is a group. The previous example of $\mathcal{N} = (\mathbb{N}, +, 0)$ fails here because $\mathcal{N} \not\models \sigma_2$.

Note also that the theory is much less confining for graphs, as they have much less structure. We only need $\mathcal{M} \models \forall v_i(\neg(R(v_i, v_i)))$ for some $\mathcal{L}_{Graph}$-*structure*, $\mathcal{M}$.

One final definition is necessary for our main theorem:

Definition 8. *We define the quantifier depth of a formula,* depth(ϕ)*, recursively:*

1. $\text{depth}(\phi) = 0$ *if* $\phi \in \text{Atom}(\mathcal{L})$
2. $\text{depth}(\phi) = \max(\text{depth}(\xi), \text{depth}(\psi))$ *if* $\phi = \xi \wedge \psi$
3. $\text{depth}(\phi) = \max(\text{depth}(\xi), \text{depth}(\psi))$ *if* $\phi = \xi \vee \psi$
4. $\text{depth}(\phi) = \text{depth}(\psi)$ *if* $\phi = \neg\psi$
5. $\text{depth}(\phi) = 1 + \text{depth}(\psi)$ *if* $\phi = \forall\psi$
6. $\text{depth}(\phi) = 1 + \text{depth}(\psi)$ *if* $\phi = \exists\psi$

Example 4. We will calculate depth(ϕ) where $\phi = \forall x_1(\exists x_2(\neg(x_1 = x_2) \wedge \forall x_3((x_1 = x_3) \vee (x_2 = x_3))))$

$$\begin{aligned}
\text{depth}(\phi) &= \text{depth}(\forall x_1(\exists x_2(\neg(x_1 = x_2) \wedge \forall x_3((x_1 = x_3) \vee (x_2 = x_3))))) \\
&= 1 + \text{depth}(\exists x_2(\neg(x_1 = x_2) \wedge \forall x_3((x_1 = x_3) \vee (x_2 = x_3)))) \\
&= 1 + 1 + \text{depth}(\neg(x_1 = x_2) \wedge \forall x_3((x_1 = x_3) \vee (x_2 = x_3))) \\
&= 2 + \max(\text{depth}(\neg(x_1 = x_2)), \text{depth}(\forall x_3((x_1 = x_3) \vee (x_2 = x_3)))) \\
&= 2 + \max(\text{depth}(x_1 = x_2), 1 + \text{depth}((x_1 = x_3) \vee (x_2 = x_3))) \\
&= 2 + \max(0, 1 + \max(\text{depth}(x_1 = x_3), \text{depth}(x_2 = x_3))) \\
&= 2 + \max(0, 1 + \max(0, 0)) \\
&= 2 + \max(0, 1 + 0) \\
&= 2 + 1 \\
&= 3
\end{aligned}$$

Exercise 1. Prove that any formula, ϕ, with quantifier depth $k \geq 1$, $\exists \psi \in$ Form($\mathcal{L}$) with quantifier depth $k - 1$ and one free variable, x, such that $\emptyset \models (\phi \Leftrightarrow (\exists x \psi))$.

2 The Ehrenfeucht Game

At first I will be considering a special case of the Ehrenfeucht Game, when it is played on graphs. It can also be defined in a more general sense on models, and later, in my main proofs I will be dealing with this. Throughout the section I will be using the symbol $\sim$ as the adjacency relation in a graph.

2.1 Definition and Preliminaries

First we have to define what the Ehrenfeucht Game is. There are two players, one is called the Spoiler and the other is called the Duplicator. To make things clearer we will let the Spoiler be male and the Duplicator be female. There are also two graphs, for now we will call them G_1 and G_2. These graphs have disjoint verticies. Before the game is played it is known how many turns the game will last, let us say k. This game is then called Ehr(G_1, G_2, k).

Now the game is ready to be played. On the first move the Spoiler chooses a vertex from either graph and marks it with a 1. The Duplicator then chooses a vertex in the other graph and also marks it with a 1. On the i^{th} move the Spoiler chooses either graph and marks a vertex i, and the Duplicator responds by marking a vertex on the other graph i. For our purposes we will suppose that the same vertex is not marked by the Spoiler twice because this is a waste of a turn for him. After k turns the game stops. Now, who wins this game? Let us call all of the marked verticies on G_1 $x_1, x_2, \ldots, x_k$ and all of the marked verticies on G_2 $y_1, y_2, \ldots, y_k$. Duplicator wins if $x_i \sim x_j \Leftrightarrow y_i \sim y_j$. Also the equality relation must be preserved: $x_i = x_j \Leftrightarrow y_i = y_j$. We will say that the Duplicator wins the game Ehr(G_1, G_2, k) if there is a strategy for her that wins.

This means that there for any sequence of moves of the Spoiler the Duplicator has a response that leads to a win. Similarly we will say that the Spoiler wins if there is no such sequence of moves.

Exercise 2. Suppose the Duplicator is winning the game $\mathrm{Ehr}(G_1, G_2, k)$ after the i^{th} turn. Show that if the Spoiler chooses a vertex that was previously chosen the Duplicator has a response where she will still be winning.

Example 5 (Complete Graphs). Let K_i, K_j be the complete graphs on i and j verticies respectively. Under which conditions will the Duplicator win $\mathrm{Ehr}(K_i, K_j, k)$? First suppose $i \neq j$ and $k > \min(i, j)$. Without loss of generality suppose $i > j$. Now all the spoiler has to do is mark a vertex on the first turn and then continue marking verticies until the $j + 1^{st}$ turn. The Duplicator will have to respond with some new vertex, but she has none left, since she has already chosen j distinct verticies. Now she will have to choose some new vertex, but she has none remaining. So her choice will overlap with some vertex, lets say $y_i = y_{j+1}$. Now we have $x_i \neq x_{j+1}$ but $y_i = y_{j+1}$. So the Spoiler wins.

Now suppose $k \leq \min(i, j)$. Here it is easy to see that the Duplicator will win. The Spoiler must choose a new vertex every turn, and the Duplicator simply does the same thing. It does not matter if the Spoiler switches graphs, the Duplicator will still just choose an unmarked vertex on the other graph. Since both graphs have more verticies than the number of turns, the Duplicator will win. It is clear that this is a win for the Duplicator, because every x_i will be adjacent to every x_j as long as $i \neq j$ (since they are playing on complete graphs) and the same thing holds on G_2.

In the above example we used the inability of the Duplicator to maintain equality, but we also could have used the fact that $x_i \sim x_{j+1}$ but $y_i \nsim y_{j+1}$.

Exercise 3. Find an example where the Spoiler must use the inability of the Duplicator to maintain equality to win.

Also in the above example the Spoiler can win in another way, he could have started on the smaller graph and then switched after running out of vertices. This would have produced the same result here, but there are many cases when switching graphs is the easiest way to produce a victory for the Spoiler.

Example 6. Suppose we have two graphs, G_1 and G_2. Let G_1 have at least one isolated point and G_2 have none. Then the Spoiler wins $\mathrm{Ehr}(G_1, G_2, k)$ $\forall k \geq 2$. The Spoiler chooses the isolated point in G_1 on his first turn, the Duplicator responds with any vertex in G_2. Since G_2 has no isolated points y_1 must have a neighbor, the Spoiler chooses that neighbor. Now the Duplicator cannot respond with a neighbor of x_1 since it is an isolated point. Thus the Spoiler wins.

One other important point is that if G_1 and G_2 are isomorphic then the Duplicator will always win $\mathrm{Ehr}(G_1, G_2, k)$ simply by following the action of the isomorphism on the point that the Spoiler marks.

One more thing to consider is the subgraphs of a given graph:

Exercise 4. Consider $\mathrm{Ehr}(G_1, G_2, k)$ where H is a subgraph of G_1 on less than k verticies that does not appear in G_2. Show that the Spoiler wins this game.

Finally we will consider the seemingly simple case of paths. We consider the game $\text{Ehr}(P_n, P_m, k)$, where P_r is the path of length r. First, it is clear that if $m = n$ the duplicator will win this game for all k, since the graphs will be isomorphic.

Theorem 1. *If $n \leq 2^k + 1$ and $n < m$ then Spoiler wins* $\text{Ehr}(P_n, P_m, k+2)$.

Proof. We will document a winning strategy for the Spoiler. He will make all of his moves in P_n. The first thing to note is that if the Spoiler marks an endpoint the Duplicator must also mark an endpoint. This is clear, because the endpoints are the only verticies with only one neighbor. Thus if the Spoiler marks an endpoint and the Duplicator responds with a non-endpoint then the Spoiler just needs to mark both of the neighbors of the point that the Duplicator marked and he will win.

On his first two moves the Spoiler chooses the two endpoints of P_n. The Duplicator must respond with the endpoints of P_m by our previous observation. Now the Spoiler chooses the midpoint of P_n. Note the distances between the points on P_n: $|x_1 - x_3| \leq 2^k, |x_3 - x_2| \leq 2^k$. For any point that the Duplicator marks either $|y_1 - y_3| > 2^k$ or $|y_3 - y_2| > 2^k$.

Now we begin our induction step: Suppose that after the $s^t h$ move $\exists i$ such that $|x_i - x_{k+2-s}| \leq 2^s$ and $|y_i - y_s| > 2^s$. We want to show that Spoiler can mark a point on the $s - 1^{st}$ turn such that the induction hypothesis also holds substuting in $s - 1$ for s.

This follows the same logic as before. Just choose $x_{k+2-(s-1)}$ as the midpoint between x_i and x_{k+2-s}. Note that since $|x_i - x_{k+2-s}| \leq 2^s$ and we chose a point half way between these two points we now have $|x_i - x_{k+2-(s-1)}| \leq 2^{s-1}$ and $|x_{k+2-(s-1)} - x_{k+2-s}| \leq 2^{s-1}$. Now we have what we wanted on the Spoiler's side, but we need to show that the Duplicator cannot mess things up.

First suppose the Duplicator chooses a point outside of the interval (y_i, y_s). Then either $|y_{s-1} - y_s| > |y_i - y_s| > 2^s > 2^{s-1}$ or $|y_{s-1} - y_s| > |y_i - y_s| > 2^s > 2^{s-1}$, which is exactly what we wanted.

Now suppose the Duplicator marks a point inside the interval. Then, as in the case where there were k turns remaining, we know that

$$\begin{aligned} 2^s &< |x_i - x_{k+2-s}| \\ &= |x_i - x_{k+2-(s-1)} + x_{k+2-(s-1)} - x_{k+2-s}| \\ &\leq |x_i - x_{k+2-(s-1)}| + |x_{k+2-(s-1)} - x_{k+2-s}| \text{ (by the Triangle Inequality)} \end{aligned}$$

Thus either $|x_i - x_{k+2-(s-1)}| > 2^{s-1}$ or $|x_{k+2-(s-1)} - x_{k+2-s}| > 2^{s-1}$, which again is what we wanted.

So the induction hypothesis holds for all s. When $s = 0$ we have $\exists i$ such that $|x_i - x_{k+2}| \leq 2^0 = 1$ and $|y_i - y_{k+2}| > 1$. Thus the Spoiler wins, because $x_i \sim x_{k+2}$ and $y_i \nsim y_{k+2}$. □

Now we show that the Duplicator wins in another case:

Theorem 2. *If $m, n > 2^{k+1} + 1$ then the Duplicator wins* $\text{Ehr}(P_m, P_n, k+2)$.

Proof. In this proof the Duplicator will be using an Inside-Outside strategy. Basically the Duplicator is only concerned about moves that the Spoiler makes that are close enough for the Spoiler to exploit. If they are not close enough the Duplicator just also makes a move that is not close enough on the other graph. The complicated part of this proof is figuring out exactly what close enough means in this case. The proof proceeds as follows.

Call a position equivalent if there are i turns remaining and $\forall j, k \leq i$ either $|x_j - x_k| = |y_j - y_k|$ or $|x_j - x_k|, |y_j - y_k| > 2^i$. Here the two cases are the inside case and the outside case respectively. The notion of closeness that we are using is that two points are close with s moves remaining if they are 2^{k+2-s} or less apart. This distance is chosen because it is the largest distance that the Spoiler can exploit with his strategy in the last turn. Thus if they are inside this distance the Duplicator must match the distance between the points exactly.

On the first two moves we will assume that the Spoiler marks the endpoints, because these two moves are necessarily responded to by the endpoints of the Duplicator's graph, and if he never marks them then it will not change the rest of the strategy, you can just consider the game without the endpoints marked starting after the second move.

Suppose that with $s + 1$ moves remaining the Duplicator has maintained equivalent positions. Now consider the $k + 2 - s^{th}$ move of the Spoiler. There are only two possibilities, that the move is Inside or Outside.

First suppose that the Spoiler moves inside. That means that $\exists i$ such that $|x_i - x_{k+2-s}| \leq 2^{s-1}$. So now the Duplicator moves in the same way and marks y_{k+2-s} such that it is on the same side of the closest endpoint relative to y_i and also $|y_i - y_{k+2-s}| = |x_i - x_{k+2-s}|$. Now suppose $\exists j \neq i$ such that x_j and x_{k+2-s} are close (i.e. $|x_j - x_{k+2-s}| \leq 2^{s-1}|$.) Then we have that $|x_j - x_{k+2-s}| \leq 2^{s-1}|$ and $|x_i - x_{k+2-s}| \leq 2^{s-1}$ so

$$\begin{aligned}
2^s &= 2(2^{s-1}) \\
&= 2^{s-1} + 2^{s-1} \\
&\geq |x_i - x_{k+2-s}| + |x_{k+2-s} - x_j| \\
&\geq |x_i - x_{k+2-s} + x_{k+2-s} - x_j| \\
&= |x_i - x_j|
\end{aligned}$$

So we know that x_i and x_j were close in the last stage. Thus $|x_i - x_j| = |y_i - y_j|$ and we marked y_{k+2-s} so that $|y_i - y_{k+2-s}| = |x_i - x_{k+2-s}|$. Now putting these together we have $|y_j - y_{k+2-s}| = |x_j - x_{k+2-s}|$ and thus the positions are still equivalent. Note also here that the imporance of maintaining distance like this is that at the last step where we will have that two points are close if they are 2^0 apart, i.e. if they are neighbors. Thus if we are also able to maintain equivalence on points that the Spoiler marks that are Outside we will have shown that two points if two points are neighbors on one graph then they are neighbors on the other graph. Also note that the strategy does not assume that the Spoiler always marks his point in G_1, he can switch the graph he is marking in and the proof still holds so far.

Now we need to consider the strategy for responding to a point that the Spoiler marks Outside. But this is easy, the Duplicator just also needs to choose a point that is Outside. The only thing that we have to do is show that it is always possible for the Duplicator to choose an outside point. Suppose that the positions are equivalent with k turns remaining. So far the only points to be marked will be the endpoints, and our notion of closeness is if the marked point is within 2^{k-1} of a marked point. Thus there are $2(2^{k-1}+1) = 2^k + 2 < m, n$ possible Inside points. Since this is less than m and n we know that there will be an Outside point for the Duplicator to mark. Now consider the next move. We will have to worry about more than just the endpoints, there will be one point somewhere on the path, but also our notion of closeness will have changed. So now we have that there are less than $2(2^{k-2}+1) + 2^{k-1} = 2^k + 2$ inside points again. In general we have to count the two endpoints (which exclude an area of possible outside points only half of a different point) and all of the other points. So in general with s turns remaining there are $2(2^{s-1}+1) + (k-s)(2^s+1)$ inside points. Now we only need to show that this is always less than $2^{k+1}+1$ and then we will know that there are always points to choose on the outside. Below in the first line we make use of the fact that $\forall n \in \mathbb{N}\ n < 2^n$.

$$\begin{aligned} 2(2^{s-1}+1) + (k-s)(2^s+1) &\leq 2^s + 2 + 2^{k-s}(2^s+1) \\ &= 2^s + 2 + 2^k + 2^{k-s} \\ &< 2^k + 2^k + 2\ \forall x, y > 0\ 2^x + 2^y < 2^{x+y} \\ &= 2^{k+1} + 2 \end{aligned}$$

Thus since $m, n > 2^{k+1} + 1$ we know that there is at least one point that can be chosen that is outside at any point (and probably many more, as a generous approximation was made.)

Now we are done, since we have shown that if equivalence can be maintained until the s^{th} turn it can also be maintained on the $s + 1^{st}$ turn. □

Exercise 5. Prove that if $m, n > 2^{k+1} + 1$ the Duplicator wins $\text{Ehr}(C_m, C_n, k)$. (Where C_i is the cycle of length i.)

2.2 The Eherenfeucht Game On Models

Now we move to a more generalized version of the Eherenfeucht game. So far we have been only playing the game on two graphs, but the game can be generalized if we consider two models of the same language. The game is still the same as before, just with changes that force the duplicator to maintain all of the structure that can be expressed in a given language.

So when dealing with models at the end of the game the Duplicator wins if these conditions are met:

Let $\mathcal{M}, \mathcal{N}$ be $\mathcal{L}$-*structures*. Let $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$ be the elements of the universe selected in $\mathcal{M}$ and $\mathcal{N}$ respectively.

1. $R(x_{i_1}, \ldots, x_{i_{n_R}}) \Leftrightarrow R(y_{i_1}, \ldots, y_{i_{n_R}})\ \forall R \in \mathcal{R}, 1 \leq i_1, \ldots, i_{n_R} \leq k$

2. $f(x_{i_1}, \ldots, x_{i_{n_f}}) = x_{i_{n_f+1}} \Leftrightarrow f(y_{i_1}, \ldots, y_{i_{n_f}}) = y_{i_{n_f+1}} \, \forall f \in \mathcal{F}, 1 \leq i_1, \ldots, i_{n_f}, i_{n_f+1} \leq k$

3. $x_i = c \Leftrightarrow y_i = c \; \forall c \in \mathcal{C}, 1 \leq i \leq k$

2.2.1 The k-Ehrenfeucht Value

We have looked at a few specific cases of the Ehrenfeucht game, but it can also be thought of in much more general terms. The first thing to do is to create classes of $\mathcal{L}$-*structures*, where playing the Ehrenfeucht game with k turns on any two $\mathcal{L}$-*structures* from the same class will be a win for the Duplicator. You may wonder why we are looking for wins for the Duplicator and not for the Spoiler. It is because when the Duplicator wins there is some relationship between the two structures, and it is exploring this relationship with first order logic that will be our main theorem.

The first thing that we need to do is to define an equivalence relation that will break the $\mathcal{L}$-*structures* into disjoint classes. We do this as follows:

Definition 9. *For two $\mathcal{L}$-structures, $\mathcal{M}$, $\mathcal{N}$ we say that $\mathcal{M} \equiv_k \mathcal{N}$ if the Duplicator wins* $\text{Ehr}(\mathcal{M}, \mathcal{N}, k)$*. We say that $(\mathcal{M}, x_1, \ldots, x_s) \equiv_k (\mathcal{N}, y_1, \ldots, y_s)$ if the Duplicator wins* $\text{Ehr}(\mathcal{M}, \mathcal{N}, k)$*, with $x_1, \ldots, x_s, y_1, \ldots, y_s$ already marked.*

Note that this definition specifies the amount of turns that the game will be played. Also it is not completely obvious that this is an equivalence relation. Of course we know the relation is reflexive, since any graph is isomorphic to itself. Also it is clear that the relation is symmetric, since the order that the graphs are listed in has no effect on the game.

Exercise 6. Prove that $\equiv_k$ is transitive.

We must also have some way to refer to the equivalence classes, here this is done by defining the notion of k-Ehrenfeucht values.

Definition 10. *The set of different equivalence values on graphs with no marked verticies for any k is called* $\text{Ehrv}[k]$*. The set of different equivalence values on graphs with s marked verticies for any k is called* $\text{Ehrv}[k, s]$*.*

This equivalence relation completely partitions the set of $\mathcal{L}$-*structures*. This is because for every $\mathcal{L}$-*structure*, $\mathcal{M}$, $\mathcal{M} \equiv_k \mathcal{M}$, so every $\mathcal{L}$-*structure* has a k-Ehrenfeucht value.

Example 7. We will examine the case of $\mathcal{L}_{Graph}$. Let $\sigma_1 = \forall x(\neg(x \sim x))$. We will look at $k = 2$ and all $\mathcal{L}_{Graph}$-*structures*, $\mathcal{M}$, where $\mathcal{M} \models \sigma_1$. We impose this limitation on the models because the adjacency relation in graph theory can not be reflexive. What are the 2-Ehrenfeucht values in this case?

1. The graph on 0 verticies.

2. The graph on 1 vertex.

3. A complete graph with 2 or more verticies.

4. An empty graph with 2 or more verticies.

5. All verticies focal or mixed, at least one of each.

6. All verticies isolated or mixed, at least one of each.

7. All verticies mixed.

It is easy to see that the Spoiler can win in a graph from one class is paired with a graph from another class. For example, if a complete graph and an empty graph are paired the Spoiler can either choose two neighbors in the complete graph or choose any vertex in the empty graph and then choose a neighbor of the point that the Duplicator chose in the complete graph.

Also all graphs must fall into one of these categories, so this is a complete characterization of the 2-Ehrenfeucht values in this case.

In this case there are only 7 Ehrenfeucht values, but will there always even be finitely many? In most cases there will be infinitely many $\mathcal{L}$-*structures*, so it is not clear that for any k there are finitely many k-Ehrenfeucht values.

Theorem 3. *For any finite language $\mathcal{L}$ and any finite k,* $\text{Ehrv}[k]$ *and* $\text{Ehrv}[k,s]$ *are finite.*

Proof. We will prove this by induction. First note that $\text{Ehrv}[k]$ is the same as $\text{Ehrv}[k,0]$. We will do induction on s, starting from the value of $s=k$. When $s=k$ all of the points are already marked. So in graph theory this simply becomes the question of how many graphs there are on k verticies. This number would not necessarily be the amount of distinct k-Ehrenfeucht values, but it would have to be an upper bound, as any k points that are chosen from a graph form a subgraph on k points.

Translating this notion to the more general setting of $\mathcal{L}$-*structures* we must simply count all of the possible $\mathcal{L}$-*structures* on k points. Then if any k points are chosen from any $\mathcal{L}$-*structure* it will be guaranteed to be one of the structures we are counting. We can break this up by the different kinds of structure that are possible:

1. First look at the possible structure imposed by the relations:
   We know our language is finite, so we have $\mathcal{R}=R_1,\ldots,R_n$. Now how many different relations are possible on k points? Well, for a relation R of arity n_R there are k^{n_R} points in its domain. In a relation each point must either hold or not hold, so there are $2^{k^{n_R}}$ different values of that relation on k points. So we have a total of $\mu_{\mathcal{R}}=\prod_{R\in\mathcal{R}}2^{k^{n_R}}$ possible relations.

2. Now we have to do the same thing for functions: The structure imposed by our function is if $f(x_1,\ldots,x_{n_f})=x_i$. Now for any f we have k possibilities for each coordinate of the input and k possibilites for the output. So for any function we have an upper bound of k^{n_f+1} possible implementations. Then for all functions we have an upper bound of $\mu_{\mathcal{F}}=\prod_{f\in\mathcal{F}}k^{n_f+1}$.

3. Finally we have to find an upper bound for the constants: We know how many total constants there are, so we just need to count the different ways that they could appear in our k chosen points. Let $n = |\mathcal{C}|$. For all $0 \leq i \leq k$ we have $k!\binom{n}{i}$ ways that the constants can appear. This is a total of $\mu_{\mathcal{C}} = \prod_{i=0}^{k} k!\binom{n}{i}$.

Now since all of these events are independent of one another we have an upper bound of $\mu_{\mathcal{R}}\mu_{\mathcal{F}}\mu_{\mathcal{C}}$ possible $\mathcal{L}$-*structures* on k points. This is an upper bound for the amount of k-Ehrenfeucht values with k marked points because we know that each $\mathcal{L}$-*structure* will for the Duplicator when paired with itself, so maximally there are $\mu_{\mathcal{R}}\mu_{\mathcal{F}}\mu_{\mathcal{C}}$ k-Ehrenfeucht values of size one.

Suppose that $\text{Ehrv}[k, s+1]$ is finite. We need to show that $\text{Ehrv}[k, s]$ is finite. Well, now we can take some $\alpha \in \text{Ehrv}[k, s]$. We will relate α to a subset of $\text{Ehrv}[k, s+1]$. Let $(x_1, \ldots, x_s) \in \alpha$. Now let $\chi_{x_1,\ldots,x_s} = \{\beta \mid (x_1, \ldots, x_s, x) \in \beta \text{ where x ranges over all possible values}\}$. Suppose that for some $y_1, \ldots, y_s$, $\chi_{x_1,\ldots,x_s} = \chi_{y_1,\ldots,y_s}$. Then $\forall x \exists y((x_1, \ldots, x_s, x) \equiv_k (y_1, \ldots, y_s, y))$ and similarly $\forall y \exists x((x_1, \ldots, x_s, x) \equiv_k (y_1, \ldots, y_s, y))$. Why? This is because they have exactly the same value for χ, so the choice of any x value puts $(x_1, \ldots, x_s, x)$ in some equivalence class for $\text{Ehrv}[k, s+1]$, but we know that there is a choice for y which will put $(y_1, \ldots, y_s, y)$ into that same class, because they have the same value for χ. Thus we have that $(x_1, \ldots, x_s) \equiv_k (y_1, \ldots, y_s)$ and that $(y_1, \ldots, y_s) \in \alpha$. It is easy to see that if $(x_1, \ldots, x_k) \equiv_k (y_1, \ldots, y_k)$ then $\chi_{x_1,\ldots,x_k} = \chi_{y_1,\ldots,y_k}$.

We have shown that if two s-tuples have the same value for χ then they are in the same equivalence class in $\text{Ehrv}[k, s]$. This means that there is an upper-bound of $|\text{Ehrv}[k, s]|$ given by all subsets of $\text{Ehrv}[k, s+1]$. Thus $|\text{Ehrv}[k, s]| < 2^{|\text{Ehrv}[k,s+1]|}$. Now our induction step is complete, because we know that $|\text{Ehrv}[k, s]|$ is finite.

Finally, by induction we have our result. Our upper bound is very large, but it is finite which is what we require. □

3 The Connection Between the Ehrenfeucht Game and First Order Logic

Now we can prove the main theorem of this paper. The connection between first order logic and the Ehrenfeucht game has already appeared partially. The Duplicator only wins the Ehrenfeucht game if the structure of the $\mathcal{L}$-*structures* on the marked points is the same. This structure can be expressed using first order logic in the language, and thus you would expect some relationship between the sentences that are true on the $\mathcal{L}$-*structures* of a particular Ehrenfeucht value.

Theorem 4. *Let $\mathcal{L}$ be a finite language and $\mathcal{M}, \mathcal{N}$ be $\mathcal{L}$-structures.*

1. $(\mathcal{M}, x_1, \ldots, x_s) \equiv_k (\mathcal{N}, y_1, \ldots, y_s)$ if an only if all first order formulas of quantifer depth $k - s$ with s free variables have the same truth value on $\mathcal{M}$ and $\mathcal{N}$ when the free varibles are given the values $x_1, \ldots, x_s$ and $y_1, \ldots, y_s$ respectively.

2. $\forall \alpha \in \mathrm{Ehrv}[k,s]$ *there is a formula of quantifier depth* $k-s$ *with* s *free variables,* $\mathcal{A}(v_1,\ldots,v_s)$ *such that for any* $\mathcal{L}$*-structure,* $\mathcal{M}$*, with marked verticies* $x_1,\ldots,x_s$ $\mathcal{M}$ *has the k-Ehrenfeucht value* α *if and only if* $\mathcal{M} \models \mathcal{A}(x_1,\ldots,x_s)$.

Proof. We will begin by proving the second statement by induction. First note that if $k = s$ the expressible structure of the $\mathcal{L}$*-structure* is completely determined. This is because we have k marked points in a $\mathcal{L}$*-structure*, $\mathcal{M}$, so we can actually just list the conditions that are required of a different $\mathcal{L}$*-structure*, $\mathcal{N}$, with k marked points so that they have the same k-Ehrenfeucht value. To do this we just form a first order sentence with k free variables that lists all of the structure in $\mathcal{M}$. This will be exactly the right sentence since two $\mathcal{L}$*-structures* must have exactly the same structure on their k marked points for the Duplicator to win.

1. The structure that we need to preserve for relations is just whether they hold or not on a given tuple of our k points. Define the following two sentences:

$$\sigma_{\mathcal{R}} = \bigwedge_{1 \le i_1,\ldots,i_{n_R} \le k, R \in \mathcal{R}, \mathcal{M} \models R(x_{i_1},\ldots,x_{i_{n_R}})} R(v_{i_1},\ldots,v_{i_{n_R}})$$

$$\sigma_{\neg\mathcal{R}} = \bigwedge_{1 \le i_1,\ldots,i_{n_R} \le k, R \in \mathcal{R}, \mathcal{M} \models \neg R(x_{i_1},\ldots,x_{i_{n_R}})} \neg R(v_{i_1},\ldots,v_{i_{n_R}})$$

   Then if we let $\tau_R = \sigma_{\mathcal{R}} \wedge \sigma_{\neg\mathcal{R}}$ we have captured the structure of the relations on the k marked points of $\mathcal{M}$.

2. For functions we need to make sure that they are the same when the range and domain are restricted to our k points. We must again define a sentence:

$$\tau_{\mathcal{F}} = \bigwedge_{1 \le i_1,\ldots,i_{n_f}, i_{n_f+1} \le k, f \in \mathcal{F}, \mathcal{M} \models f(x_{i_1},\ldots,x_{i_{n_f}}) = x_{i_{n_f+1}}} f(v_{i_1},\ldots,v_{i_{n_f}}) = v_{i_{n_f+1}}$$

3. Finally we have to deal with constants. The only structure imposed by constants is if $x_i = c_j$ for some i and j. We have to note all of these occurances. Let $\tau_{\mathcal{C}} = \bigwedge_{1 \le i \le k, 1 \le j \le |\mathcal{C}|, \mathcal{M} \models x_i = c_j} v_i = c_j$.

Let $\tau = \tau_{\mathcal{R}} \wedge \tau_{\mathcal{F}} \wedge \tau_{\mathcal{C}}$. Now we have accounted for all of the possible types of structure on the k marked points. It is important to note that since our language is finite τ must be of finite length. So we have a formula of quantifer depth $k - k = 0$ with k free variables where for any $\mathcal{L}$*-structures* with k marked points, $(\mathcal{N}, y_1,\ldots,y_k)$ $\mathcal{N} \models \sigma(y_1,\ldots,y_k)$ exactly when $(\mathcal{M}, x_1,\ldots,x_k) \equiv_k (\mathcal{N}, y_1,\ldots,y_k)$.

Now we have proven the base case of our induction. So assume that for every $\beta \in \mathrm{Ehrv}[k, s+1]$ there is a first order sentence of quantifier depth $k-s-1$ with

$s+1$ free variables such that $(\mathcal{M}, x_1, \dots, x_s) \in \beta$ exactly when $(\mathcal{M}, x_1, \dots, x_s) \in \beta$.

We use a similar method to the one used in proving that Ehrv$[k]$ is finite. Again the key fact is that the subsets of Ehrv$[k, s+1]$ correspond to values of Ehrv$[k, s]$. We begin by taking any $\alpha \in$ Ehrv$[k, s]$. Now take some $(x_1, \dots, x_s) \in \alpha$. Now for every $\beta \in$ Ehrv$[k, s+1]$ we know by induction that there is a first order formula with quantifier depth $k - s - 1$ and $s + 1$ free variables that is true only for its elements, call this formula $\mathcal{A}_\beta$. Now we can write a formula $\exists x(\mathcal{A}_\beta(x_1, \dots, x_s, x))$. The truth value of this formula is determined by the choice of β. Let Yes$[\alpha]$ be the set of β on which this sentence is true, and No$[\alpha]$ be the set of β on which it is false. Then we simply say that $\mathcal{A}_\alpha(x_1, \dots, x_s) = \bigwedge_{\beta \in \text{Yes}[\alpha]} \exists x(\mathcal{A}_\beta(x_1, \dots, x_s, x)) \wedge \bigwedge_{\beta \in \text{No}[\alpha]} \neg\exists x(\mathcal{A}_\beta(x_1, \dots, x_s, x))$.

This sentence determines the Ehrenfeucht value of any $\mathcal{L}$-*structure* with s marked points that satisfies it, because it associates it with other $\mathcal{L}$-*structures* with exactly the same set of Yes$[\alpha]$ and No$[\alpha]$. Thus if any new point is marked by the Spoiler for any two structures satisfying this sentence there must be a response for the Duplicator by the same logic as in the proof for finiteness of Ehr$[k, s]$. So the second part is proven.

Now to prove the first part. Assume that we have two $\mathcal{L}$-*structures*, $(\mathcal{M}, x_1, \dots, x_s), (\mathcal{N}, y_1, \dots, y_s)$ such that for all first order sentences, ϕ with s free variables and quantifier depth $k - s$ we have $(\mathcal{M} \models \phi(x_1, \dots, x_s)) \Leftrightarrow (\mathcal{N} \models \phi(y_1, \dots, y_s))$. They must satisfy some $\mathcal{A}_\alpha$ because the Ehrenfeucht values form a partition of the $\mathcal{L}$-*structures* with s marked verticies. Thus since they satisfy the same sentences, they must satisfy the same $\mathcal{A}_\alpha$ and have the same Ehrenfeucht value.

Finally suppose that $(\mathcal{M}, x_1, \dots, x_s) \equiv_k (\mathcal{N}, y_1, \dots, y_s)$. We will prove by induction that all first order sentences with quantifier depth $k - s$ and s free variables have the same truth value. First we do the case where $s = k$. In this case we just have boolean combinations of the atomic formulas. But we know that the atomic formulas with k free variables all have the same truth value on $\mathcal{M}$ and $\mathcal{N}$ when $x_1, \dots, x_k, y_1, \dots, y_k$ are used as the free variables, because this is precisely the condition required for them to have the same k-Ehrenfeucht value. Thus they have the same truth value on any formula with quantifier depth 0 and k free variables.

Now suppose that the statement is true for $s + 1$. Let ϕ be any formula with quantifier depth $k - s$ and s free variables. Then by Exercise 1 there is a formula ψ with quantifier depth $k - s - 1$ and $s + 1$ free variables such that $\emptyset \models (\phi \Leftrightarrow (\exists x(\psi)))$. By the induction assumption we know that the truth value of ψ is determined by the Ehrenfeucht value of $(\mathcal{M}, x_1, \dots, x_s)$ and $(\mathcal{N}, x_1, \dots, x_s)$. Thus the truth value of $\exists x(\psi)$ is determined, and they must be the same on both structures. □

Corollary 1. Let $\mathcal{L}$ be a finite language and $\mathcal{M}, \mathcal{N}$ be $\mathcal{L}$-*structures*.

1. $\mathcal{M} \equiv_k \mathcal{N}$ if an only if all first order sentences have the same truth value on $\mathcal{M}$ and $\mathcal{N}$.

2. $\forall \alpha \in \text{Ehrv}[k]$ there is a first order sentence, σ, such that for any $\mathcal{L}$-*structure*, $\mathcal{M}$, $\mathcal{M}$ has the k-Ehrenfeucht value α if and only if $\mathcal{M} \models \sigma$.

4 Applications

Now we can look at some applications of the last theorem. The first thing to note is that this provides a way for us to prove that some property of a model is not expressable in first order logic. We will move back to graph theory for some examples. Recall the result that for any fixed $k \in \mathbb{N}$ $\exists n$ such that $\forall i, j \geq n, C_i \equiv_k C_j$

Example 8 (Conectedness). A slight modification of the proof for cycles can be made to prove that for any fixed $k \in \mathbb{N}$ $\exists n$ such that $\forall i, j, m \geq n, C_i \equiv_k (C_j \cup C_m)$. Now we have two graphs in the same equivalence class, so all first order sentences must have the same truth value on them. But C_i is connected while $C_j \cup C_m$ (the graph with two disjoint cycles of length j and m) is not. Thus connectedness cannot be expressible as a first order sentence.

Exercise 7 (2-Colorability). Show that 2-Colorability is also not a first order property.

References

[1] David Marker *Model Theory: An Introduction*, chapters 1–2, pp. 1–69, Springer, New York, New York, 2002.

[2] Joel Spencer *The Strange Logic of Random Graphs*, chapters 0–3, pp. 1–65, Springer, Berlin, Germany, 2001.