Lesson 3: Ehrenfeucht-Fraïssé games

Theme: Limitations of first-order logic and relational algebra.

1 Partial isomoprhism

Let $\mathcal{A} = (A, R_1, \ldots, R_k)$ and $\mathcal{B} = (B, R_1, \ldots, R_k)$ be two databases over vocabulary $\tau = \{R_1, \ldots, R_k\}$. For $\bar{a} = (a_1, \ldots, a_n) \in A^n$ and $\bar{b} = (b_1, \ldots, b_n) \in B^n$, we write $\bar{a} \mapsto \bar{b}$ to denote the function that maps a_i to b_i. We call $\bar{a} \mapsto \bar{b}$ a *partial isomorphism*, if the following holds.

- $a_i = a_j$ if and only if $b_i = b_j$.
- For every $R_l \in \tau$, for every $1 \leqslant i_1, \ldots, i_{\mathrm{ar}(R_l)} \leqslant n$,

$$a_{i_1}, \ldots, a_{\mathrm{ar}(R_l)} \in R_l^{\mathcal{A}} \quad \text{if and only if} \quad b_{i_1}, \ldots, b_{\mathrm{ar}(R_l)} \in R_l^{\mathcal{B}}$$

2 Ehrenfeucht-Fraïssé (EF) games

The game. An Ehrenfeucht-Fraïssé (EF) game is played on two databases $\mathcal{A} = (A, R_1, \ldots, R_k)$ and $\mathcal{B} = (B, R_1, \ldots, R_k)$. There are two players: *spoiler* and *duplicator*. An n-round EF game consists of n rounds, and each round consists of the following steps.

- The spoiler picks a structure (either $\mathcal{A}$ or $\mathcal{B}$), and makes a move by picking an element in that database: either $a \in \mathcal{A}$ or $b \in \mathcal{B}$.
- The duplicator responds by picking an element in the other database.

The winning rule. Let $\bar{a} = (a_1, \ldots, a_n)$ and $\bar{b} = (b_1, \ldots, b_n)$ be the elements pick on $\mathcal{A}$ and $\mathcal{B}$, respectively, where a_i and b_i are the elements pick on the i^{th} round. Duplicator *wins*, if $\bar{a} \mapsto \bar{b}$ is a partial isomorphism. Otherwise, Spoiler *wins*.

We say that Duplicator has winning strategy in an n-round EF game (played on $\mathcal{A}$ and $\mathcal{B}$), if Duplicator can play in a way that guarantees he/she wins in n rounds, regardless of how Spoiler plays. We denote it by $\mathcal{A} \equiv_n \mathcal{B}$.

A simple, but useful, modification of EF games. Note that in the EF game described above, the game always starts "fresh," i.e., none of the elements from $\mathcal{A}$ or $\mathcal{B}$ are chosen in advance. In this course, it is useful to define the EF games where some elements are already chosen *before* the game starts.

The formal definition is as follows. As before, let $\mathcal{A} = (A, R_1, \ldots, R_k)$ and $\mathcal{B} = (B, R_1, \ldots, R_k)$ be two databases. Let $\bar{c} = (c_1, \ldots, c_m) \in A^m$ and $\bar{d} = (d_1, \ldots, d_m) \in B^m$ be the elements from $\mathcal{A}$ and $\mathcal{B}$, chosen before the game starts.

The rule of the game is the same as before, i.e., the Spoiler and Duplicator alternately pick elements from $\mathcal{A}$ and $\mathcal{B}$. After n rounds, suppose $\bar{a} = (a_1, \ldots, a_n)$ and $\bar{b} = (b_1, \ldots, b_n)$ be the elements chosen from $\mathcal{A}$ and $\mathcal{B}$, respectively. Duplicator wins, if $(\bar{c}, \bar{a}) \mapsto (\bar{d}, \bar{b})$ is a partial isomoprhism. Otherwise, Spoiler wins. That Duplicator has winning strategy in an n-round game played on $(\mathcal{A}, \bar{c})$ and $(\mathcal{B}, \bar{d})$ is defined similarly, in which case we denote by $(\mathcal{A}, \bar{c}) \equiv_n (\mathcal{B}, \bar{d})$.

Remark 3.1 If $(\mathcal{A}, \bar{c}) \equiv_n (\mathcal{B}, \bar{d})$, then $(\mathcal{A}, \bar{c}) \equiv_m (\mathcal{B}, \bar{d})$, for every $m \leqslant n$.

3 EF games and first-order logic

The *quantifier rank* of a formula α, denoted by $\mathrm{qr}(\alpha)$, is defined inductively as follows.

- The quantifier rank of an atomic formula is zero.
- $\mathrm{qr}(\neg\beta) = \mathrm{qr}(\beta)$.
- $\mathrm{qr}(\beta \wedge \gamma) = \mathrm{qr}(\beta \vee \gamma) = \max(\mathrm{qr}(\beta), \mathrm{qr}(\gamma))$.
- $\mathrm{qr}(\exists x\ \beta) = \mathrm{qr}(\beta) + 1$.

Theorem 3.2 *The following two statements are equivalent.*

- $(\mathcal{A}, \bar{c}) \equiv_n (\mathcal{B}, \bar{d})$.
- *For every formula $\varphi(\bar{x})$ of quantifier rank $\leqslant n$,*

$$(\mathcal{A}, \bar{x} \mapsto \bar{c}) \ \models\ \varphi(\bar{x}) \quad \textit{if and only if} \quad (\mathcal{B}, \bar{x} \mapsto \bar{d}) \ \models\ \varphi(\bar{x})$$

Proof. The proof is by induction on n. The base case $n = 0$ is straightforward, due to the definition of partial isomoprhism.

For the induction hypothesis, we assume that the theorem holds for n. For the induction step, we will show it for the case $n + 1$.

First we show that the first statement implies the second. Assume that $(\mathcal{A}, \bar{c}) \equiv_{n+1} (\mathcal{B}, \bar{d})$. This means that for every Spoiler's choice $a \in A$, Duplicator has an answer $b \in B$ such that $(\mathcal{A}, \bar{c}, a) \equiv_n (\mathcal{B}, \bar{d}, b)$.

Now, let $\exists y\varphi(\bar{x}, y)$ be of quantifier rank $n + 1$. If $\mathcal{A}, \bar{c} \models \exists y\varphi(\bar{x}, y)$, by definition, there is $a \in A$ such that $\mathcal{A}, \bar{c}, a \models \varphi(\bar{x}, y)$. Let b be Duplicator's answer so that $(\mathcal{A}, \bar{c}, a) \equiv_n (\mathcal{B}, \bar{d}, b)$. By the induction hypothesis, $(\mathcal{B}, \bar{d}, b) \models \varphi(\bar{x}, y)$, which means $(\mathcal{B}, \bar{d}) \models \exists y\varphi(\bar{x}, y)$. That $(\mathcal{B}, \bar{d}) \models \exists y\varphi(\bar{x}, y)$ implies $\mathcal{A}, \bar{c} \models \exists y\varphi(\bar{x}, y)$ is similar.

Now, we show that the second statement implies the first. Suppose that for every formula $\varphi(\bar{x})$ of quantifier rank $\leqslant n + 1$,

$$(\mathcal{A}, \bar{x} \mapsto \bar{c}) \ \models\ \varphi(\bar{x}) \quad \text{if and only if} \quad (\mathcal{B}, \bar{x} \mapsto \bar{d}) \ \models\ \varphi(\bar{x}) \tag{1}$$

Assume to the contrary that $(\mathcal{A}, \bar{c}) \not\equiv_{n+1} (\mathcal{B}, \bar{d})$. This means there is $\alpha \in A$ such that for every $\beta \in B$, we have

$$(\mathcal{A}, \bar{c}, \alpha) \not\equiv_n (\mathcal{B}, \bar{d}, \beta).$$

By the induction hypothesis, for every $\beta \in B$, there is a formula $\psi_\beta(\bar{x}, y)$ of quantifier rank $\leqslant n$ such that

$$(\mathcal{A}, \bar{x}, y \mapsto \bar{c}, \alpha) \ \models\ \psi_\beta(\bar{x}, y) \quad \text{and} \quad (\mathcal{B}, \bar{x}, y \mapsto \bar{d}, \beta) \ \not\models\ \psi_\beta(\bar{x}, y)$$

We negate the formula $\psi_\beta(\bar{x}, y)$ in case:

$$(\mathcal{A}, \bar{x}, y \mapsto \bar{c}, \alpha) \ \not\models\ \psi_\beta(\bar{x}, y) \quad \text{and} \quad (\mathcal{B}, \bar{x}, y \mapsto \bar{d}, \beta) \ \models\ \psi_\beta(\bar{x}, y)$$

Now, let $B = \{\beta_1, \ldots, \beta_m\}$. Consider the formula:

$$\varphi(\bar{x}) \quad := \quad \exists y \bigwedge_{i=1}^{m} \ \psi_{\beta_i}(\bar{x}, y).$$

Then,

$$(\mathcal{A}, \bar{x} \mapsto \bar{c}) \ \models\ \varphi(\bar{x}) \quad \text{and} \quad (\mathcal{B}, \bar{x} \mapsto \bar{d}) \ \not\models\ \varphi(\bar{x})$$

Note that the quantifier rank of $\varphi(\bar{x})$ is $\leqslant n+1$. So, this contradicts the assumption (1), and this completes our proof. ■

Remark 3.3 EF games are a very useful tool to show that certain queries can not be expressed by first-order formula, thus, also relational algebra expression. To prove a query $Q(\bar{x})$ cannot be expressed by FO formulas, typically the proof proceeds as follows.

(1) Assume to the contrary that $Q(\bar{x})$ can be expressed by an FO formula $\varphi(\bar{x})$, i.e., for every database DB, $Q(\mathsf{DB}) = \varphi(\mathsf{DB})$.

    Let n be the quantifier rank of φ.

(2) Construct two databases $(\mathcal{A}, \bar{a})$ and $(\mathcal{B}, \bar{b})$ such that $(\mathcal{A}, \bar{x} \mapsto \bar{a}) \models \varphi$, but $(\mathcal{B}, \bar{x} \mapsto \bar{b}) \not\models \varphi$.

(3) Describe the winning strategy for Duplicator on n-round EF game played on $(\mathcal{A}, \bar{a})$ and $(\mathcal{B}, \bar{b})$.

(4) Note that items (2) and (3) contradict each other, thus, we can conclude that $Q(\bar{x})$ can not be expressed by an FO formula.

For a more comprehensive treatment of EF games, see, for example, [1, 2, 3, 4].

Note that in our proof of Theorem 3.2 above, we assume the domain is finite. In general Theorem 3.2 also holds for infinite domain, but the proof is more complicated. Please see the references below for more details.

References

[1] H.-D. Ebbinghaus and J. Flum. *Finite model theory.* Perspectives in Mathematical Logic. Springer, 1995.

[2] N. Immerman. *Descriptive complexity.* Graduate texts in computer science. Springer, 1999.

[3] L. Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2004.

[4] W. Rautenberg. *A Concise Introduction to Mathematical Logic (3rd ed.).* Universitext. Springer, 2010.