Logic, descriptive complexity and theory of databases

8 novembre 2010

Lecture 6
Lecturer : Luc Segoufin
Scribe : Arnaud de Mesmay

1 Recap

We have already seen the following results :
- FO $\subseteq$ LOGSPACE, has 0-1 Laws, Locality and Ehrenfeucht-Fraïssé Games.
- ESO = NP
- LFP=IFP $\subseteq$ PFP. They all satisfy 0-1 Laws (hence can't compute Parity). Can compute Connectivity (they are designed for that !)
- LFP = IFP $\subseteq$ PTIME.
- PFP $\subseteq$ PSPACE.
- Immerman-Vardi Theorem : On <u>ordered</u> structures ,

$$LFP = IFP = PTIME.$$

$$PFP = PSPACE.$$

2 Abiteboul-Vianu Theorem

The goal of this course is to prove the following theorem.

[2.A] Theorem ***(Abiteboul-Vianu 91)***
LFP=PFP if and only if PTIME=PSPACE

A natural idea to prove PTIME $\neq$ PSPACE would be to pick a PSPACE-complete problem (Quantified Boolean Formula..) and try to prove that it cannot be computed in PTIME. But it's very (too ?) hard. This theorem enables us to pick a PSPACE-complete problem which is provably in PFP (like the Game of Life) and prove that it is not in LFP, reducing a complexity problem expressed in terms of Turing machines to an expressivity problem : the one of proving whether some semantics to compute fixed points is stronger than another one.

Proof of [2.A].

There are two directions, the first one is easy, the reverse one is harder.
First direction : Assume LFP=PFP.
Fix σ, take P a property over σ-structures such that $P \in PSPACE$.
- On input I, compute a linear order on the domain of I, which gives us a new ordered structure I^*.
-

$$\begin{aligned} P \in PSPACE \text{ on } I^* \quad & \underbrace{\rightarrow}_{I-V thm} \quad \exists \Psi \in PFP, I^* \models P \text{ iff } I^* \models \Psi. \\ & \underbrace{\rightarrow}_{LFP=PFP} \quad \exists \varphi \in LFP, I^* \models \Psi \text{ iff } I^* \models \varphi. \\ & \rightarrow \quad \exists \text{ PTIME Turing Machine }, I^* \models \varphi \text{ iff M accepts } I^*. \end{aligned}$$

Reverse direction :

Let $\varphi \in PFP$, let k be the number of variables used in φ. Let us assume to simplify things that $k >$ maximum arity in σ.

Recall first the two equivalent meanings of the equivalence relation $\overset{k}{\equiv}$:

$$(I, \overline{a}) \overset{k}{\equiv} (J, \overline{b})$$

1. $\forall \varphi(\overline{x}) \in FO$ using k variables,

$$I \models \varphi(\overline{a}) \text{ iff } J \models \varphi(\overline{b})$$

2. D wins the k-pebble game on (I, J) with initial configuration $(\overline{a}, \overline{b})$.

The proof is divided in three steps. The first step of the proof is to show that this equivalence relation can be expressed in LFP :

[2.B] Proposition *(1)*

$\exists \eta(\overline{x}, \overline{y}) \in LFP$,

$$\forall I, \overline{a}, \overline{b}, I \models \eta(\overline{a}, \overline{b}) \textit{ iff } (I, \overline{a}) \overset{k}{\equiv} (I, \overline{b}).$$

The second step is to show that one can define a preorder compatible with $\overset{k}{\equiv}$ on I^k in LFP.

[2.C] Proposition *(2)*

$\exists \chi(\overline{x}, \overline{y}) \in LFP$,

$$\forall I, \chi \textit{ defines a preorder on } I^k \textit{ with } \overset{k}{\equiv} \textit{ as associated equivalence relation}$$

The last step of the proof needs some more definitions :

Starting with a signature $\sigma = (R_1..R_l)$, define a new signature, called canonical :

$$\sigma^* = (<, U, U_1..U_l, S_1..S_k, P_1..P_t)$$

where $<$ is binary, the U relations are unary, the S and P relations are binary and $t = k^k$.

We can now build a structure on σ^* from a structure on σ as follows :

$I(\sigma)$	$\rightarrow$	$I^*(\sigma^*)$
V is the universe of I		$V^k / \overset{k}{\equiv}$ universe of I^*
		$<$ is χ.
		$U([\overline{a}])$ iff $a_1 = a_2$.
		$U_i([\overline{a}])$ iff $(a_1..a_m) \in R_i$ where $m \leq k$ is the arity of R_i.
		$S_j([\overline{a}], [\overline{b}])$ iff $\forall \alpha, a_\alpha = b_\alpha$ except maybe for $\alpha = j$.
		$\Pi : \{1..k\} \rightarrow \{1..k\}, P_\Pi([\overline{a}], [\overline{b}])$ iff $\overline{b} = \Pi(\overline{a})$.

The idea is that these logics cannot distinguish things equivalent for $\overset{k}{\equiv}$, so we have to quotient. The magic of the construction is that in this new universe $V^k / \overset{k}{\equiv}$, we can have a linear order expressible in LFP. The other definitions are just technicalities which will be justified during the proof.

The third proposition states that any property using k-variables (defined with partial fixpoints) can be lifted from σ to σ^* and the converse is also true.

[2.D] Proposition *(3)*

1. $\forall \varphi \in PFP(\sigma)$ *using k-variables,*

$$\exists \varphi^* \in PFP(\sigma^*), \forall I, I \models \varphi \textit{ iff } I^* \models \varphi^*.$$

2. $\forall \Psi^* \in LFP(\sigma^*)$,

$$\exists \Psi \in LFP(\sigma), \forall I, I^* \models \Psi^* \textit{ iff } I \models \Psi.$$

Let us now prove the Abiteboul-Vianu theorem assuming these 3 propositions.

Assume PTIME=PSPACE. Take $\varphi \in PFP$, k the number of variables of φ. By Prop 3.1, $\exists \varphi^*, \forall I, I \models \varphi$ iff $I^* \models \varphi^*$.

$$PFP \subseteq PSPACE \to \varphi^* \in PSPACE = PTIME.$$

I^* is ordered, hence the Immerman-Vardi theorem tells us that $\exists \Psi^* \in LFP$,

$$I^* \models \varphi^* \text{ iff } I^* \models \Psi^*$$

And by prop 3.2,

$$\exists \Psi \in LFP, I^* \models \Psi^* \text{ iff } I \models \Psi.$$

Altogether we have shown that $\forall \varphi \in PFP, \exists \Psi \in LFP, \quad I \models \varphi$ iff $I \models \Psi$. Hence LFP=PFP. □

Let us now prove the propositions. As the reader might have guessed, the harder one is the second one, because building the linear order is the key step of the proof.

Onwards with the proof of Proposition 3 :

Proof of [2.D].

We start with 3.2, which works by induction on Ψ^*, it is just about translating naturally the formulas on σ^* into formulas on σ. To each variable of Ψ^* we associate a tuple of k variables in Ψ. We denote by $\to$ this translation process.

- $x = y$, hence with proposition 1, $\to \eta(\overline{x}, \overline{y})$.
- $x < y$, hence with proposition 2, $\to \chi(\overline{x}, \overline{y})$.
- $U(x) \to x_1 = x_2$.
- $U_i(x) \to R_i(x_1..x_m)$.
- $S_j(x, y) \to \bigwedge_{i \neq j} x_i = y_i$.
- $P_\Pi(x, y) \to \bigwedge_i y_i = x_{\Pi(i)}$.
- $\exists x \varphi^*(x) \to \exists \overline{x} \varphi(\overline{x})$.
- $\wedge \to \wedge$
- $\neg \to \neg$
- $\mu \to \mu$

3.1 is proved again by induction on φ, we translate φ which uses k free variables into φ^* using only 1 free variable, with the intended meaning that this only free variable matches the k original free variables.

- $x_i = x_j \to \exists y, P_\Pi(x, y) \wedge U(y)$ where Π is a function sending i to 1 and j to 2.
- $R_i(x_{i_1}..x_{i_n}) \to \exists y, P_\Pi(x, y) \wedge U_i(y)$ where Π is a function sending i_1 to 1, i_2 to 2.. i_m to m.
- $\neg, \wedge \to \neg, \wedge$.
- $\exists x_i \varphi_1(\overline{x}) \to \exists y, S_i(x, y) \wedge \varphi_1^*(y)$. (The predicate S_i states exactly that x and y differ only at position i).

□

Proof of [2.B].

We define an atomic k-type to be a consistent set of atomic formulas[1] with k free variables, i.e. for every tuple $\overline{x}$ there is either $R_i(\overline{x})$ or $\neg R_i(\overline{x})$ in a k-type. The k-type is viewed as the conjunction of its formulas. There are finitely many of them, we denote them with $\alpha_1..\alpha_t$.

We will define the formula $\neg\eta$ with a Least Fixed Point based on the game formulation of $\stackrel{k}{\equiv}$. The base case is the case where the spoiler wins immediately, which means that there is no partial isomorphism, which can be stated as follows with the types :

$$\Psi_0(\overline{x},\overline{y}) : \bigwedge_{i\neq j} \alpha_i(\overline{x}) \wedge \alpha_j(\overline{y})$$

We now compute the formula with Least Fixed Point, it is just about stating the rules of the pebbles game in the first order language :

$$\Psi(R,\overline{x},\overline{y}) : \Psi_0(\overline{x},\overline{y}) \vee \bigvee_{i\leq k} \exists x_i \forall y_i R(\overline{x},\overline{y}) \vee \bigvee_{i\leq k} \exists y_i \forall x_i R(\overline{x},\overline{y})$$

By induction one can show that if we assume that $R(\overline{x},\overline{y})$ means that the spoiler wins in $\leq i$ steps then $\Psi(R,\overline{x},\overline{y})$ states that the spoiler wins in $\leq i+1$ steps.

By setting : $\eta(\overline{x},\overline{y}) : \neg\mu_R(\Psi(R,\overline{x},\overline{y}))$, we obtain the desired property. □

The proof of proposition 2 uses the same kind of ideas but is somewhat trickier.

Proof of [2.C].

We denote again the atomic k-types by $\alpha_1..\alpha_t$ and add the notations :

$$\overline{a} = (a_1..a_k)$$

$$\overline{a}_{i\leftarrow a} = (a_1..a_{i-1}, a, a_{i+1}..a_k)$$

We define our preorder using a fixpoint such that the preorder $<_j$, obtained at step j, is a preorder whose associated equivalence relation is the complement of the relation R_j computed at the j^{th} stage of the computation of the formula η in Proposition [2.C]. In the beginning, it is just about ordering k-types, which can be done arbitrarily since they are explicitly defined :

$$\theta_0(\overline{x},\overline{y}) : \bigwedge_{i<j} \alpha_i(\overline{x}) \wedge \alpha_k(\overline{y})$$

We then define the preorder by refining at each step the tuples it orders, and then taking the limit with a fixed point. The general idea is the following : At each step, we have to order the tuples which were equivalent (according to the R of Proposition [2.C]) at step j but become nonequivalent at stage $j+1$. Depending on the pebble the spoiler moves, the linear ordering of one tuple may evolve in one direction or the other one, so we have to choose which pebble is moved and how it is moved in a canonical way ; this canonical way will be to move the smallest possible pebble to the smallest possible place in the ordering. Thus we need formulas to check whether two tuples have been differentiated yet (eq), whether they will be differentiated at the next stage (γ_i), to choose the smallest pebble to move (β_i) and a last one to figure out the new order at the step $j+1$ (δ_i).

$$eq(\overline{x},\overline{y}) : \neg S(\overline{x},\overline{y}) \wedge \neg S(\overline{y},\overline{x})$$

$$\gamma_i(\overline{x},\overline{y}) : \forall x_i \exists y_i eq(\overline{x},\overline{y}) \wedge \forall y_i \exists x_i eq(\overline{x},\overline{y})$$

$$\beta_i(\overline{x},\overline{y}) : \bigwedge_{p<i} \gamma_p(\overline{x},\overline{y}) \wedge \neg\gamma_i(\overline{x},\overline{y})$$

$$\delta_i(x,\overline{x},\overline{y}) : \forall y \left[S(\overline{x}_{i\leftarrow x},\overline{y}_{i\leftarrow y}) \vee S(\overline{y}_{i\leftarrow y},\overline{x}_{i\leftarrow y})\right]$$

We can now define θ, and χ by taking its fixed point.

[1] atom or negated atom

$$\theta(S,\overline{x},\overline{y}) : \theta_0(\overline{x},\overline{y}) \vee \neg eq(\overline{x},\overline{y}) \wedge \bigvee_{i\leq k} \beta_i(\overline{x},\overline{y}) \wedge \exists x \delta_i(x,\overline{x},\overline{y}) \wedge \forall y S(\overline{x}_{i\leftarrow x},\overline{y}_{i\leftarrow x})$$

The reader may notice that because of the negations, this is not a Least Fixed Point formula but using Inflationary Fixed Point semantics instead it still works (by the Gurevich-Shelah theorem).

□

Though the technicalities in this proof may be initially scary, the general idea that a linear order can be computed with least/inflationary fixed points on the k-tuples quotiented by the $\overset{k}{\equiv}$ relation is worth remembering.

3 Monadic second order logic

MSO is second order logic with quantification over sets, the general syntax is as follows : $\exists S, \forall S, \exists x, \forall y, S(x), x \in S, \wedge, \vee, \neg,$ atoms.

Examples :

- 3-Colorability :

$$\begin{aligned} \exists S_1,S_2,S_3, \quad \forall x, \quad & S_1(x) \vee S_2(x) \vee S_3(x) \wedge \\ \forall x, \quad & S_1(x) \rightarrow \neg S_2(x) \wedge \neg S_3(x) \wedge \\ & S_2(x) \rightarrow ... \wedge \\ & S_3(x) \rightarrow ... \wedge \\ \forall x,y \quad & \bigwedge_i (S_i(x) \wedge S_i(y) \rightarrow \neg E(x,y)) \end{aligned}$$

- Non-connectivity :

$$\exists S, \exists x \in S \wedge \exists x \notin S \wedge \forall x,y E(x,y) \rightarrow (S(x) \wedge S(y) \vee \neg S(x) \wedge \neg S(y))$$

- $(aa)^* \in MSO$:

$$\exists S, \forall x P_a(x) \wedge \min \in S \wedge \forall x S(x) \leftrightarrow \neg S(x+1) \wedge \max \in S$$

The complexity results concerning MSO are summarized in the following theorem :

[3.A] Theorem ***(Complexity of MSO)***

$MSO \subseteq PSPACE$ $MSO \subseteq PH$

The following theorem is left as an exercise :

[3.B] Theorem ***(0-1 laws)***

MSO does not verify 0-1 laws

The main result of this section (which will be only be proved in the next course) is the following :

[3.C] Theorem ***(Ehrenfeucht-Fraïssé)***

There is an equivalent E-F game for MSO.

Let's end this lecture with the proof of [3.A] :

We already know that $FO \subseteq LOGSPACE$, and a formula of MSO looks like $\exists S \varphi(S)$, hence it is natural to try all the sets, which can be done in PSPACE because each set requires $n \log n$ memory. In total, the memory needed is $f(|\varphi|) n \log n$.