

TD5

Matthieu BOYER, Grégoire LE CORRE

17 mars 2024

1 Exercise 1

Evaluating Datalog is : for each tuple in the database, we check in polynomial time if it verifies each rule in Σ . The number of rule being finite, datalog-evaluation is in P .

2 Exercise 2

When we use combined complexity, we need to take into account the size of q . Thus, Datalog evaluation is EXPTIME.

We will reduce from finite Halt problem $P(T, I, N)$ which sees if T accepts I in N steps. We take here $N = 2^{|I|^k}$. $|I|$ is the number of database constants.

We represent the step of the computation in binary over $|I|^k$ digits.

We define predicates for successorship ($x + 1 = y$), equality with 0 and $2^i - 1$ for numbers encoded on i bits by induction on i depending on the bit we add in the head of our numbers.

We can thus define an order from the successorship relation on n^k bits thus we get a way to make sure the transitions are consistent : from step i we can only get to step $i + 1$.

We can encode our Turing Machine on an input $a_1 \dots a_n$. We write B_i the encoding of i on k digits in base n : $\text{state}_q(X)$ is the predicate *the machine is in state q after X steps*, $\text{symbol}_\sigma(X, Y)$ is the predicate *the tape cell at position Y holds σ after X steps*, $\text{head}(X, Y)$ is the predicate *the reading head is at position Y after X steps*

$$\begin{aligned} & \text{state}_{q_0}(B_0) && \text{where } q_0 \text{ is initial} \\ & \text{head}(B_0, B_0) \\ & \text{symbol}_{\sigma_i}(B_0, B_i) && \forall i \in \{1, \dots, n\} \\ & \text{symbol}_-(B_0, Y) \leftarrow \leq^{n^k} (B_{n+1}, Y) \end{aligned}$$

For each transition $\langle q, \sigma, q', \sigma', d \rangle$ we add rules of the form :

$$\text{symbol}_{\sigma'}(X, Y) \leftarrow \text{succ}^{n^k}(X, X') \wedge \text{head}(X, Y) \wedge \text{symbol}_\sigma(X, Y) \wedge \text{state}_q(X)$$

$$\text{state}_{q'}(X') \leftarrow \text{succ}^{n^k}(X, X') \wedge \text{head}(X, Y) \wedge \text{symbol}_\sigma(X, Y) \wedge \text{state}_q(X)$$

We use similar rules depending on the direction d to set the new position of the head.

We also need to add rules for preservation of unaltered cells :

$$\text{symbol}_\sigma(X', Y) \leftarrow \text{succ}^{n^k}(X, X') \wedge \text{symbol}_\sigma(X, Y) \wedge \text{head}(X, Z) \wedge \text{succ}^{n^k}(Z, Z') \wedge \leq^{n^k}(Z', Y)$$

$$\text{symbol}_\sigma(X', Y) \leftarrow \text{succ}^{n^k}(X, X') \wedge \text{symbol}_\sigma(X, Y) \wedge \text{head}(X, Z) \wedge \text{succ}^{n^k}(Z, Z') \wedge \leq^{n^k}(Y, Z')$$

Finally, the machine accepts iff there is $X \leq 2^{n^k}$ such that $\text{state}_{q_{acc}}(X)$.

3 Exercise 3

- We rewrite $\{C\} \rightarrow \{B\}$ as an EGD :

$$\forall a, a', b, b', c, d_1, d_2, e, e', f, f', R(a, b, c, d_1, e, f) \wedge R(a', b', c, d_2, e', f') \Rightarrow d_1 = d_2$$

- We rewrite $\{D, E\} \rightarrow \{F\}$ as an EGD :

$$\forall a, a', b, b', c, c', d, e, f_1, f_2, R(a, b, c, d, e, f_1) \wedge R(a', b', c', d, e, f_2) \rightarrow f_1 = f_2$$

- We rewrite $\{A\} \twoheadrightarrow \{C, F\}$ as a TGD :

$$\forall a, b, b', c_1, c_2, d, d', e, e', f_1, f_2, (R(a, b, c_1, d, e, f_1) \wedge R(a, b', c_2, d', e', f_2) \rightarrow R(a, b', c_1, d', e', f_1))$$

4 Exercise 4

We need to show that (a, b, c, d, e, f) is in the database knowing that (a, b, c, d', e', f') and (a, b', c', d, e, f) are in the database.

We have the table :

A	B	C	D	E	F
a	b	c	d'	e'	f'
a	b'	c'	d	e	f

We use the third rule :

A	B	C	D	E	F
a	b	c	d'	e'	f'
a	b'	c'	d	e	f
a	b	c'	d'	e'	f
a	b'	c	d	e	f'

We use the second rule :

A	B	C	D	E	F
a	b	c	d'	e'	f'
a	b'	c'	d	e	f
a	b	c'	d'	e'	f'
a	b'	c	d	e	f

Then, using the first rule :

A	B	C	D	E	F
a	b	c	d'	e'	f'
a	b'	c'	d	e	f
a	b'	c'	d'	e'	f'
a	b	c	d	e	f

Thus, the decomposition is lossless

5 Exercise 5

If Σ is datalog, there are no existentially quantified variables. Thus, there are no special edges in the graph. Thus, there is no cycle featuring a special edge in its graph.

6 Exercise 6

Comme on considère un ensemble de TDG (en particulier sans EGD), alors à chaque application l'ensemble des relations ne fait que grandir au sens de l'inclusion, ce qui fait que la même TDG ne pourra pas être appliquée deux fois avec le même homéomorphisme.

7 Exercise 7

On commence par remarquer qu'il n'y a que des arcs normaux dans la composante fortement connexe car autrement on aurait un cycle comportant un arc spécial, ce qui contredirait l'hypothèse de faiblement acyclique. (Pas fini)

8 Exercise 8

On commence par décomposer le graphe de l'ensemble des TDG en composantes fortement connexes. On obtient donc une structure de graphe acyclique, dans lequel les nœuds sont des composantes fortement connexes. D'après la question précédente, les nœuds ne possédant pas de prédécesseur ont un domaine actif borné. Toujours en appliquant le résultat de la question précédente par récurrence dans l'ordre d'un tri topologique de notre graphe, on montre que l'ensemble des nœuds ont un domaine actif borné. Comme chaque application d'une TDG augmente strictement la taille de notre base de donnée, on en déduit que la chase termine en un temps fini.