Courcelle's Theorem

Home assignment to hand in before or on October 10, 2017.

October	25	26	27	28	29	30	
							1
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15

Electronic versions (PDF only) can be sent by email to ⟨sylvain.schmitz@lsv.fr⟩; paper versions should be handed in on the 10th or put in my mailbox at LSV, ENS Paris-Saclay. **No delays**. The numbers in the margins next to exercises are indications of time and difficulty, not necessarily of the points you might earn answering them.

We show in this homework a landmark result in graph algorithmics, where monadic second-order logic and tree automata play a central role, namely *Courcelle's Theorem.*

Theorem 1 (Courcelle's Theorem)**.** *Fix a counting MSO sentence φ on graphs and a natural number $k > 0$. The following problem can be solved in linear time $O(f(|\varphi|, k)\cdot|G|)$ for some computable function f:*

input: *A graph G of treewidth at most k.*

question: *Is G a model of φ, i.e. $G \models \varphi$?*

Importantly, neither k nor φ are part of the input in the above problem, ensuring $f(k, |\varphi|)$ is a constant. Put differently, Courcelle's Theorem proves that counting MSO model-checking on graphs is *fixed parameter tractable* (FPT), where the formula and treewidth are taken as parameters.

The properties and algorithmics of treewidth are studied in more details in MPRI course 2.29.1 *Graph Algorithms*, while results closely related to Courcelle's Theorem are applied to the modelling and verification of concurrent and distributed systems in MPRI course 2.8.1 *Non-Sequential Theory of Distributed Systems*.

1 Counting MSO on Ranked Trees

Counting MSO (CMSO) is an extension of monadic second-order logic, where we can furthermore measure the cardinality of sets quantified by second-order variables.

Syntax. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two infinite countable disjoint sets of first-order and second-order variables. The set of *CMSO formulæ on trees* over a ranked alphabet $\mathcal{F}$ is defined by the abstract syntax

$$\psi ::= x \downarrow_i x' \mid P_f(x) \mid x = x' \mid x \in X \mid \mathrm{card}_{q,r}(X) \mid \neg\psi \mid \psi \wedge \psi \mid \exists x.\psi \mid \exists X.\psi$$

where $1 \leq i \leq m$, $f \in \mathcal{F}$, $x, x' \in \mathcal{X}_1$, $X \in \mathcal{X}_2$, and $q > r \geq 0$. Free variables are defined as usual; a *sentence* is a formula without free variables. The *size* of a CMSO formula is its term size, with q, r encoded in unary.

Semantics. Given a tree $t \in T(\mathcal{F})$ (seen as a function from a set of positions $\mathrm{Pos}(t) \subseteq \mathbb{N}^*_{>0}$ to $\mathcal{F}$), and two valuations $\nu_1 \colon \mathcal{X}_1 \to \mathrm{Pos}(t)$ and $\nu_2 \colon \mathcal{X}_2 \to 2^{\mathrm{Pos}(t)}$, we say that t *satisfies* ψ and write $t \models_{\nu_1,\nu_2} \psi$ in the following situations: first, the relations specific to trees in $T(\mathcal{F})$:

$$\begin{aligned}
&t \models_{\nu_1,\nu_2} x \downarrow_i x' && \text{if } \nu_1(x') = \nu_1(x)\, i \;, \\
&t \models_{\nu_1,\nu_2} P_f(x) && \text{if } t(\nu_1(x)) = f \;,
\end{aligned}$$

then the usual MSO constucts:

$$\begin{aligned}
&t \models_{\nu_1,\nu_2} x = x' && \text{if } \nu_1(x) = \nu_1(x') \;, \\
&t \models_{\nu_1,\nu_2} x \in X && \text{if } \nu_1(x) \in \nu_2(X) \;, \\
&t \models_{\nu_1,\nu_2} \neg\psi && \text{if } t \not\models_{\nu_1,\nu_2} \psi \;, \\
&t \models_{\nu_1,\nu_2} \psi \wedge \psi' && \text{if } t \models_{\nu_1,\nu_2} \psi \text{ and } t \models_{\nu_1,\nu_2} \psi' \;, \\
&t \models_{\nu_1,\nu_2} \exists x.\psi && \text{if } \exists p \in \mathrm{Pos}(t),\ t \models_{\nu_1[x\mapsto p],\nu_2} \psi \;, \\
&t \models_{\nu_1,\nu_2} \exists X.\psi && \text{if } \exists P \subseteq \mathrm{Pos}(t),\ t \models_{\nu_1,\nu_2[X\mapsto P]} \psi \;,
\end{aligned}$$

and finally the counting predicates:

$$t \models_{\nu_1,\nu_2} \mathrm{card}_{q,r}(X) \qquad \text{if } |\nu_2(X)| \equiv r \bmod q \;.$$

When ψ is a sentence, satisfaction does not depend on the valuations ν_1 and ν_2, and we write more simply $t \models \psi$.

Exercise 1 (From CMSO to NFTA)**.** The inductive construction of an NFTA $\mathcal{A}_\psi$ for an MSO formula ψ seen in class can be extended to handle CMSO formulæ as well. We only need to consider an additional base case for a CMSO formula $\psi \stackrel{\text{def}}{=} \mathrm{card}_{q,r}(X)$ with a single free second-order variable $X \in \mathcal{X}_2$.

[1] 1. Show how to construct an NFTA $\mathcal{A}_{\mathrm{card}_{q,r}(X)}$ over the ranked alphabet $\mathcal{F} \times \{0,1\}$ where each $(f^{(n)}, b) \in \mathcal{F}_n \times \{0,1\}$ has arity n, such that

$$t \in L(\mathcal{A}_{\mathrm{card}_{q,r}(X)}) \quad \text{if and only if} \quad |\{p \in \mathrm{Pos}(t) : \pi_2(t(p)) = 1\}| \equiv r \bmod q \quad (*)$$

where 'π_2' denotes the projection $\mathcal{F} \times \{0,1\} \to \{0,1\}$.

[2] 2. Assume $\mathcal{F}$ contains at least one constant and one symbol of arity greater than 0. Show that any NFTA satisfying $(*)$ must have at least q states.

Using the constructions seen in class and the previous questions, we obtain an algorithm for constructing NFTA from CMSO formulæ:

Fact 1. *Let ψ be a CMSO sentence over $T(\mathcal{F})$. We can construct an NFTA $\mathcal{A}_\psi$ of size $g(|\psi|)$ for some computable function g such that, for all $t \in T(\mathcal{F})$, $t \models \psi$ if and only if $t \in L(\mathcal{A}_\psi)$.*

Exercise 2 (Relations in CMSO)**.** Let $\mathcal{F}$ be a finite ranked alphabet. Consider a CMSO formula $\psi(x_1, \ldots, x_r)$ with r free first-order variables (and no other free variable). It defines an r-ary relation on the positions of a tree $t \in T(\mathcal{F})$:

$$[\![\psi]\!]_t \stackrel{\text{def}}{=} \{(p_1, \ldots, p_r) \in (\mathrm{Pos}(t))^r : t \models_{\nu_1[x_1 \mapsto p_1, \ldots, x_r \mapsto p_r], \nu_2} \psi(x_1, \ldots, x_r)\} .$$

[2] 1. Let $\psi(x_1, x_2)$ be a CMSO formula. Define a CMSO formula $\mathrm{tc}_\psi(z_1, z_2)$ such that for all $t \in T(\mathcal{F})$, $[\![\mathrm{tc}_\psi]\!]_t = [\![\psi]\!]_t^+$ the transitive closure of $[\![\psi]\!]_t$.

[3] 2. The *document order* $\ll$ on a tree $t \in T(\mathcal{F})$ is the smallest transitive relation on $\mathrm{Pos}(t)$ such that $p \ll p\,i$ for all $i \in \mathbb{N}_{>0}$ and $p\,i\,p' \ll p\,j$ for all $i < j \in \mathbb{N}_{>0}$ and $p' \in \mathbb{N}_{>0}^*$.

   (a) Show that $\ll$ is a strict total order on $\mathrm{Pos}(t)$.

   (b) Provide a CMSO formula $x_1 \ll x_2$ with $[\![\ll]\!]_t = \ll$ for all $t \in T(\mathcal{F})$.

2 Treewidth

We consider in this assignment (finite undirected simple) graphs $G = (V, E)$, defined by a finite set V of vertices and a symmetric irreflexive set $E \subseteq V \times V$ of edges. We shall use a definition of treewidth that is easier to manipulate in our tree setting, based on a graph algebra.

Sourced Graphs. Let $k > 0$ and $\mathcal{Y}_k \stackrel{\text{def}}{=} \{y_1, \ldots, y_{2k}\}$ be a set of $2k$ *sources.* A *k-sourced graph* (V, E, s) is a finite graph (V, E) together with an injective partial function $s\colon \mathcal{Y}_k \to V$ with a domain of cardinal $|\mathrm{dom}\ s| \leq k$ (see Figure 1 for an example, where the vertices in the range $\mathrm{rng}\ s$ of s appear in red); a graph can be seen as a sourced graph where s has an empty domain.

Fusion. Given two k-sourced graphs $G = (V, E, s)$ and $G' = (V', E', s')$ and a subset $Y \subseteq \mathcal{Y}_k$ of cardinal $|Y| \leq k$, their *Y-fusion* $G \oplus_Y G'$ is a k-sourced graph where

1. the vertices of G and G' with the same sources are identified, and
2. we then forget the sources from Y; these remain as plain vertices.

An example of a fusion is displayed in Figure 2.

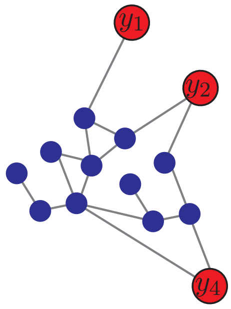

Figure 1: A 3-sourced graph with $\mathrm{dom}\, s = \{y_1, y_2, y_4\}$.

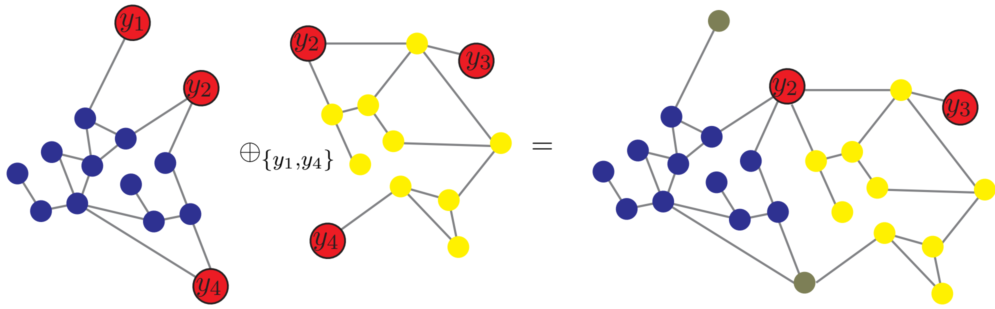

Figure 2: Example of a fusion; here $F = \{y_2, y_4\}$ and the forgotten $\{y_1, y_4\}$ appear in khaki.

In order to define this formally, we consider the intersection $F \stackrel{\text{def}}{=} (\mathrm{dom}\, s) \cap (\mathrm{dom}\, s')$ of the domains of s, s'; then $G \oplus_Y G' = (V'', E'', s'')$ where

$$
\begin{aligned}
V'' &\stackrel{\text{def}}{=} V \uplus (V' \setminus s'(F)) ,\\
E'' &\stackrel{\text{def}}{=} E \cup \{(v, v'), (v', v) : ((v, v') \in E' \text{ and } v' \in (V' \setminus s'(F)))\\
&\qquad\qquad\qquad\qquad \text{or } (\exists y \in F \,.\, (v, s'(y)) \in E' \text{ and } v' = s(y)\} ,\\
\mathrm{dom}\, s'' &\stackrel{\text{def}}{=} ((\mathrm{dom}\, s) \cup (\mathrm{dom}\, s')) \setminus Y ,\\
s''(y) &\stackrel{\text{def}}{=} \begin{cases} s(y) & \text{if } y \in (\mathrm{dom}\, s) \setminus Y \\ s'(y) & \text{if } y \in (\mathrm{dom}\, s) \setminus (F \cup Y) . \end{cases}
\end{aligned}
$$

Exercise 3 (Graph Algebra)**.** Let $k > 0$. We let $\mathcal{B}_k \stackrel{\text{def}}{=} \{(V, E, s)\ k\text{-sourced graph} : |V| \leq k + 1\}$ denote the set of k-sourced graphs of size at most $k + 1$; those graphs are called *bags* and this is a finite set up to isomorphism for every fixed k. We define the finite ranked alphabet $\Sigma_k \stackrel{\text{def}}{=} \mathcal{B}_k \cup \{\oplus_Y : Y \subseteq \mathcal{Y}_k \text{ and } |Y| \leq k\}$, where the bags of $\mathcal{B}_k$ are treated as atomic symbols of arity 0 and the '$\oplus_Y$' symbols have arity 2.

A term $t \in T(\Sigma_k)$ denotes a k-sourced graph $\gamma(t)$ defined by $\gamma(B) \stackrel{\text{def}}{=} B$ for all $B \in \mathcal{B}_k$ and $\gamma(t_1 \oplus_Y t_2) \stackrel{\text{def}}{=} \gamma(t_1) \oplus_Y \gamma(t_2)$. A graph G has *treewidth* k if k is minimal such that $G = \gamma(t)$ for some term $t \in \Sigma_k$. Clearly, a graph $G = (V, E)$ has treewidth at most $|V|$.

[1] 1. Let $k = 2$; what is the graph denoted by the term $(B_1 \oplus_{\{y_2\}} B_2) \oplus_{\{y_1,y_3\}} (B_1 \oplus_{\{y_2\}} B_2)$, where B_1 and B_2 are displayed in Figure 3?

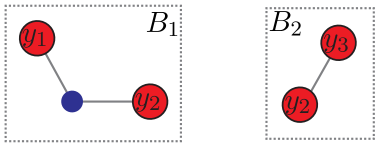

Figure 3: Two 2-sourced bags for Exercise 3.1.

[3] 2. A graph (V, E) in which $V \neq \emptyset$ and any two vertices are connected by exactly one path is *tree-shaped*. Show that any tree-shaped graph can be denoted by a term over Σ_1. Is any graph denoted by a term over Σ_1 tree-shaped?

Exercise 4 (CMSO on $T(\Sigma_k)$)**.** The aim of this exercise is to write CMSO formulæ on terms in $T(\Sigma_k)$, which denote interesting properties of the denoted graph.

Notations. Observe that, up to isomorphism, the vertex set of a bag in $\mathcal{B}_k$ can be taken as a subset of $\{0, \ldots, k\}$. We are going to see the vertex set V and the edge set E in any bag (V, E, s) from $\mathcal{B}_k$ as predicates: for all $0 \leq i, j \leq k$, $V(i)$ holds if the vertex is defined and $E(i, j)$ holds if $V(i)$ and $V(j)$ hold and the edge is defined.

[1] 1. For $0 \leq i, j \leq k$ and $1 \leq n \leq 2k$, define CMSO formulæ $V_i(x)$ (resp. $E_{i,j}(x)$, resp. $Y_{i,n}(x)$) such that, for all $t \in T(\mathcal{F})$, $p \in [\![V_i]\!]_t$ (resp. $p \in [\![E_{i,j}]\!]_t$, resp. $p \in [\![Y_{i,n}]\!]_t$) if and only if p is labelled by a bag $B = (V, E, s)$ where $V(i)$ (resp. $E(i, j)$, resp. $s(y_n) = i$) holds.

2. The idea in the following will be to identify each vertex in $\gamma(t)$ by a *representative*: a pair (p, i) where $V(i)$ holds in the bag labelling position $p \in \mathrm{Pos}(t)$. We denote by $\gamma(p, i)$ the vertex in $\gamma(t)$ represented by (p, i).

   An issue with this idea is that two representatives (p, i) and (p', j) might identify the same vertex of $\gamma(t)$. For instance, if $B \stackrel{\text{def}}{=} (\{0, 1\}, \{(0, 1), (1, 0)\}, \{y_1 \mapsto 0\})$, then $t \stackrel{\text{def}}{=} B \oplus_{\{y_1\}} B$ denotes a path of length three, whose middle vertex occurs at index 0 in both copies of B (at positions 1 and 2) in t; formally $\gamma(1, 0) = \gamma(2, 0)$.

[3] For $0 \leq i, j \leq k$, define a CMSO formula $\mathrm{eq}_{i,j}(x_1, x_2)$ with two free first-order variables such that, for all $t \in T(\mathcal{F})$, $(p, p') \in [\![\mathrm{eq}_{i,j}]\!]_t$ if and only if $\gamma(p, i) = \gamma(p', j)$.

[1] 3. For $0 \leq i, j \leq k$, define a CMSO formula $e_{i,j}(x_1, x_2)$ such that, for all $t \in T(\mathcal{F})$, $(p, p') \in [\![e_{i,j}]\!]_t$ if and only if there is an edge in the resulting graph $\gamma(t)$ between $\gamma(p, i)$ and $\gamma(p', j)$.

4. Given a vertex of $\gamma(t)$, we want to choose a *canonical* representative (p, i), where p is chosen minimal with respect to the document order $\ll$ among all the positions p' such that $\gamma(p, i) = \gamma(p', j)$ for some $0 \leq j \leq k$.

[1] Define a CMSO formula $\text{canonical}_i(x)$ such that, for all $t \in T(\mathcal{F})$, $p \in [\![\text{canonical}_i]\!]_t$ if and only if (p, i) is a canonical representative in $\gamma(t)$.

Computing Treewidths. Although deciding whether the treewidth of a graph is at most k is NP-complete when k is part of the input, when k is considered as *fixed* this can be checked in linear time and a term in Σ_k can be computed:

Fact 2. *Let $k > 0$. If G has treewidth at most k, then a term $t \in T(\Sigma_k)$ denoting $G = \gamma(t)$ can be computed in time $O(h(k) \cdot |G|)$ for some computable function h.*

3 Counting MSO on Graphs

Syntax. The set of CMSO formulæ on graphs is defined by the abstract syntax

$$\varphi ::= e(x, x') \mid x = x' \mid x \in X \mid \text{card}_{q,r}(X) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi \mid \exists X.\varphi$$

where $x, x' \in \mathcal{X}_1$, $X \in \mathcal{X}_2$, and $q, r \in \mathbb{N}$.

Semantics. The semantics on graphs $G = (V, E)$ are similar to those on trees, with a binary relation 'e' denoting the edge relation; now $\nu_1 \colon \mathcal{X}_1 \to V$ and $\nu_2 \colon \mathcal{X}_2 \to 2^V$ and

$$G \models_{\nu_1,\nu_2} e(x, x') \qquad\qquad \text{if } (\nu_1(x), \nu_1(x')) \in E\ .$$

Exercise 5 (Graph Properties in CMSO)**.** Let us get acquainted with CMSO on graphs.

[1] 1. Define a CMSO sentence φ_{3c} such that $G \models \varphi_{3c}$ if and only if G is 3-colourable.

[1] 2. Define a CMSO sentence φ_{Ec} such that $G \models \varphi_{Ec}$ if and only if G has an Eulerian cycle.

Exercise 6 (Interpreting Graphs in Trees)**.** The aim of this objective is to construct from a CMSO formulæ φ on graphs a CMSO sentence ψ on trees in $T(\Sigma_k)$, which encodes the same property.

We shall proceed by induction on φ. The translation has to handle CMSO formulæ with free variables, and the correction of the translation will need to translate between valuations in graphs and in trees.

first-order variables: we manipulate in ψ the *representatives* defined in Exercise 4: in order to represent a first-order variable from φ in a tree t, we need in ψ both a first-order variable ranging over positions of t and an index $0 \leq i \leq k$. The translation therefore maintains a *variable index* $I \colon \text{fv}_1(\psi) \to \{0, \ldots, k\}$, which gives the index associated to each free first-order variable of ψ.

second-order variables: we also use representatives, and each variable $X \in \mathcal{X}_2$ of φ is encoded as $k+1$ second-order variables $X_0, \ldots, X_k$ in ψ, such that a representative (p, i) for p in the valuation of X_i will stand for a vertex in the valuation of X.

The outcome of this exercise is a family of translations $\Psi_I(\varphi)$ such that, assuming $G = \gamma(t)$,

$$G \models_{\nu_1',\nu_2'} \varphi \quad \text{if and only if} \quad t \models_{\nu_1,\nu_2} \Psi_I(\varphi) \ , \tag{†}$$

where

$$\nu_1'(x) \stackrel{\text{def}}{=} \gamma(\nu_1(x), I(x)) \ , \qquad \nu_2'(X) \stackrel{\text{def}}{=} \{\gamma(p,i) : 0 \leq i \leq k \text{ and } p \in \nu_2(X_i)\} \ . \tag{‡}$$

To give you a taste of the translation, here are a few cases:

$$\begin{aligned}
\Psi_I(e(x_1,x_2)) &\stackrel{\text{def}}{=} e_{I(x_1),I(x_2)}(x_1,x_2) \ , && \text{(using } e_{i,j} \text{ from Exercise 4.3)}\\
\Psi_I(\neg\varphi) &\stackrel{\text{def}}{=} \neg\Psi_I(\varphi) \ ,\\
\Psi_I(\varphi \wedge \varphi') &\stackrel{\text{def}}{=} \Psi_I(\varphi) \wedge \Psi_I(\varphi') \ ,\\
\Psi_I(\exists x.\varphi) &\stackrel{\text{def}}{=} \exists x. \bigvee_{0\leq i\leq k} V_i(x) \wedge \Psi_{I[x\mapsto i]}(\varphi) \ .
\end{aligned}$$

[1] 1. Define $\Psi_I(x_1 = x_2)$ and prove it correct.

[2] 2. Define $\Psi_I(\exists X.\varphi)$ and $\Psi_I(\mathrm{card}_{q,r}(X))$ and prove it correct, assuming by induction hypothesis that $\Psi_I(\varphi)$ is correct. *Hint: You need to ensure* $|\nu_2'(X)| = \sum_{0\leq i\leq k} |\nu_2(X_i)|$.

[1] 3. Define $\Psi_I(x \in X)$ and prove it correct.

Exercise 7 (Proof of Courcelle's Theorem)**.** We arrive at last to the proof of Courcelle's Theorem. Let us fix $k > 0$ and a CMSO sentence φ on graphs.

Algorithm. We perform a number of pre-treatments:

Step 1. Construct a CMSO sentence $\Psi(\varphi)$ such that, for all graphs G and all terms $t \in T(\Sigma_k)$ with $G = \gamma(t)$, $G \models \varphi$ if and only if $t \models \Psi(\varphi)$, using Exercise 6.

Step 2. Construct a NFTA $\mathcal{A}_{\Psi(\varphi)}$ such that for all $t \in T(\Sigma_k)$, $t \models \Psi(\varphi)$ if and only if $t \in L(\mathcal{A}_{\Psi(\varphi)})$, using Fact 1.

Step 3. Determinise the NFTA $\mathcal{A}_{\Psi(\varphi)}$, yielding an equivalent DFTA $\mathcal{A}^d_{\Psi(\varphi)}$.

The algorithm proving Courcelle's Theorem then processes its input graph G:

Step 4. Compute a term $t \in T(\Sigma_k)$ with $\gamma(t) = G$ using Fact 2.

Step 5. Check whether $t \in L(\mathcal{A}^d_{\Psi(\varphi)})$, which is an instance of the Membership Problem for DFTA.

[1] Complete the proof by justifying the correctness of the algorithm and the complexity statement of Theorem 1.