Courcelle’s Theorem Made Dynamic

Context: Given a decision problem, at what cost can we update our decision when one bit of the problem input is modified?

Dynamic complexity class: If precomputing auxiliary data in $C$ helps us treating input updates in $C'$, we say that the dynamic problem is in $\text{Dyn}(C|C')$.

Example: Reachability in acyclic graphs is in $\text{Dyn}(\text{NL},\text{FO})$ [5]

Decision problem: Given two vertices $s,t$ of an acyclic graph $G = (V, E)$, does there exist a path from $s$ to $t$ in $G$?

Input updates: Edge deletion or insertion (without creating cycles)

Auxiliary predicate: $R(x,y)$ = “There exists a path from $x$ to $y$”.

Courcelle’s theorem

Ingredients: A graph $G = (V, E)$, a tree-decomposition $D$ of width $\kappa$ of $G$, a succinct encoding $\mathcal{D}$ of $D$ and an MSO formula $\varphi$.

Tree-decomposition of width $\kappa$ of $G$: Pair $D = (T, \text{bag})$, where $T = (N,E)$ is an ordered binary tree and $\text{bag}$ is a mapping $N \rightarrow 2^V$ such that:

1. for each vertex $v \in V$, the set $\{ n \in N \mid v \in \text{bag}(n) \}$ is connected and non-empty;
2. for each edge $e = (v_1, v_2) \in E$, the set $\{ n \in N \mid (v_1, v_2) \subseteq \text{bag}(n) \}$ is non-empty;
3. for each node $n \in N$, the set $\text{bag}(n)$ is of cardinality at most $\kappa + 1$.

Succinct encoding of $D$: Triple $(\chi, \lambda, \pi)$, where $\lambda : V \rightarrow \{ 0, \ldots, \kappa \}$, $\chi : N \rightarrow 2^{V \setminus \chi(n)} \times 2^{V \setminus \lambda(n)}$ are mappings such that, for each node $n \in N$:

1. the restriction of $\chi$ to $\text{bag}(n)$ is injective (hence $\chi$ is a proper coloring of $G$);
2. $\chi(n) = \{ (v) : v \in \text{bag}(n) \}$, where $\text{bag}(n) = \text{bag}(n') \cap \text{bag}(n)$ if $n'$ is $n$'s parent; and, for each node $n \in N$,
3. $\lambda(n) = \{ (v, e) : (v, e) \in E \cap \text{bag}(n') \}$.

Labeling every node $n$ in $T$ with the pair $(\lambda(n), \chi(n))$ gives a succinctly encoded tree-decomposition of $G$.

Example: Succinctly encoded tree-decomposition of width 2

MSO formula: Formula over graphs with quantification on (sets of) edges and vertices

Example: The graph $G$ is strongly connected iff $G$ satisfies the formula

$$\varphi \equiv \forall X \subseteq V \forall x, y \in V, x \neq y \in X \forall y \in X \forall \exists u, v \in V. E(u, v) \land u \notin X \land v \notin X$$

Theorem statement [5]

Given an integer $\kappa$ and an MSO formula $\varphi$, there exists a tree automaton $\mathcal{A}_\varphi$ such that, for all graphs $G$ and all succinctly encoded tree-decompositions $T$ of width $\kappa$ of $G$,

$G$ satisfies $\varphi$ iff $\mathcal{A}_\varphi$ accepts $T$ succinct.

Sequentially simulating runs of tree automata

Context: Bottom-up, deterministic automata perform computations in a distributed way. How can we simulate them on a single (sequential) computation thread?

Tree automata and distributed computation: The run of the tree automaton $A = (\Sigma, Q, \delta, \ell, f)$ on a labeled tree $T = (\mathcal{N}, \mathcal{E}, \Sigma)$ is the mapping $\rho : \mathcal{N} \rightarrow \mathcal{Q}$ such that:

1. $\rho(n) = \delta(n, \ell(n), \lambda(n))$ for all leaves $n$ with label $\lambda(n) \in \Sigma$;
2. $\rho(n) = \delta(n, \rho(m_1), \rho(m_2))$ for all nodes $n$ with label $\lambda(n)$ and children $m_1$ and $m_2$.

The automaton $A$ accepts the tree $T$, with root $r$, iff $\rho(r) \in F$.

Slicing $T$: Choose subsets $S_0, \ldots, S_k$ of $\mathcal{N}$ such that $S_0 = \emptyset$, $S_k = \{ r \}$ and, for $k \geq 1$:

1. there is a unique node $n_0 \in S_0 \setminus S_{k-1}$;
2. its children (if any) belong to $S_{k-1}$.

Sequential simulation: Compute the run restrictions $\rho|_{S_k}$ for $0 \leq k \leq \ell$:

1. the initial restriction $\rho|_{S_0}$ is fixed;
2. $\rho|_{S_k}$ determines whether $A$ accepts $T$;
3. $\rho|_{S_k}$, depends on $\rho|_{S_k}$ and $\lambda(n_{k+1})$ only: we set $\rho|_{S_k} = \Pi_1(\rho|_{S_k}, \lambda(n_{k+1}))$.

Sequential computations vs Dyck-path reachability

Dyck words: Well-parenthesized words (with multiple kinds of parentheses)

Dyck paths in a labeled graph: Paths whose labels are Dyck words

Example: There are 7 Dyck paths in this graph. Will you find them all?

Making Courcelle’s theorem dynamic

Using two more ingredients in addition to the above constructions:

1. Computing logarithmic-depth tree-decompositions of width $4\epsilon + 3$ in $\ell$ [2, 4];
2. Solving Dyck-path reachability problems in acyclic graphs in $\text{Dyn}(\text{LogCFL},\text{FO})$ [6].

Dynamic Courcelle’s theorem statement [1]

Let $\kappa$ and $\varphi$ be fixed. Given a maximal graph $G = (V, E)$ of tree-width $\kappa$, an initial subgraph $G_0 = (V, E)$ with $E \subseteq E^*$, and updating $G$ by adding/deleting edges $e \in E^*$ checking whether $G$ satisfies $\varphi$ is feasible in $\text{Dyn}(L,\text{FO})$.

References