There is a computable function $f$ such that the problem of deciding, given a word $w$ and an MSO sentence $\phi$ whether,

$$S_w \models \phi$$

can be decided in time $O(f(l)n)$ where $l$ is the length of $\phi$ and $n$ is the length of $w$.

The algorithm proceeds by constructing, from $\phi$ an automaton $A_\phi$ such that the language recognized by $A_\phi$ is

$$\{ w \mid S_w \models \phi \}$$

then running $A_\phi$ on $w$.

**The automaton $A_\phi$**

The states of $A_\phi$ are the equivalence classes of $\equiv_{m}^{\text{MSO}}$ where $m$ is the quantifier rank of $\phi$.

We write $\text{Type}_{m}^{\text{MSO}}(A)$ for the set of all formulas $\phi$ with $\text{qr}(\phi) \leq m$ such that $A \models \phi$.

$A \equiv_{m}^{\text{MSO}} B$ is equivalent to

$$\text{Type}_{m}^{\text{MSO}}(A) = \text{Type}_{m}^{\text{MSO}}(B)$$

There is a single formula $\theta_A$ that characterizes $\text{Type}_{m}^{\text{MSO}}(A)$.

It turns out that we can compute $\theta_{S_w \cdot a}$ from $\theta_{S_w}$.

**Trees**

An (undirected) forest is an acyclic graph and a tree is a connected forest.

We next aim to show that there is an algorithm that decides, given a tree $T$ and an MSO sentence $\phi$ whether

$$T \models \phi$$

and runs in time $O(f(l)n)$ where $l$ is the length of $\phi$ and $n$ is the size of $T$. 
Rooted Directed Trees

A rooted, directed tree \((T, a)\) is a directed graph with a distinguished vertex \(a\) such that for every vertex \(v\) there is a unique directed path from \(a\) to \(v\).

We will actually see that \(\text{MSO}\) satisfaction is \(\text{FPT}\) for rooted, directed trees.

The result for undirected trees follows, as any undirected tree can be turned into a rooted directed one by choosing any vertex as a root and directing edges away from it.

Sums of Rooted Trees

Given rooted, directed trees \((T, a)\) and \((S, b)\) we define the sum \((T, a) \oplus (S, b)\) to be the rooted directed tree which is obtained by taking the disjoint union of the two trees while identifying the roots.

That is,

- the set of vertices of \((T, a) \oplus (S, b)\) is \(V(T) \cup V(S) \setminus \{b\}\).
- the set of edges is \(E(T) \cup E(S) \cup \{(a, v) \mid (b, v) \in E(S)\}\).

Congruence

If \((T_1, a_1) \equiv_{\text{MSO}} (T_2, a_2)\) and \((S_1, b_1) \equiv_{\text{MSO}} (S_2, b_2)\), then

\((T_1, a_1) \oplus (S_1, b_1) \equiv_{\text{MSO}} (T_2, a_2) \oplus (S_2, b_2)\).

This can be proved by an application of Ehrenfeucht games.

Moreover (though we skip the proof), \(\text{Type}_{\text{MSO}}((T, a) \oplus (S, b))\) can be computed from \(\text{Type}_{\text{MSO}}((T, a))\) and \(\text{Type}_{\text{MSO}}((S, b))\).

Add Root

For any rooted, directed tree \((T, a)\) define \(r(T, a)\) to be rooted directed tree obtained by adding to \((T, a)\) a new vertex, which is the root and whose only child is \(a\).

That is,

- the vertices of \(r(T, a)\) are \(V(T) \cup \{a'\}\) where \(a'\) is not in \(V(T)\);
- the root of \(r(T, a)\) is \(a'\); and
- the edges of \(r(T, a)\) are \(E(T) \cup \{(a', a)\}\).

Again, \(\text{Type}_{\text{MSO}}(r(T, a))\) can be computed from \(\text{Type}_{\text{MSO}}(T, a)\).
**MSO satisfaction is FPT on Trees**

Any *rooted, directed tree* \((T, a)\) can be obtained from *singleton trees* by a sequence of applications of \(\oplus\) and \(r\).

The length of the sequence is linear in the size of \(T\).

We can compute \(\text{Type}_{\text{MSO}}^m(T, a)\) in linear time.

---

**The Method of Decompositions**

Suppose \(C\) is a class of graphs such that there is a finite class \(B\) and a finite collection \(\text{Op}\) of operations such that:

- \(C\) is contained in the closure of \(B\) under the operations in \(\text{Op}\);
- there is a polynomial-time algorithm which computes, for any \(G \in C\), an \(\text{Op}\)-decomposition of \(G\) over \(B\); and
- for each \(m\), the equivalence class \(\equiv_{m}^\text{MSO}\) is an *effective* congruence with respect to to all operations \(o \in \text{Op}\) (i.e., the \(\equiv_{m}^\text{MSO}\)-type of \(o(G_1, \ldots, G_s)\) can be computed from the \(\equiv_{m}^\text{MSO}\)-types of \(G_1, \ldots, G_s\)).

Then, \(\text{MSO}\) satisfaction is fixed-parameter tractable on \(C\).

---

**Treewidth**

The *treewidth* of an undirected graph is a measure of how tree-like the graph is.

A graph has treewidth \(k\) if it can be covered by subgraphs of at most \(k + 1\) nodes in a tree-like fashion.

This gives a *tree decomposition* of the graph.
Dynamic Programming

It has long been known that graphs of small treewidth admit efficient dynamic programming algorithms for intractable problems.

In general, these algorithms proceed bottom-up along a tree decomposition of $G$.

At any stage, a small set of vertices form the “interface” to the rest of the graph.

This allows a recursive decomposition of the problem.

Treewidth

Looking at the decomposition bottom-up, a graph of treewidth $k$ is obtained from graphs with at most $k + 1$ nodes through a finite sequence of applications of the operation of taking sums over sets of at most $k$ elements.

$$G_1 \oplus_X G_2$$

$|X| \leq k$

We let $T_k$ denote the class of graphs $G$ such that $\text{tw}(G) \leq k$.

Treewidth

More formally,

Consider graphs with up to $k + 1$ distinguished vertices $C = \{c_0, \ldots, c_k\}$.

Define a merge operation $(G \oplus_C H)$ that forms the union of $G$ and $H$ disjointly apart from $C$.

Also define erase$_i(G)$ that erases the name $c_i$.

Then a graph $G$ is in $T_k$ if it can be formed from graphs with at most $k + 1$ vertices through a sequence of such operations.

Congruence

• Any $G \in T_k$ is obtained from $B_k$ by finitely many applications of the operations erase$_i$ and $\oplus_C$.

• If $G_1, \rho_1 \equiv_m^{\text{MSO}} G_2, \rho_2$, then
  
  $$\text{erase}_i(G_1, \rho_1) \equiv_m^{\text{MSO}} \text{erase}_i(G_2, \rho_2)$$

• If $G_1, \rho_1 \equiv_m^{\text{MSO}} G_2, \rho_2$, and $H_1, \sigma_1 \equiv_m^{\text{MSO}} H_2, \sigma_2$ then
  
  $$(G_1, \rho_1) \oplus_C (H_1, \sigma_1) \equiv_m^{\text{MSO}} (G_2, \rho_2) \oplus_C (H_2, \sigma_2)$$

Note: a special case of this is that $\equiv_m^{\text{MSO}}$ is a congruence for disjoint union of graphs.
Courcelle’s Theorem

**Theorem (Courcelle)**

For any MSO sentence $\phi$ and any $k$ there is a linear time algorithm that decides, given $G \in T_k$ whether $G \models \phi$.

Given $G \in T_k$ and $\phi$, compute:
- from $G$ a labelled tree $T$; and
- from $\phi$ a bottom-up tree automaton $A$

such that $A$ accepts $T$ if, and only if, $G \models \phi$.

Bounded Degree Graphs

**Theorem (Seese)**

For every sentence $\phi$ of FO and every $k$ there is a linear time algorithm which, given a graph $G \in D_k$ determines whether $G \models \phi$.

A proof is based on locality of first-order logic.

To be precise a strengthening of Hanf’s theorem.

Note: this is not true for MSO unless P = NP.

Construct, for any graph $G$, a graph $G'$ such that $\Delta(G') \leq 5$ and $G'$ is 3-colourable if $G$ is, and the map $G \mapsto G'$ is polynomial-time computable.

Bounded Degree Graphs

In a graph $G = (V, E)$ the **degree** of a vertex $v \in V$ is the number of neighbours of $v$, i.e.

$|\{u \in V \mid (u, v) \in E\}|$.

We write $\delta(G)$ for the smallest degree of any vertex in $G$.

We write $\Delta(G)$ for the largest degree of any vertex in $G$.

$D_k$—the class of graphs $G$ with $\Delta(G) \leq k$.

Hanf Types

For an element $a$ in a structure $A$, define

$N_r^A(a)$—the substructure of $A$ generated by the elements whose distance from $a$ (in $GA$) is at most $r$.

We say $A$ and $B$ are Hanf equivalent with radius $r$ and threshold $q$ ($A \simeq_{r,q} B$) if, for every $a \in A$ the two sets

$\{a' \in a \mid N_r^A(a) \cong N_r^A(a')\}$ and $\{b \in B \mid N_r^B(a) \cong N_r^B(b)\}$

either have the same size or both have size greater than $q$;

and, similarly for every $b \in B$. 

**Hanf Locality Theorem**

**Theorem (Hanf)**
For every vocabulary $\sigma$ and every $m$ there are $r \leq 3^m$ and $q \leq m$ such that for any $\sigma$-structures $A$ and $B$: if $A \simeq_{r,q} B$ then $A \equiv_m B$.

In other words, if $r \geq 3^m$, the equivalence relation $\simeq_{r,m}$ is a refinement of $\equiv_m$.

For $A \in D_k$:
$N^A_r(a)$ has at most $kr + 1$ elements
each $\simeq_{r,m}$ has finite index.
Each $\simeq_{r,m}$-class $t$ can be characterised by a finite table, $I_t$, giving isomorphism types of neighbourhoods and numbers of their occurrences up to threshold $m$.

**Satisfaction on $D_k$**

For a sentence $\phi$ of FO, we can compute a set of tables $\{I_1, \ldots, I_s\}$ describing $\simeq_{r,m}$-classes consistent with it.

This computation is independent of any structure $A$.

Given a structure $A \in D_k$,

- for each $a$, determine the isomorphism type of $N^A_r(a)$
- construct the table describing the $\simeq_{r,m}$-class of $A$.
- compare against $\{I_1, \ldots, I_s\}$ to determine whether $A \models \phi$.

For fixed $k, r, m$, this requires time linear in the size of $A$.

**Note:** satisfaction for FO is in $O(f(l, k)n)$.

---

**Reading List for this Handout**

1. Libkin. Sections 7.6 and 7.6