Courcelle's Theorem, tree automata, hypergraphs, and matroids

Dillon Mayhew

Victoria University of Wellington
New Zealand

Joint work with Daryl Funk (Douglas College),
Mike Newman (University of Ottawa),
Geoff Whittle (Victoria University of Wellington).

Courcelle's Theorem

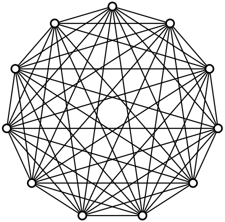
Courcelle's Theorem (1990)

Let φ be a sentence in MS_2, the monadic second-order logic of graphs.

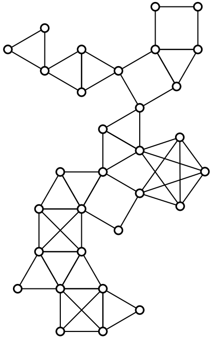
Let $\mathcal{G}$ be a class of graphs with bounded tree-width.

There is a polynomial-time algorithm that tests graphs in $\mathcal{G}$ and decides whether they satisfy φ.

Tree-width



High tree-width



Low tree-width

Monadic second-order logic

$$
\begin{aligned}
&\exists E_1\ \forall v\ \exists e_1\ \exists e_2\ (e_1 \in E_1 \wedge e_2 \in E_1 \wedge e_1 \neq e_2 \\
&\quad \wedge \mathrm{inc}(v, e_1) \wedge \mathrm{inc}(v, e_2) \\
&\qquad \wedge \forall e_3\ (e_3 \in E_1 \wedge e_3 \neq e_1 \wedge e_3 \neq e_2 \rightarrow \neg\, \mathrm{inc}(v, e_3))) \\
&\qquad\quad \wedge \forall V_1\ \forall V_2 \\
&\qquad\qquad (\exists v_1\ \exists v_2\ (v_1 \in V_1 \wedge v_2 \in V_2) \\
&\qquad\qquad\quad \wedge \forall v\ (v \in V_1 \vee v \in V_2 \wedge \neg(v \in V_1 \wedge v \in V_2))) \\
&\qquad\qquad\qquad \rightarrow \\
&\qquad\qquad\qquad\quad (\exists e\ (e \in E_1 \wedge \exists v_1\ \exists v_2 \\
&\qquad\qquad\qquad\qquad (v_1 \in V_1 \wedge v_2 \in V_2 \wedge \mathrm{inc}(v_1, e) \wedge \mathrm{inc}(v_2, e))))
\end{aligned}
$$

Monadic second-order logic

$$
\begin{aligned}
&\exists E_1\ \forall v\ \exists e_1\ \exists e_2\ (e_1 \in E_1 \wedge e_2 \in E_1 \wedge e_1 \neq e_2 \\
&\quad \wedge \mathrm{inc}(v, e_1) \wedge \mathrm{inc}(v, e_2) \\
&\quad\quad \wedge \forall e_3\ (e_3 \in E_1 \wedge e_3 \neq e_1 \wedge e_3 \neq e_2 \to \neg\, \mathrm{inc}(v, e_3))) \\
&\quad\quad\quad \wedge \forall V_1\ \forall V_2 \\
&\quad\quad\quad\quad (\exists v_1\ \exists v_2\ (v_1 \in V_1 \wedge v_2 \in V_2) \\
&\quad\quad\quad\quad\quad \wedge \forall v\ (v \in V_1 \vee v \in V_2 \wedge \neg(v \in V_1 \wedge v \in V_2))) \\
&\quad\quad\quad\quad\quad\quad \to \\
&\quad\quad\quad\quad\quad\quad\quad (\exists e\ (e \in E_1 \wedge \exists v_1\ \exists v_2 \\
&\quad\quad\quad\quad\quad\quad\quad\quad (v_1 \in V_1 \wedge v_2 \in V_2 \wedge \mathrm{inc}(v_1, e) \wedge \mathrm{inc}(v_2, e))))
\end{aligned}
$$

"There exists a set of edges, E_1, such that every vertex is incident with exactly two edges in E_1, and whenever (V_1, V_2) is a partition of the vertices, there is an edge in E_1 that is incident with vertices in both V_1 and V_2."

Monadic second-order logic

$$
\begin{aligned}
&\exists E_1\ \forall v\ \exists e_1\ \exists e_2\ (e_1 \in E_1 \wedge e_2 \in E_1 \wedge e_1 \neq e_2\\
&\quad \wedge \mathrm{inc}(v, e_1) \wedge \mathrm{inc}(v, e_2)\\
&\qquad \wedge \forall e_3\ (e_3 \in E_1 \wedge e_3 \neq e_1 \wedge e_3 \neq e_2 \rightarrow \neg\, \mathrm{inc}(v, e_3)))\\
&\qquad\quad \wedge \forall V_1\ \forall V_2\\
&\qquad\qquad (\exists v_1\ \exists v_2\ (v_1 \in V_1 \wedge v_2 \in V_2)\\
&\qquad\qquad\quad \wedge \forall v\ (v \in V_1 \vee v \in V_2 \wedge \neg(v \in V_1 \wedge v \in V_2)))\\
&\qquad\qquad\qquad \rightarrow\\
&\qquad\qquad\qquad\quad (\exists e\ (e \in E_1 \wedge \exists v_1\ \exists v_2\\
&\qquad\qquad\qquad\qquad (v_1 \in V_1 \wedge v_2 \in V_2 \wedge \mathrm{inc}(v_1, e) \wedge \mathrm{inc}(v_2, e))))
\end{aligned}
$$

In other words, the graph is hamiltonian.

Courcelle's Theorem

Courcelle's Theorem (1990)

Let φ be a sentence in MS_2, the monadic second-order logic of graphs.

Let $\mathcal{G}$ be a class of graphs with bounded tree-width.

There is a polynomial-time algorithm that tests graphs in $\mathcal{G}$ and decides whether they satisfy φ.

Courcelle's Theorem means that we can test intractable properties efficiently, if we limit the structural complexity of the input graph.

Matroids

A matroid is a structured hypergraph (set-system).

Definition

A matroid is a pair, $(E, \mathcal{I})$, where E is a finite set (the ground set), and $\mathcal{I}$ is a family of subsets (the independent sets), satisfying:

- $\emptyset \in \mathcal{I}$,
- $I_1 \in \mathcal{I}$ and $I_2 \subseteq I_1$ implies $I_2 \in \mathcal{I}$,
- $I_1, I_2 \in \mathcal{I}$ and $|I_2| < |I_1|$ implies there exists $e \in I_1 - I_2$ such that $I_2 \cup \{e\} \in \mathcal{I}$.

Matroids

A matroid is a structured hypergraph (set-system).

Definition

A matroid is a pair, $(E, \mathcal{I})$, where E is a finite set (the ground set), and $\mathcal{I}$ is a family of subsets (the independent sets), satisfying:

- $\emptyset \in \mathcal{I}$,
- $I_1 \in \mathcal{I}$ and $I_2 \subseteq I_1$ implies $I_2 \in \mathcal{I}$,
- $I_1, I_2 \in \mathcal{I}$ and $|I_2| < |I_1|$ implies there exists $e \in I_1 - I_2$ such that $I_2 \cup \{e\} \in \mathcal{I}$.

Definition

A maximal independent set is a basis. A subset of E that is not independent is dependent. A minimal dependent set is a circuit.

Graphic matroids

Example

Let G be a graph with edge set E. Then

$$(E, \{I \subseteq E \colon G[I] \text{ does not contain a cycle}\})$$

is a graphic matroid.

The bases of a graphic matroid are maximal forests. The circuits are cycles in the graph.

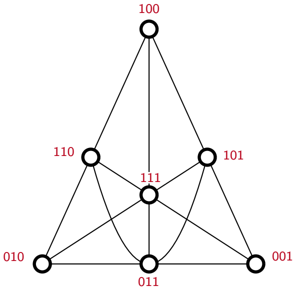
Representable matroids

Example

Let $\mathbb{F}$ be a field. Let E be a finite subset of the vector space $\mathbb{F}^n$. Then

$$(E, \{I \subseteq E \colon I \text{ is linearly independent}\})$$

is an $\mathbb{F}$-representable matroid.



Matroids and the Robertson-Seymour project

Statements about graphs are sometimes special cases of more general statements about matroids.

Matroids and the Robertson-Seymour project

Statements about graphs are sometimes special cases of more general statements about matroids.

The Graphs Minors Project of Robertson and Seymour provides a qualitative structural description of any proper minor-closed class of graphs.

Among other consequences, we can deduce that in any infinite collection of graphs, one is isomorphic to a minor of another.

Matroids and the Robertson-Seymour project

Statements about graphs are sometimes special cases of more general statements about matroids.

The Graphs Minors Project of Robertson and Seymour provides a qualitative structural description of any proper minor-closed class of graphs.

Among other consequences, we can deduce that in any infinite collection of graphs, one is isomorphic to a minor of another.

Geelen, Gerards, and Whittle have now established a qualitative structural description of any proper minor-closed class of $\mathbb{F}$-representable matroids, when $\mathbb{F}$ is a finite field.

We can deduce that in any infinite collection of $\mathbb{F}$-representable matroids, one is isomorphic to a minor of another.

Hliněný's Theorem

Hliněný's Theorem (2006)

Let φ be a sentence in MS_0, the monadic second-order logic of matroids.

Let $\mathcal{M}$ be a class of $\mathbb{F}$-representable matroids with bounded branch-width, where $\mathbb{F}$ is a finite field.

There is a polynomial-time algorithm that tests matroids in $\mathcal{M}$ and decides whether they satisfy φ.

Monadic second-order logic

$$\forall X_1 \forall X_2 \ (\mathsf{Ind}(X_1) \wedge X_2 \subseteq X_1 \rightarrow \mathsf{Ind}(X_2))$$

Monadic second-order logic

$$\forall X_1 \forall X_2 \; (\mathrm{Ind}(X_1) \wedge X_2 \subseteq X_1 \rightarrow \mathrm{Ind}(X_2))$$

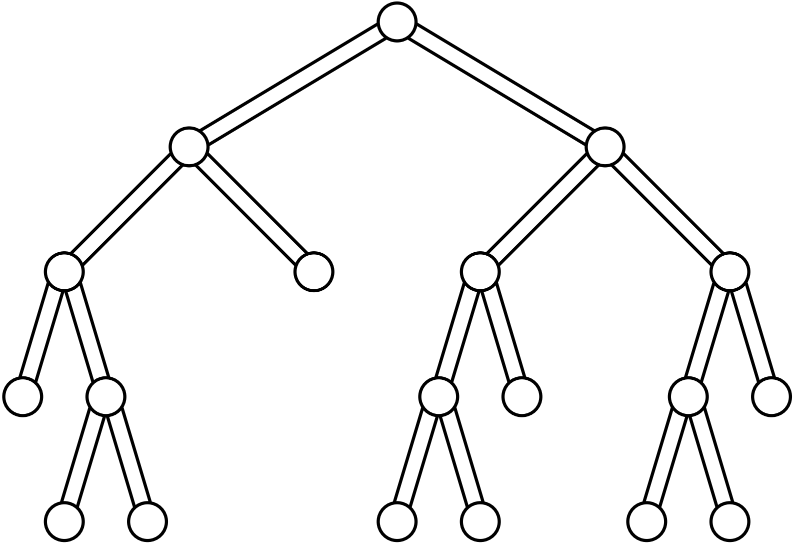
This expresses the second matroid axiom: every subset of an independent set is independent.

Tree automata

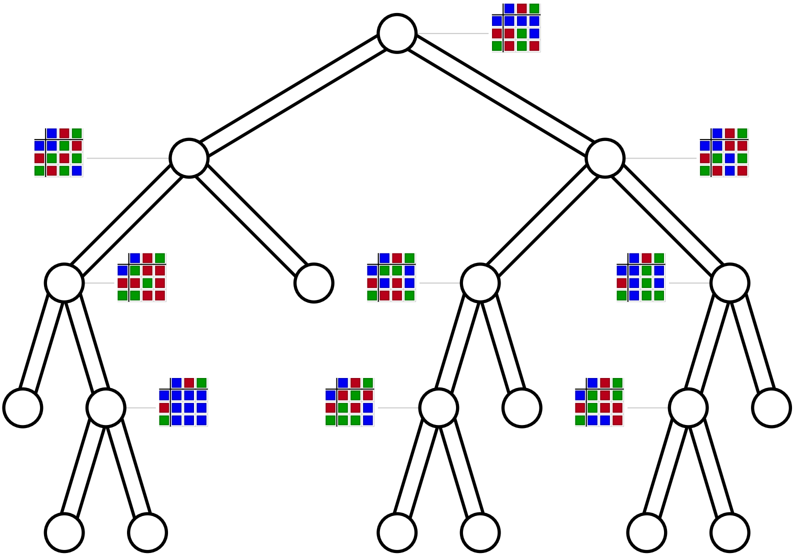
A tree automaton consists of a set of states (colours)

and a distinguished subset of accepting states.

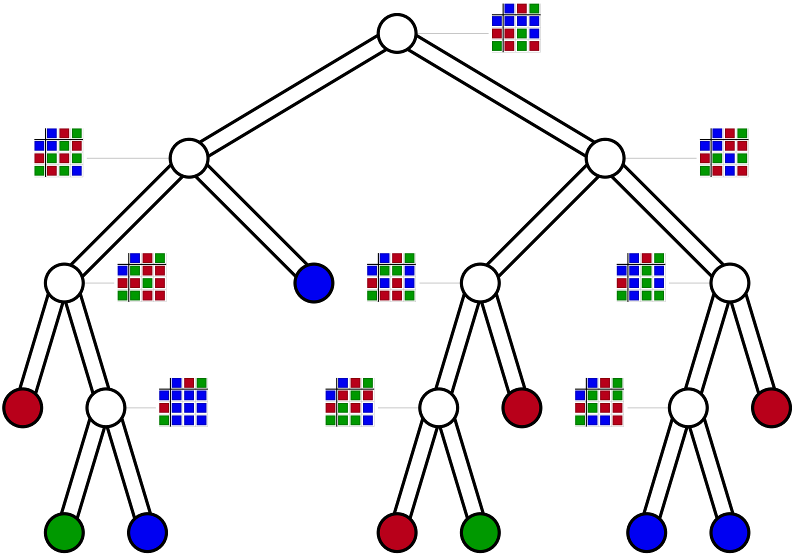
Tree automata



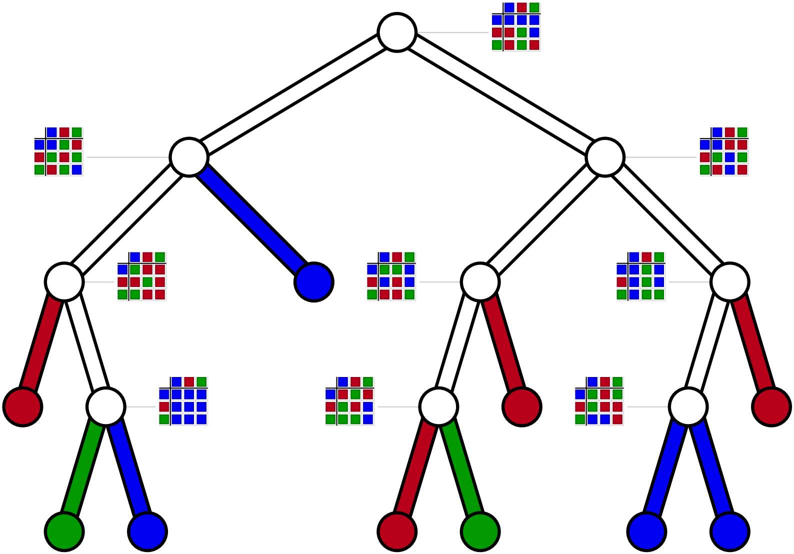
Tree automata



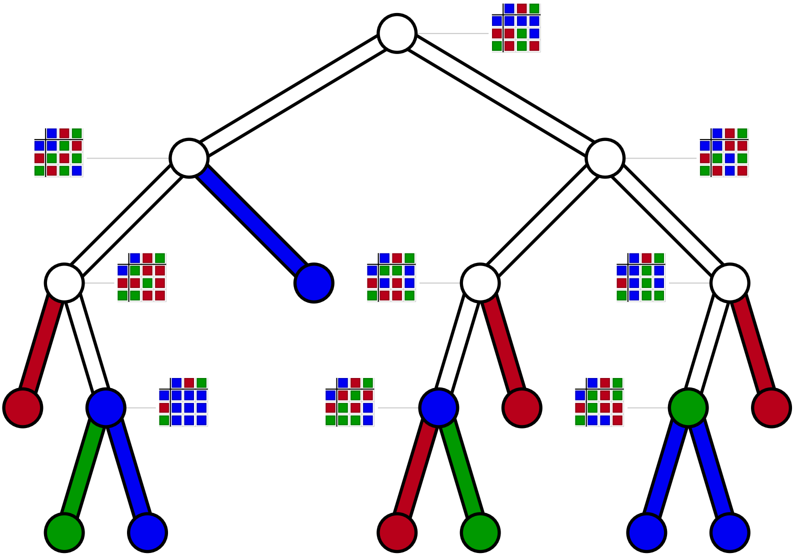
Tree automata



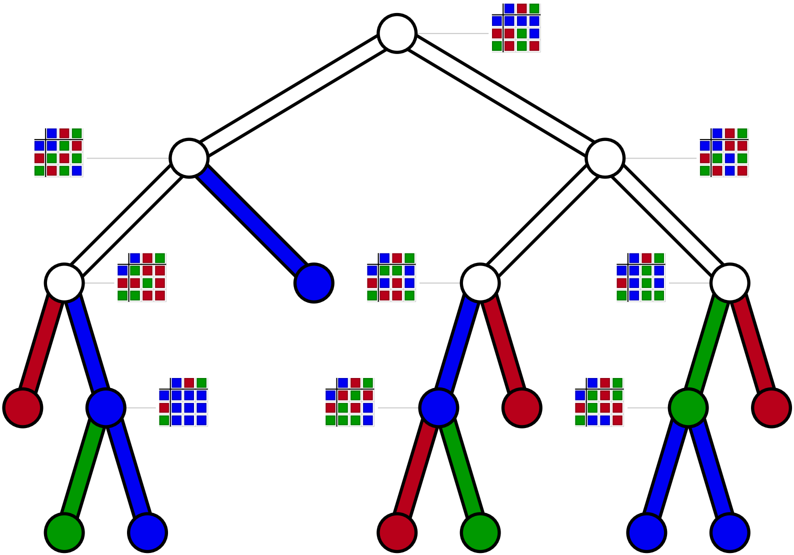
Tree automata



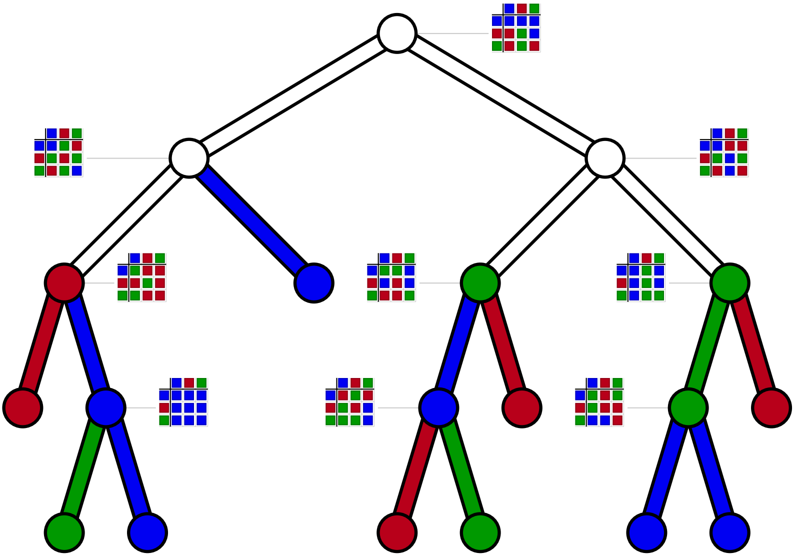
Tree automata



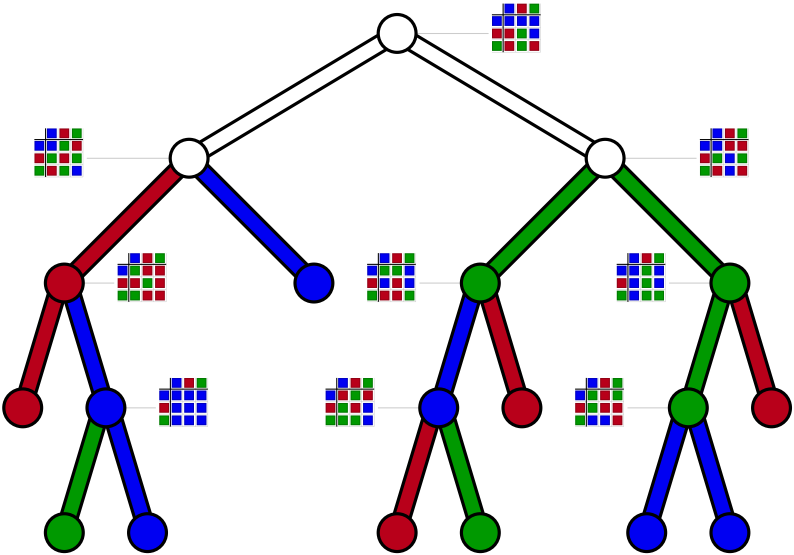
Tree automata



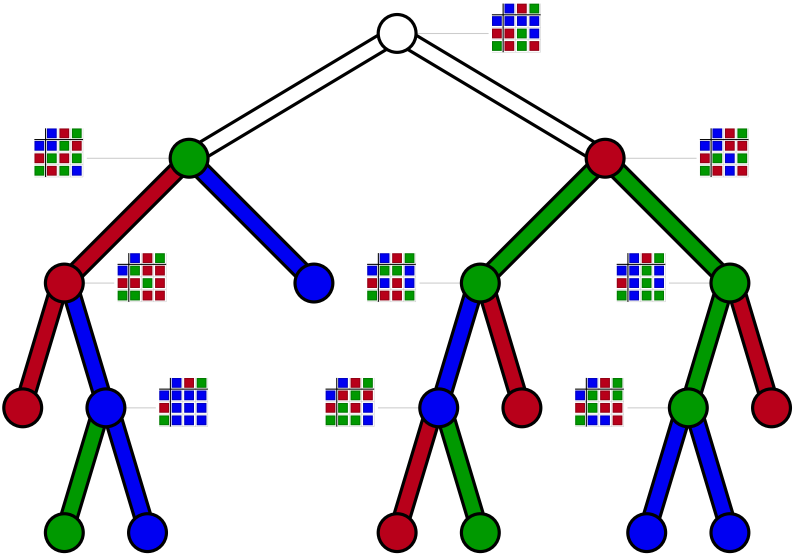
Tree automata



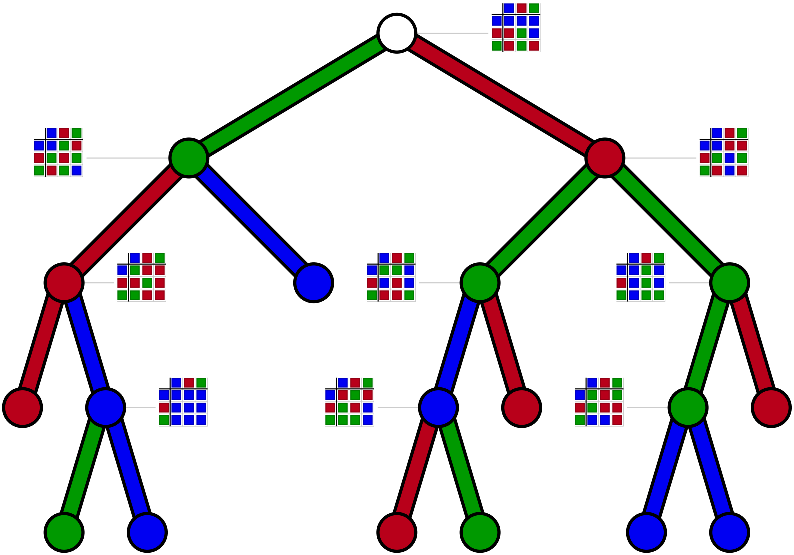
Tree automata



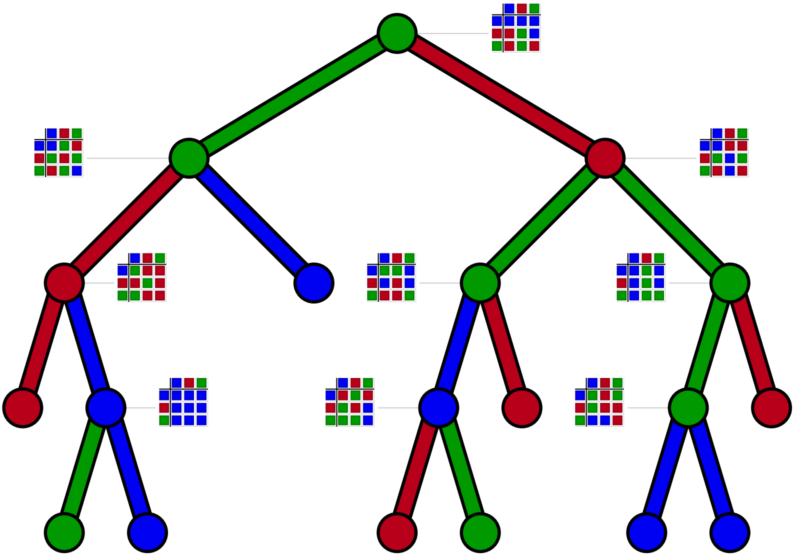
Tree automata



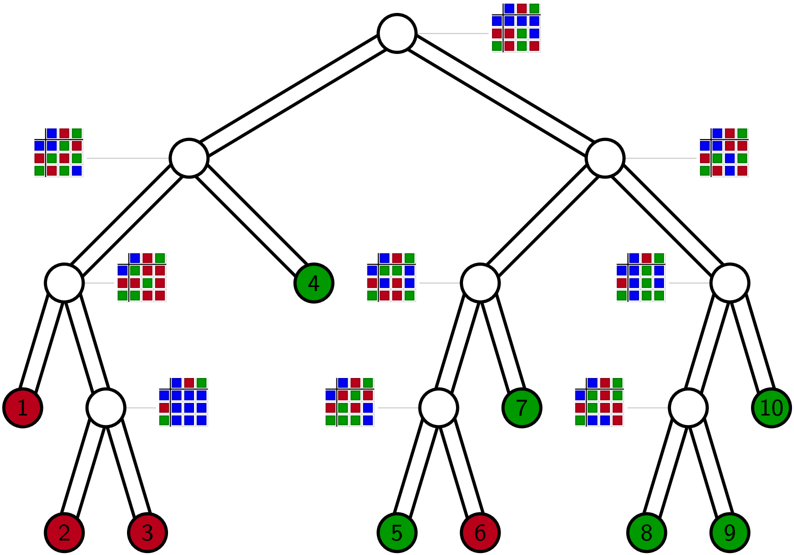
Tree automata



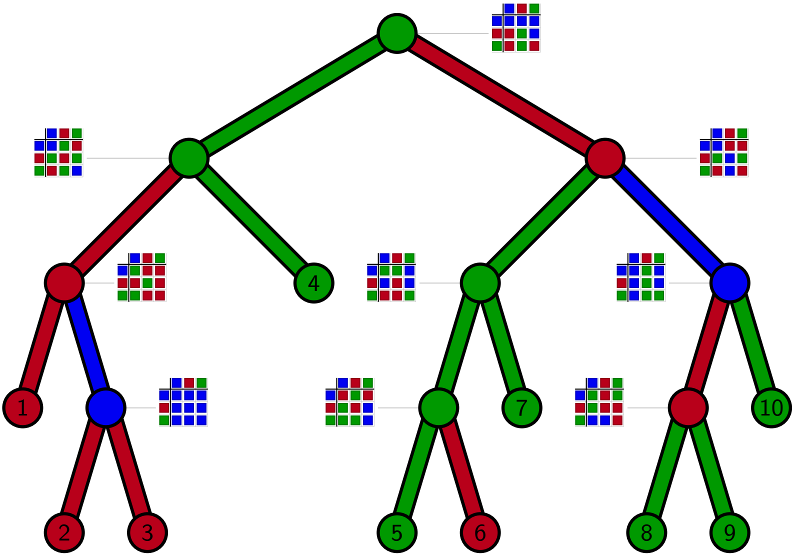
Tree automata



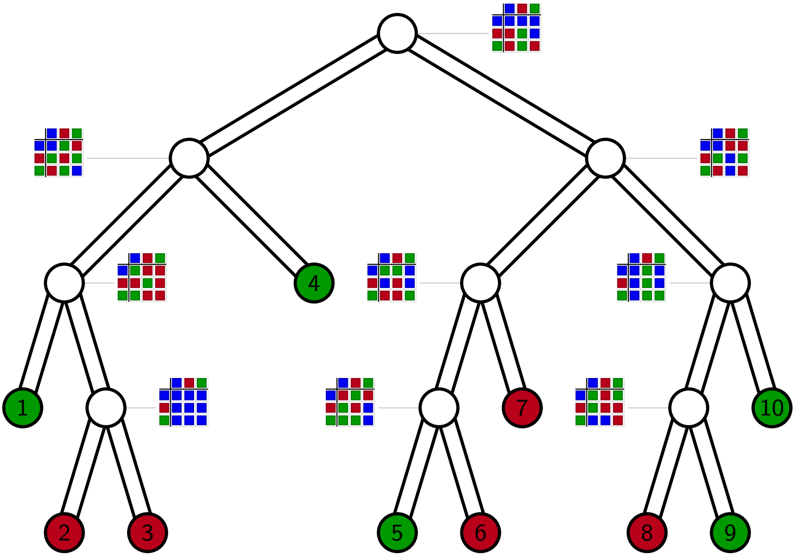
Tree automata



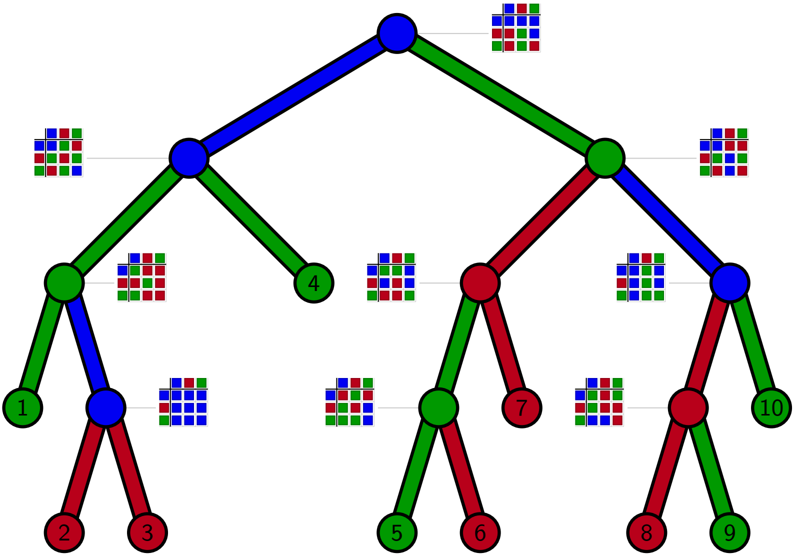
Tree automata



Tree automata



Tree automata



Automatic set-systems

Given an automaton with two distinguished colours that encode subsets, there is a corresponding set-system on the leaf-set of any tree.

What families of set-systems arise in this way?

Automatic set-systems

Given an automaton with two distinguished colours that encode subsets, there is a corresponding set-system on the leaf-set of any tree.

What families of set-systems arise in this way?

Definition

Let $\mathcal{M}$ be family of set-systems.

Assume there is an automaton, A, and for every $(E, \mathcal{I}) \in \mathcal{M}$, there is a tree, T_M, with leaf-set E, such that A accepts the subsets in $\mathcal{I}$ and rejects the subsets not in $\mathcal{I}$.

Then we say that $\mathcal{M}$ is automatic.

What families of set-systems are automatic?

Characterising automatic set-systems

Theorem

Let $\mathcal{M}$ be a family of set-systems. Then $\mathcal{M}$ is automatic if and only if it has bounded decomposition-width.

Decomposition-width

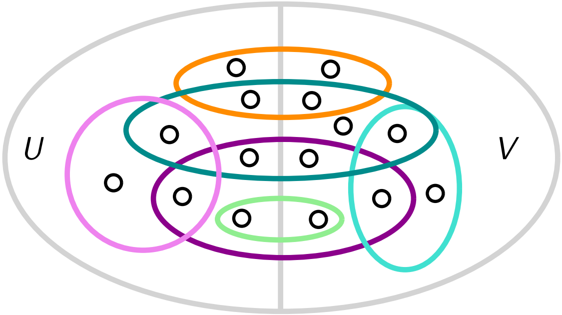
Definition

Let $(E, \mathcal{I})$ be a set-system, and let (U, V) be a partition of E.

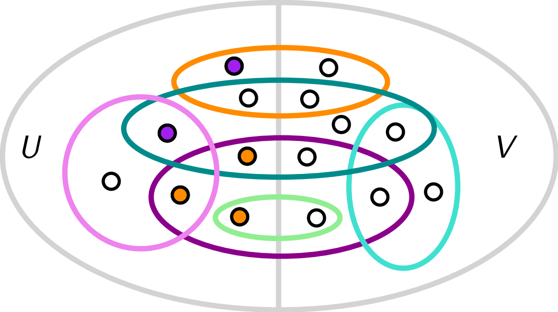
We define $\sim_U$, an equivalence relation on subsets of U.

Subsets $X, X' \subseteq U$ satisfy $X \sim_U X'$ if, whenever Z is a subset of V, both $X \cup Z$ and $X' \cup Z$ are in $\mathcal{I}$, or neither are.

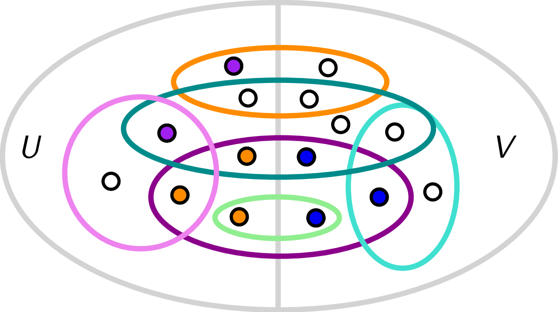
Decomposition-width



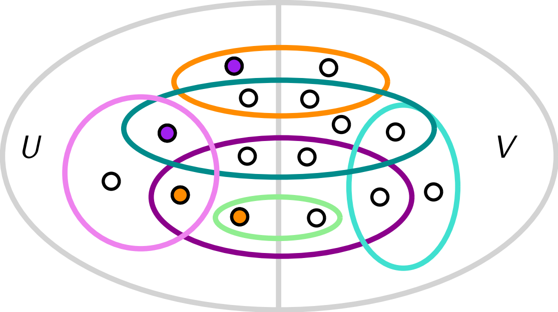
Decomposition-width



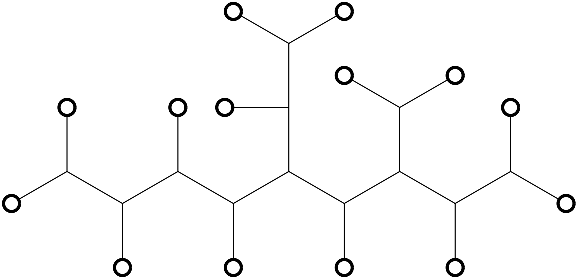
Decomposition-width



Decomposition-width

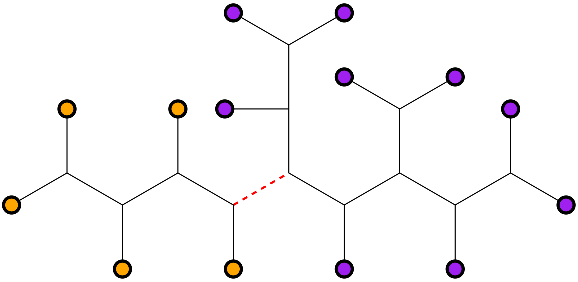


Decomposition-width



A decomposition of a set-system $(E, \mathcal{I})$ is a bijection between E and the leaves of a tree where every non-leaf vertex has degree three.

Decomposition-width



A decomposition of a set-system $(E, \mathcal{I})$ is a bijection between E and the leaves of a tree where every non-leaf vertex has degree three.

A displayed set is any set corresponding to the leaves in a connected component created by deleting an edge of the tree.

Characterising automatic set-systems

Theorem

Let $\mathcal{M}$ be a family of set-systems. Then $\mathcal{M}$ is automatic if and only if it has bounded decomposition-width.

Characterising automatic set-systems

Theorem

Let $\mathcal{M}$ be a family of set-systems. Then $\mathcal{M}$ is automatic if and only if it has bounded decomposition-width.

Definition

If $\mathcal{M}$ has bounded decomposition-width, then there is an integer, K, and for every $M = (E, \mathcal{I}) \in \mathcal{M}$, we have a decomposition of M such that whenever U is a displayed set, then $\sim_U$ has at most K equivalence classes.

An equivalent notion of decomposition-width was discussed by Král and Strozecki.

Tree automata and Hliněný's Theorem

If $\mathcal{M}$ is an automatic family of set-systems, then there is an automaton, A, and for every $(E, \mathcal{I}) \in \mathcal{M}$, there is a tree, T_M, with leaf-set E, such that A accepts the subsets in $\mathcal{I}$ and rejects the subsets not in $\mathcal{I}$.

Tree automata and Hliněný's Theorem

If $\mathcal{M}$ is an automatic family of set-systems, then there is an automaton, A, and for every $(E, \mathcal{I}) \in \mathcal{M}$, there is a tree, T_M, with leaf-set E, such that A accepts the subsets in $\mathcal{I}$ and rejects the subsets not in $\mathcal{I}$.

In this case, we can quickly test whether a subset of E is in $\mathcal{I}$: colour the leaves appropriately, and then just run A.

Tree automata and Hliněný's Theorem

If $\mathcal{M}$ is an automatic family of set-systems, then there is an automaton, A, and for every $(E, \mathcal{I}) \in \mathcal{M}$, there is a tree, T_M, with leaf-set E, such that A accepts the subsets in $\mathcal{I}$ and rejects the subsets not in $\mathcal{I}$.

In this case, we can quickly test whether a subset of E is in $\mathcal{I}$: colour the leaves appropriately, and then just run A.

We can use a bootstrapping procedure to build an automaton that will quickly test any given MS_0 sentence for set-systems in $\mathcal{M}$.

Tree automata and Hliněný's Theorem

If $\mathcal{M}$ is an automatic family of set-systems, then there is an automaton, A, and for every $(E, \mathcal{I}) \in \mathcal{M}$, there is a tree, T_M, with leaf-set E, such that A accepts the subsets in $\mathcal{I}$ and rejects the subsets not in $\mathcal{I}$.

In this case, we can quickly test whether a subset of E is in $\mathcal{I}$: colour the leaves appropriately, and then just run A.

We can use a bootstrapping procedure to build an automaton that will quickly test any given MS_0 sentence for set-systems in $\mathcal{M}$.

This explains why tree automata are at the heart of Hliněný's Theorem.

The challenge is constructing T_M, given $M \in \mathcal{M}$.

Our conclusions

We have reproved Courcelle's Theorem and Hliněný's Theorem.

Our conclusions

We have reproved Courcelle's Theorem and Hliněný's Theorem.

We have developed general tools which we can use to extend Hliněný's Theorem to new classes.

Our conclusions

We have reproved Courcelle's Theorem and Hliněný's Theorem.

We have developed general tools which we can use to extend Hliněný's Theorem to new classes.

We have done this for the following classes.

- Gain-graphic matroids (with gain labels from a finite group)
- Bicircular matroids
- Lattice-path matroids
- Principal transversal matroids

Our conclusions

We have reproved Courcelle's Theorem and Hliněný's Theorem.

We have developed general tools which we can use to extend Hliněný's Theorem to new classes.

We have done this for the following classes.

- Gain-graphic matroids (with gain labels from a finite group)
- Bicircular matroids
- Lattice-path matroids
- Principal transversal matroids

We have some negative results: if $\mathbb{F}$ is an infinite field, then Hliněný's Theorem cannot extend to the class of $\mathbb{F}$-representable matroids. Nor can it extend to the class of gain-graphic matroids over an infinite group (assuming $\mathsf{P} \neq \mathsf{NP}$).