Logic for algorithms

Rod Downey
Victoria University
Wellington
Joint with Benjamin Burton, Queensland University
Montpellier, June 2016.

This lecture:

- In this lecture I have chosen some material which is relevant to the algorithms and logic groups here at Montpellier.
- The new work a combination of logic, complexity theory and low dimensional topology, so I will explain each bit separately.

Parameterized complexity

- A mathematical idealization is to identify “Feasible” with P
- With this assumption, the theory of NP-hardness is an excellent vehicle for mapping an outer boundary of intractability, for all practical purposes.
- Indeed, assuming the reasonable current working assumption that NTM acceptance is $\Omega(2^n)$, NP-hardness allows for practical lower bound for exact solution for problems.
- A very difficult practical and theoretical problem is “How can we deal with P?”.
- More importantly how can we deal with $P - FEASIBLE$, and map a further boundary of intractability.

Revisioning complexity theory

- When is the only thing you know about a problem is the input size
- Answer : only cryptography, and this is by design.
- For practical problems, the world comes equipped with many many additional parameters.
- As we soon see, sensitizing the run times to parameters allows the development of a distinctive and often useful toolkit.
- In particular, focusing on the paramaters as a standard attack method for practice can be systematized, and sometimes even automated.
- The theory equips us with both a positive and negative tool kit.
- We are far from undrstanding the complexity of real data, and whether things like P vs NP even matters.

Parameters

- Without even going into details, think of all the graphs you have given names to and each has a relevant parameter: planar, bounded genus, bounded cutwidth, pathwidth, treewidth, degree, interval, etc, etc. In numerical analysis the degree of precision etc.
- Also nature is kind in that for many practical problems the input (often designed by us) is nicely ordered.

Two Basic Examples

- VERTEX COVER
  Input: A Graph G.
  Parameter : A positive integer k.
  Question: Does G have a size k vertex cover? (Vertices cover edges.)
- DOMINATING SET
  Input: A Graph G.
  Parameter : A positive integer k.
  Question: Does G have a size k dominating set? (Vertices cover vertices.)

- VERTEX COVER is solvable by an algorithm $\mathfrak{O}$ in time $f(k)|G|$, a behaviour we call fixed parameter tractability, (Specifically $1.275^k k^2 + c|G|$, with c a small absolute constant independent of k.)
- Whereas the only known algorithm for DOMINATING SET is complete search of the possible k-subsets, which takes time $\Omega(|G|^k)$.

- In the below I will mostly talk for convenience about graphs.
- I could just as easily be talking about many other areas.
- In the Computer Journal alone, there is biological, artificial intelligence, constraint satisfaction, geometric problems, scheduling, cognitive science, voting, combinatorial optimzation, phylogeny.

Basic Definition(s)

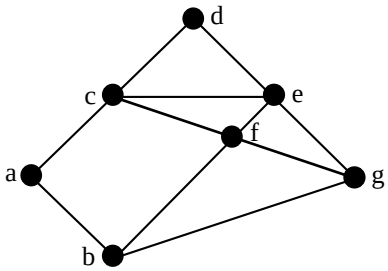
- Setting : Languages $L \subseteq \Sigma^* \times \Sigma^*$.
- Example (Graph, Parameter).
- We say that a language L is fixed parameter tractable if there is a algorithm M, a constant C and a function f such that for all x, k,

$$(x, k) \in L \text{ iff } M(x) = 1 \text{ and}$$

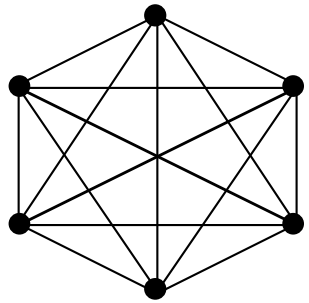
$$\text{the running time of } M(x) \text{ is} f(k)|x|^C.$$

- E.g. VERTEX COVER has $C = 1$. Vertex Cover has been implemented in computational biology for k up to about 7000 and n large.
- One example: Langston et. al. 2008 Innovative computational methods for transcriptomic data analysis: A case study in the use of FPT for practical algorithm design and implementation. in *The Computer Journal*, 51(1):26–38, 2008.

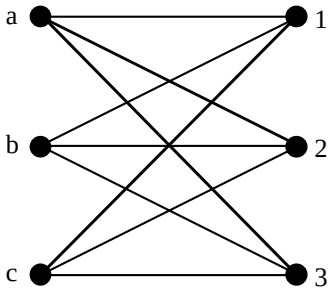
OTHER POSSIBLE VERSIONS



VERTEX COVER : Vertices cover edges.
Example: {c, f, b, e, h}.

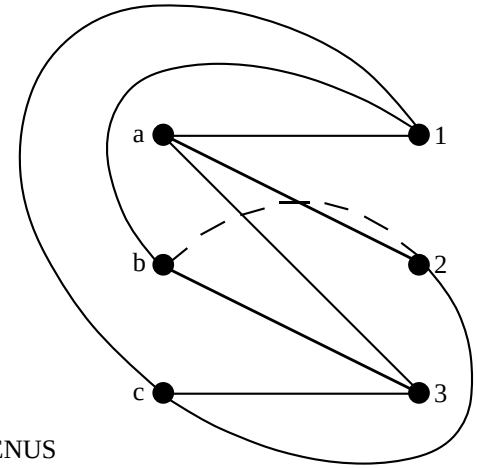


GRAPH LINKING NUMBER :
K_6 has linking number 1.



$K_{3,3}$

GRAPH GENUS



$K_{3,3}$ has genus 1 by
putting all lines except
$< b, 2 >$ on a sphere
and $< b, 2 >$ on a handle.

FIGURE: Examples of FPT problems

- There has been now over 20 years of research into this area.
- What is not well known is that we now more or less can match upper and lower bounds for various algorithms, assuming a certain complexity hypothesis, like P$\neq$NP, (namely $M[1] \neq FPT$, no $2^{o(n)}$ algorithm for n-variable 3SAT. Impagliazzo, Paturi and Zane)
- For example, (Cai and Juedes) Assuming this, there is no $2^{o(k)}|G|^c$ algorithm for k-PATH in a graph G.
- There is known a $2^{O(k)}|G|^c$ algorithm!
- That is, upper and lower bounds are matching up to big O.
- In case you are interested, I have order forms for an excellent book....

Positive techniques

- Elementary ones that work.
- Fancy metatechniques that almost work.
- Exotic methods that are even non-constructive. We know something is in P but don't know the algorithm. E.g. embeddable on a torus, linkless embedding.
- Logical metatheorems that we thought did not work but now have some implementations.
- We care about the last one here. But I will begin with one easy algorithm.
- Limits

Metatheorems: Logic I

- One of the key realizations is that many things we are interested in are constructed inductively using a finite set of "operators"
- This means that they can (i) be thought of as a "parse" formal language, and (ii) can attract "metrics" as parameters for the size of this "language".
- This allows for interactions complexity, automata, definability.
- Ideas go back to Büchi, Rabin, Kleene, McNaughton etc.

Metatheorems: Logic

- (First order Logic)
  1. **Atomic formulas:** $x = y$ and $R(x_1, ..., x_k)$, where R is a k-ary relation symbol and $x, y, x_1, ..., x_k$ are individual variables, are FO-formulas.
  2. **Conjunction, Disjunction:** If ϕ and ψ are FO-formulas, then $\phi \wedge \psi$ is an FO-formula and $\phi \vee \psi$ is an FO-formula.
  3. **Negation:** If ϕ is an FO-formula, then $\neg\phi$ is an FO-formula.
  4. **Quantification:** If ϕ is an FO-formula and x is an individual variable, then $\exists x\ \phi$ is an FO-formula and $\forall x\ \phi$ is an FO-formula.
- Eg We can state that a graph has a clique of size k using an FO-formula,

$$\exists x_1 ... x_k \bigwedge_{1 \leq i \leq j \leq k} E(x_i, x_j)$$

- Of course Turing in 1936 proved FO logic is undecidable.

Monadic Second Order Logic

- Two sorted structure with variables for sets of objects.
- 1. **Additional atomic formulas:** For all set variables X and individual variables y, Xy is an MSO-formula.
  2. **Set quantification:** If ϕ is an MSO-formula and X is a set variable, then $\exists X\ \phi$ is an MSO -formula, and $\forall X\ \phi$ is an MSO-formula.
- Eg k-colorability

$$\exists X_1,,,\exists X_k \left(\forall x \bigvee_{i=1}^{k} X_i x \wedge \forall x \forall y \left(E(x,y) \rightarrow \bigwedge_{i=1}^{k} \neg (X_i x \wedge X_i y) \right) \right)$$

- Want to be careful as to what the primitives are. Eg. MSO graphs, MSO strings, etc. Many questions involve the translations between.
- For example, Büchi's theorem for strings uses the ordering of strings, what about graphs of bounded treewidth (below)? No order there, etc. More later.

Model Checking

- Instance: A structure $\mathcal{A} \in \mathcal{D}$, and a sentence (no free variables) $\phi \in \Phi$.
  Question: Does $\mathcal{A}$ satisfy ϕ?
- PSPACE-complete for FO and MSO.

- Logicians such as Büchi, Rabin and others realized the conection between MSO and complexity, especially on formal languages. Below we will discuss this in the situation where objects of interest have parse languages, meaning that they are build up in an inductive manner.
- MSO is central to program verification, and makes logic the calculus of computer science. See articles by Moshe Vardi.
- That is, the connection is way beyond mere complexity.

A SEMINAL THEOREM

- The FO logic of $\mathbb{N}, <$
- You can say things like $\exists x \forall y (x < y)$, etc. and intrpret in $\{0, 1, 2, \dots\} = \mathbb{N}$.
- The monadic second order of strings w over Σ^*, say $\Sigma = \{a, b, c\}$.
- Have $P_a(x)$ with the interpretation "position x is labelled by a, etc. The domain is the set of numbers $\{0, \dots, |w| - 1\}$. Typically also have $succ(x, y)$
- $\exists x \exists y \exists z (succ(x, y) \wedge succ(y, z) \wedge Q_c(z))$

We say that $L \subseteq \Sigma^*$ is MSO-definable if there is a MSO setence φ such that $\sigma \in L$ iff $\sigma \models \varphi$.

THEOREM (BÜCHI, 1960)

L is MSO-definable iff L is regular.

This has also been proven for tree languages and leaf to root automata.

Bounded Width Metrics

- Graphs constructed inductively. Treewidth, Pathwidth, Branschwidth, Cliquewidth mixed width etc. k-Inductive graphs, plus old favourites such as planarity etc, which can be viewed as local width.
- Example:

Definition

[Tree decomposition and Treewidth] Let $G = (V, E)$ be a graph. A tree decomposition, TD, of G is a pair $(T, \mathcal{X})$ where

1. $T = (I, F)$ is a tree, and
2. $\mathcal{X} = \{X_i \mid i \in I\}$ is a family of subsets of V, one for each node of T, such that

(i) $\bigcup_{i \in I} X_i = V$,

(ii) *for every edge* $\{v, w\} \in E$, *there is an* $i \in I$ *with* $v \in X_i$ *and* $w \in X_i$, *and*

(iii) *for all* $i, j, k \in I$, *if* j *is on the path from* i *to* k *in* T, *then* $X_i \cap X_k \subseteq X_j$.

- This gives the following well-known definition.

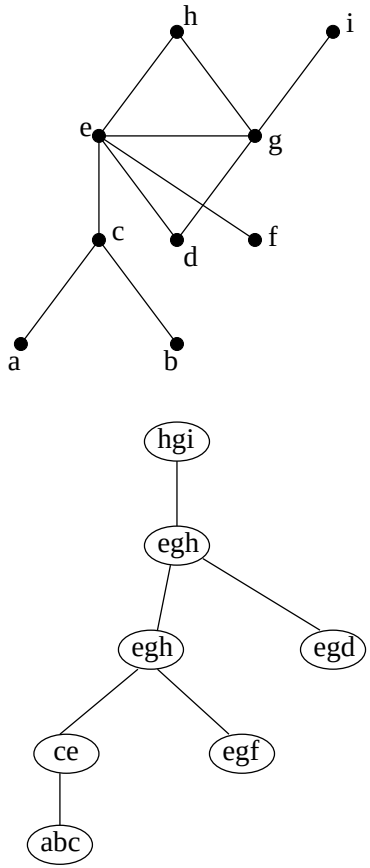
DEFINITION

The width of a tree decomposition $((I, F), \{X_i \mid i \in I\})$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph G, denoted by $tw(G)$, is the minimum width over all possible tree decompositions of G.

The canonical method

- The following refers to any of these inductively defined graphs families. Notes that many commercial constructions of, for example chips are inductively defined.
  1. Find a bounded-width tree (path) decomposition of the input graph that exhibits the underlying tree (path) structure.
  2. Perform dynamic programming on this decomposition to solve the problem.

An example for Independent Set



$\emptyset$	a	b	c	ab	ac	bc	abc
0	1	1	1	2	-	-	-

Bodlaender's Theorem

- The following theorem is shows that treewidth is FPT. Improves many earlier results showing this. The constant is about 2^{35k^3}.

Theorem (Bodlaender)

k-Treewidth is linear time FPT. Moreover a tree decomposition can be fornd in the same time.

- Not practical because of large hidden O term.
- Unknown if there is a practical FPT treewidth algorithm
- Nevertheless approximation and algorithms specific to known decomps run well at least sometimes.

Courcelle's and Seese's Theorems

Theorem (Courcelle 1990)

The model-checking problem for MSO restricted to graphs of bounded treewidth is linear-time fixed-parameter tractable.

Detlef Seese has proved a converse to Courcelle's theorem.

Theorem (Seese 1991)

Suppose that $\mathcal{F}$ is any family of graphs for which the model-checking problem for MSO is decidable, then there is a number n such that, for all $G \in \mathcal{F}$, the treewidth of G is less than n.

Proving Courcelle's Theorem

- There are several ways to do this.
- Fellows and Langston reduced this to a string theorem.
- Language $\Sigma = \{\emptyset, p(i), j(i,t), \oplus\}$.
- Boundaried graphs with boundary $\{1, \ldots, k\}$.
- Interpretation $\emptyset$ creates a boundary, $j(i,t)$ says join i to t, $p(i)$ says creat a new vertex labelled i and remove the old label, $\oplus$ allows us to *glue* along the boundary.
- The a treewidth k graph is a tree of such labels, and pathwidth k is a string of such with no $\oplus$.
- To prove Courcelle's Theorem you can filter through the corresponding Büchi's Theorem.
- Realize, however, that monadic second order logic in the language of graphs is not the same as that for strings or trees; and moreover, many strings can correspond to the same graph.

A question

Question Does the full analog of Büchi's Theorem hold for graphs of bounded treewidth?
There are several claims in the literature, but no full proof.

FO Logic

- We can define the "treewidth growth rate" by looking at diameter d around vertices of G. If we have a class of graphs where the treewidth of graphs in the graph at diameter d is bounded by $h(d)$, we say that the class has bounded local treewidth.

Theorem (Flum-Grohe)

If a class of graphs has bounded local treewidth, then FO model checking is linear time decidable.

- Filters through Gaifman Locality Theorem which states that FO can't talk about large diameters.

Nowhere Density

- Recently Nesetril and Ossona de Mendez introduce a sweping generilization of the above, excluding a minor (Thilkos etc), and lots of other structure theorems.
- A depth r minor $G \preceq_r H$, is a collection of branch sets $\{T_v : v \in V(H)\}$, each of which is a tree of radius at most r. This is the size of the "folio".
- A class $\mathcal{C}$ of graphs is nowhere dense if for each r there is a graph $G = G_r$ such that for all $H \in \mathcal{C}$, $G \not\preceq_r H$.

Theorem (Grohe, Kreutzer and Siebertz)

There is an algorithm decising FO model checking in time $O(n^{1+\epsilon})$ for nowhere dense graphs.

- The proof is a complex variation on the above with new finite model theory and new graph structure theory.
- Dvorak, D. Kral, and R. Thomas have proven that for classes closed under taking subgraphs, this is optimal.

There is more

- Generate graphs using $\emptyset$, $J(i,t)$ join everything with label i to label t, $r(i,j)$ relabel i to j.
- The width is the number of colours i needed.
- Cliques have cliquewidth 2.

Theorem (Courcelle, Makowsky and Rotics 2000)

MSO_1 is linear time decidable for graphs of bounded cliquewidth. (Only sets of vertices allowed.)

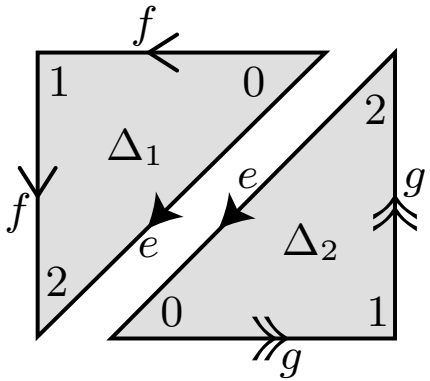
Graphs?

- The treewidth methodology has been applied in lots of areas. Finite model theory, braids, knots (Makowski, Rotics etc), matroids,
- Can be used for hard problems like counting.
- No general metheorem.
- Generalized to other parse notions like Cliquewidth d-Degeneracy, Nowhere Density the idea being that formal languages and automata are basic.
- Here I will look at a new application.

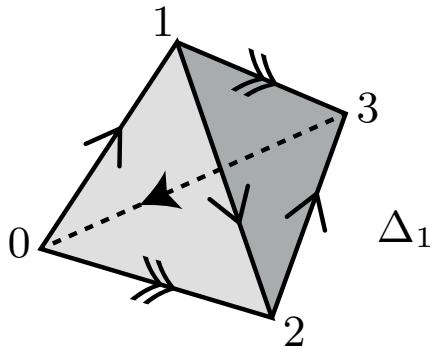
Triangulations

- 1. A *d-dimensional triangulation* consists of a collection of d-dimensional simplices $\Delta_1, \ldots, \Delta_n$ some or all of whose facets (ie. $d-1$-dimensional faces) are affinely identified.
  2. Each facet F may only be identified with at most one other facet F' of a d-simplex. (This could be the same d-simplex but not the same facet.)
- There are $\binom{d+1}{i+1}$ many i-faces of the d-simplices (where a 0-face is a vertex, a 1-face an edge etc).
- A *d-manifold triangulation* is simply a d-dimensional triangulation whose underlying topological space is a d-manifold when using the quotient topology.
- There are $\binom{d+1}{i+1}$ many i-faces of the d-simplices (where a 0-face is a vertex, a 1-face an edge etc).
- The idea is to have a metatheorem for such traingulations.

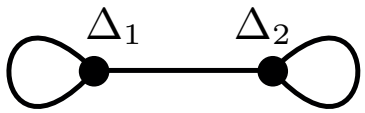
An example-Klein Bottle



(a) A Klein bottle $\mathcal{K}$



(b) A one-tetrahedron solid torus



(c) The dual graph $\mathcal{D}(\mathcal{K})$

FIGURE: Examples of triangulations

$$\Delta_1 : 02 \longleftrightarrow \Delta_2 : 20, \quad \Delta_1 : 01 \longleftrightarrow \Delta_1 : 12, \quad \Delta_2 : 01 \longleftrightarrow \Delta_2 : 12.$$

- The resulting triangulation has one vertex (since all three vertices of Δ_1 and all three vertices of Δ_2 become identified together), and three edges (labelled e, f, g in the diagram).
- The dual graph $\mathcal{D}(T)$ of a triangulation T is the multigraph whose nodes correspond to simplices and whose edges correspond to identified pairs of facets.
  Above we have the dual graph of the Klein bottle.

The language

- Standard boolean operations.
- for each $i \in [0, d]$ variables for i-faces of a triangulation, and ones for sets of them.
- for each $i \in [0, d]$, and each ordered sequence $\pi_0, \ldots, \pi_i$ of distinct integers from $\{0, \ldots, d\}$, a subface relation $\leq_{\pi_0 \ldots \pi_i}$.
- The interpretation $f \leq_{\pi_0 \ldots \pi_i} s$ indicates that f is a subface of the triangulation, s a simplex of the triangulation, and f is identified with th subface of s formed by the simplicial vertices $\pi_0, \ldots, \pi_i$ in the way that vertices of the face $0, \ldots, i$ of the face f correspond to vertices $\pi_0, \ldots, \pi_i$ of the simplex s.

Orientability

Recall: 2-dimensional triangulation is orientable if and only if each triangle can be assigned an orientation (clockwise or anticlockwise) so that adjacent triangles have compatible orientations, as illustrated in Figure 3 below.

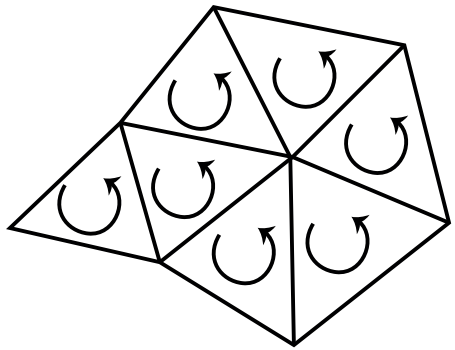


Figure: Adjacent triangles have compatible orientations

Theorem (Burton and D)

For any fixed dimension d, let K denote the class of d-dimensional triangulations whose dual graphs have universally bounded treewidth. Then for a fixed MSO ϕ, and triangulation $T \in K$, we can decide if T satisfies ϕ in linear time.

- We also show that the optimization problems can be solved.
- The method is one of reduction to the graph version.
- We apply this to various problems on these objects. including things called Taut Angle Structure, Discrete Morse Matching, Turaev-Viro Invariants, of which I know nothing.

- Many Thanks