Automata based graph algorithms for logically defined problems

Part 3: beyond MSO logic for graph properties and *functions* on graphs.

Summary

A fly-automaton for regularity of graphs (*not an MSO property*).

Boolean and first-order constructions of properties and functions, and their interpretations in terms of fly-automata.

Monadic-second order constructions.

Implementation (in AUTOGRAPH).

Conclusions and call for interesting problems to handle in this way.

Part 4 (Bonus) : Edge quantification and special tree-width.

The example of regularity

This property is *not MSO* ($K_{n,m}$ is regular $\leftarrow\rightarrow$ n = m).

Information q(u) relative to G(t/u) :

(deg(a), …, deg(d), #(a), …., #(d)) or *Error*

where deg(p) = common degree of all p-vertices,

#(p) = number of p-vertices.

Error : two p-vertices have different degrees

(G(t) cannot be regular, because all its p-vertices are linked to the same vertices outside of t/u, where t is an *irredundant* term; easy preprocessing).

Some "programmed" transitions :

For $Add_{a,b}$: deg(a) := deg(a) + #(b), deg(b) := deg(b) + #(a)

other values of deg and # are not modified.

For $\oplus$: for each a, #(a) := #(a) $_1$ + #(a) $_2$

yields *Error* if deg(a) $_1$ $\neq$ deg(a) $_2$ for some a.

A state is *Error* or belongs to $[0,n]^{2k}$, n = number of vertices of G(t), k = number of labels in t. Firing a transition takes time O(k.log(n)). We have a P-FA.

Application: Partition into 2 regular graphs : $\exists X\ (Reg[X] \wedge Reg[X^c])$, is an XP-decidable property (not FPT because of $\exists X$).

Boolean and first-order compositions of properties and functions, and of automata.

$P \wedge Q$, $P \vee Q$, $\neg P$, $g(\alpha_1, \ldots, \alpha_p)$ where g is poly-time computable (can be a relation such as a comparison of numbers),

$P[X \cap Y]$: property of subgraph induced on $X \cap Y$ (*set term*)

First-order (FO) quantifications : $\exists \underline{x}.P(\underline{x})$, $\underline{x}$ =tuple of FO var.

Set of satisfying assignments : Sat $\underline{x}.P(\underline{x})$ (*a query*)

Number of satisfying assignments : $\# \underline{x}.P(\underline{x})$.

Set of values $\alpha(\underline{x})$ such that $P(\underline{x})$ is true.

Min (Max) of values $\alpha(\underline{x})$ such that $P(\underline{x})$ is true.

Type of automata	Finite	P-FA	FPT-FA	XP-FA
$P\wedge Q, P\vee Q, \neg P, P[X\cap Y]$	Finite	P	FPT	XP
$g(\alpha_1, \ldots, \alpha_p)$, $\alpha(X\cap Y)$		P	FPT	XP
$\exists\, \underline{x}.P(\underline{x})$, $\forall\, \underline{x}.P(\underline{x})$	Finite	P	FPT	XP
Sat $\underline{x}.P(\underline{x})$, # $\underline{x}.P(\underline{x})$	P	P	FPT	XP
SetVal $\underline{x}.\alpha(\underline{x})$ / $P(\underline{x})$		P	FPT	XP

Finite : Finite signature and sets of states.

We have “nice preservations” of the types of automata.

Main proof ideas for $\exists\, \underline{x}.P(\underline{x})$, $\#\, \underline{x}.P(\underline{x})$, Sat $\underline{x}.P(\underline{x})$,

where $\underline{x}$ is a p-tuple of first-order variables.

Projection pr : $F^{(p)} \rightarrow F$, a relabelling of $F^{(p)}$ designed to handle p variables. From a deterministic automaton *A* over $F^{(p)}$ for $P(\underline{x})$, we get a FA pr(*A*), that is not deterministic and decides $\exists\, \underline{x}.P(\underline{x})$. But, its *nondeterminism degree* is $\leq n^p$, hence *polynomially bounded* in the number n of vertices. We get a P-FA or an FPT-FA, or an XP-FA if *A* is so.

Same idea for Sat $\underline{x}.P(\underline{x})$. For $\#\, \underline{x}.P(\underline{x})$, we transform pr(*A*) into a *deterministic FA* that counts the number of its accepting runs.

Monadic second-order constructions

The spectrum $Sp\underline{X}.P(\underline{X})$ of a property $P(\underline{X})$ is the set of tuples of cardinalities of the components of the $\underline{X}$ satisfying $P(\underline{X})$.

The multispectrum $MSp\underline{X}.P(\underline{X})$ is the corresponding multiset of tuples of $Sp\underline{X}.P(\underline{X})$. If $\underline{X} = X$ (one component), it is :

the set of pairs (m, i) such that $i > 0$ is

the number of sets X of cardinality m that satisfy $P(X)$.

If $\underline{X}$ is a p-tuple and n is the number of vertices, a multispectrum is a function $[0,n]^p \rightarrow [0,2^{p.n}]$; it can be encoded in size $O(n^p.\log(2^{p.n})) = O(n^{p+1})$.

Type of automaton *A* for P(X̲)	Finite	P-FA	FPT-FA	XP-FA
∃X̲.P(X̲), ∀X̲.P(X̲)	F	P	FPT	XP
MSp X̲.P(X̲), Sp X̲.P(X̲), # X̲.P(X̲), MaxCard X.P(X)	P	P	FPT	XP

P-FA means : *A* is P-FA *and* pr(*A*) has a polynomially bounded nondeterminism degree. Similar for FPT and XP. (FPT- or XP-bounded nondeterminism degree). Hence, there are more contraints for the preservation of types of automata than for FO constructions.

Some examples : (1) *Equitable p-coloring* (not MSO) :

$\exists X_1,\ldots,X_p$ (Partition$(X_1,\ldots,X_p) \wedge$ Stable$[X_1] \wedge\ldots\wedge$ Stable$[X_p]$

$\wedge\ |X_1| = \ldots = |X_{i-1}| \geq |X_i| = \ldots = |X_p| \geq |X_1| - 1)$.

It is FPT (for fixed p).

(2) A P-FA computable (*not MSO*) function : the *generalized degree*:

$e(X,Y)$ = number of edges between X and Y, where $X \cap Y = \emptyset$.

Its minimimal value such that $P[X] \wedge Q[X^c]$ where P, Q are MSO properties (for an example) is XP-computable.

(3) *Counting p-colorings* with particular properties, e.g., acyclic or equitable.

(4) One can minimize (or maximize) the use of a particular color. Minimization gives a “distance to p-colorability”.

(5) Covering the edges of a graph by p cliques whose vertex sets satisfying MSO constraints

The system AUTOGRAPH (by Irène Durand)

Fly-automata for basic graph properties :

Clique, Stable (no edge), Link(X,Y), NoCycle,

Connected, *Regular*, Partition(X, Y, Z), etc…

and functions :

#Link(X,Y) (number of edges between X and Y)

Maximum degree, etc..

Procedures for combining fly-automata (combinations of descriptions)

(defined in Part 2)

product : for $P \wedge Q$, $P \vee Q$, $g(\alpha_1, \ldots, \alpha_p)$

$A \to A/X$: for $P \to P[X]$, (P in induced subgraph on X)

$A \to A/(X \cap Y) \cup (Y \cap Z)^c$ for relativization to *set terms*

image automaton: $A \to pr(A)$: in the transitions of A, each function symbol f is replaced by pr(f) ;

makes pr(A) *nondeterministic* : for $P(\underline{X}) \to \exists \underline{X}.P(\underline{X})$

Procedures : to build automata that compute functions:

#$\underline{X}$.P($\underline{X}$) : the number of tuples $\underline{X}$ that satisfy P($\underline{X}$) in the input term or graph.

Sp$\underline{X}$.P($\underline{X}$) defined as the set of tuples of cardinalities of the components of the $\underline{X}$ that satisfy P($\underline{X}$).

MSp$\underline{X}$.P($\underline{X}$) defined as the corresponding multiset.

SetVal$\underline{X}$.α($\underline{X}$)/P($\underline{X}$) defined as the set of values of α($\underline{X}$) for the tuples $\underline{X}$ that satisfy P($\underline{X}$).

For each case, a procedure transforms FA for P($\underline{X}$) and α($\underline{X}$) into ones computing the associated functions.

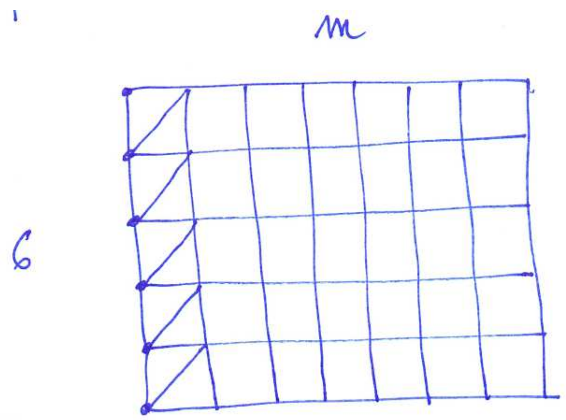
(These transformations do not depend on P($\underline{X}$) and α($\underline{X}$), but only on the relevant signature.)

Counting 3-colorings of grids

grid	time	number
6x8	7 s (*)	$\simeq 3\times10^{10}$
6x60	120 s	$\simeq 4\times10^{73}$
6x525	300 s	

(*) : also spectrum.

Counting 3-colorings of modified grids



Modified grid	Time	Number
6x10	3 s	$\simeq 7\times10^{9}$
6x20	8 s	$\simeq 10^{22}$
6x30	13 s	$\simeq 2\times10^{34}$
6x40	18 s	$\simeq 2\times10^{46}$

Exact numbers are computed.

Spectra of 3-colorings of modified grids (set of triples of cardinalities of colors)

Modified grid	Time	Number of triples
6x10	5 s	253
6x20	207 s	1378
6x30	??	Out of memory

Conclusion

We get XP algorithms in most cases, that can be obtained independently. We get FPT ones in some cases.

We have tools for constructing automata from logical descriptions → flexibility. Constructions of automata are implemented. Tests have been made for colorability and connectedness problems.

Thank you for suggesting algorithmic problems that could fit in this framework.

Further topics to present or to investigate :

Using *directed acyclic graphs* for sharing identical subterms. Deterministic FA run on dags. The question is to transform a term so as to get a dag with “few” nodes, perhaps by loosing on the number of labels.

Handling directly tree-decompositions in the same way, and the variant of monadic second-order logic allowing edge set quantifications.

A technical difficulty comes from *parallel composition* (the operation G//H) because one vertex "comes from" several positions in the term. The notion of special tree-width (weaker than tree-width) handles this problem. See Part 4 of this set of slides.

Comparing several logical expressions of a same problem.

For example the number cc(G) of connected components of a graph G can be computed as :

$\#X.(\mathrm{Conn}[X] \wedge X \text{ not empty} \wedge \neg \mathrm{Link}(X,X^c)),$

or as $\log_2(\#X.\neg \mathrm{Link}(X,X^c)),$

or by a FA (already sketched in Part 1) that modifies the one for connectedness.

Is there a generalization to other functions of this observation ?