Coq/Ssreflect for large case-based graph theory proofs *

Ricardo D. Katz[1,3] and Daniel Severín[2,3]

[1] Centro Internacional Franco-Argentino de Ciencias de la Información y de Sistemas (CIFASIS), Argentina
katz@cifasis-conicet.gov.ar
[2] Facultad de Cs. Exactas, Ingeniería y Agrimensura, Universidad Nacional de Rosario, Argentina
daniel@fceia.unr.edu.ar
[3] CONICET, Argentina

Introduction. In a research in progress on a graph parameter, we came across two problems in which the use of proof assistants turns out to be especially useful.

The first one concerns some long proofs, involving a large number of cases that can be checked mechanically. In such cases the use of a proof assistant is convenient because readers or reviewers should not need to check a large number of cases to be sure of the veracity of the corresponding results, and focus their attention on those that could be more interesting or important to check or learn. An example of this is given by one of the longest proofs ever formalized in Coq: the *Four Color Theorem* [2]. We recall that this theorem was first proved by K. Appel and W. Haken and it had the peculiarity that it was necessary to prove thousands of cases, called *configurations*, although this task could be carried out mechanically by a computer. Later, the number of configurations was reduced to 633 but it is still a large number to be considered "manually verifiable". The formalization of such a result in a proof assistant such as Coq has the advantage of reducing trust only to the proof assistant, without involving other software. That is, skeptical readers neither need to trust in a program that checks all the configurations nor have to make their own program to convince themselves of the correctness of the theorem, they only need to trust in the proof assistant.

The second one concerns the possibility of generating a *certificate* of a claim on the numerical value of a given graph parameter. This idea came to our mind after knowing many works about optimization problems in graphs where the authors give, for example, a new bound of some parameter that improves the existing ones, but one has to trust what the authors claim. The generation of certificates has already been explored in other contexts, e.g. Laurent Thery's coqprime tool certifies the primality of big numbers.

The Maximum Weighted Irredundant Set (MWIS) problem. In order to describe our work in more detail, we next introduce the considered problem.

Given a simple graph $G = (V, E)$ and a vertex $v \in V$, we denote by $N[v]$ the *closed neighborhood* $\{u : (v, u) \in E\} \cup \{v\}$ of v. If $v \in V$ and $u \in N[v]$, then it is said that v *dominates* u. Given a set $D \subseteq V$ and a vertex $v \in D$, $s_D(v) \doteq N[v] - (\bigcup_{u \in D, u \neq v} N[u])$ is the set of *private vertices* of v in D (it contains those vertices only dominated by v). The set D is called *irredundant* if $s_D(v) \neq \emptyset$ for all $v \in D$. In other words, each vertex of D must dominate at least one vertex not dominated by any other vertex of D. The *upper irredundance number* $IR(G)$ is the maximum cardinality of an irredundant set in G. This parameter can be generalized as follows: if a positive weight $w(v)$ is associated with each vertex v of G, then the objective is to find an irredundant set D such that $\sum_{v \in D} w(v)$, i.e., the weight of D, is maximized. This is the parameter associated with the MWIS problem, which is denoted by $IR_w(G)$. Its definition was previously formalized in Coq/Ssreflect, as part of a generalization of the *Cockayne-Hedetniemi domination chain* [4], now part of the graph library made by Doczkal and Pous [1].

Some complexity results on the MWIS problem are based on a *transformed* graph, denoted $G' = (V', E')$, which is defined in terms of G as follows: $V' \doteq \{uv : u \in V,\ v \in N[u]\}$ and $E' \doteq \{(uv, zr) : uv, zr \in V',\ uv \neq zr,\ v \in N[z] \vee r \in N[u]\}$. In particular, two complexity results require the proof of the following lemmas:

Lemma 1. *G has a $K_{1,3}$, a bull, a P_6 or a $\overline{C}_6$ if and only if G' has a $K_{1,3}$.*

Lemma 2. *G has a co-paw, a G_1^7, or a G_2^7 if and only if G' has a co-paw.*

where the mentioned graphs are displayed in Figure 1. The sufficiency parts of these lemmas are rather easy

*This work was supported by grant PID-UNR 80020180100091UR.

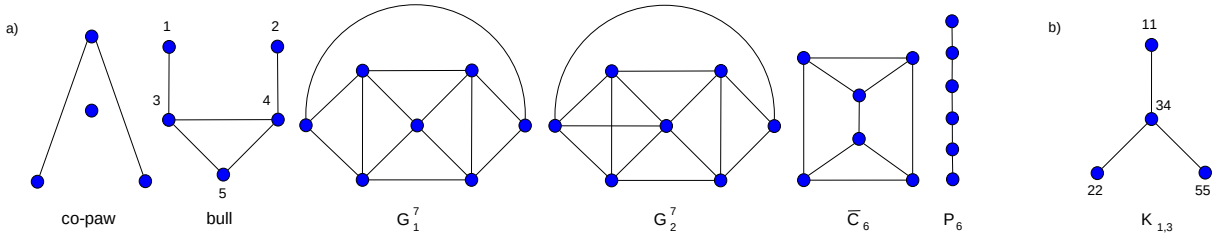

Figure 1: a) The co-paw, bull, two graphs of order 7, $\overline{C}_6$ and P_6, b) a $K_{1,3}$ in G' induced by a bull in G.

to prove (e.g., Figure 1(b) shows which $K_{1,3}$ has G' if G contains a bull whose vertices are enumerated as in Figure 1(a)) but, on the contrary, we were unable to find simple proofs of their necessity parts. The current proofs of the latter are based on a large number of cases that (although each one can be systematically proved) makes it almost impossible to present a proper pen-and-paper version of them. Indeed, due to the number of cases, it is difficult to be sure that all the possible cases are considered and that each of them is tackled properly. It would be even harder for a reader to check such proofs.

Formalization. For the work in progress, we formalized several aspects of the MWIS parameter in the folder `mwis` from [3] and, in particular, the two lemmas above. It consists of 7108 lines of Coq code (ommiting blank lines and comments), most of which are devoted to proving only the necessity parts of Lemmas 1 and 2, called `G'clawfree_rev` and `G'copawfree_rev` respectively. Other parts of interest of our code are: the construction of the transformed graph G' (see above) given in section `TrfGraph_construction`, the definition of the graphs of Figure 1 among others in section `Graph_definitions` and the sufficiency parts of Lemmas 1 and 2, called `G'clawfree` and `G'copawfree` respectively.

Certificates. On the other hand, we have devised a tool to generate a Coq file that certifies the solution obtained for a given instance, see folder `solver`. Note that this feature is available thanks to the formalization of the existing theory. The file `mwis/check_ir.v` of [3] contains some functions and results that allow us to check that a given irredundant set is indeed irredundant for a given graph, and has a certain cardinality/weight depending on whether we want to get a lower bound of $IR(G)$ or $IR_w(G)$. Finally, the files in the folder `certs` contain the certificates generated by our tool for several instances used for benchmarking purposes. For example, `certs/myciel4.v` has the definition of Mycielsky graph of order 23 (`inst`), an irredundant set composed of 11 vertices (`inst_set`), the same set as a list (`inst_list`) and the proof that $IR(G) \geq 11$. Below, we display the statements that give rise to that result. The whole file is quite intuitive, so much so that even a graph theorist who is barely familiarized with Coq can understand the main parts.

```
Fact inst_list_eq_inst_set : inst_list =i inst_set.
Fact inst_set_card : #|inst_set| = 11.
Fact inst_set_is_irr : @irredundant inst inst_set.
Fact IR_lb : IR inst ≥ 11.
```

Considering a list-version of the irredundant set allows us to prove some facts by computation.

We would like to thank the anonymous referees for their suggestions, in particular one of them dramatically improved the checking time of certificates.

References

[1] C. Doczkal and D. Pous. Graph theory in coq: Minors, treewidth, and isomorphisms. *J. Autom. Reasoning*, 64:795–825, 2020.

[2] G. Gonthier. Formal proof - the Four-Color Theorem. *Notices Amer. Math. Soc.*, 55:1382–1393, 2008.

[3] R. Katz and D. Severín. Supplementary material: https://github.com/aureus123/graph-theory/tree/mwis.

[4] D. Severín. Formalization of the domination chain with weighted parameters (short paper). *Lebniz. Int. Proc. Inform.*, 141:36:1–36:7, 2019.