Reasoning about connectivity without paths

Alberto Casagrande and Eugenio G. Omodeo

UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dip. Matematica e Geoscienze — DMI

Reasoning about connectivity without paths[1]

Alberto Casagrande and Eugenio G. Omodeo

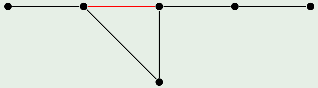
UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Dip. Matematica e Geoscienze — DMI

[1]Work partially funded by: INdAM/GNCS 2013, FRA-UniTS 2012 PUMA

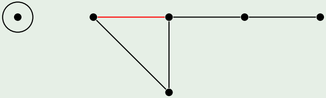
Every connected graph has non-cut vertices

Example



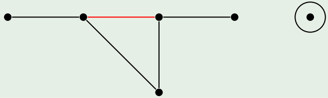
Every connected graph has non-cut vertices

Example



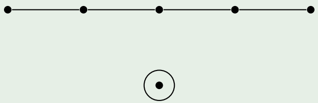
Every connected graph has non-cut vertices

Example



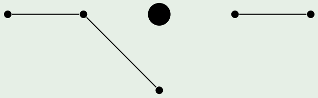
Every connected graph has non-cut vertices

Example



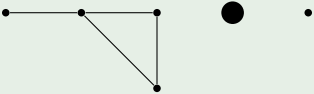
Every connected graph has non-cut vertices

Example



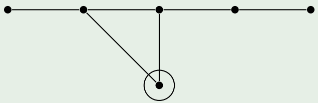
Every connected graph has non-cut vertices

Example



Every connected graph has non-cut vertices

Example



More generally:

'Every connected (finite) $\boxed{\text{hyper}}$graph has at least one vertex whose removal does not disrupt connectivity'

(Ideal) itinerary of this talk

I Connectivity and non-cut vertices

II Applications (one in particular. . .)

III The proof assistant Ref

IV Our proof-verification experiment
`http://www2.units.it/eomodeo/NonCutVertices.html`
`http://aetnanova.units.it/scenarios/NonCutVertices/`

$\boxed{\text{Connectivity}}$ plays a crucial role in many fields.

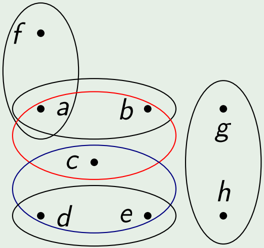
Es. The number of connected components of a graph

- is a topological invariant;
- corresponds to the multiplicity of the eigenvalue 0 in its Laplacian;
- is related to the number of its claw-free subgraphs [CPR07].

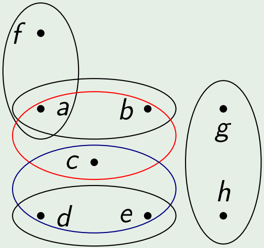
$\therefore$ Large scale proof-verification efforts [Wie07, SCO11] must formally investigate this notion.

How we see a hypergraph

Example

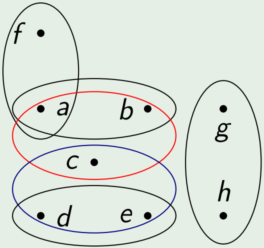


EXAMPLE



The edges of G belong to G .

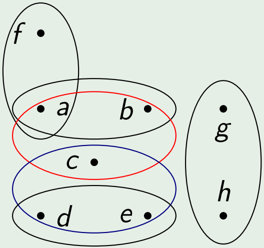
Example



$$\text{The} \left\{ \begin{array}{c} \text{vertices} \\ \text{or} \\ \text{nodes} \end{array} \right\} \text{of } G \text{ belong to } \{v : e \in G, v \in e\}\,.$$

How we see a hypergraph

Example



In a graph, the edges have cardinality 2 .

-
-
-

Def. $\mathsf{Conn}(G) \quad \leftrightarrow_{\mathsf{Def}} \quad \{p \subseteq G \mid \mathsf{nodes}(p) \cap \mathsf{nodes}(G \backslash p) = \emptyset\} \subseteq \{\emptyset, G\} \;\&\; \mathsf{HGraph}(G)$

Our formal definition of connectivity

.

Def. $\mathsf{nodes}(G) \quad =_{\mathsf{Def}} \quad \bigcup G$

.

Def. $\mathsf{Conn}(G) \quad \leftrightarrow_{\mathsf{Def}} \quad \{p \subseteq G \mid \mathsf{nodes}(p) \cap \mathsf{nodes}(G \setminus p) = \emptyset\} = \{\emptyset, G\}$ & $\mathsf{HGraph}(G)$

.

Def. $\mathsf{nodes}(G) =_{\mathsf{Def}} \bigcup G$

Def. $\mathsf{HGraph}(G) \leftrightarrow_{\mathsf{Def}} \langle \forall e \in G \mid \mathsf{CardAtLeast2}(e) \rangle \;\&\; \mathsf{Finite}\big(\mathsf{nodes}(G)\big)$

Def. $\mathsf{Conn}(G) \leftrightarrow_{\mathsf{Def}} \{p \subseteq G \mid \mathsf{nodes}(p) \cap \mathsf{nodes}(G \setminus p) = \emptyset\} = \{\emptyset, G\} \;\&\; \mathsf{HGraph}(G)$

Our formal definition of connectivity

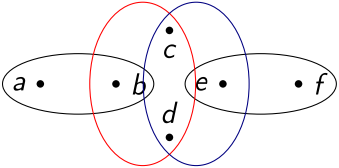
Def. $\mathsf{CardAtLeast2}(\mathsf{E}) \leftrightarrow_{\mathsf{Def}}$ $\mathsf{E} \not\subseteq \{\mathbf{arb}(\mathsf{E})\}$

Def. $\mathsf{nodes}(\mathsf{G}) =_{\mathsf{Def}}$ $\bigcup \mathsf{G}$

Def. $\mathsf{HGraph}(\mathsf{G}) \leftrightarrow_{\mathsf{Def}}$ $\langle \forall \mathsf{e} \in \mathsf{G} \mid \mathsf{CardAtLeast2}(\mathsf{e}) \rangle \ \& \ \mathsf{Finite}\big(\mathsf{nodes}(\mathsf{G})\big)$

Def. $\mathsf{Conn}(\mathsf{G}) \leftrightarrow_{\mathsf{Def}}$ $\{\mathsf{p} \subseteq \mathsf{G} \mid \mathsf{nodes}(\mathsf{p}) \cap \mathsf{nodes}(\mathsf{G} \setminus \mathsf{p}) = \emptyset\} = \{\emptyset, \mathsf{G}\} \ \& \ \mathsf{HGraph}(\mathsf{G})$

Does this hypergraph have a spanning tree ?



-
-
-

Thm. $\mathsf{Conn}(\mathsf{G})$ & $\mathsf{G} \not\subseteq \{\mathbf{arb}(\mathsf{G})\} \to \langle \exists \mathsf{v} \in \mathsf{nodes}(\mathsf{G}) \mid \mathsf{NonCut}(\mathsf{G}, \mathsf{v}) \rangle$

.

.

Def. $\mathsf{NonCut}(\mathsf{G},\mathsf{V}) \quad \leftrightarrow_{\mathsf{Def}} \quad \mathsf{Conn}\big(\mathsf{rmv}(\mathsf{G},\mathsf{V})\big) \,\&\, \mathsf{lost}(\mathsf{G},\mathsf{V}) = \emptyset$

$$\boxed{\text{Thm.} \quad \mathsf{Conn}(\mathsf{G}) \,\&\, \mathsf{G} \not\subseteq \{\mathbf{arb}(\mathsf{G})\} \rightarrow \big\langle \exists \mathsf{v} \in \mathsf{nodes}(\mathsf{G}) \mid \mathsf{NonCut}(\mathsf{G},\mathsf{v})\big\rangle}$$

Existence of non-cut vertices

Def. $\mathsf{rmv}(G, V) \quad =_{\mathsf{Def}} \quad \{e \backslash \{V\} : e \in G \mid \mathsf{CardAtLeast2}(e \backslash \{V\})\}$

.

Def. $\mathsf{NonCut}(G, V) \quad \leftrightarrow_{\mathsf{Def}} \quad \mathsf{Conn}\big(\mathsf{rmv}(G, V)\big) \,\&\, \mathsf{lost}(G, V) = \emptyset$

Thm. $\mathsf{Conn}(G) \,\&\, G \not\subseteq \{\mathbf{arb}(G)\} \rightarrow \big\langle \exists v \in \mathsf{nodes}(G) \mid \mathsf{NonCut}(G, v)\big\rangle$

Existence of non-cut vertices

Def. $\mathsf{rmv}(G, V) =_{\mathsf{Def}} \{e \backslash \{V\} : e \in G \mid \mathsf{CardAtLeast2}(e \backslash \{V\})\}$

Def. $\mathsf{lost}(G, V) =_{\mathsf{Def}} \mathsf{nodes}(G) \backslash \big(\mathsf{nodes}\big(\mathsf{rmv}(G, V)\big) \cup \{V\}\big)$

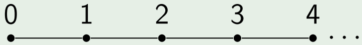
Def. $\mathsf{NonCut}(G, V) \leftrightarrow_{\mathsf{Def}} \mathsf{Conn}\big(\mathsf{rmv}(G, V)\big) \;\&\; \mathsf{lost}(G, V) = \emptyset$

Thm. $\mathsf{Conn}(G) \;\&\; G \not\subseteq \{\mathbf{arb}(G)\} \rightarrow \big\langle \exists v \in \mathsf{nodes}(G) \mid \mathsf{NonCut}(G, v) \big\rangle$

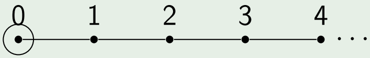
How can we:

- *walk* along an infinite acyclic path?
- *visit* all vertices of a finite acyclic path?

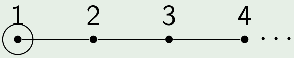
Example



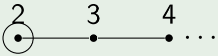
Example



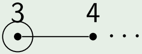
Example



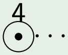
EXAMPLE



Example



Example



Example

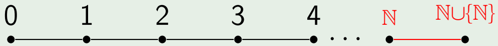
This amounts to repeatedly picking and removing a non-cut vertex (the *only one*, in this case) from a graph (*infinite* in this case)

Example

This amounts to repeatedly picking and removing a non-cut vertex (the *only one*, in this case) from a graph (*infinite* in this case)

Why such a silly example ?

Example



Where does the difference between $\mathbb{N}$ and $\mathbb{N} \cup \{\mathbb{N}, \mathbb{N} \cup \{\mathbb{N}\}\}$ lie ?

A more revealing application

Example

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

A more revealing application

Example

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

1. Picking & removing a $\boxed{\text{non-cut}}$ vertex from a connected graph

A more revealing application

Example

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

1. Picking & removing a $\boxed{\text{non-cut}}$ vertex from a connected graph
2. $\boxed{\text{recursively}}$ getting a spanning tree for the resulting graph

EXAMPLE

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

1. Picking & removing a $\boxed{\text{non-cut}}$ vertex from a connected graph
2. $\boxed{\text{recursively}}$ getting a spanning tree for the resulting graph
3. restoring the removed vertex, along with $\boxed{\text{one}}$ of the edges incident to it

A more revealing application

Example

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

1. Picking & removing a $\boxed{\text{non-cut}}$ vertex from a connected graph
2. $\boxed{\text{recursively}}$ getting a spanning tree for the resulting graph
3. restoring the removed vertex, along with $\boxed{\text{one}}$ of the edges incident to it

0. In the $\boxed{\text{base case}}$, the spanning tree consists of the (sole) edge

Example

We can get a $\boxed{\text{spanning tree}}$ for a connected nonnull graph by:

1. Picking & removing a $\boxed{\text{non-cut}}$ vertex from a connected graph
2. $\boxed{\text{recursively}}$ getting a spanning tree for the resulting graph
3. restoring the removed vertex, along with $\boxed{\text{one}}$ of the edges incident to it

0. In the $\boxed{\text{base case}}$, the spanning tree consists of the (sole) edge

This example is paradigmatic:

Inductive proofs on connected graphs usually follow this pattern

An achievement, but also a pending proof obligation:

An achievement, but also a pending proof obligation:


Set Graphs. III. Proof Pearl: Claw-Free Graphs Mirrored into Transitive Hereditarily Finite Sets

Eugenio G. Omodeo & Alexandru I. Tomescu

Journal of Automated Reasoning

ISSN 0168-7433
Volume 52
Number 1

J Autom Reasoning (2014) 52:1-29
DOI 10.1007/s10817-012-9272-3

Journal of Automated Reasoning

Springer Vol. 52 No. 1 2014

Springer

Two earlier proof-pearl scenarios

Two fully formal reconstructions of results on *connected claw-free graphs* have been achieved by means of Ref.

Two fully formal reconstructions of results on *connected claw-free graphs* have been achieved by means of Ref.

E. G. Omodeo and A. I. Tomescu.
Set graphs. III. Proof Pearl: Claw-free graphs mirrored into transitive hereditarily finite sets.
J. Autom. Reason., 52(1), pp.1–29, 2014.
Cf. http://www2.units.it/eomodeo/ClawFreeness.html

E. G. Omodeo and A. I. Tomescu.
Appendix: Claw-free graphs as sets.
In: M. Davis, E. Schonberg (eds.) From Linear Operators to Computational Biology: Essays in Memory of Jacob T. Schwartz, pp. 131–167, Springer, 2012.

Two earlier proof-pearl scenarios

Two fully formal reconstructions of results on *connected claw-free graphs* have been achieved by means of Ref.

E. G. Omodeo and A. I. Tomescu.
Set graphs. V. On representing graphs as membership digraphs.
To appear on *J. Log. Comput.*
Cf. `http://www2.units.it/eomodeo/GraphsViaMembership.html`

E. G. Omodeo and A. I. Tomescu.
Set graphs. III. Proof Pearl: Claw-free graphs mirrored into transitive hereditarily finite sets.
J. Autom. Reason., 52(1), pp.1–29, 2014.
Cf. `http://www2.units.it/eomodeo/ClawFreeness.html`

E. G. Omodeo and A. I. Tomescu.
Appendix: Claw-free graphs as sets.
In: M. Davis, E. Schonberg (eds.) From Linear Operators to Computational Biology: Essays in Memory of Jacob T. Schwartz, pp. 131–167, Springer, 2012.

DEFINITION

A graph (V, E) is said to be $\boxed{\textit{claw-free}}$ if none of its subgraphs induced by 4 vertices has the shape of a 'Y'

Definition

A graph (V, E) is said to be $\boxed{\textit{claw-free}}$ if none of its subgraphs induced by 4 vertices has the shape of a 'Y'

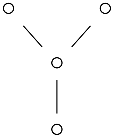


Figure: Forbidden claw $K_{1,3}$

Definition

A graph (V, E) is said to be $\boxed{\textit{claw-free}}$ if none of its subgraphs induced by 4 vertices has the shape of a 'Y'

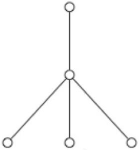


Figure: Worse than a claw

Definition

A graph (V, E) is said to be $\boxed{\textit{claw-free}}$ if none of its subgraphs induced by 4 vertices has the shape of a 'Y'

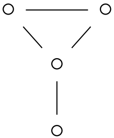
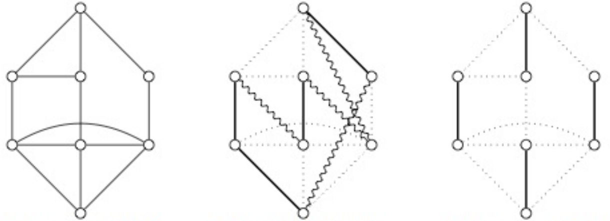


Figure: A claw-free graph

Classical results about claw-free graphs

"Every connected claw-free graph admits a perfect matching and has a Hamiltonian cycle in its square".

(1970s / 1980s)

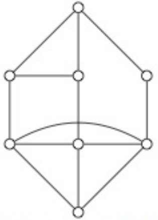


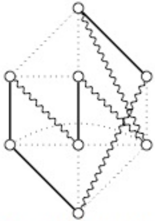
(a) a connected claw-free graph G (b) a Hamiltonian cycle in G^2 (c) a perfect matching in G

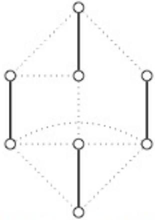
Classical results about claw-free graphs

"Every connected claw-free graph admits a perfect matching and has a Hamiltonian cycle in its square".

(1970s / 1980s)


(a) a connected claw-free graph G

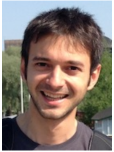

(b) a Hamiltonian cycle in G^2


(c) a perfect matching in G

Also: Each connected claw-free graph has a vertex-pancyclic square

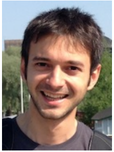
Novel proofs based on a repr'n theorem

Martin Milanič and A. I. Tomescu found new, simpler proofs of the results just mentioned via a theorem about the representation of edges as (directed!) membership arcs.



Martin Milanič and A. I. Tomescu found new, simpler proofs of the results just mentioned via a theorem about the representation of edges as (directed!) membership arcs.

M. Milanič and A. I. Tomescu.
Set graphs. I. Hereditarily finite sets and $\boxed{\text{extensional}}$ $\boxed{\text{acyclic}}$ orientations.
Discrete Applied Mathematics, 161(4-5):677–690, 2013.

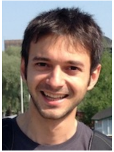


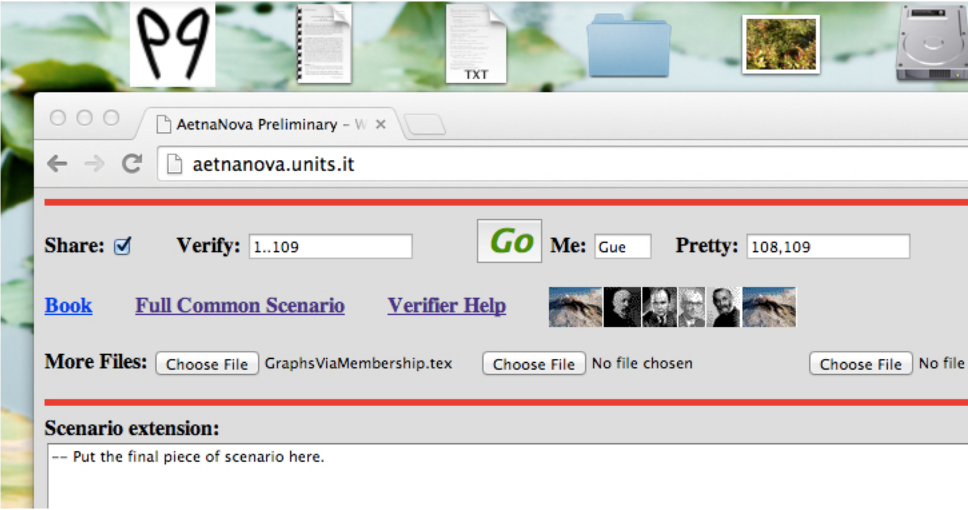
Novel proofs based on a repr'n theorem

Martin Milanič and A. I. Tomescu found new, simpler proofs of the results just mentioned via a theorem about the representation of edges as (directed!) membership arcs.

M. Milanič and A. I. Tomescu.
Set graphs. I. Hereditarily finite sets and $\boxed{\text{extensional}}$ $\boxed{\text{acyclic}}$ orientations.
Discrete Applied Mathematics, 161(4-5):677–690, 2013.

A. I. Tomescu.
A simpler proof for vertex-pancyclicity of squares of connected claw-free graphs.
Discrete Mathematics, 312(15):2388–2391, 2012.





(On-line worksheet)

Share: ☐ **Verify:** 1..2 **Go** **Me:** Eug **Pretty:**

Book Full Common Scenario Verifier Help

More Files: Choose File No file chosen Choose File No file chosen

Scenario extension:

```
--BEGIN HERE
--
Def pow: [Family of all subsets of a given set] pow(S) := { x : x •incin S }
--
Def trans: [Transitivity w.r.t. membership] Trans(T) := {y in T | y •nincin T} = 0
--
Theorem trans_1: [Peddicord's lemma] (Trans(T) & (S •incin T) & (S /= T)) •imp (0 /= (T-S) * pow(S)). Proof:
Suppose_not(t0,s0) ==> AUTO
   Use_def(pow(s0)) ==> AUTO
   Loc_def ==> Stat1: a = arb(t0-s0)
   Use_def(Trans) ==> Stat2:  ({y in t0 | y •nincin t0} = 0) & (a notin {z: z •incin s0})
   (a,a)-->Stat2(Stat1) ==> false
   Discharge ==> QED
--
Theorem trans_2: [Peddicord's lemma, main corollary] Trans(T) •imp (0 in T). Proof:
Suppose_not(t0) ==> AUTO
   (t0,0)-->Ttrans_1 ==> Stat1: 0 /= t0 * pow(0)
   Use_def(pow(0)) ==> AUTO
   z-->Stat1 ==> Stat2: (z in { x : x •incin 0 }) & (z in t0)
   z1-->Stat2 ==> false
   Discharge ==> QED

--END HERE
```

Misc. Init. Output ○ **Good Pfs.** ○ **Bad** ⊙

	Thm. No.	Thm. Name	Line No.	F/A	Line
⊙	2	Ttrans_2	2	F	(t0,0)-->Ttrans_1 ==> Stat1: 0 /= t0 * pow(0) Attempt to derive MLS contradiction has failed

++++++++ Ttrans_2#2 -- 26 ****** Error verifying step: 2 of theorem Ttrans_2

(t0,0)-->Ttrans_1 ==> Stat1: 0 /= t0 * pow(0)

Attempt to derive MLS contradiction has failed

reduced blobbed statement was: BLB_{1} & $\neg\{\} \in T_0$ & $\neg\{\} \neq T_0 \cap BLB\_{2}$ & (BLB_{1} & $\{\} \neq T_0 \rightarrow \{\} \neq$

Misc. Init. Output ○ **Good Pfs.** ○ **Bad** ◉

	Thm. No.	Thm. Name	Line No.	F/A	Line
◉	2	Ttrans_2	2	F	(t0,0)-->Ttrans_1 ==> Stat1: 0 /= t0 * pow(0)
					Attempt to derive MLS contradiction has failed

++++++++ Ttrans_2#2 -- 26 ****** Error verifying step: 2 of theorem Ttrans_2

(t0,0)-->Ttrans_1 ==> Stat1: 0 /= t0 * pow(0)

Attempt to derive MLS contradiction has failed

reduced blobbed statement was: $\mathrm{BLB\_}_1$ & $\neg\{\} \in T_0$ & $\neg\{\} \neq T_0 \cap \mathrm{BLB\_}_2$ & ($\mathrm{BLB\_}_1$ & $\{\} \neq T_0 \rightarrow \{\} \neq$

Misc. Init. Output ○ **Good Pfs.** ◉ **Bad** ○

Thm. No.	Thm. Name	Time	Details	
1	Ttrans_1	#6: t20	Z4: 7	S5: 8
2	Ttrans_2			

DEFINITION: (SHORTHAND)

-- After the celebrated paper *Sur les ensembles fini* (Tarski, 1924)

Def. Finite(F) $\leftrightarrow_{\text{Def}}$ $\langle \forall g \in \mathcal{P}(\mathcal{P}(F)) \setminus \{\emptyset\},\ \exists m \mid g \cap \mathcal{P}(m) = \{m\} \rangle$

DEFINITION: ($\in$-RECURSION HERE!)

-- "The cardinality of S exceeds M"

Def. $\mathsf{Exc}(\mathsf{S},\mathsf{M}) \leftrightarrow_{\mathsf{Def}} \mathsf{S}\neq\emptyset \;\&\; \big\{p \in \mathsf{M} \mid \neg\mathsf{Exc}\big(\mathsf{S}\setminus\{\mathbf{arb}(\mathsf{S})\}, p\big)\big\} = \emptyset$

THEOREM AND PROOF: (MONOTONICITY OF FINITUDE)

Thm fin_0. $Y \supseteq X \mathbin{\&} \text{Finite}(Y) \rightarrow \text{Finite}(X)$. Proof:

$\text{Suppose\_not}(y_0, x_0) \Longrightarrow \quad y_0 \supseteq x_0 \mathbin{\&} \text{Finite}(y_0) \mathbin{\&} \neg\text{Finite}(x_0)$

$\langle y_0, x_0 \rangle \hookrightarrow T\text{pow}_1 \Longrightarrow \quad \mathcal{P}y_0 \supseteq \mathcal{P}x_0$

$\text{Use\_def}(\text{Finite}) \Longrightarrow \quad \textit{Stat1} : \neg\langle \forall g \in \mathcal{P}(\mathcal{P}x_0)\backslash\{\emptyset\}, \exists m \mid$
$g \cap \mathcal{P}m = \{m\}\rangle \mathbin{\&} \langle \forall g' \in \mathcal{P}(\mathcal{P}y_0)\backslash\{\emptyset\}, \exists m \mid g' \cap \mathcal{P}m = \{m\}\rangle$

$\langle \mathcal{P}y_0, \mathcal{P}x_0 \rangle \hookrightarrow T\text{pow}_1 \Longrightarrow \quad \mathcal{P}(\mathcal{P}y_0) \supseteq \mathcal{P}(\mathcal{P}x_0)$

$\langle g_0, g_0 \rangle \hookrightarrow \textit{Stat1}(\textit{Stat1}\star) \Longrightarrow \quad \neg\langle \exists m \mid g_0 \cap \mathcal{P}m = \{m\}\rangle \mathbin{\&}$
$\langle \exists m \mid g_0 \cap \mathcal{P}m = \{m\}\rangle$

Discharge $\Longrightarrow$ Qed

4^{th}, MAJOR CONSTITUENT OF A SCENARIO (EXAMPLE)

A CONSTRUCT FOR PROOF REUSE

Theory finite_image (s_0 , $g(X)$)

    Finite(s_0)

End finite_image

Enter_theory finite_image

$\vdots \quad \vdots \quad \vdots \quad \vdots$

Enter_theory Set_theory

4^{th}, major constituent of a scenario (example)

A construct for proof reuse

Theory finite_image (s_0 , $g(X)$)

  Finite(s_0)

End finite_image

Enter_theory finite_image

$\vdots \quad \vdots \quad \vdots \quad \vdots$

Enter_theory Set_theory

Within a scenario, the discourse can momentarily digress into a 'Theory' that enforces certain local assumptions.
At the end of the digression, the upper theory will be re-entered.

A CONSTRUCT FOR PROOF REUSE

Theory finite_image $(s_0\,,\ g(X))$

$\quad \mathrm{Finite}(s_0)$

$\Longrightarrow \quad (f_\Theta)$

$\quad \mathrm{Finite}(\{\, g(x) : x \in s_0 \,\})$

$\quad f_\Theta \subseteq s_0 \;\;\&\;\; \langle \forall t \subseteq f_\Theta \mid g(t) = g(s_0) \leftrightarrow t = f_\Theta \rangle$

End finite_image

As an outcome of the digression, the Theory will be able to instantiate new theorems

A construct for proof reuse

Theory finite_image $(s_0, g(X))$

$\quad \mathsf{Finite}(s_0)$

$\Longrightarrow \; (f_\Theta)$

$\quad \mathsf{Finite}(\{g(x) : x \in s_0\})$

$\quad f_\Theta \subseteq s_0 \;\; \& \;\; \langle \forall t \subseteq f_\Theta \mid g(t) = g(s_0) \leftrightarrow t = f_\Theta \rangle$

End finite_image

As an outcome of the digression, the Theory will be able to instantiate new theorems: possibly involving new symbols, whose definition it encapsulates.

Our experiment, in digits

The script-file containing our verified formal derivation of the existence of non-cut vertices in hypergraphs:

- comprises 13 definitions;
- proves 46 theorems (only two whose length exceeds 50 lines),
- organized in 3 Theorys.
- Its processing takes ca. 4 seconds;
- the overall number of proof lines is 905.

http://www2.units.it/eomodeo/NonCutVertices.html

http://aetnanova.units.it/scenarios/NonCutVertices/

Proof-verification can highly benefit from representation theorems of the kind illustrated by the Milanič–Tomescu result about connected, claw-free graphs.

Conclusions and future work

Proof-verification can highly benefit from representation theorems of the kind illustrated by the Milanič–Tomescu result about connected, claw-free graphs.

On the human side, such results disclose new insights by shedding light on a discipline from unusual angles

Proof-verification can highly benefit from representation theorems of the kind illustrated by the Milanič–Tomescu result about connected, claw-free graphs.

ON THE HUMAN SIDE, such results disclose new insights by shedding light on a discipline from unusual angles

ON THE TECHNOLOGICAL SIDE, they enable the transfer of proof methods from one realm of mathematics to another.

Proof-verification can highly benefit from representation theorems of the kind illustrated by the Milanič–Tomescu result about connected, claw-free graphs.

On the human side, such results disclose new insights by shedding light on a discipline from unusual angles

on the technological side, they enable the transfer of proof methods from one realm of mathematics to another.

> This contribution closes a cycle of activities related to claw-free graphs. . .

Proof-verification can highly benefit from representation theorems of the kind illustrated by the Milanič–Tomescu result about connected, claw-free graphs.

On the human side, such results disclose new insights by shedding light on a discipline from unusual angles

on the technological side, they enable the transfer of proof methods from one realm of mathematics to another.

> This contribution closes a cycle of activities related to claw-free graphs. . .
> . . . and paves the way to an extensive exploration on how to formalize hypergraphs .

Thank you for your attention!

Jacob T. Schwartz, Domenico Cantone, and Eugenio G. Omodeo. *Computational Logic and Set Theory – Applying formalized Logic to Analysis*. Springer, 2011. Foreword by Martin Davis.

Freek Wiedijk. The QED Manifesto revisited. *Studies in Logic, Grammar and Rhetoric*, 10(23):121–133, 2007.

Gab-Byung Chae, Edgar M. Palmer, and Robert W. Robinson. Counting labeled general cubic graphs. *Discrete Mathematics*, 307(23):2979–2992, 2007.