Unique perfect matchings, structure from acyclicity and proof nets

Nguyễn Lê Thành Dũng (a.k.a. Tito) — nltd@nguyentito.eu
LIPN, Université Paris 13

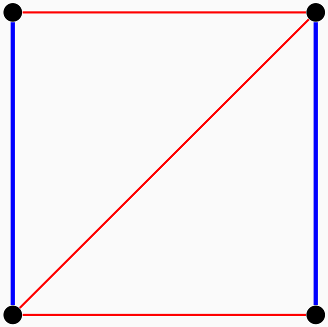
Computational Logic and Applications, Versailles, July 2nd, 2019

Perfect matchings (1)

Definition

A *perfect matching* is a set of edges in a graph such that each vertex is incident to exactly one edge in the matching.

Example below: blue edges form a perfect matching

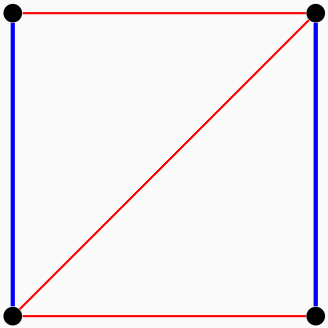


Perfect matchings (2)

An *alternating path* (resp. cycle) is a path (resp. cycle) which

- has no vertex repetitions
- alternates between edges inside and outside the matching

$\exists$ alternating cycle $\Leftrightarrow$ the perfect matching is not *unique*

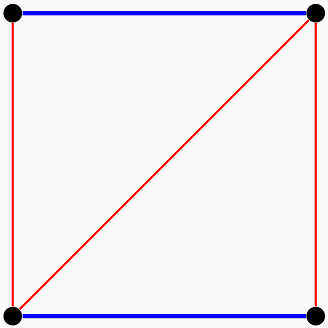


Perfect matchings (2)

An *alternating path* (resp. cycle) is a path (resp. cycle) which

- has no vertex repetitions
- alternates between edges inside and outside the matching

$\exists$ alternating cycle $\Leftrightarrow$ the perfect matching is not *unique*



Structure from acyclicity for perfect matchings

Lemma (Berge 1957[1])

No alternating cycle $\iff$ unique *perfect matching*

Theorem (Kotzig)

Every unique perfect matching contains a bridge.

Putting this together:

$$\textit{absence} \text{ of alt. cycle} \implies \textit{existence} \text{ of bridge (in matching)}$$

[1] According to Wikipedia, observed already in 1891 by Petersen.

Structure from acyclicity everywhere

Theorem (Kotzig)

Absence *of alt. cycle* $\implies$ existence *of bridge in matching*.

Szeider 2004: there are a lot of theorems of this kind that are actually *equivalent* to Kotzig's theorem.

Example:

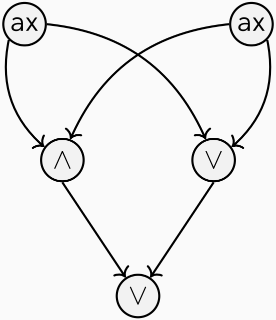
Theorem (Yeo 1997)

Every edge-colored graph ($G = (V, E)$ *with coloring* $c : E \to C$) *with no* properly colored cycle ($c(e_i) \neq c(e_{i+1})$) *contains a* color-separating vertex.

This talk: another instance from the *proof theory* of *linear logic*.

Proof structures

A *proof structure* is a DAG with node labels in $\{\mathsf{ax}, \vee, \wedge\}$.



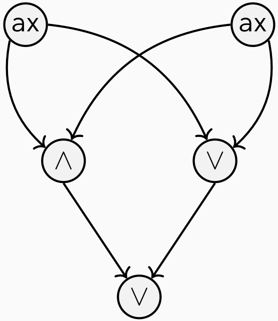
It's supposed to represent a proof in a fragment of linear logic (here, of $(A \wedge B) \vee (A^{\perp} \vee B^{\perp})$), but it might not be a *correct* proof

The correctness criterion

We need to add a condition to ensure correctness

$\longrightarrow$ Danos–Regnier *switching acyclicity*:

no undirected cycle using ≤ 1 incoming edge of each $\vee$



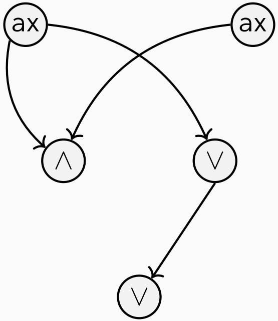
(Switching: delete 1 of the 2 incoming edges of each $\vee$ vertex)

The correctness criterion

We need to add a condition to ensure correctness

$\longrightarrow$ Danos–Regnier *switching acyclicity*:

no undirected cycle using ≤ 1 incoming edge of each $\vee$



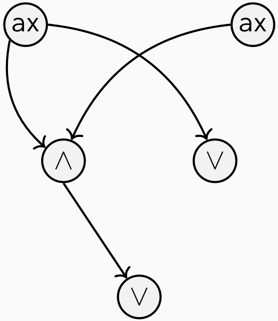
(Switching: delete 1 of the 2 incoming edges of each $\vee$ vertex)

The correctness criterion

We need to add a condition to ensure correctness

$\longrightarrow$ Danos–Regnier *switching acyclicity*:

no undirected cycle using ≤ 1 incoming edge of each $\vee$



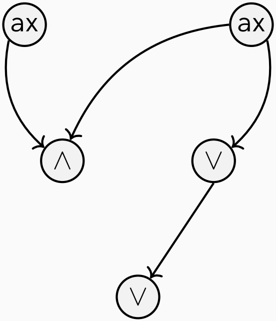
(Switching: delete 1 of the 2 incoming edges of each $\vee$ vertex)

The correctness criterion

We need to add a condition to ensure correctness

$\longrightarrow$ Danos–Regnier *switching acyclicity*:

no undirected cycle using ≤ 1 incoming edge of each $\vee$



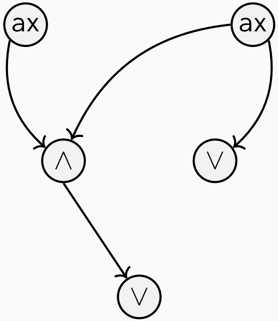
(Switching: delete 1 of the 2 incoming edges of each $\vee$ vertex)

The correctness criterion

We need to add a condition to ensure correctness

$\longrightarrow$ Danos–Regnier *switching acyclicity*:

no undirected cycle using ≤ 1 incoming edge of each $\vee$



(Switching: delete 1 of the 2 incoming edges of each $\vee$ vertex)

Proof nets and the sequentialization theorem

A *proof net* is a correct proof structure.

How do we know that this is the right notion of correctness?

Proof nets and the sequentialization theorem

A *proof net* is a correct proof structure.

How do we know that this is the right notion of correctness?

Compare with another proof formalism: *sequent calculus*.

Theorem

A proof structure is correct (i.e. switching acyclic) iff it is the translation of some proof in the MLL+Mix sequent calculus.

MLL+Mix is a fragment/variant of linear logic, extending the linear λ-calculus (proofs-as-programs correspondence)

Proof nets and the sequentialization theorem

A *proof net* is a correct proof structure.
How do we know that this is the right notion of correctness?
Compare with another proof formalism: *sequent calculus*.

Theorem

A proof structure is correct (i.e. switching acyclic) iff it is the translation of some proof in the MLL+Mix sequent calculus.

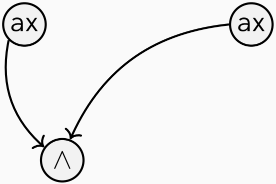
MLL+Mix is a fragment/variant of linear logic, extending the linear λ-calculus (proofs-as-programs correspondence)

structure from acyclicity for proof nets
=
sequentialization theorem

Sequentialized proof nets

Sequent calculus proofs are *inductively generated*:

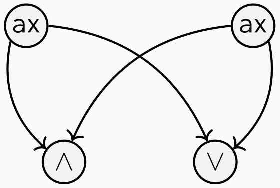
$$\dfrac{\dfrac{}{\vdash A, A^{\perp}}\ \mathsf{ax} \qquad \dfrac{}{\vdash B, B^{\perp}}\ \mathsf{ax}}{\vdash A \wedge B, A^{\perp}, B^{\perp}}\ \wedge$$



Sequentialized proof nets

Sequent calculus proofs are *inductively generated*:

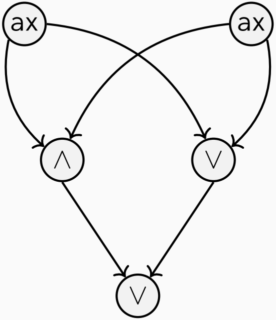
$$\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^\perp}\,\mathsf{ax}}{\dfrac{\vdash A \wedge B, A^\perp, B^\perp}{\vdash A \wedge B, A^\perp \vee B^\perp}\,\vee}\,\wedge$$



Sequentialized proof nets

Sequent calculus proofs are *inductively generated*:

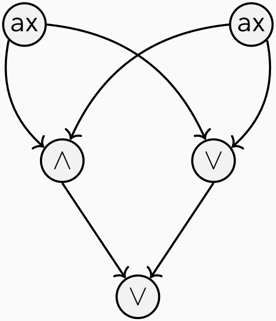
$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^\perp}\,\mathsf{ax}}{\vdash A \wedge B, A^\perp, B^\perp}\,\wedge}{\vdash A \wedge B, A^\perp \vee B^\perp}\,\vee}{\vdash (A \wedge B) \vee (A^\perp \vee B^\perp)}\,\vee
$$



Sequentialized proof nets

Sequent calculus proofs are *inductively generated*:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash A, A^\perp}\,\mathsf{ax} \qquad \dfrac{}{\vdash B, B^\perp}\,\mathsf{ax}}{\vdash A \wedge B, A^\perp, B^\perp}\,\wedge}{\vdash A \wedge B, A^\perp \vee B^\perp}\,\vee}{\vdash (A \wedge B) \vee (A^\perp \vee B^\perp)}\,\vee$$



structure from acyclicity for proof nets

=

"splitting lemma": switching acyclic $\implies$ $\exists$ final inductive rule

Proof net correctness vs perfect matching uniqueness

In the mid-90's, Christian Retoré introduced "R&B-graphs":
a translation *proof structures* $\rightsquigarrow$ *graphs w/ perfect matchings*

Theorem (Retoré's correctness criterion)

A proof structure is correct (for MLL+Mix) *iff the perfect matching of its R&B-graph is* unique*, i.e. has no alternating cycle.*

Proof net correctness vs perfect matching uniqueness

In the mid-90's, Christian Retoré introduced "R&B-graphs":
a translation *proof structures* $\rightsquigarrow$ *graphs w/ perfect matchings*

Theorem (Retoré's correctness criterion)

A proof structure is correct (for MLL+Mix) *iff the perfect matching of its R&B-graph is* unique*, i.e. has no alternating cycle.*

Corollary (N. 2018, but could have been discovered in 1999!)

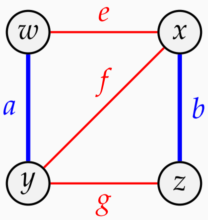
Correctness for MLL+Mix can be decided in linear time.

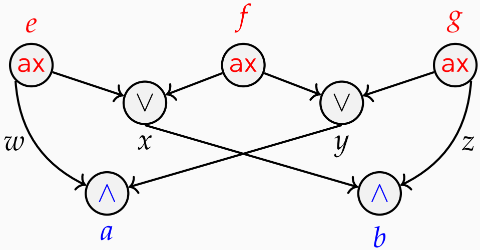
Proof (by direct reduction).

- R&B-graphs can be computed in linear time
- there is a linear time algorithm for PM uniqueness (Gabow, Kaplan & Tarjan 1999) □

Reduction perfect matchings → proof structures

New: MLL+Mix correctness is *equivalent* to PM uniqueness.





On sequentialization for unique perfect matchings

Another remark by Retoré: unique perfect matchings admit a "sequentialization", i.e. an inductive characterization.

Corollary (of Kotzig's theorem)

A perfect matching M is unique iff iterative deletion of bridges in M (with their endpoints) reaches the empty graph.

Another remark by Retoré: unique perfect matchings admit a "sequentialization", i.e. an inductive characterization.

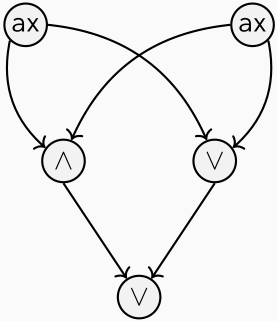
Corollary (of Kotzig's theorem)

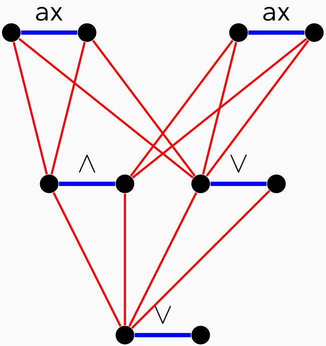
A perfect matching M is unique iff iterative deletion of bridges in M (with their endpoints) reaches the empty graph.

- A mismatch: $\{$sequentializations of a proof net$\} \ncong$ $\{$sequentializations of its "R&B-graph"$\}$
- We fix this with another reduction $\{$proof structures$\} \to \{$graphs w/ PMs$\}$: *graphification*

Graphification of proof structures (1)

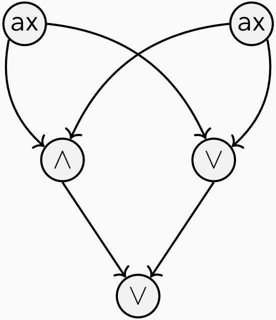
- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*

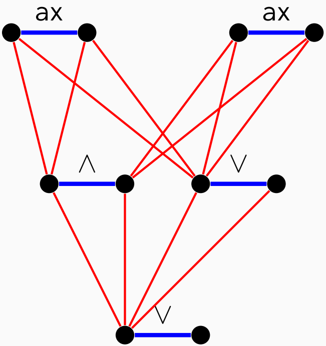




Graphification of proof structures (1)

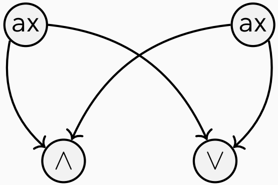
- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*

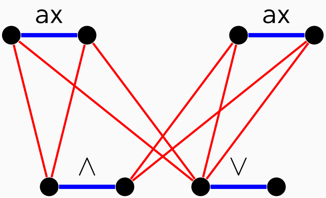




Graphification of proof structures (1)

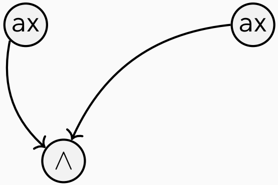
- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*

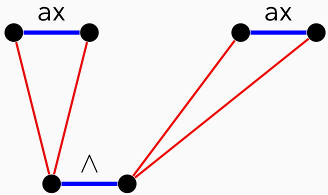




Graphification of proof structures (1)

- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*



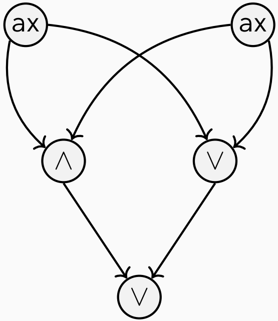


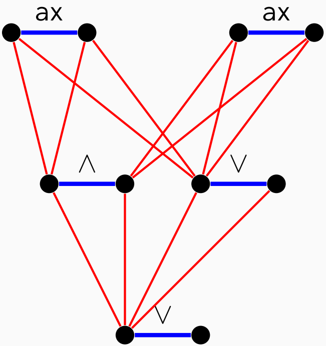
Graphification of proof structures (1)

- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*

Graphification of proof structures (1)

- Matching edges correspond to vertices
- *Bridges* correspond to *splitting terminal vertices*





Correctness criterion is still uniqueness of PM i.e. no alt cycle

Graphifications of proof nets (2)

Theorem

The sequentializations of a proof structure are in bijection with the sequentializations of its graphification.

In particular if one set is $\neq \emptyset$ so is the other, therefore:

Corollary (Sequentialization theorem for MLL+Mix)

Switching acyclic $\Leftrightarrow$ *MLL+Mix sequentializable.*

New proof, immediate from graph-theoretic analogue.

Graphifications of proof nets (2)

Theorem

The sequentializations of a proof structure are in bijection with the sequentializations of its graphification.

In particular if one set is $\neq \emptyset$ so is the other, therefore:

Corollary (Sequentialization theorem for MLL+Mix)

Switching acyclic $\Leftrightarrow$ MLL+Mix sequentializable.

New proof, immediate from graph-theoretic analogue.

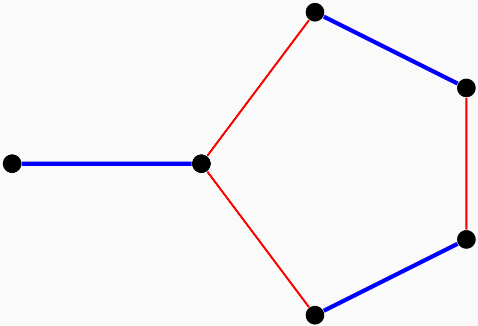
Next: a theorem on graphs inspired by linear logic.

Blossoms in matching theory

A key concept in combinatorial matching algorithms,
e.g. testing PM uniqueness: *blossoms*[2]

Definition

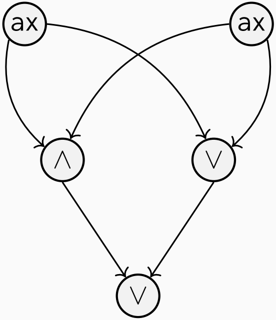
A *blossom* is a cycle with exactly 1 vertex matched outside.

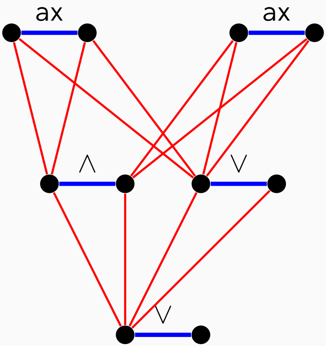


[2] Edmonds, *Paths, trees and flowers*, Canadian J. Math., 1965

Blossoms vs. dependencies

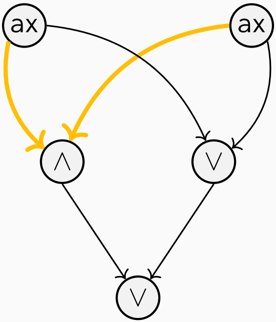
Blossoms of graphification $\rightsquigarrow$ predecessors and dependencies

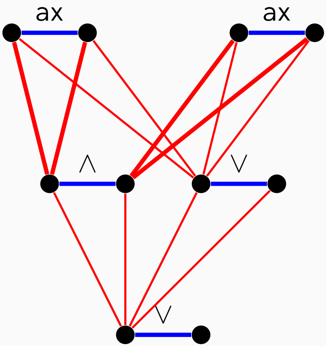




Blossoms vs. dependencies

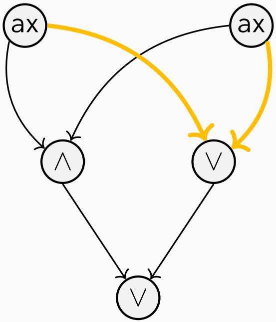
Blossoms of graphification ⇝ **predecessors** and dependencies

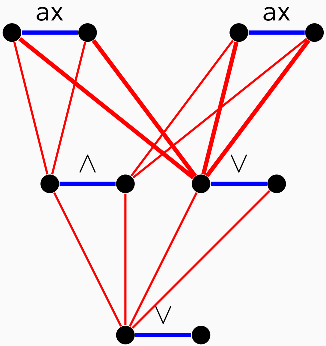




Blossoms vs. dependencies

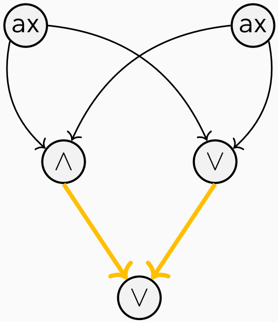
Blossoms of graphification $\rightsquigarrow$ **predecessors** and dependencies

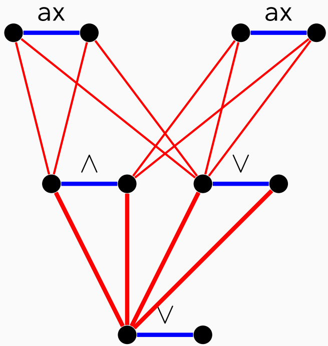




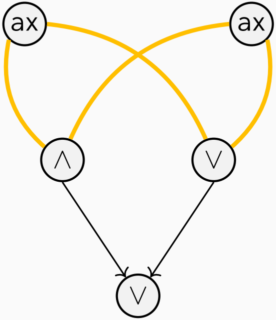
Blossoms vs. dependencies

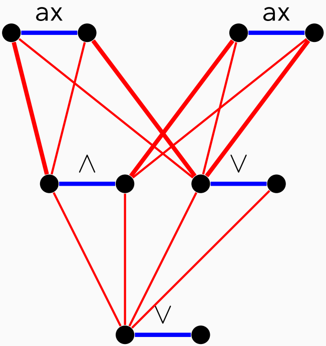
Blossoms of graphification $\rightsquigarrow$ **predecessors** and dependencies





Blossoms of graphification ⇝ predecessors and **dependencies**



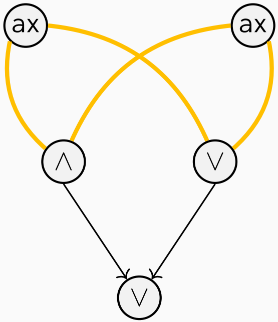


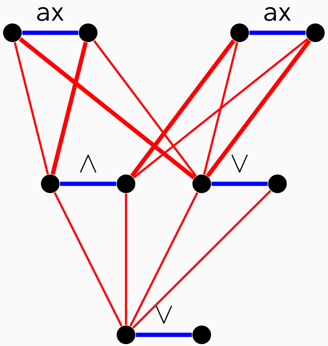
Definition

A ∨-vertex u *depends upon* a vertex v if there is a switching path between the premises of u going through v.

Blossoms vs. dependencies

Blossoms of graphification ⇝ predecessors and **dependencies**





Definition

A ∨-vertex u *depends upon* a vertex v if there is a switching path between the premises of u going through v.

Kingdom ordering of proof nets

Definition

A $\vee$-vertex u *depends upon* a vertex v if there is a switching path between the premises of u going through v.

This notion has already been used before!

Definition (Kingdom ordering of a proof net)

Let l, l' be vertices of a MLL+Mix proof net π. We define $u \ll_\pi v$ iff every sequentialization of π introduces u above v.

Theorem (Bellin 1997)

$\ll_\pi$ is the transitive closure of (predecessor relation) $\cup$ (dependency relation).

Bellin's theorem for unique perfect matchings

Theorem (N. 2018; Bellin's theorem, rephrased)

Let G be a graph, M be a unique PM of G and $e, e' \in M$. TFAE:

- *every bridge deletion sequence reaching $\varnothing$ deletes e before e';*
- *there exists a sequence $e_1, \ldots, e_n \in M$ such that*
  - *$e_1 = e$ and $e_n = e'$,*
  - *for all $i < n$, e_i is the stem of some blossom containing e_{i+1}.*

(Think of perfect elimination orderings of chordal graphs)

Simpler statement: transitive closure of only 1 relation!

Conclusion

Unique perfect matchings: the right graph-theoretic counterpart for the statics of MLL+Mix proof nets

- Statics: no account of computational content (cut-elimination)
- Not a combinatorial bijection, but both algorithmic reductions and transfer of structural properties