Removing Cycles from Proofs

CSL 2017

Andrea Aler Tubella, Alessio Guglielmi, and **Benjamin Ralph**

Université Paris Diderot, University of Bath, and University of Bath

August 21, 2017

Cycles in Proofs

A cycle in a proof is a cycle in a path traced by propositional variables or atom occurrences.

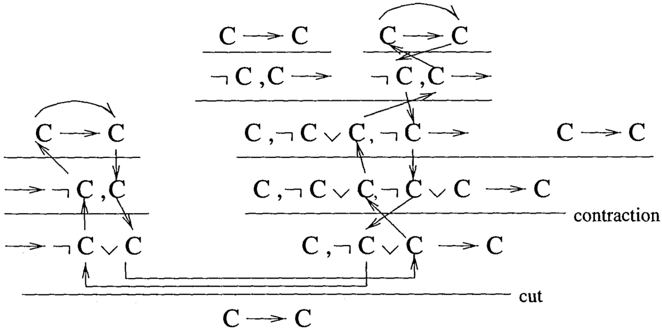
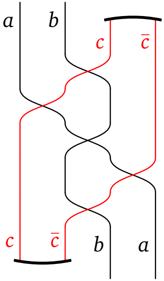


Figure: [2]

Cycles in Proofs

A cycle in a proof is a cycle in a path traced by propositional variables or atom occurrences.

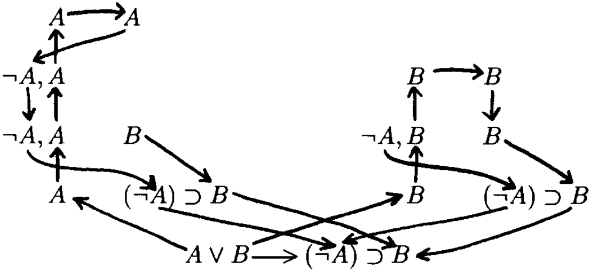
$$
\mathsf{2s}\,\dfrac{\mathsf{m}\,\dfrac{\mathsf{2s}\,\dfrac{a \wedge b \wedge \boxed{\mathsf{ai}{\downarrow}\,\dfrac{\mathsf{t}}{c \vee \bar{c}}}}{\boxed{=\,\dfrac{a \wedge c}{c \wedge a}} \vee (b \wedge \bar{c})}}{[c \vee b] \wedge \boxed{=\,\dfrac{a \vee \bar{c}}{\bar{c} \vee a}}}}{\boxed{\mathsf{ai}{\uparrow}\,\dfrac{c \wedge \bar{c}}{\mathsf{f}}} \vee b \vee a}
$$



Previous work on cycles

Buss introduced the notion of a logical flow graph for sequent calculus proofs [1].

$$
\dfrac{
\dfrac{\dfrac{\dfrac{A \rightarrow A}{\neg A,\, A \rightarrow}}{\neg A,\ A \rightarrow B}}{A \rightarrow (\neg A) \supset B}
\qquad
\dfrac{\dfrac{B \rightarrow B}{\neg A,\, B \rightarrow B}}{B \rightarrow (\neg A) \supset B}
}{A \vee B \rightarrow (\neg A) \supset B}
$$

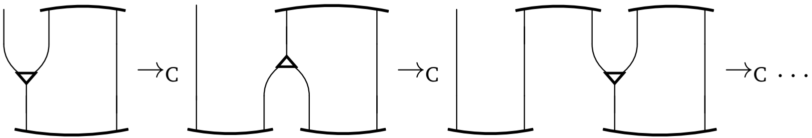


Questions about logical flow graphs

- “Does the presence of cycles in proofs of propositional logic help to shorten the length of a proof?” [2]
- “Given a proof Π (possibly containing cycles), is there an acyclic proof for the same sequent only polynomially larger than Π” [1]
- A. Yes: In the sequent calculus we can transform a proof with n cycles into an acyclic one with an expansion bounded by a polynomial of degree $n + 1$ [2].

Our interest in cycles

- In open deduction, an important part of normalisation is decomposition, pushing contractions through a proof.
- If contractions get stuck in a cycle, they have an infinite path ahead of them.



- Therefore, eliminating cycles is a crucial part of normalisation.

Open Deduction

Open deduction is a deep inference proof formalism that gives us more ways of composing derivations.

Given derivations $\begin{matrix} A \\ \Phi \| \\ B \end{matrix}$ and $\begin{matrix} C \\ \Psi \| \\ D \end{matrix}$ we can compose them with:

1. An inference rule $\sigma : B/C$:

$$\begin{array}{c} A \\ \Phi \| \\ \sigma \dfrac{B}{C} \\ \Psi \| \\ D \end{array}$$

2. A binary logical relation $\star$:

$$\begin{matrix} A \\ \Phi \| \\ B \end{matrix} \star \begin{matrix} C \\ \| \Psi \\ D \end{matrix} = \begin{matrix} A \star C \\ \Phi{\star}\Psi \| \\ B \star D \end{matrix}$$

SKS

- The *structural* rules (all atomic):

$$\text{ai}\!\downarrow \frac{\mathsf{t}}{a \vee \bar{a}} \qquad \text{ac}\!\downarrow \frac{a \vee a}{a} \qquad \text{aw}\!\downarrow \frac{\mathsf{f}}{a}$$

identity *contraction* *weakening*

$$\text{ai}\!\uparrow \frac{a \wedge \bar{a}}{\mathsf{f}} \qquad \text{ac}\!\uparrow \frac{a}{a \wedge a} \qquad \text{aw}\!\uparrow \frac{a}{\mathsf{t}}$$

cut *cocontraction* *coweakening*

- The *logical* rules (all linear):

$$\mathsf{s}\,\frac{A \wedge [B \vee C]}{(A \wedge B) \vee C} \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{[A \vee C] \wedge [B \vee D]}$$

switch *medial*

- The *unit* rules:

$$A \wedge \mathsf{t} = A \qquad A \vee \mathsf{f} = A$$

Atomic Flows

Atomic flows [3] are a geometric invariant of SKS propositional logic proofs. Only structural information about the proof is conserved.

$\mathsf{ai}\!\downarrow \dfrac{\mathsf{t}}{a \vee \bar{a}}$	$\mathsf{ac}\!\downarrow \dfrac{a \vee a}{a}$	$\mathsf{aw}\!\downarrow \dfrac{\mathsf{f}}{a}$
identity	*contraction*	*weakening*
$\mathsf{ai}\!\uparrow \dfrac{a \wedge \bar{a}}{\mathsf{f}}$	$\mathsf{ac}\!\uparrow \dfrac{a}{a \wedge a}$	$\mathsf{aw}\!\uparrow \dfrac{a}{\mathsf{t}}$
cut	*cocontraction*	*coweakening*

Atomic Flows

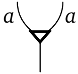
Atomic flows [3] are a geometric invariant of SKS propositional logic proofs. Only structural information about the proof is conserved.

$$\mathsf{ai}\!\downarrow \frac{\mathsf{t}}{a \vee \bar{a}}$$

identity

$$\mathsf{ac}\!\downarrow \frac{a \vee a}{a}$$

contraction



$$\mathsf{aw}\!\downarrow \frac{\mathsf{f}}{a}$$

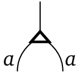
weakening

$$\mathsf{ai}\!\uparrow \frac{a \wedge \bar{a}}{\mathsf{f}}$$

cut

$$\mathsf{ac}\!\uparrow \frac{a}{a \wedge a}$$

cocontraction



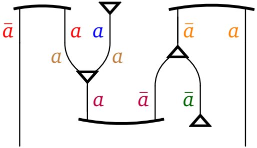
$$\mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}}$$

coweakening

Composition of proofs naturally corresponds to composition of flows.

Open Deduction Examples

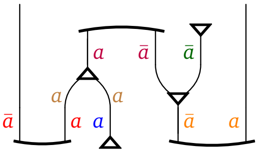
$$
\dfrac{
\left(=\dfrac{\left(\mathsf{ai}\!\downarrow\dfrac{\mathsf{t}}{\bar a \vee a}\right) \vee \left(\mathsf{aw}\!\downarrow\dfrac{\mathsf{f}}{a}\right)}{\bar a \vee \left(\mathsf{ac}\!\downarrow\dfrac{a \vee a}{a}\right)}\right)
\wedge
\left(\mathsf{ai}\!\downarrow\dfrac{\mathsf{t}}{\left(\mathsf{ac}\!\uparrow\dfrac{\bar a}{\bar a \wedge \left(\mathsf{aw}\!\uparrow\dfrac{\bar a}{\mathsf{t}}\right)}\right) \vee a}\right)
}{
\mathsf{2s}\dfrac{[\bar a \vee a] \wedge [a \vee \bar a]}{\bar a \vee \left(\mathsf{ai}\!\uparrow\dfrac{a \wedge \bar a}{\mathsf{f}}\right) \vee a}
}
$$



The proof above has premise $[\mathsf{t} \vee \mathsf{f}] \wedge \mathsf{t} = \mathsf{t}$ and conclusion $\bar a \vee \mathsf{f} \vee a = \bar a \vee a$.

Open Deduction Examples

$$\dfrac{\mathsf{2s}\,\dfrac{\bar a \wedge \left[\mathsf{ai}\!\downarrow \dfrac{\mathsf t}{a \vee \bar a}\right] \wedge a}{(\bar a \wedge a) \vee (a \wedge \bar a)}}{\left[= \dfrac{\bar a \wedge \left[\mathsf{ac}\!\uparrow \dfrac{a}{a\wedge a}\right]}{\left[\mathsf{ai}\!\uparrow\dfrac{\bar a\wedge a}{\mathsf f}\right]\wedge\left[\mathsf{aw}\!\uparrow\dfrac{a}{\mathsf t}\right]}\right] \vee \left[\mathsf{ai}\!\uparrow\dfrac{\left[\mathsf{ac}\!\downarrow\dfrac{\bar a \vee \left[\mathsf{aw}\!\downarrow\dfrac{\mathsf f}{\bar a}\right]}{\bar a}\right]\wedge a}{\mathsf f}\right]}$$



The derivation above has premise $\bar a \wedge \mathsf{t} \wedge a = \bar a \wedge a$ and conclusion $(\mathsf f \wedge \mathsf t) \vee \mathsf f = \mathsf f$.

Normalisation for SKS

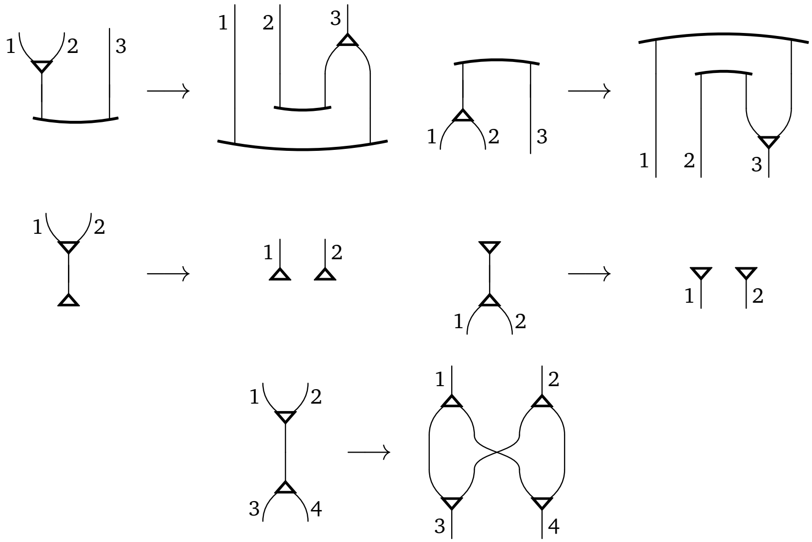
Recently, Aler Tubella showed that proofs in many different logics with certain properties can be normalised using the same techniques [4]:

$$\begin{array}{c} \phi_0 \,\|\, \mathsf{SKS} \\ A \end{array} \xrightarrow{(1)} \begin{array}{c} \phi_1 \,\|\, \mathsf{SKS}\setminus\{\mathsf{ac}\downarrow,\mathsf{ac}\uparrow\} \\ A' \\ \| \{\mathsf{ac}\downarrow\} \\ A \end{array} \xrightarrow{(2)} \begin{array}{c} \phi_2 \,\|\, \mathsf{KS}\setminus\{\mathsf{ac}\downarrow\} \\ A' \\ \| \{\mathsf{ac}\downarrow\} \\ A \end{array}$$

1. Decomposition: (co)contractions are pushed through the proof until they reach the bottom. This procedure can increase the size of the proof exponentially.
2. Linear Cut Elimination: A technique called “splitting” is used to eliminate cuts from the proof. Provided that decomposition has taken place first, this procedure does not significantly increase the size of the proof.

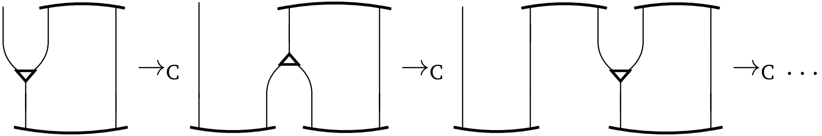
Decomposition for SKS

We define the rewriting system C:



Cycles cause non-termination

If we apply C to a proof with cycles, the rewriting system will not terminate.



Thus, if we want to use decomposition as the first part of our normalisation, we need a procedure for removing cycles.

Normalisation for SKS

Therefore we need to refine the normalisation procedure:

$$\begin{array}{c} \phi_0 \,\|\, \mathsf{SKS} \\ A \end{array} \quad \xrightarrow{(1)} \quad \begin{array}{c} \phi_1 \,\|\, \mathsf{SKS}\setminus\{\mathsf{ac}{\downarrow},\mathsf{ac}{\uparrow}\} \\ A' \\ \| \{\mathsf{ac}{\downarrow}\} \\ A \end{array} \quad \xrightarrow{(2)} \quad \begin{array}{c} \phi_3 \,\|\, \mathsf{KS}\setminus\{\mathsf{ac}{\downarrow}\} \\ A' \\ \| \{\mathsf{ac}{\downarrow}\} \\ A \end{array}$$

1. Decomposition
2. Linear Cut Elimination

Normalisation for SKS

Therefore we need to refine the normalisation procedure:

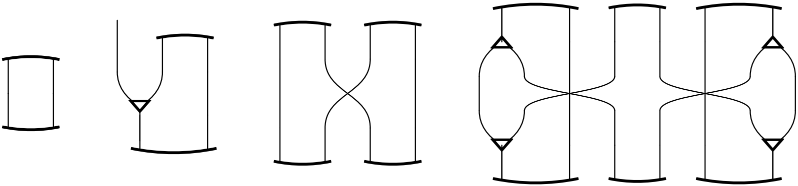
$$\begin{array}{c} \phi_0 \; \Vert \; \mathsf{SKS} \\ A \end{array} \xrightarrow{(1)} \begin{array}{c} \phi_1 \; \Vert \; \mathsf{SKS} \\ A \end{array} \xrightarrow{(2)} \begin{array}{c} \phi_2 \; \Vert \; \mathsf{SKS} \setminus \{\mathsf{ac}\downarrow, \mathsf{ac}\uparrow\} \\ A' \\ \Vert \; \{\mathsf{ac}\downarrow\} \\ A \end{array} \xrightarrow{(3)} \begin{array}{c} \phi_3 \; \Vert \; \mathsf{KS} \setminus \{\mathsf{ac}\downarrow\} \\ A' \\ \Vert \; \{\mathsf{ac}\downarrow\} \\ A \end{array}$$

1. Cycle Removal
2. Decomposition
3. Linear Cut Elimination

Our paper sets out how to remove cycles from proofs in SKS in a local way.

Different types of cycle

There are various different shapes and sizes cycles can come in:



For this talk, we assume there is only one cycle in the proof and it contains only one identity and one cut.

Critical Medial

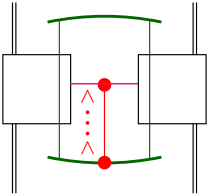
For a cycle to occur in a proof, two edges of an atomic flow that were related by $\vee$ at the top of a connected component have to be connected by $\wedge$ at the bottom of the flow.

$$\mathsf{m}\,\frac{(A\{a\} \wedge B) \vee (C \wedge D\{\bar{a}\})}{[A\{a\} \vee C] \wedge [B \vee D\{\bar{a}\}]}$$

We call this medial a critical medial for the cycle. For this talk, we will assume there is only one critical medial per cycle.

Trace the ∧-flow

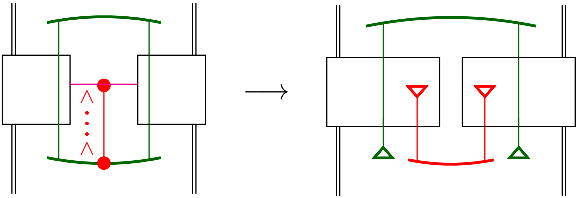
Starting from the bottom, colour red each conjunction (∧) connecting two green edges until the critical medial for the cycle.



We call the trace of ∧-s the ∧-*flow*.

Transformation along the $\wedge$-flow

By pushing the critical medial along the $\wedge$-flow in a certain way, we can break the cycle.

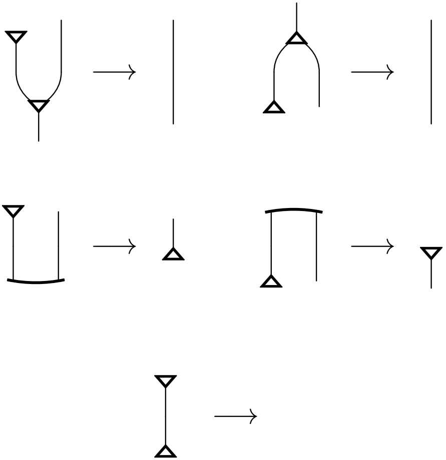


However, when the critical medial hits the cut, an unsound rule is created:

$$\neq \frac{\left(\boxed{\mathsf{aw}\!\uparrow \frac{\bar{a}}{\mathsf{t}}} \wedge a\right) \vee \left(\bar{a} \wedge \boxed{\mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}}}\right)}{\mathsf{f}}$$

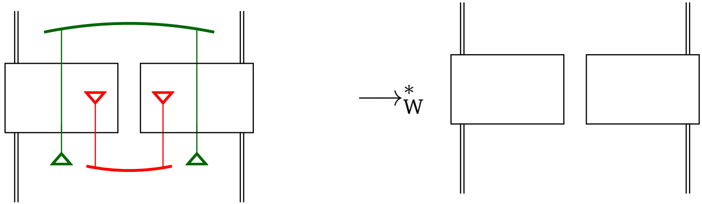
Rewriting System W

We define a new rewriting system, W:



Solving the problem

We can use rewriting system W to solve the problem:

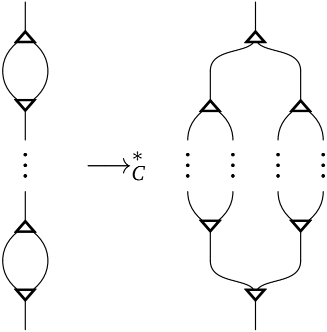


$$\neq \frac{\left(\boxed{\mathsf{aw}\!\uparrow \frac{\bar{a}}{\mathsf{t}}} \wedge a\right) \vee \left(\bar{a} \wedge \boxed{\mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}}}\right)}{\mathsf{f}} \quad \longrightarrow \quad = \frac{(\mathsf{t} \wedge \mathsf{f}) \vee (\mathsf{f} \wedge \mathsf{t})}{\mathsf{f}}$$

The cycle is now successfully removed!

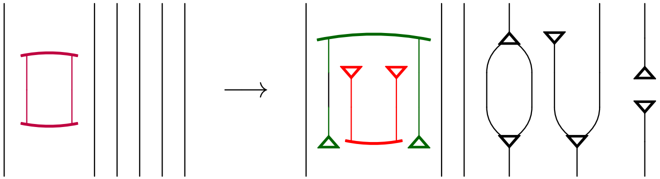
Complexity creation

Rewriting system C can cause exponential blow up in a proof, if there are “sausages” in the flow.



Creation of complexity through cycle removal

In the process of transforming along the $\wedge$-flow, various different things can happen to other edges that are involved in the critical medial.



Note that a sausage can be created. Further work is needed to explore how this and other methods of cycle removal affect the complexity of the whole normalisation procedure.

Bibliography I

[1] S. R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53(1):75–102, 1991.

[2] A. Carbone. The cost of a cycle is a square. *The Journal of Symbolic Logic*, 67(01):35–60, 2002.

[3] A. Guglielmi, T. Gundersen, and L. Straßburger. Breaking paths in atomic flows for classical logic. In *Logic in Computer Science (LICS), 2010 25th Annual IEEE Symposium on*, pages 284–293. IEEE, 2010.

[4] A. A. Tubella. *A study of normalisation through subatomic logic*. Thesis, University of Bath, 2017.