The Composition of Combinatorial Flows

Giti Omidvar and Lutz Straßburger
Inria Saclay
Ecole Polytechnique

GT Scalp, Marseille, 15 February 2023
Preliminaries: Open Deduction

\[
\begin{align*}
\frac{t}{a \lor \overline{a}} \quad & \frac{(A \lor B) \land C}{A \lor (B \land C)} \quad \frac{\overline{a} \land a}{f} \\
\frac{a \lor a}{a} \quad & \frac{(A \land C) \lor (B \land D)}{(A \lor B) \land (C \lor D)} \quad \frac{a}{a \land a} \\
\frac{f}{a} \quad & \frac{f}{t} \quad \frac{a}{t}
\end{align*}
\]
1. We cannot read back a proof from atomic flows
2. yanking is not possible

Alessio Guglielmi and Tom Gundersen, LMCS 2008
Anupam Das, RTA 2013
1. We cannot read back a proof from atomic flows
2. Yanking is not possible

Alessio Guglielmi and Tom Gundersen, LMCS 2008
Anupam Das, RTA 2013
Preliminaries: Open Deduction and Atomic Flows

1. We cannot read back a proof from atomic flows
2. yanking is not possible

Alessio Guglielmi and Tom Gundersen, LMCS 2008

Alessio Guglielmi and Tom Gundersen, RTA 2013
Preliminaries: Open Deduction and Atomic Flows

\[ a \land \overline{a} \quad \overline{a} \quad \overline{a} \]

[Diagram]

1. We cannot read back a proof from atomic flows
2. Yanking is not possible
Preliminaries: Open Deduction and Atomic Flows

1. We cannot read back a proof from atomic flows
1. We cannot read back a proof from atomic flows

2. Yanking is not possible
• Total separation of linear part and resource management of the proof $\rightarrow$ size explosion
From Open Deduction to Preflows

Corresponds to B-nets: François Lamarche and Lutz Straßburger, TLCA 2005
From Open Deduction to Preflows

\[ \begin{align*}
\frac{a \land a}{a} & \quad \text{ac} \\
\frac{a}{a} & \\
\frac{t}{\bar{a} \lor a} & \quad \text{ai} \\
\frac{a \lor (\bar{a} \lor a)}{a} & \\
\frac{\bar{a} \lor a}{a} & \\
\frac{a \lor \bar{a}}{f} & \quad \text{ai} \\
\frac{a \lor \bar{a}}{a} & \\
\frac{\bar{a} \lor a}{f} & \\
\frac{a \land \bar{a}}{a} & \\
\end{align*} \]

Corresponds to B-nets

\[ \frac{a \land \bar{a}}{t} \quad \text{ai} \]

\[ \frac{\bar{a} \lor a}{f} \]

\[ \frac{a \lor \bar{a}}{a} \land (\bar{a} \lor a) \]

\[ \frac{a \land \bar{a}}{a} \lor (a \land \bar{a}) \]

\[ \frac{a}{a} \lor (\bar{a} \land a) \]

\[ \frac{\bar{a} \lor a}{a} \land (\bar{a} \lor a) \]

\[ \frac{a \lor \bar{a}}{a} \lor (a \lor \bar{a}) \]

\[ \frac{\bar{a} \lor a}{a} \land (\bar{a} \lor a) \]
From Open Deduction to Preflows

B-nets: François Lamarche and Lutz Straßburger, TLCA 2005
From Open Deduction to Combinatorial Flows

Corresponds to B-nets: François Lamarche and Lutz Straßburger, TLCA 2005
From Open Deduction to Combinatorial Flows

Corresponds to B-nets

B-nets: François Lamarche and Lutz Straßburger, TLCA 2005
Formulas:

\[ A, B := t \mid f \mid a \mid \bar{a} \mid A \lor B \mid A \land B \]
Formulas:

\[ A, B := t \mid f \mid a \mid \overline{a} \mid A \lor B \mid A \land B \]

\[
\begin{align*}
A \land B &\equiv B \land A \\
A \lor B &\equiv B \lor A \\
(A \land B) \land C &\equiv A \land (B \land C) \\
(A \lor B) \lor C &\equiv A \lor (B \lor C) \\
A \land t &\equiv A \\
A \lor f &\equiv A \\
t \lor t &\equiv t \\
f \land f &\equiv f
\end{align*}
\]
Formulas:

\[
A, B := t \mid f \mid a \mid \overline{a} \mid A \lor B \mid A \land B
\]

\[
A \land B \equiv B \land A \quad (A \land B) \land C \equiv A \land (B \land C) \quad A \land t \equiv A \quad t \lor t \equiv t
\]

\[
A \lor B \equiv B \lor A \quad (A \lor B) \lor C \equiv A \lor (B \lor C) \quad A \lor f \equiv A \quad f \land f \equiv f
\]

Unit-free Formulas:

\[
A, B := a \mid \overline{a} \mid A \lor B \mid A \land B
\]
Formulas:

\[ A, B := t \mid f \mid a \mid \overline{a} \mid A \lor B \mid A \land B \]

\[ A \land B \equiv B \land A \quad (A \land B) \land C \equiv A \land (B \land C) \quad A \land t \equiv A \quad t \lor t \equiv t \]

\[ A \lor B \equiv B \lor A \quad (A \lor B) \lor C \equiv A \lor (B \lor C) \quad A \lor f \equiv A \quad f \land f \equiv f \]

Unit-free Formulas:

\[ A, B := a \mid \overline{a} \mid A \lor B \mid A \land B \]

Pure Formulas: \( A \equiv t \) or \( A \equiv f \) or \( A \) is equivalent to a unit-free formula.
Graph of a formula

- $G(t) = G(f)$: empty graph
- $G(a)$:
  - $a$
  - $\bar{a}$
- $G(A \lor B)$:
  - $G(A)$
  - $G(B)$
- $G(A \land B)$:
  - $G(A)$
  - $G(B)$
Graph of a formula

- $\mathcal{G}(t) = \mathcal{G}(f)$: empty graph
Graph of a formula

- $G(t) = G(f)$: empty graph
- $G(a)$:
  - $\bar{a}$
Graph of a formula

- $G(t) = G(f)$: empty graph
- $G(a)$:
  - $a$
- $G(\bar{a})$:
  - $\bar{a}$
Graph of a formula

- $G(t) = G(f)$: empty graph
- $G(a)$:
  - $a$
- $G(\bar{a})$:
  - $\bar{a}$
- $G(A \lor B)$:
  - $G(A)$
  - $G(B)$
Graph of a formula

- $G(t) = G(f)$: empty graph
- $G(a)$:
  - $a$
- $G(\bar{a})$:
  - $\bar{a}$
- $G(A \lor B)$:
  - $G(A)$
  - $G(B)$
- $G(A \land B)$:
  - $G(A)$
  - $G(B)$
Graph of a formula

\[(\neg d \lor \neg \neg d) \land (\neg \neg c \land \neg a) \lor (\neg c \lor \neg a) \lor (\neg a \land \neg \neg a)\]

A cograph is a graph without $P_4$.

Theorem
A graph $G$ is graph of a formula $A$ if and only if $G$ is a cograph.
Graph of a formula

\[((d \lor \overline{d}) \land (\overline{c} \land a)) \lor ((c \lor \overline{a}) \lor (a \land \overline{a}))\]

A **cograph** is a graph without \(\mathcal{P}_4\):

\[\begin{array}{c}
\overline{d} & \overline{a} & a \\
\overline{d} & \overline{a} & a \\
\overline{d} & \overline{d} & \overline{c} \\
\overline{c} & \overline{a} & c \\
\end{array}\]

\[\begin{array}{c}
u \\
z & w \\
\end{array}\]
Theorem

A graph $G$ is graph of a formula $A$ if and only if $G$ is a cograph.

A cograph is a graph without $P_4$: 

$$((d \lor \bar{d}) \land (\bar{c} \land a)) \lor ((c \lor \bar{a}) \lor (a \land \bar{a}))$$
Multiplicative Flows

A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$
Multiplicative Flows

A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\varepsilon$-acyclic.

**correctness criterion:**

![Diagram](attachment:image.png)
A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\alpha$-acyclic.

correctness criterion:
A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $\mathcal{G}(\phi)$ is $\omega$-acyclic.

**correctness criterion:**

Christian Retore, Theoretical Computer Science 2003 (Handsome Proof Nets)
Multiplicative Flows

A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\tilde{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\alpha$-acyclic.

**correctness criterion:**

![Correctness Criterion Diagram](image)
A triple $\phi = \langle A, B, B_\phi \rangle$ is an $m$-flow if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\varepsilon$-acyclic.

correctness criterion:
Multiplicative Flows

A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\omega$-acyclic.

correctness criterion:
Multiplicative Flows

A triple $\phi = \langle A, B, B_\phi \rangle$ is an **m-flow** if $A$ and $B$ are pure formulas, $B_\phi$ is a perfect matching on the atom occurrences of $\bar{A} \lor B$ such that the underlying RB-cograph $G(\phi)$ is $\approx$-acyclic.

$$t = (((d \lor d) \land (\bar{c} \land a)) \lor ((c \lor \bar{a}) \lor (a \land \bar{a})))$$

**correctness criterion:**

![Correctness Diagram](Image)

Christian Retore, Theoretical Computer Science 2003 (Handsome Proof Nets)
Multiplicative Flows

**Theorem**

Let $D \| \{ai\downarrow, ai\uparrow, s, \text{mix}\}$ be a derivation. If $A$ and $B$ are pure, then the translation of $D$ is an $m$-flow.

**Theorem**

Let $\phi = \langle A, B, B_\phi \rangle$ be an $m$-flow. Then there is a derivation $D \| \{ai\downarrow, ai\uparrow, s, \text{mix}\}$ whose translation is $\phi$.

$$
\begin{align*}
&\begin{array}{c}
 t \\
 a \lor \overline{a}
\end{array} \\
&\frac{a}{ai\downarrow} \\
&\frac{(A \lor B) \land C}{A \lor (B \land C)} \\
&\frac{\overline{a} \land a}{\text{s}} \\
&\frac{f}{ai\uparrow} \\
&\frac{f}{\text{mix}} \\
&\frac{t}{t}
\end{align*}
$$
\[ (a \land \overline{a}) \lor a \equiv a \land a \]

\[ a \lor (a \land \overline{a}) \lor a \equiv a \lor (a \land \overline{a}) \lor (a \land a) \]
Additive Flows

A triple $\phi = \langle A, B, f_{\phi} \rangle$ is an $a^\perp$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f_{\phi}$ is a skew fibration $f_{\phi} : G(A) \to G(B)$.
Additive Flows

- A triple \( \phi = \langle A, B, f_\phi \rangle \) is an \( a^\downarrow \)-flow if \( A \) and \( B \) are pure, and \( A \neq t \), and \( f_\phi \) is a skew fibration \( f_\phi : G(A) \rightarrow G(B) \).

A skew fibration is a graph homomorphism \( f : G \rightarrow H \) such that
Additive Flows

- A triple $\phi = \langle A, B, f^\downarrow \rangle$ is an $a^\downarrow$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f^\downarrow$ is a skew fibration $f^\downarrow : G(A) \to G(B)$.

A **skew fibration** is a graph homomorphism $f : G \to H$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in \mathcal{E}_H$,
Additive Flows

- A triple $\phi = \langle A, B, f^{\phi}_\downarrow \rangle$ is an $a^{\downarrow}$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f^{\phi}_\downarrow$ is a skew fibration $f^{\phi}_\downarrow : G(A) \to G(B)$.

A **skew fibration** is a graph homomorphism $f : G \to H$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in E_H$, there exists $z \in G$ with the edge $vz \in E_G$
Additive Flows

• A triple $\phi = \langle A, B, f_\phi \rangle$ is an $a^\downarrow$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f_\phi$ is a skew fibration $f_\phi : \mathcal{G}(A) \to \mathcal{G}(B)$.

A skew fibration is a graph homomorphism $f : \mathcal{G} \to \mathcal{H}$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in E_H$, there exists $z \in G$ with the edge $vz \in E_G$ such that the edge $f(z)w$ does not exist in $\mathcal{H}$.
Additive Flows

- A triple $\phi = \langle A, B, f_\phi \rangle$ is an $a^{\downarrow}$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f_\phi$ is a skew fibration $f_\phi : G(A) \to G(B)$.

A skew fibration is a graph homomorphism $f : G \to H$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in E_H$, there exists $z \in G$ with the edge $vz \in E_G$ such that the edge $f(z)w$ does not exist in $H$. 
• A triple $\phi = \langle A, B, f_\phi \downarrow \rangle$ is an $a_\downarrow$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f_\phi \downarrow$ is a skew fibration $f_\phi \downarrow : G(A) \to G(B)$.

A skew fibration is a graph homomorphism $f : G \to H$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in E_H$, there exists $z \in G$ with the edge $vz \in E_G$ such that the edge $f(z)w$ does not exist in $H$. 
Additive Flows

- A triple $\phi = \langle A, B, f_\phi^\downarrow \rangle$ is an $a^\downarrow$-flow if $A$ and $B$ are pure, and $A \neq t$, and $f_\phi^\downarrow$ is a skew fibration $f_\phi^\downarrow : G(A) \to G(B)$.

- A triple $\phi = \langle C, D, f_\phi^\uparrow \rangle$ is an $a^\uparrow$-flow if $C$ and $D$ are pure, and $D \neq f$, and $f_\phi^\uparrow$ is a skew fibration $f_\phi^\uparrow : G(D) \to G(C)$.

A skew fibration is a graph homomorphism $f : G \to H$ such that for every $v \in V_G$ and $w \in V_H$, with $f(v)w \in E_H$, there exists $z \in G$ with the edge $vz \in E_G$ such that the edge $f(z)w$ does not exist in $H$. 

\[
\begin{array}{c}
(a \land (b \lor c)) \lor (b \land c) \\
(a \lor b) \land (a \lor b) \land (a \lor c) \\
a^\uparrow\text{-flow}
\end{array}
\quad
\begin{array}{c}
\bar{b} \lor \bar{b} \lor (b \land a) \lor (b \land a) \\
b \lor c \lor ((b \lor b) \land a) \\
a^\downarrow\text{-flow}
\end{array}
\quad
\begin{array}{c}
\bar{b} \lor \bar{b} \lor (b \land a) \lor (b \land a) \\
(b \land c) \lor ((b \lor b) \land a) \\
\text{not a skew fibration}
\end{array}
\]
Additive Flows

**Theorem**

Let $\mathcal{D} \parallel \{\text{aw}_\downarrow, \text{ac}_\downarrow, m\}$ be a derivation. If $A$ and $B$ are pure, then translation of $\mathcal{D}$ is an $\text{a}_\downarrow$-flow. Dually, if $A$ and $B$ are pure in $\mathcal{D} \parallel \{\text{aw}_\uparrow, \text{ac}_\uparrow, m\}$ then translation of $\mathcal{D}$ is an $\text{a}_\uparrow$-flow.

**Theorem**

Let $\phi = \langle A, B, f^\downarrow \rangle$ be an $\text{a}_\downarrow$-flow. Then there is a derivation $\mathcal{D} \parallel \{\text{aw}_\downarrow, \text{ac}_\downarrow, m\}$ whose translation is $\phi$. For every $\text{a}_\uparrow$-flow $\psi$ we have $\mathcal{D} \parallel \{\text{aw}_\uparrow, \text{ac}_\uparrow, m\}$ whose translation is $\psi$.

\[
\begin{align*}
\text{aw}_\downarrow : & \frac{a \lor a}{a} \quad \text{ac}_\downarrow : \frac{f}{a} \\
\text{aw}_\uparrow : & \frac{(A \land C) \lor (B \land D)}{(A \lor B) \land (C \lor D)} \quad \text{m} : \frac{a}{t} \\
\text{ac}_\uparrow : & \frac{a}{a \land a}
\end{align*}
\]
**Purification**

**Pure Formulas:** $A \equiv t$ or $A \equiv f$ or $A$ is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
Purification

**Pure** Formulas: \( A \equiv t \) or \( A \equiv f \) or \( A \) is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
Purification

**Pure Formulas:** \( A \equiv t \) or \( A \equiv f \) or \( A \) is equivalent to a unit-free formula.

Slice of a Combinatorial flow:

\[ a \lor a \quad (a \lor a) \land a \quad a \lor f \quad (a \lor a) \land f \quad \text{not pure} \]
**Purification**

**Pure Formulas:** $A \equiv t$ or $A \equiv f$ or $A$ is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
Purification

Pure Formulas: $A \equiv t$ or $A \equiv f$ or $A$ is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
Purification

**Pure** Formulas: \( A \equiv t \) or \( A \equiv f \) or \( A \) is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
**Purification**

**Pure Formulas:** \( A \equiv t \quad \text{or} \quad A \equiv f \quad \text{or} \quad A \) is equivalent to a unit-free formula.

Slice of a Combinatorial flow:
Purification of a formula:

\[ A \land t \leadsto A \]
\[ t \land A \leadsto A \]
\[ A \lor t \leadsto t \]
\[ t \lor A \leadsto t \]
\[ A \lor f \leadsto A \]
\[ f \lor A \leadsto A \]
\[ A \land f \leadsto f \]
\[ f \land A \leadsto f \]

Purification of combinatorial flows:

\[ (a \lor \bar{a}) \land a \]
\[ a \land \bar{a} \]
\[ a \lor \bar{a} \]
\[ a \lor f \]
\[ a \lor a \]
Purification of a formula:

\[
\begin{align*}
A \land t & \leadsto A \\
A \lor f & \leadsto A \\
\end{align*}
\]

Purification of combinatorial flows:
Purification of a formula:

\[
\begin{align*}
A \land t & \leadsto A \\
A \lor f & \leadsto A \\
A \lor t & \leadsto t \\
f \land f & \leadsto f \\
\end{align*}
\]

Purification of combinatorial flows:
Purification Example
Purification Example
Purification Example
Normalization (Work in Progress)
Normalization Example
Normalization Example
Normalization Example

\[ (b \land \overline{a}) \lor (a \land c) \]

\[ (b \lor a) \land (\overline{a} \lor c) \]

\[ b \lor c \]

\[ \vdash \]

\[ \land \]

\[ \lor \]
Normalization Example

\[(b \land \bar{a}) \lor (a \land c) \Rightarrow \bar{a} \lor (a \land c) \Rightarrow \bar{a} \lor c \Rightarrow \bar{a} \land c\]
Normalization Example

\[(b \land \bar{a}) \lor (a \land c) \Rightarrow b \lor c\]

\[(b \land a) \land (a \lor c) \Rightarrow (b \land \bar{a}) \land (a \lor c)\]

\[(b \lor a) \land (\bar{a} \lor c) \Rightarrow b \lor c \land (\bar{a} \lor c)\]

\[(b \lor a) \land (a \lor c) \Rightarrow (b \lor a) \land (a \lor c)\]

\[b \lor c \Rightarrow b \lor c\]
Normalization (Work in Progress)
Normalization (Work in Progress)

NOT confluent
NOT confluent and NOT terminating
What to remember from this talk?
What to remember from this talk?

≠  vs.  =
Future Work

- Normalization Termination
- Proof identity
- Other Logics (For example: Modal Logic and Intuitionistic Logic)
What to remember from this talk?

≠ vs. =
Normalization is not Confluent and not Terminating
Normalization is not Confluent and not Terminating

\[ a \lor \overline{a} \lor a \lor \overline{a} \lor (a \lor \overline{a}) \land (a \lor \overline{a}) \Rightarrow \]

\[ a \lor \overline{a} \lor a \lor \overline{a} \lor (a \lor \overline{a}) \land (a \lor \overline{a}) \]

\[ \]
Normalization is not Confluent and not Terminating
Yanking example5

\[ \bar{a} \lor b \lor (\bar{b} \lor (\bar{b} \land \bar{b})) \lor ((\bar{b} \lor \bar{b}) \land a) \lor \bar{b} \land \bar{b} \lor (b \lor (b \lor b)) \lor ((b \lor b) \land a) \lor \bar{b} \land \bar{b} \]

\[ \bar{a} \lor b \lor \bar{b} \lor b \land \bar{b} \lor (\bar{b} \land \bar{b}) \lor ((\bar{b} \lor \bar{b}) \land a) \lor \bar{a} \land \bar{b} \lor b \land b \]

\[ \bar{a} \lor b \lor \bar{b} \lor b \land \bar{b} \lor (\bar{b} \land \bar{b}) \lor ((\bar{b} \lor \bar{b}) \land a) \lor \bar{a} \land \bar{b} \lor b \land b \]

\[ \bar{a} \land b \lor b \land \bar{b} \lor \bar{a} \land \bar{b} \lor b \land b \]

\[ \bar{a} \land b \lor b \land \bar{b} \lor \bar{a} \land \bar{b} \lor b \land b \]

\[ \bar{a} \land b \lor b \land \bar{b} \lor \bar{a} \land \bar{b} \lor b \land b \]
A combinatorial proof with cuts for the sequent $\Gamma$ is a combinatorial proof for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots C_n$ are cut formulas. (Everything is unit-free)
A combinatorial proof with cuts for the sequent $\Gamma$ is a combinatorial proof for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots C_n$ are cut formulas. (Everything is unit-free)

Translating a combinatorial proof with cuts of $\bar{A}, B$ to a combinatorial flow from $A$ to $B$: 

\[
\begin{align*}
\phi & \quad \phi' \\
H_A \lor H_B & \quad H_A \\
\psi_A^{\downarrow} \lor \psi_{BC}^{\downarrow} & \quad H_A \\
\bar{A} \lor B \lor (C_1 \land \bar{C}_1) \lor \cdots \lor (C_n \land \bar{C}_n) & \quad B \lor (C_1 \land \bar{C}_1) \lor \cdots \lor (C_n \land \bar{C}_n)
\end{align*}
\]
A combinatorial proof with cuts for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots, C_n$ are cut formulas. (Everything is unit-free)

Translating a combinatorial flow with premise $A$ and conclusion $B$ to a combinatorial proof with cuts for $\bar{A}, B$:
A combinatorial proof with cuts for the sequent $\Gamma$ is a combinatorial proof for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots C_n$ are cut formulas. (Everything is unit-free)

Translating a combinatorial flow with premise $A$ and conclusion $B$ to a combinatorial proof with cuts for $\bar{A}, B$:
A combinatorial proof with cuts for the sequent $\Gamma$ is a combinatorial proof for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots, C_n$ are cut formulas. (Everything is unit-free)

Translating a combinatorial flow with premise $A$ and conclusion $B$ to a combinatorial proof with cuts for $\bar{A}, B$:
Combinatorial Flows vs. Combinatorial Proofs with cuts

A combinatorial proof with cuts for the sequent $\Gamma$ is a combinatorial proof for the sequent $\Gamma, C_1 \land \bar{C}_1, \ldots, C_n \land \bar{C}_n$ where $C_1, \ldots, C_n$ are cut formulas. (Everything is unit-free)

Translating a combinatorial flow with premise $A$ and conclusion $B$ to a combinatorial proof with cuts for $\bar{A}, B$: 

\[
\begin{array}{c}
\phi \\
\psi \\
\phi_B^\perp \lor \phi_{BC} \\
\psi_B^\perp \lor \psi_{DC'} \\
A \lor B \lor (C_1 \land \bar{C}_1) \lor \ldots \\
\bar{A} \lor (B \land \bar{B}) \lor D \lor (C_i' \land \bar{C}_i) \lor (C'_i \land \bar{C}_i') \lor \ldots
\end{array}
\]