# Formalization of Graph Theory in HOL: the Maxflow - Mincut Theorem

Niccolò Veltri

September 15, 2012

# Maxflow - Mincut theorem

## Problem

Given a network with only one source and only one sink, find a maximum value flow for the network.

Maxflow - Mincut theorem gives the solution to this issue.

- Important theorem in graph theory and linear programming.
- Can be used to derive Menger's theorem and König's theorem.
- Numerous applications.

# Maxflow - Mincut theorem

## Problem

Given a network with only one source and only one sink, find a maximum value flow for the network.

Maxflow - Mincut theorem gives the solution to this issue.

- Important theorem in graph theory and linear programming.
- Can be used to derive Menger's theorem and König's theorem.
- Numerous applications.

# Maxflow - Mincut theorem

## Problem

Given a network with only one source and only one sink, find a maximum value flow for the network.

Maxflow - Mincut theorem gives the solution to this issue.

- Important theorem in graph theory and linear programming.
- Can be used to derive Menger's theorem and König's theorem.
- Numerous applications.

# The theorem's statement

**Theorem (MAXFLOW_MINCUT)**

*In a network N the maximum flow value is equal to the minimum capacity of a s-t cut of N.*

# Necessary definitions

- Directed graph [DIR_GRAPH]

```
|- DIR_GRAPH v e <=>
   (!x y. x,y IN e ==> x IN v /\ y IN v /\ ~(x = y))
```

- Network [NET]

```
|- NET v e s t c <=>
   DIR_GRAPH v e /\
   s IN v /\
   t IN v /\
   ~(s = t) /\
   (!x y. x,y IN e ==> &0 <= c (x,y)) /\
   (!x y. ~(x,y IN e) ==> c (x,y) = &0)
```

# Necessary definitions

- Directed graph [DIR_GRAPH]

```
|- DIR_GRAPH v e <=>
   (!x y. x,y IN e ==> x IN v /\ y IN v /\ ~(x = y))
```

- Network [NET]

```
|- NET v e s t c <=>
   DIR_GRAPH v e /\
   s IN v /\
   t IN v /\
   ~(s = t) /\
   (!x y. x,y IN e ==> &0 <= c (x,y)) /\
   (!x y. ~(x,y IN e) ==> c (x,y) = &0)
```

# Necessary definitions

- Flow [FLOW]

```
|- FLOW v e s t c f <=>
   NET v e s t c /\
   (!x y. x,y IN e ==> f (x,y) <= c (x,y)) /\
   (!x y. x,y IN e ==> &0 <= f (x,y)) /\
   (!w. w IN v /\ ~(w = s) /\ ~(w = t)
        ==> isum (IN_NODES e w) f = isum (OUT_NODES e w) f)
```

- Flow value [FLOW_VALUE]

```
|- FLOW_VALUE e s =
   (\f. isum (OUT_NODES e s) f - isum (IN_NODES e s) f)
```

# Necessary definitions

- Flow `[FLOW]`

```
|- FLOW v e s t c f <=>
   NET v e s t c /\
   (!x y. x,y IN e ==> f (x,y) <= c (x,y)) /\
   (!x y. x,y IN e ==> &0 <= f (x,y)) /\
   (!w. w IN v /\ ~(w = s) /\ ~(w = t)
        ==> isum (IN_NODES e w) f = isum (OUT_NODES e w) f)
```

- Flow value `[FLOW_VALUE]`

```
|- FLOW_VALUE e s =
   (\f. isum (OUT_NODES e s) f - isum (IN_NODES e s) f)
```

# Flow variation in the sink

For the property of conservation of flow in inner vertexes we have [FLOW_IN_SOURCE_EQ_FLOW_OUT_END]

```
|- !v e s t c f.
       FINITE v /\ FLOW v e s t c f
       ==> FLOW_VALUE e s f =
           isum (IN_NODES e t) f -
           isum (OUT_NODES e t) f
```

# Necessary definitions

- Directed walk `[DIR_WALK]`

```
|- (!a b e. DIR_WALK e a b [] <=> a = b) /\
   (!a b e h h1 h2 t.
        DIR_WALK e a b (CONS (h,h1,h2) t) <=>
        a = h /\
        h = h1 /\
        h1,h2 IN e /\
        DIR_WALK (e DELETE (a,h2)) h2 b t)
```

- *s*-*t* cut `[CUT]`

```
|- CUT v e s t c z <=>
   NET v e s t c /\
   z SUBSET e /\
   (!l. DIR_WALK e s t l ==> (?q. q IN z /\ MEM (FST q,q) l))
```

# Necessary definitions

- Directed walk [DIR_WALK]

```
|- (!a b e. DIR_WALK e a b [] <=> a = b) /\
   (!a b e h h1 h2 t.
        DIR_WALK e a b (CONS (h,h1,h2) t) <=>
        a = h /\
        h = h1 /\
        h1,h2 IN e /\
        DIR_WALK (e DELETE (a,h2)) h2 b t)
```

- *s*-*t* cut [CUT]

```
|- CUT v e s t c z <=>
   NET v e s t c /\
   z SUBSET e /\
   (!l. DIR_WALK e s t l ==> (?q. q IN z /\ MEM (FST q,q) l))
```

# The theorem's statement in HOL Light

```
|- !v e s t c.
       FINITE v /\ NET v e s t c
       ==> (?f z.
                FLOW v e s t c f /\
                CUT v e s t c z /\
                FLOW_VALUE e s f = isum z c /\
                (!f'. FLOW v e s t c f'
                      ==> FLOW_VALUE e s f' <=
                          FLOW_VALUE e s f) /\
                (!z'. CUT v e s t c z'
                      ==> isum z c <= isum z' c))
```

# Generalized source

A *generalized source* of a network is a subset of vertexes such that it contains the source but not the sink. For a generalized source $V'$ it is valid that

$$v(f) = \sum_{v \in V'} \left( \sum_{e \in U(v)} f(e) - \sum_{e \in E(v)} f(e) \right).$$

$E(w) \leftrightsquigarrow$ `IN_NODES e w`
$U(w) \leftrightsquigarrow$ `OUT_NODES e w`

# Arcs pointing in (out) a subset of vertexes

Let $V'$ be a subset of vertexes in a network. We define $E(V')$ (respectively $U(V')$) as the the set of arcs that point in (out) $V'$:

```
|- IN_ARCS e v' = {x,y | x,y IN e /\ y IN v' /\ ~(x IN v')}

|- OUT_ARCS e v' = {x,y | x,y IN e /\ x IN v' /\ ~(y IN v')}
```

# Flow variation in a neighbourhood of the source

A generalized source behaves like a source
[FLOW_VALUE_FOR_SUBSET]:

```
|- !c v s t e f v'.
       FINITE v /\
       FLOW v e s t c f /\
       v' SUBSET v /\
       s IN v' /\
       ~(t IN v')
       ==> FLOW_VALUE e s f =
           isum (OUT_ARCS e v') f - isum (IN_ARCS e v') f
```

# The flow value never exceeds the capacity of a *s*-*t* cut

First important lemma
[FLOW_VALUE_IS_BOUNDED_BY_CUT_CAPACITY]:

```
|- !v e s t c f z.
      FINITE v /\ FLOW v e s t c f /\ CUT v e s t c z
      ==> FLOW_VALUE e s f <= isum z c
```

# The flow value never exceeds the capacity of a $s$-$t$ cut

Let $Z$ be a $s$-$t$ cut for the considered network.
Let $S$ be the set of vertexes connected to the source through directed walk that not contains arcs in $Z$.
If we prove that $U(S) \subseteq Z$ then the theorem `FLOW_VALUE_IS_BOUNDED_BY_CUT_CAPACITY` is proved.

$$v(f) = \sum_{e \in U(S)} f(e) - \sum_{e \in E(S)} f(e) \leq \sum_{e \in U(S)} f(e) \leq \sum_{e \in Z} f(e) \leq \sum_{e \in Z} c(e) = c(Z).$$

# The flow value never exceeds the capacity of a $s$-$t$ cut

Let $Z$ be a $s$-$t$ cut for the considered network.
Let $S$ be the set of vertexes connected to the source through directed walk that not contains arcs in $Z$.
If we prove that $U(S) \subseteq Z$ then the theorem `FLOW_VALUE_IS_BOUNDED_BY_CUT_CAPACITY` is proved.

$$v(f) = \sum_{e \in U(S)} f(e) - \sum_{e \in E(S)} f(e) \leq \sum_{e \in U(S)} f(e) \leq \sum_{e \in Z} f(e) \leq \sum_{e \in Z} c(e) = c(Z).$$

# The flow value never exceeds the capacity of a $s$-$t$ cut

To prove that $U(S) \subseteq Z$ we had to prove some basic properties of directed walks:

- DIR_WALK_EXTENSION

```
|- !l e a b c.
        DIR_WALK e a b l /\ b,c IN e /\ ~MEM (b,b,c) l
        ==> DIR_WALK e a c (APPEND l [b,b,c])
```

- DIR_WALK_SUBSET

```
|- !l a b e e'.
        e' SUBSET e /\ DIR_WALK e' a b l ==> DIR_WALK e a b l
```

# The flow value never exceeds the capacity of a *s*-*t* cut

To prove that $U(S) \subseteq Z$ we had to prove some basic properties of directed walks:

- DIR_WALK_EXTENSION

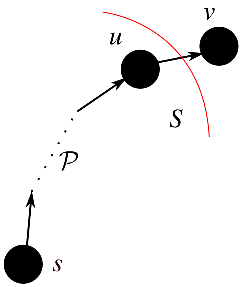
```
|- !l e a b c.
        DIR_WALK e a b l /\ b,c IN e /\ ~MEM (b,b,c) l
        ==> DIR_WALK e a c (APPEND l [b,b,c])
```

- DIR_WALK_SUBSET

```
|- !l a b e e'.
        e' SUBSET e /\ DIR_WALK e' a b l ==> DIR_WALK e a b l
```

# The flow value never exceeds the capacity of a $s$-$t$ cut

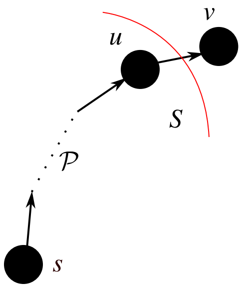
Let's take $e = (u, v) \in U(S)$. Then exists a directed walk $\mathcal{P}$ from $s$ to $u$ that not contains arcs in $Z$.




DIR_WALK_EXTENSION $\Rightarrow$ $e \in Z$

# The flow value never exceeds the capacity of a $s$-$t$ cut

Let's take $e = (u, v) \in U(S)$. Then exists a directed walk $\mathcal{P}$ from $s$ to $u$ that not contains arcs in $Z$.




$\texttt{DIR\_WALK\_EXTENSION} \quad \Rightarrow \quad e \in Z$

# Existence of a flow with greater value

Let's now see under which conditions, given a flow for a network, we can construct a flow with greater value. The key to solve this issue lies in the definition below of the set $S_f$ [SF] of a given flow $f$.

```
|- SF c f v e s = {w | w IN v /\ (?j. DIR_REACH c f v e s w j)}
```

where

```
|- (!a b v e c f. DIR_REACH c f v e a b [] <=> a = b /\ b IN v) /\
   (!a b v e c f h h1 h2 t.
         DIR_REACH c f v e a b (CONS (h,h1,h2) t) <=>
         a IN v /\
         a = h /\
         h1,h2 IN e /\
         (a = h1 /\
          f (h1,h2) < c (h1,h2) /\
          DIR_REACH c f (v DELETE a) (e DELETE (h1,h2)) h2 b t \/
          a = h2 /\
          &0 < f (h1,h2) /\
          DIR_REACH c f (v DELETE a) (e DELETE (h1,h2)) h1 b t))
```

# Existence of a flow with greater value

Second important result [GREATER_FLOW]:

```
|- !v e s t c f.
      FINITE v /\ FLOW v e s t c f /\ t IN SF c f v e s
      ==> (?f'.
              FLOW v e s t c f' /\
              FLOW_VALUE e s f < FLOW_VALUE e s f')
```

# $(i)$ Existence of $f'$

$t \in S_f$, then exists a DIR_REACH between the sorce and the sink.
Let's define for every arc $e$ in this DIR_REACH

$$\delta(e) = \begin{cases} c(e) - f(e) & \text{if } e \text{ points right} \\ f(e) & \text{if } e \text{ points left} \end{cases}$$

$\delta(e) > 0 \Rightarrow \delta = \min\{\delta(e) \mid e \text{ in the DIR\_REACH}\} > 0.$

# $(i)$ Existence of $f'$

$t \in S_f$, then exists a `DIR_REACH` between the sorce and the sink.
Let's define for every arc $e$ in this `DIR_REACH`

$$\delta(e) = \begin{cases} c(e) - f(e) & \text{if } e \text{ points right} \\ f(e) & \text{if } e \text{ points left} \end{cases}$$

$\delta(e) > 0 \Rightarrow \delta = \min\{\delta(e) \mid e \text{ in the } \texttt{DIR\_REACH}\} > 0.$

# $(i)$ Existence of $f'$

$$f'(e) = \begin{cases} f(e) + \delta & \text{if } e \text{ in the } \texttt{DIR\_REACH} \text{ and points right,} \\ f(e) - \delta & \text{if } e \text{ in the } \texttt{DIR\_REACH} \text{ and points left,} \\ f(e) & \text{if } e \text{ not in the } \texttt{DIR\_REACH}. \end{cases}$$

# $(ii)$ $f'$ is a flow

1. $f'(e) \leq c(e)$ for every $e \in E$;
2. $\sum_{e \in E(v)} f'(e) = \sum_{e \in U(v)} f'(e)$ for every $v \in V, v \neq s, t$.
   That is equivalent to prove that in the `DIR_REACH` there are as many arcs immediately preceding a vertex $v$ as arcs immediately following $v$.

# $(ii)$ $f'$ is a flow

1. $f'(e) \leq c(e)$ for every $e \in E$;
2. $\sum_{e \in E(v)} f'(e) = \sum_{e \in U(v)} f'(e)$ for every $v \in V, v \neq s, t$.
   That is equivalent to prove that in the `DIR_REACH` there are as many arcs immediately preceding a vertex $v$ as arcs immediately following $v$.

# $(ii)$ $f'$ is a flow

We choose the following bijection

```
\p. @q. (FST q = w /\ MEM (FST q,q) j) \/ (SND q = w /\ MEM (SND q,q) j)
```

The fundamental result turned out to be:

- EXISTS_ARC_IN_DIR_REACH

```
|- !l c f a b v e x w.
       DIR_REACH c f v e a b l /\
       ~(w = b) /\
       (MEM (x,x,w) l \/ MEM (x,w,x) l)
       ==> (?y. MEM (w,w,y) l \/ MEM (w,y,w) l)
```

# $(ii)$ $f'$ is a flow

We choose the following bijection

```
\p. @q. (FST q = w /\ MEM (FST q,q) j) \/ (SND q = w /\ MEM (SND q,q) j)
```

The fundamental result turned out to be:

- EXISTS_ARC_IN_DIR_REACH

```
|- !l c f a b v e x w.
        DIR_REACH c f v e a b l /\
        ~(w = b) /\
        (MEM (x,x,w) l \/ MEM (x,w,x) l)
        ==> (?y. MEM (w,w,y) l \/ MEM (w,y,w) l)
```

# $(iii)$ $f'$ has greater value than $f$

Let's $e_u$ be the last arc in the DIR_REACH. Let's assume it points right.

$v(f)$ measures the flow variation in the sink $+$

NO_ARC_BEGINS_WITH_TERMINAL $\Rightarrow$

$$
\begin{aligned}
v(f') &= \sum_{e \in E(t)} f'(e) - \sum_{e \in U(t)} f'(e) \\
&= \sum_{e \in E(t) \setminus e_u} f(e) + f'(e_u) - \sum_{e \in U(t)} f(e) \\
&= v(f) + \delta
\end{aligned}
$$

Similarly if $e_u$ points left.

# $(iii)$ $f'$ has greater value than $f$

Let's $e_u$ be the last arc in the DIR_REACH. Let's assume it points right.

$v(f)$ measures the flow variation in the sink $+$

NO_ARC_BEGINS_WITH_TERMINAL $\Rightarrow$

$$
\begin{aligned}
v(f') &= \sum_{e \in E(t)} f'(e) - \sum_{e \in U(t)} f'(e) \\
&= \sum_{e \in E(t) \setminus e_u} f(e) + f'(e_u) - \sum_{e \in U(t)} f(e) \\
&= v(f) + \delta
\end{aligned}
$$

Similarly if $e_u$ points left.

# Maxflow - Mincut theorem

The results proven give us the following `EXISTS_FLOW_WITHOUT_TERMINAL_IN_SF` :

```
|- !v e s t c.
        FINITE v /\ NET v e s t c
        ==> (?f. FLOW v e s t c f /\ ~(t IN SF c f v e s))
```

# Maxflow - Mincut theorem

We need to prove that the value of the flow found in `EXISTS_FLOW_WITHOUT_TERMINAL_IN_SF` is equal to the capacity of some *s*-*t* cut.

The theorem `OUT_ARCS_SF_IS_CUT` finds the cut:

```
|- !v e s t c f.
      FLOW v e s t c f /\ ~(t IN SF c f v e s)
      ==> CUT v e s t c (OUT_ARCS e (SF c f v e s))
```

# Maxflow - Mincut theorem

We need to prove that the value of the flow found in EXISTS_FLOW_WITHOUT_TERMINAL_IN_SF is equal to the capacity of some *s*-*t* cut.
The theorem OUT_ARCS_SF_IS_CUT finds the cut:

```
|- !v e s t c f.
       FLOW v e s t c f /\ ~(t IN SF c f v e s)
       ==> CUT v e s t c (OUT_ARCS e (SF c f v e s))
```

# Maxflow - Mincut theorem

The proof of OUT_ARCS_SF_IS_CUT is based on the theorem DIVIDED_DIR_WALK:

```
|- !l v e a b v'.
       DIR_GRAPH v e /\
       DIR_WALK e a b l /\
       a IN v' /\
       ~(b IN v') /\
       v' SUBSET v
       ==> (?x y l1 l2.
                l = APPEND l1 (CONS (x,x,y) l2) /\
                x IN v' /\
                ~(y IN v'))
```

# Maxflow - Mincut theorem

We just need to prove that $v(f) = c(U(S_f))$
[FLOW_VALUE_EQ_CUT_CAPACITY]:

```
|- !v e s t c f.
      FINITE v /\ FLOW v e s t c f /\ ~(t IN SF c f v e s)
      ==> FLOW_VALUE e s f = isum (OUT_ARCS e (SF c f v e s)) c
```

# Maxflow - Mincut theorem

If $e = (x, y) \in U(S_f)$ then exists a `DIR_REACH` from the source to $x$ and doesn't exists a `DIR_REACH` from the source to $y$. If $f(e) < c(e)$ then exists a `DIR_REACH` from the source to $y$, which is absurd.
Then $f(e) = c(e)$.
Similarly if $e \in E(S_f)$ then $f(e) = 0$.

# Maxflow - Mincut theorem

Therefore

$$v(f) = \sum_{e \in U(S_f)} f(e) - \sum_{e \in E(S_f)} f(e) = \sum_{e \in U(S_f)} c(e) = c(U(S_f))$$

and the theorem `MAXFLOW_MINCUT` is proven.

# De Bruijn's factor

The *De Bruijn factor* is the quotient of the size of a formalization of a mathematical text and the size of its informal original.
In HOL Light it's about 4.

- The De Bruijn factor of the Maxflow - Mincut theorem is 4;
- The formalization in HOL Light of the Maxflow - Mincut theorem consists in 3027 source lines of code, of which more than half are needed to prove the lemma on the exstence of a flow with greater value.

# De Bruijn's factor

The *De Bruijn factor* is the quotient of the size of a formalization of a mathematical text and the size of its informal original.
In HOL Light it's about 4.

- The De Bruijn factor of the Maxflow - Mincut theorem is 4;
- The formalization in HOL Light of the Maxflow - Mincut theorem consists in 3027 source lines of code, of which more than half are needed to prove the lemma on the exstence of a flow with greater value.

# What else is in the library

**Theorem**

*Let* $G = (V, E)$ *be a finite graph, then* $\sum_{v \in V} d_G(v) = 2|E|$.

**Theorem**

*Let* $G = (V, E)$ *a finite connected graph, then* $|E| \geq |V| - 1$.

**Theorem**

*Given a graph* $G = (V, E)$, *exists a polynomial* $P(G, c)$, *called chromatic polynomial, that counts the vertex coloring of G using c colors.*