Equivalence Between Systems Stronger Than Resolution

Maria Luisa Bonet
CS, UPC, Barcelona, Spain

Jordi Levy
IIIA, CSIC, Barcelona, Spain

SAT 2020, July 7-9, Alghero, Italy

SAT Solvers Based on Proof Systems

SAT solvers have 2 components:

- One tries to find a satisfying **assignment** (solves **SAT**)
- Another tries to find a **proof** (solves **TAUT**)

SAT Solvers Based on Proof Systems

SAT solvers have 2 components:

- One tries to find a satisfying **assignment** (solves **SAT**)
- Another tries to find a **proof** (solves **TAUT**)

Why base a SAT solver in a certain proof system?

- For most known proof systems there are **hard to prove** principles
- Most proof systems are difficult to **automatize** (able to find a proof in polynomial time on the size of the smallest proof)

SAT Solvers Based on Proof Systems

SAT solvers have 2 components:

- One tries to find a satisfying **assignment** (solves **SAT**)
- Another tries to find a **proof** (solves **TAUT**)

Why base a SAT solver in a certain proof system?

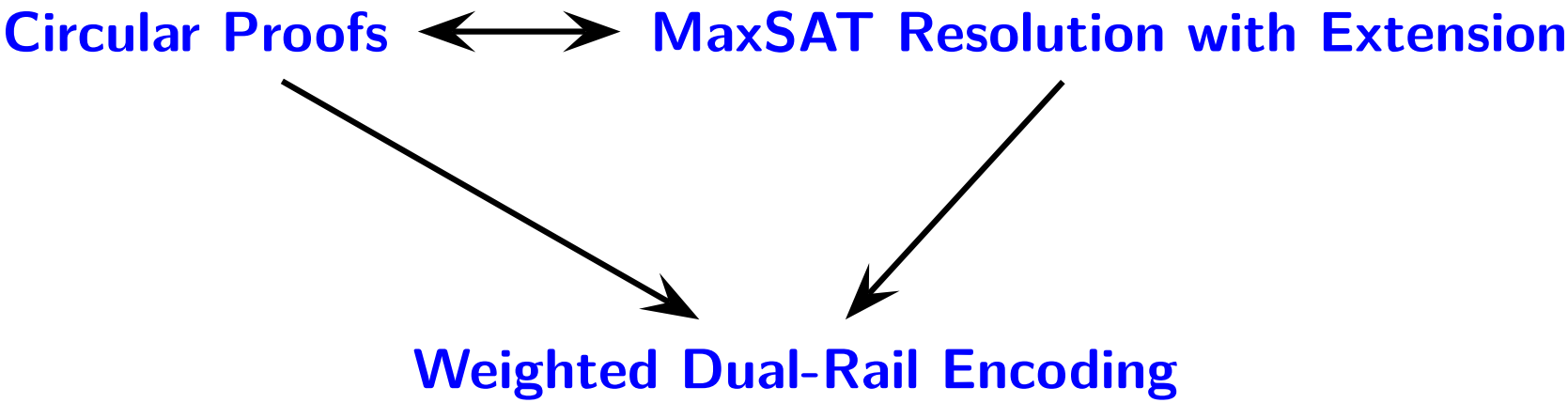
- For most known proof systems there are **hard to prove** principles
- Most proof systems are difficult to **automatize** (able to find a proof in polynomial time on the size of the smallest proof)

Aim: Find a proof system slightly stronger that resolution

- Able to "efficiently" proof **PHP**
- A proof system "easy" to **automatize**

Overview

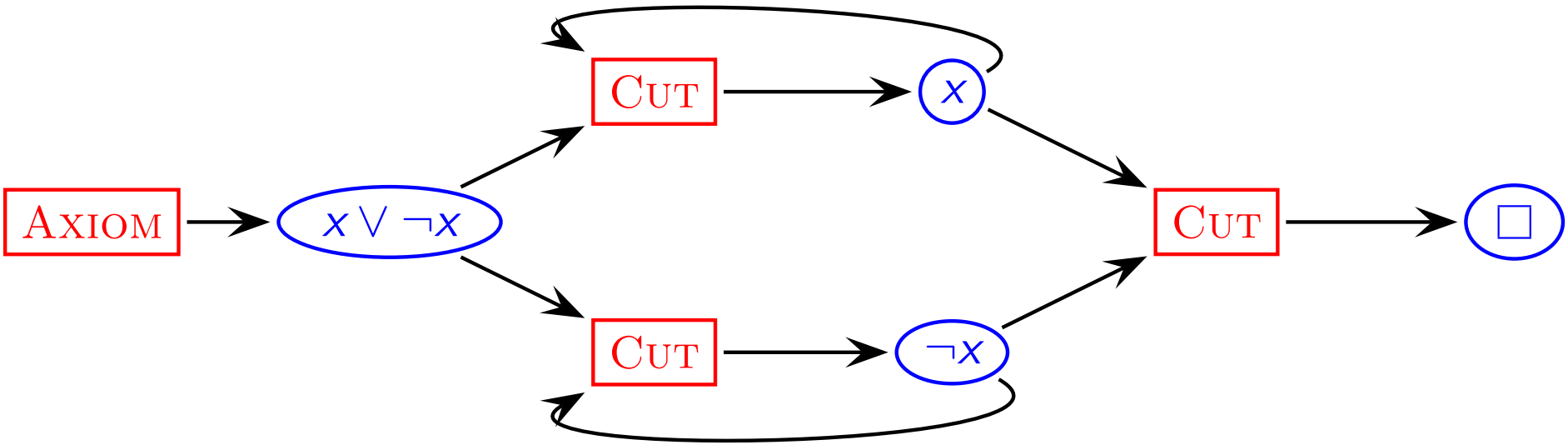
- We analyze the relative strength of 3 proof systems
- All of them efficiently prove **PHP** and simulate **Resolution**



Circular Proofs [Atserias & Lauria]

$$\frac{}{x \vee \neg x}\ \textsc{Axiom} \qquad \frac{x \vee A \qquad \neg x \vee A}{A}\ \textsc{Cut}$$

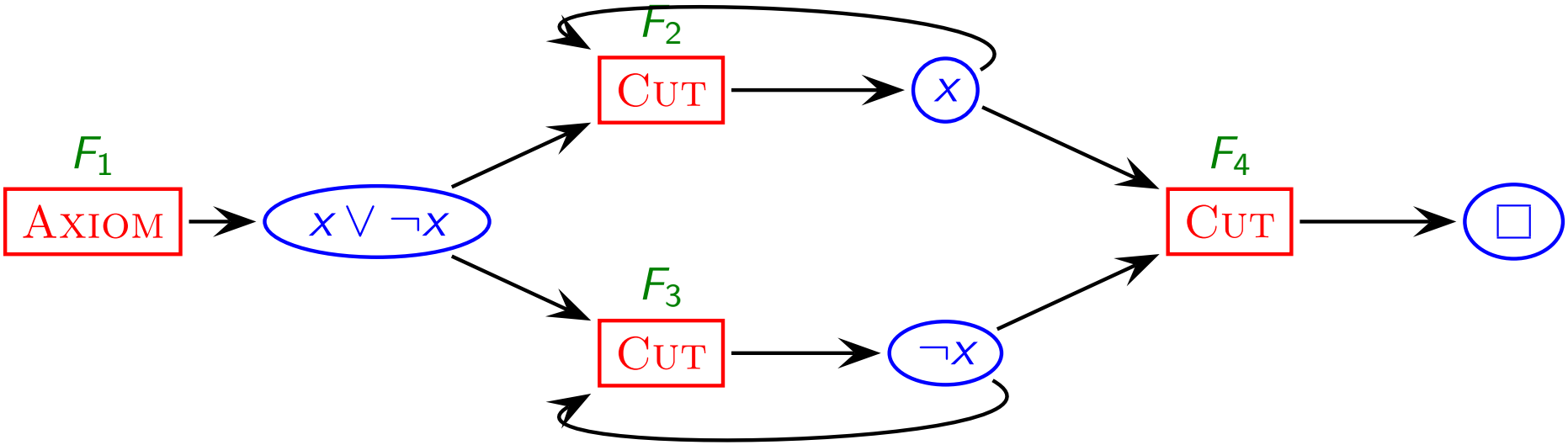
$$\frac{A}{x \vee A \qquad \neg x \vee A}\ \textsc{Split}$$



Circular Proofs [Atserias & Lauria]

$$\frac{}{x \vee \neg x}\ \textsc{Axiom} \qquad \frac{x \vee A \qquad \neg x \vee A}{A}\ \textsc{Cut}$$

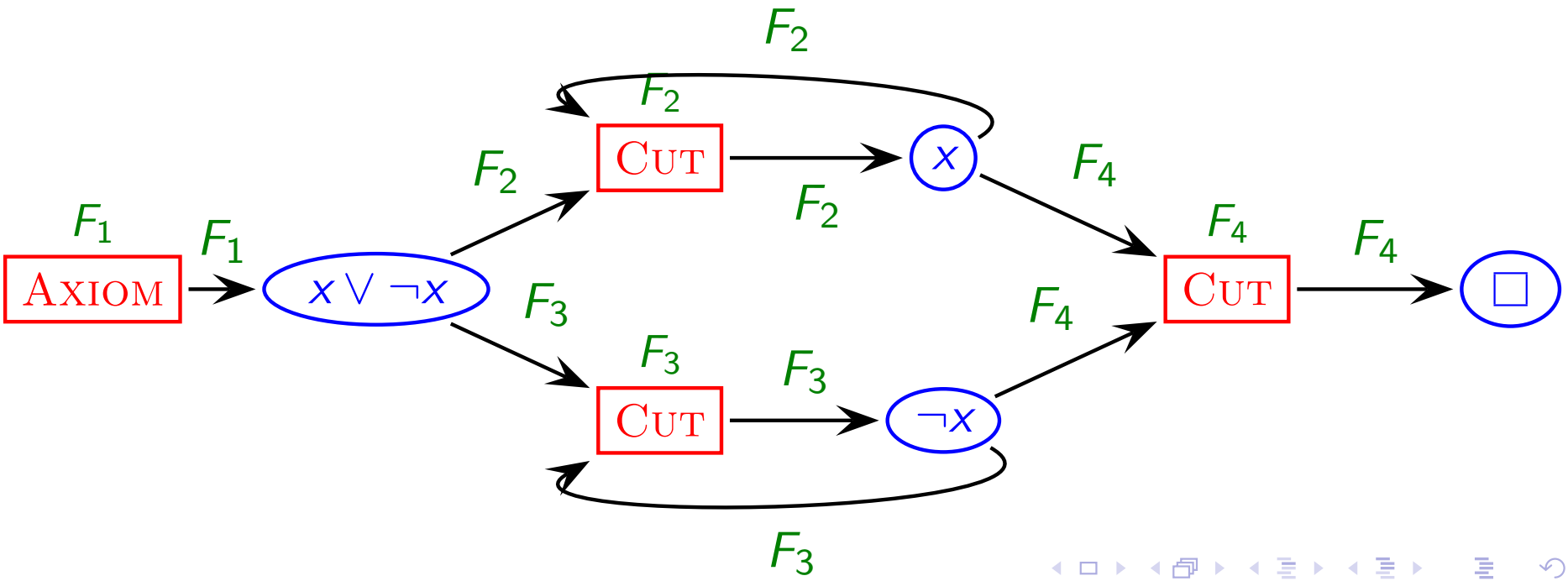
$$\frac{A}{x \vee A \qquad \neg x \vee A}\ \textsc{Split}$$



Circular Proofs [Atserias & Lauria]

$$\frac{}{x \vee \neg x}\ \textsc{Axiom} \qquad \frac{x \vee A \qquad \neg x \vee A}{A}\ \textsc{Cut}$$

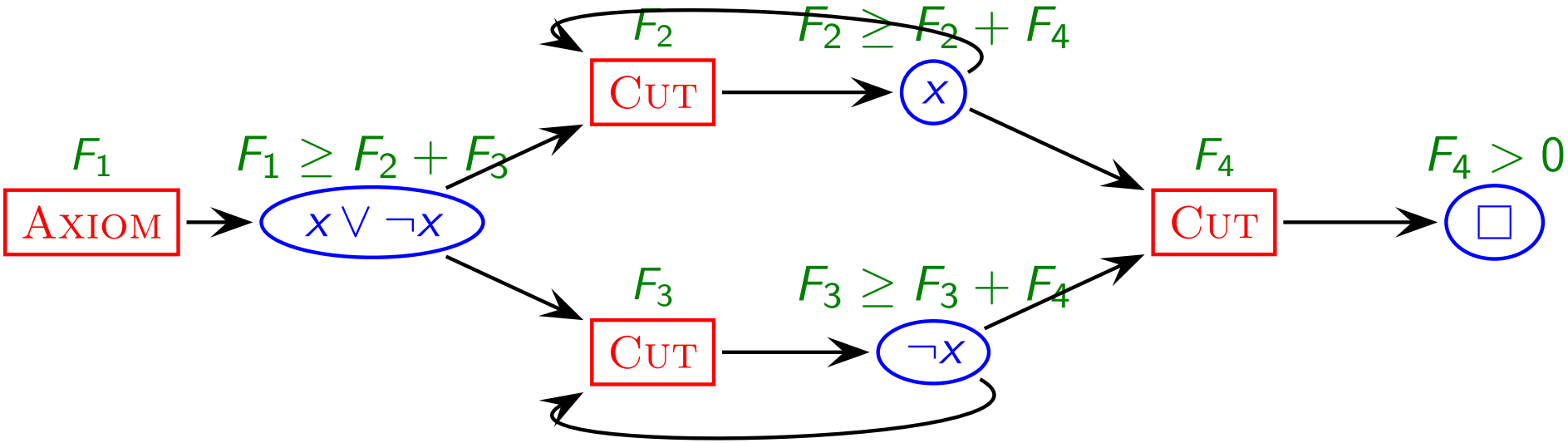
$$\frac{A}{x \vee A \qquad \neg x \vee A}\ \textsc{Split}$$



Circular Proofs [Atserias & Lauria]

$$\frac{}{x \vee \neg x}\ \textsc{Axiom} \qquad \frac{x \vee A \qquad \neg x \vee A}{A}\ \textsc{Cut}$$

$$\frac{A}{x \vee A \qquad \neg x \vee A}\ \textsc{Split}$$



MaxSAT Resolution with Extension [Larrosa & Rollon]

$$\frac{(x \vee A, u) \qquad (\neg x \vee B, u)}{\begin{array}{c}(A \vee B, u) \\ (x \vee A \vee \overline{B}, u) \\ (\neg x \vee B \vee \overline{A}, u)\end{array}} \text{ MaxSAT Resolution}$$

$$\frac{}{(A, -u) \qquad (x \vee A, u) \qquad (\neg x \vee A, u)} \text{ Extension}$$

- Clauses have a (positive or negative) **weight** associated
- Proceed **replacing** premises by consequences
- Clauses may be combined or decomposed:

$$\frac{(C, u) \qquad (C, v)}{(C, u + v)} \text{ Fold} \qquad \frac{(C, u + v)}{(C, u) \qquad (C, v)} \text{ Unfold}$$

(Weighted) Dual-Rail Encoding [Marques & Bonet et al.]

$$\frac{(x \vee A, u) \qquad (\neg x \vee B, u)}{\begin{array}{c}(A \vee B, u) \\ (x \vee A \vee \overline{B}, u) \\ (\neg x \vee B \vee \overline{A}, u)\end{array}}$$ MaxSAT Resolution

- Clauses have a **positive weight** associated
- Proceed **replacing** premises by consequences
- Clauses may be combined or decomposed

Dual-Rail Encoding:

- Change x_i by p_i, and $\neg x_i$ by n_i
- Add clauses $(\neg p_i \vee \neg n_i, \infty)$, (p_i, u_i) and (n_i, u_i)
- We can derive $1 + \sum_{i=1}^{n} u_i$ empty clauses

Split or Extension

Lemma

The Split *and* Extension *rules are equivalent in weighted proof*

$$\frac{(A, u)}{(A, u)} \quad \text{Extension}$$

$$\frac{(A, -u)\ (x \vee A, u)\ (\neg x \vee A, u)}{(x \vee A, u)\ (\neg x \vee A, u)} \quad \text{Fold}$$

$$\frac{\dfrac{}{(A, u)\ (A, -u)}}{(A, -u)\ (x \vee A, u)\ (\neg x \vee A, u)} \quad \begin{matrix}\text{Unfold} \\ \text{Split}\end{matrix}$$

Generalizing Weighted Proofs Using $\mathcal{R}$

$$\frac{(x \vee A, u) \qquad (\neg x \vee B, u)}{\begin{array}{c}(A \vee B, u) \\ (x \vee A \vee \overline{B}, u) \\ (\neg x \vee B \vee \overline{A}, u)\end{array}}$$ MaxSAT Resolution

$$\frac{(A, u)}{(x \vee A, u) \qquad (\neg x \vee A, u)}$$ Split *replacing* Extension

- Clauses have a (positive or negative) **weight** associated
- Proceed **replacing** premises by consequences
- Clauses may be combined or decomposed:

$$\frac{(C, u) \qquad (C, v)}{(C, u + v)}$$ Fold $$\frac{(C, u + v)}{(C, u) \qquad (C, v)}$$ Unfold

Generalizing Weighted Proofs Using $\mathcal{R}$

$$\frac{(x \vee A, u) \qquad (\neg x \vee B, u)}{\begin{array}{c}(A \vee B, u) \\ (x \vee A \vee \overline{B}, u) \\ (\neg x \vee B \vee \overline{A}, u)\end{array}} \text{ MaxSAT Resolution}$$

$$\frac{(A, u)}{(x \vee A, u) \qquad (\neg x \vee A, u)} \text{ Split } \textit{replacing} \text{ Extension}$$

$\mathcal{R}$

- Clauses have a (positive or negative) **weight** associated
- Proceed **replacing** premises by consequences in rules of $\mathcal{R}$ with same positive weight $u > 0$
- Clauses may be combined or decomposed:

$$\frac{(C, u) \qquad (C, v)}{(C, u + v)} \text{ Fold} \qquad \frac{(C, u + v)}{(C, u) \qquad (C, v)} \text{ Unfold}$$

MaxSAT Resolution or Symmetric Cut

Lemma

Weighted proofs using $\mathcal{R} = \{\text{MaxSAT Resolution, Split}\}$ *are poly-equivalent to proofs using* $\mathcal{R} = \{\text{Symmetric Cut, Split}\}$.

$$\frac{(x \vee A, u)\ (\neg x \vee B, u)}{(x \vee A \vee \neg b_1, u)\ (x \vee A \vee b_1, u)\ (\neg x \vee B, u)}\ \text{Split}$$

$$\frac{(x \vee A \vee \neg b_1, u)\ (x \vee A \vee b_1 \vee \neg b_2, u)\ (x \vee A \vee b_1 \vee b_2, u)\ (\neg x \vee B, u)}{(x \vee A \vee B, u)\ (x \vee A \vee \neg b_1, u) \cdots (x \vee A \vee b_1 \vee \cdots \vee b_{s-1} \vee \neg b_s, u)\ (\neg x \vee B, u)}\ (s-2) \times \text{Split}$$

$$\frac{(x \vee A \vee B, u)\ (\neg x \vee A \vee B, u)\ (x \vee A \vee \overline{B}, u)\ (\neg x \vee B \vee \overline{A}, u)}{(A \vee B, u)\ (x \vee A \vee \overline{B}, u)\ (\neg x \vee B \vee \overline{A}, u)}\ r \times \text{Split}\ \text{Symmetric Cut}$$

Our Results

Lemma

Weighted proofs *and* **Circular proofs** *using the* **same set** *of inference rules are polynomially equivalent*

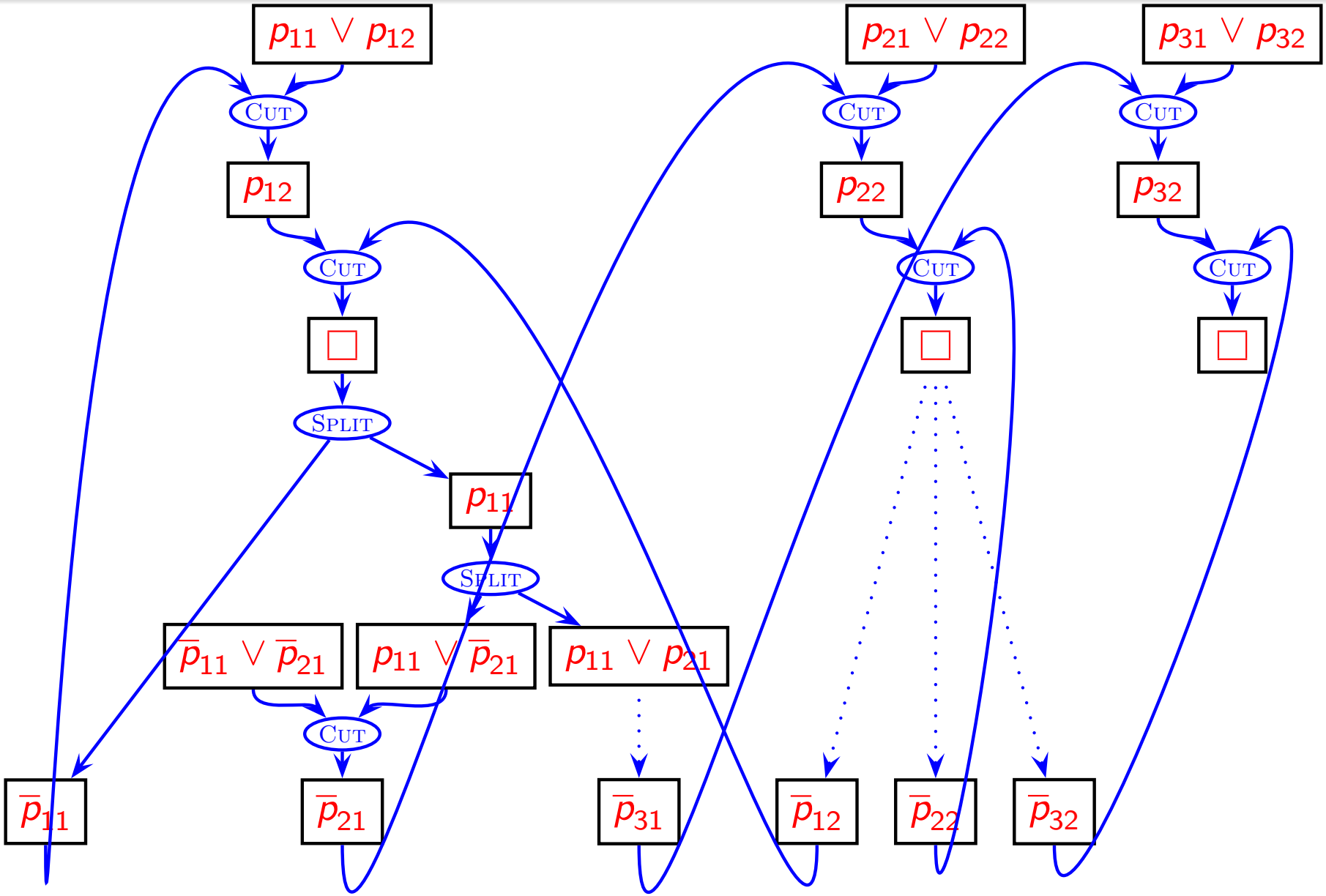
Theorem

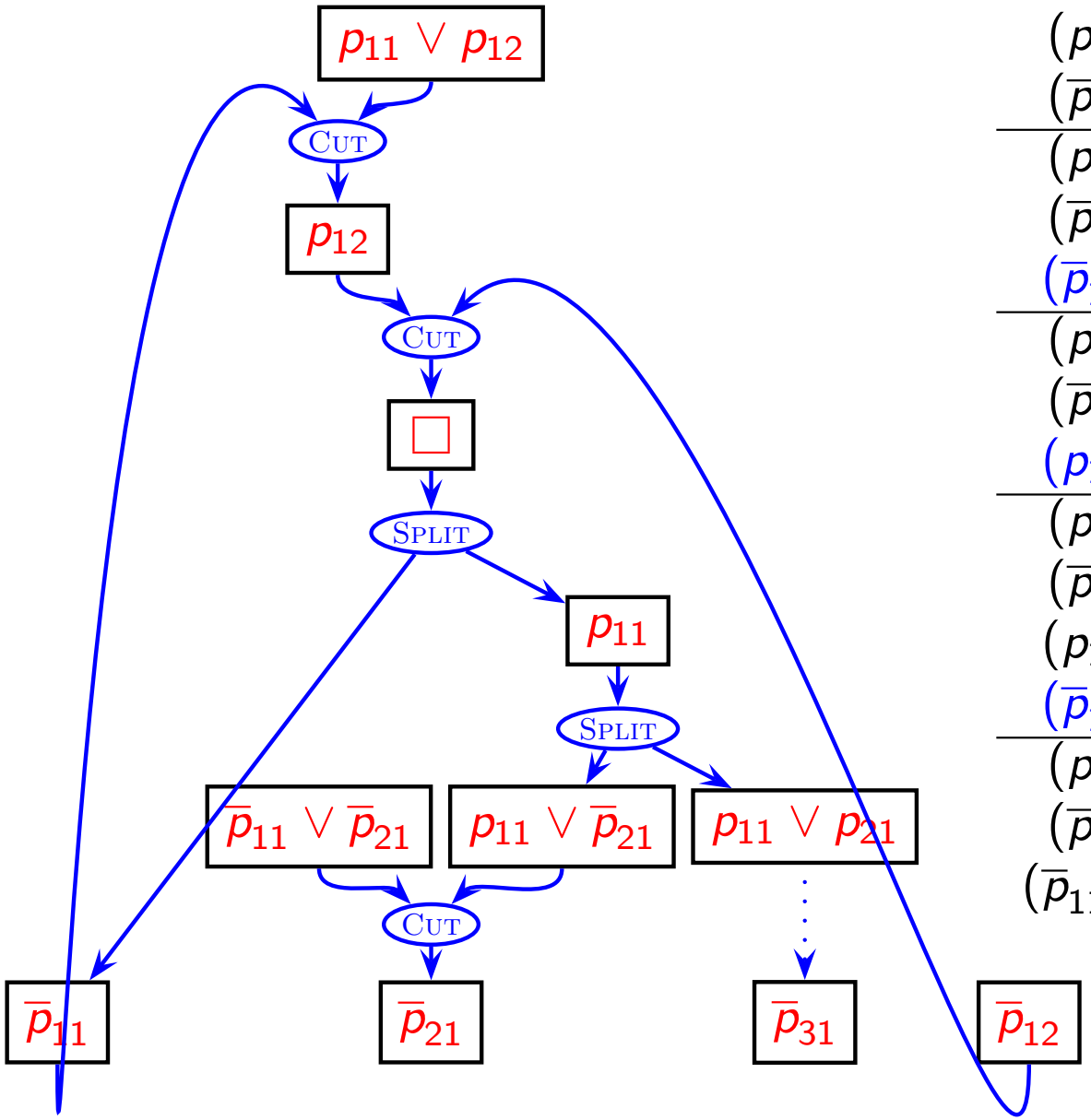
MaxSAT Resolution with Extension *and* **Circular Resolution** *are polynomially equivalent*

Theorem

The **weighted proof using** $\mathcal{R} = \{$**MaxSAT Resolution**, **0-Split**$\}$ *polynomially simulates the* **Weighted Dual-Rail MaxSAT**

$$\frac{\square}{x \qquad \neg x}\ 0\text{-Split}$$


$p_{11} \vee p_{12}$
$p_{21} \vee p_{22}$
$p_{31} \vee p_{32}$
Cut
p_{12}
p_{22}
p_{32}
□
Split
p_{11}
$\overline{p}_{11} \vee \overline{p}_{21}$
$p_{11} \vee \overline{p}_{21}$
$p_{11} \vee p_{21}$
$\overline{p}_{11}$
$\overline{p}_{21}$
$\overline{p}_{31}$
$\overline{p}_{12}$
$\overline{p}_{22}$
$\overline{p}_{32}$



$$\frac{(p_{11} \vee p_{12}, 1), \cdots \quad (\overline{p}_{11} \vee \overline{p}_{21}, 1), \cdots}{(p_{11} \vee p_{12}, 1), \cdots \quad (\overline{p}_{11} \vee \overline{p}_{21}, 1), \cdots} \text{ Unfold}$$

$$\frac{(\overline{p}_{11}, 1), (\overline{p}_{11}, -1)}{(p_{21} \vee p_{22}, 1), \cdots \quad (\overline{p}_{11} \vee \overline{p}_{21}, 1), \cdots} \text{ Cut}$$

$$\frac{(p_{12}, 1), (\overline{p}_{11}, -1)}{(p_{21} \vee p_{22}, 1), \cdots \quad (\overline{p}_{11} \vee \overline{p}_{21}, 1), \cdots} \text{ Unfold}$$

$$\frac{(\overline{p}_{12}, 1), (\overline{p}_{12}, -1)}{(p_{21} \vee p_{22}, 1), \cdots \quad (\overline{p}_{11} \vee \overline{p}_{21}, 1), \cdots} \text{ Cut}$$

$$(p_{12}, 1), (\overline{p}_{11}, -1)$$

$$(\overline{p}_{11}, -1), (\overline{p}_{12}, -1)$$

$$(\square, 1)$$

Conclusions and Further Work

- Circular proofs and weighted proofs are equivalent and parameterizable on a set of rules $\mathcal{R}$
- Both implement a sort of *irreversible* proofs where it makes sense to prove twice the same clause and where we can use clauses still not proved
- Split makes Symmetric Cut complete, and allow us to *reverse* its application
- 0-split is enough to prove PHP efficiently
- Further work: Design a proof system for practical SAT solvers:
  - Use some sort of Split rule
  - Use some sort of circularity or (negative) weights