# Proof logging the Generalized Totalizer Encoding

**Carlos Cantero**
KU Leuven, Belgium

**Bart Bogaerts**
KU Leuven, Belgium
Vrije Universiteit Brussel, Belgium

**Dieter Vandesande**
KU Leuven, Belgium
Vrije Universiteit Brussel, Belgium

## Abstract

There are many different pseudo-Boolean encodings, including the Generalized Totalizer Encoding, and various proof logging techniques have been developed for some of them. Those techniques are able to certify certain correctness properties in the context of, for example, pseudo-Boolean solving and MaxSAT solving. In this work, we present a cutting planes derivation of the original pseudo-Boolean constraint from the clauses generalized by its Generalized Totalizer Encoding; together with a previous proof, this derivation can be used to certify via proof logging the one-to-one correspondence of the models of a constraint and its Generalized Totalizer Encoding, a crucial correctness property for pseudo-Boolean model counting. We also show that the traditional definition of the Generalized Totalizer Encoding is incomplete in this sense, and offer a simple way to fix it.

**2012 ACM Subject Classification** Theory of computation → Constraint and logic programming; Mathematics of computing → Combinatorial optimization

**Keywords and phrases** proof logging, pseudo-boolean, totalizer, generalized totalizer, encoding, cutting planes, reification, combinatorial optimization, pseudo-boolean solving

**Digital Object Identifier** 10.4230/LIPIcs.DP.2025.

## 1 Introduction

During the last few decades, combinatorial optimization techniques have quickly improved, and we are able to solve a higher number of problems quicker and quicker every year. While this is generally a good thing, it also brings with it important technical difficulties, like safety concerns: we may be content with the low percentage of failure that plain testing allows for most algorithms, but for some sensitive cases, like scheduling of organ transplants or automated public transport, we would like to achieve zero error.

This is possible through formal verification, a series of techniques that ensure the correctness of a system with respect with a mathematical specification, proving a number of conditions are satisfied, and, in particular, allowing to prove that an algorithm will never fail. But traditional formal verification techniques, like model checking, are slow and complicated processes and can hardly keep up with the fast evolution of combinatorial optimization algorithms: an alternative is proof logging, a much more suitable verification technique for the particularities of this field.

Proof logging consists in making algorithms output a small proof of the correctness of a particular execution, which can later be checked by a much simpler proof checker, fit to be formally verified. Proof logging an algorithm does not guarantee the general correctness of the algorithm, but it does ensure the correctness of every execution accepted by the proof checker. This offers both weaker and stronger guarantees with respect to traditional formal verification: weaker because they are less general, and stronger because they also certify every execution is free from isolated problems, like bit-flips due to gamma rays. Proof logging

also offers a great opportunity for auditability, providing useful debugging information when something does go wrong.

This is why we can observe an increased interest in proof logging techniques in the last two decades, as exemplified by the mandatory introduction of proof logging for SAT solvers in the main track of the SAT competition. Another discipline that can benefit from proof logging is the field of pseudo-Boolean-to-CNF encodings: the translation of pseudo-Boolean constraints into CNF formulas. A possible use of that proof logging is the complete certification of pseudo-Boolean solvers, since a number of them work by translating the input into CNF and then applying a SAT solver to the resulting clauses: at the moment, only the proof logging of the second part of this process has been extensively adopted.

There are many different PB-to-CNF encodings, one of which is the Generalized Totalizer Encoding (GTE). First introduced for cardinality constraints (pseudo-Boolean constraints in which every weight is equal to one) in [1], it was extended to general pseudo-Boolean constraints in [4]. The task of proof logging a PB-to-CNF encoding has two directions: *constraint to clauses*, where we prove that every model (satisfying assignment) of the original pseudo-Boolean constraint is also a model of the resulting encoding, and *clauses to constraint*, in which we prove the converse.[1] The direction *constraint to clauses* of the GTE was certified in [6], and it allows us to produce proofs of unsatisfiability, which is why it has been used to certify the iterative MaxSAT solver QMaxSAT [7]. A proof of the converse direction would enable us to certify techniques that need to preserve one-to-one correspondence of models between clauses and constraint, like pseudo-Boolean model counting.

## Our contributions

In this paper, we present a proof of the *clauses to constraint* direction, to the best of our knowledge for the first time. Following the lead of [6], we will show that direction via a cutting planes derivation, in such a way that the proof can be readily transformed to a proof logging implementation in VeriPB style.

With this proof, we confirm there is a one-to-one correspondence of models between the GTE encoding and the original pseudo-Boolean constraint. Additionally, we show that the traditional definition of the GTE based in [4] is incomplete, in the sense that it does not preserve one-to-one correspondence, and we present a simple extension to the definition that makes it complete.

Let us overview the structure of the paper. In Section 2, we introduce all the necessary definitions: we explain what a pseudo-Boolean constraint is, we define the GTE, we explain how a cutting planes derivation works and we outline our proof logging method; in Section 3, we show why the traditional definition of the GTE is incomplete by means of a counterexample; finally, in Section 4, we present our proof of the *clauses to constraint* direction and we explain what a proof logging implementation certifying one-to-one correspondence would look like.

[1] More precisely, since in general the PB-to-CNF encodings introduce new variables, what we prove in the *clauses to constraint* direction is that every model of the encoding can be restricted to a model of the original constraint, so that in the end, if we prove both directions, we obtain a one-to-one correspondence of models rather than an equivalence.

## 2 Preliminaries

▶ **Definition 1** (Pseudoboolean constraint). *A pseudo-Boolean constraint is an inequality of the form*

$$\sum_{i=1}^{n} w_i \cdot l_i + j_1 * \sum_{i=1}^{m} w'_i \cdot l'_i + j_2,$$

*where all $w_i$, all $w'_i$, $j_1$ and $j_2$ are integers, $l_i$ and $l'_i$ are literals (a boolean variable or the negation of a boolean variable), and $* \in \{\leq, \geq\}$.*

▶ **Definition 2** (Normalized form). *A pseudo-Boolean constraint in normalized form has the following structure:*

$$\sum_{i=1}^{n} w_i \cdot l_i \geq j,$$

*where all $w_i$ (the* weights *of the literals) are positive integers and $j$ is a nonnegative integer.*

Since every pseudo-Boolean constraint can be easily transformed into a normalized pseudo-Boolean constraint (as detailed in [5]), from now on we will assume every constraint is in normalized form. We presume the reader is familiar with Boolean logic and the definition of CNF; note a clause $\bigvee_{i=1}^{n} l_i$ can be straightforwardly represented as a constraint: $\sum_{i=1}^{n} l_i \geq 1$.

Now, we will introduce our definition of the GTE. We begin by defining a binary-tree structure and some auxiliary concepts that will aid us in the the actual definition.

▶ **Definition 3** (Generalized totalizer tree). *A generalized totalizer tree (GTT) $T$ of a constraint $\sum_{i=1}^{n} w_i \cdot l_i \geq j$ is a binary tree the leaves of which are the literals $l_1, \ldots, l_n$ from the given constraint.*

▶ **Definition 4** (leaves). *Let $T$ be a GTT. For every leaf $l$ of a $T$ we define* $\text{leaves}(l) = \{l\}$, *and for every internal node $\eta$ of $T$ we define* $\text{leaves}(\eta) = \text{leaves}(\eta_1) \cup \text{leaves}(\eta_2)$.

▶ **Definition 5** (weight). *Let $T$ be a GTT and $\eta$ be a node of $T$. Then,* $\text{weight}(\eta)$ *represents the sum of the weights of all leaves which are descendants of $\eta$. Formally, for every leaf $l = l_i$, we define* $\text{weight}(l) = w_i$, *where $w_i$ is the weight of the literal $l_i$ in the given constraint, and for every internal node $\eta$, we say* $\text{weight}(\eta) = \text{weight}(\eta_1) + \text{weight}(\eta_2)$.

▶ **Definition 6** (sumweights). *Let $T$ be a GTT. For every node $\eta$ of $T$, the sequence* $\text{sumweights}(\eta)$ *is intended to represent all possible combinations of weights of the leaves of $\eta$ in increasing order. Formally, for every leaf $l$, we define* $\text{sumweights}(l) = \langle \text{weight}(l) \rangle$; *for every internal node $\eta$, we define* $\text{sumweights}(\eta)$ *as the sequence of all the elements from the following set in increasing order:*

$$\text{sumweights}(\eta_1) \cup \text{sumweights}(\eta_2) \cup \{w_1 + w_2 : w_1 \in \text{sumweights}(\eta_1) \wedge w_2 \in \text{sumweights}(\eta_2)\}.$$

*Let us denote the $i$-th element of* $\text{sumweights}(\eta)$ *by $s^\eta_i$, where the smallest index is $i = 1$. If the context makes $\eta$ clear, we sometimes say just $s_i$. For every node $\eta$, we convene $s^\eta_0 = 0$ and $s^\eta_{|\text{sumweights}(\eta)|+1} = s^\eta_{|\text{sumweights}(\eta)|} + 1$.*

▶ **Definition 7** (Counting variables). *Let $T$ be a GTT. We introduce a variable $y^\eta_{s_j}$ for every node $\eta$ of $T$ and every $1 \leq j \leq |\text{sumweights}(\eta)|$, with the intended meaning that the sum of the true weighted literals from* $\text{leaves}(\eta)$ *adds up to at least $s_j$. We convene $y^\eta_0 = 1$ and $y^\eta_{|\text{sumweights}(\eta)|+1} = 0$, and $y^l_{s_1} = l$ for every leaf $l$.*

▶ **Definition 8** (Generalized totalizer encoding). *Let $T$ be a GTT. For every internal node $\eta$ of $T$, its GTE is the set of the clauses*

$$C_1^\eta(a,b,c) = \bar{y}_{s_a}^{\eta_1} \vee \bar{y}_{s_b}^{\eta_2} \vee y_{s_c}^{\eta}$$
$$C_2^\eta(a,b,c) = y_{s_{a+1}}^{\eta_1} \vee y_{s_{b+1}}^{\eta_2} \vee \bar{y}_{s_{c+1}}^{\eta}$$

*for all values of $0 \leq a \leq |\text{sumweights}(\eta_1)|$ and $0 \leq b \leq |\text{sumweights}(\eta_2)|$ and $0 \leq c \leq |\text{sumweights}(\eta)|$ such that $s_a^{\eta_1} + s_b^{\eta_2} = s_c^{\eta}$, and of the clauses*

$$C_3^\eta(j) = \bar{y}_{s_j}^{\eta} \vee y_{s_{j-1}}^{\eta}$$

*for every $1 < j \leq |\text{sumweights}(\eta)|$. We have $C_1^\eta(a,b,c)$ represents $y_{s_a}^{\eta_1} \wedge y_{s_b}^{\eta_2} \rightarrow y_{s_c}^{\eta}$, while $C_2^\eta(a,b,c)$ represents $\bar{y}_{s_{a+1}}^{\eta_1} \wedge \bar{y}_{s_{b+1}}^{\eta_2} \rightarrow \bar{y}_{s_{c+1}}^{\eta}$, meaning that if the leaves of $\eta_1$ add up to at most $s_a$ and the leaves of $\eta_2$ add up to at most $s_b$, then the leaves of $\eta$ add up to at most $s_c$. $C_3^\eta(j)$ represents $y_{s_j}^{\eta} \rightarrow y_{s_{j-1}}^{\eta}$, meaning that if the sum of the true leaves of $\eta$ adds up to at least $s_j$, then it must add to at least $s_{j-1}$.*

*The GTE of $T$ is the union of the GTEs of every internal node of $T$. Finally, let $\sum_{i=1}^{n} w_i \cdot l_i \geq j$ be a constraint, let $T$ be the its GTT, let $\eta_{\text{rt}}$ be the root node of $T$ and let $0 \leq k \leq |\text{sumweights}(\eta_{\text{rt}})|$ be the least index such that $s_k^{\eta_{\text{rt}}} \geq j$ (if $j > s_{|\text{sumweights}(\eta_{\text{rt}})|}^{\eta_{\text{rt}}}$, then simply $k = |\text{sumweights}(\eta_{\text{rt}})| + 1$). Then, the GTE of $\sum_{i=1}^{n} w_i \cdot l_i \geq j$ is the union of the GTE of $T$ and $\{y_{s_k}^{\eta_{\text{rt}}}\}$.*

Now, we present the basics of the cutting planes proof system (introduced in [3]), which our proof will be based on. This proof system reasons with pseudo-Boolean constraints and is implicationally complete (meaning that given a set of constraints as input, it can derive any constraint that is implied by them), making it perfect for our proof logging purposes. This is why VeriPB (see [2]), the leading tool used for pseudo-Boolean proof logging, is based on this proof system. The main rules of the cutting planes proof system are the following:

▶ **Definition 9** (Main rules).

$$\frac{}{l \geq 0}\ \textit{Literal Axiom} \qquad \frac{\sum_{i=1}^{n} w_i \cdot l_i \geq j_1 \qquad \sum_{i=1}^{m} w_i' \cdot l_i' \geq j_2}{\sum_{i=1}^{n} w_i \cdot l_i + \sum_{i=1}^{m} w_i' \cdot l_i' \geq j_1 + j_2}\ \textit{Addition Rule}$$

$$\frac{\sum_{i=1}^{n} w_i \cdot l_i \geq j}{\sum_{i=1}^{n} (c \cdot w_i) \cdot l_i \geq c \cdot j}\ \textit{Multiplication Rule} \qquad \frac{\sum_{i=1}^{n} w_i \cdot l_i \geq j}{\sum_{i=1}^{n} \lceil w_i/c \rceil \cdot l_i \geq \lceil j/c \rceil}\ \textit{Division Rule}$$

The previous rules are enough to achieve implicational completeness, but we also include the following two rules to allow for shorter derivations:

▶ **Definition 10** (Additional rules).

$$\frac{\sum_{i=1}^{n} w_i \cdot l_i \geq j}{\sum_{i=1}^{n} \min(j, w_i) \cdot l_i \geq j}\ \textit{Saturation Rule} \qquad \frac{w \cdot l + \sum_{i=1}^{n} w_i \cdot l_i \geq j}{\sum_{i=1}^{n} w_i \cdot l_i \geq j - w}\ \textit{Weakening Rule}$$

We say there is a cutting planes proof of constraint $C$ from a set of constraints $S$ when there is a sequence of cutting planes rules that takes as input the constraints from $S$ and ends with $C$.

Then, our objective is to derive the original constraint from its GTE via a cutting planes proof. But we will prove something stronger: we will show that from the GTE of a constraint, we can derive by cutting planes all the reification constraints of the counting variables associated with the given constraint, including the constraint itself. The reification constraints of a counting variable $y^\eta_{s_j}$ assert its intended meaning:

$$y^\eta_{s_j} \longleftrightarrow \sum_{l \in \text{leaves}(\eta)} \text{weight}(l) \cdot l \geq s_j. \qquad \text{Formally,}$$

▶ **Definition 11** (Reification constraints). *For a node $\eta$ of a GTT and $0 \leq j \leq |\text{sumweights}(\eta)|$, the reification constraints of $y^\eta_{s_j}$ are:*

$$C^{\rightarrow}_{\text{reif}}(y^\eta_{s_j}) = s_j \cdot \bar{y}^\eta_{s_j} + \sum_{l \in \text{leaves}(\eta)} \text{weight}(l) \cdot l \geq s_j$$
$$C^{\leftarrow}_{\text{reif}}(y^\eta_{s_j}) = (\text{weight}(\eta) - s_j + 1) \cdot y^\eta_{s_j} + \sum_{l \in \text{leaves}(\eta)} \text{weight}(l) \cdot \bar{l} \geq \text{weight}(\eta) - s_j + 1.$$

The reification constraints are the touchstones we use when proof logging the GTE: if we can show that there is a one-to-one correspondence of models between reification constraints and their associated GTE, then we are sure that the counting variables of that GTE are actually expressing what we want them to express (including the counting variable which enforces the constraint we are encoding). A cutting planes derivation of *reification constraints to GTE* was given in [6],[2] and in this paper we will give a derivation of the converse direction.

## 3 A counterexample

The use of the clauses of kind $C^\eta_3$ in the generalized totalizer encoding is not standard. The traditional way to define the GTE, which is based on [4], introduces only clauses of kind $C^\eta_1$ and $C^\eta_2$.[3] We argue $C^\eta_3$ clauses are necessary to achieve one-to-one correspondence of the GTE and the reification constraints, as shown by the following simple counterexample: consider the constraint $2x_1 + 3x_2 \geq 3$, and let $T$ be its GTT. The only internal node $\eta$ of $T$ introduces the variables $y^\eta_2$, $y^\eta_3$ and $y^\eta_5$, and the traditional GTE of $T$ (without $C^\eta_3$ clauses) consists of the following clauses:

$$\begin{array}{lllllllll} C^\eta_1(2,0,2) & = & \bar{x}_1 \vee y^\eta_2 & C^\eta_1(0,3,3) & = & \bar{x}_2 \vee y^\eta_3 & C^\eta_1(2,3,5) & = & \bar{x}_1 \vee \bar{x}_2 \vee y^\eta_5 \\ C^\eta_2(0,0,0) & = & x_1 \vee x_2 \vee \bar{y}^\eta_2 & C^\eta_2(2,0,2) & = & x_2 \vee \bar{y}^\eta_3 & C^\eta_2(0,3,3) & = & x_1 \vee \bar{y}^\eta_5. \end{array}$$

Two assignments that satisfy every clause while failing to satisfy the reification constraints of all the counting variables are $\{x_1 = 1,\ x_2 = 0,\ y^\eta_2 = 1,\ y^\eta_3 = 0,\ y^\eta_5 = 1\}$, which fails to satisfy $C^{\rightarrow}_{\text{reif}}(y^\eta_5)$, and $\{x_1 = 0,\ x_2 = 1,\ y^\eta_2 = 0,\ y^\eta_3 = 1,\ y^\eta_5 = 0\}$, which fails to satisfy $C^{\leftarrow}_{\text{reif}}(y^\eta_2)$.

A convenient way to fix this is to introduce the clauses of kind $C^\eta_3$. Those new clauses are easily derivable from the reification constraints (we get $C^\eta_3(j)$ by adding $C^{\rightarrow}_{\text{reif}}(y^\eta_{s_j})$ to $C^{\leftarrow}_{\text{reif}}(y^\eta_{s_{j-1}})$, weakening and saturation), so the proof in [6], which uses the traditional version of the GTE, is easily generalizable to this new definition.

[2] Actually, the proof there is restricted to cardinality constraints, but it can be immediately extended to the general case.

[3] Actually, the definition in [4] works with a different normalized form and only introduces clauses of kind $C^\eta_1$, but it can be easily extended with clauses of kind $C^\eta_2$ following [1].

## 4 Proof logging

Now, we want to show a cutting planes derivation of *GTE to reification constraints*. To ease the reading, let us introduce some notation: given an internal node $\eta$,

$$\begin{array}{llllll} L^\eta = \sum_{l\in\text{leaves}(\eta_1)} \text{weight}(l)\cdot l & R^\eta = \sum_{l\in\text{leaves}(\eta_2)} \text{weight}(l)\cdot l & |L^\eta| = \text{weight}(\eta_1) \\ \bar{L}^\eta = \sum_{l\in\text{leaves}(\eta_1)} \text{weight}(l)\cdot \bar{l} & \bar{R}^\eta = \sum_{l\in\text{leaves}(\eta_2)} \text{weight}(l)\cdot \bar{l} & |R^\eta| = \text{weight}(\eta_2), \end{array}$$

where we drop the superscript when $\eta$ is clear by context.

To show *GTE to reification constraints*, we will prove two weaker propositions. The first proposition will entail GTE to $C^{\leftarrow}_{\text{reif}}$, and the second one, proven in a similar way, will entail GTE to $C^{\rightarrow}_{\text{reif}}$. But there is a complication: due to there potentially being 'gaps' between the indexes of the counting variables, for the first proposition we will need a strengthened version of $C^{\leftarrow}_{\text{reif}}$, which we will call $C^{\leftarrow}_{\text{streif}}$, meaning

$$y^\eta_{s_j} \leftarrow \sum_{l\in\text{leaves}(\eta)} \text{weight}(l)\cdot l \geq s_{j-1}+1. \qquad \text{Formally,}$$

▶ **Definition 12** (Strengthened reification constraint)**.**

$$C^{\leftarrow}_{\text{streif}}(y^\eta_{s_j}) = (\text{weight}(\eta)-s_{j-1})\cdot y^\eta_{s_j} + \sum_{l\in\text{leaves}(\eta)} \text{weight}(l)\cdot \bar{l} \geq \text{weight}(\eta) - s_{j-1}.$$

*For the case $j=0$, we convene $s_{j-1}=-1$.*

Let us begin with the first proposition, before which we will prove two auxiliary lemmas:

▶ **Lemma 13.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$, let $1\leq j\leq|\text{sumweights}(\eta)|$, let $b\in\{1,2\}$ and let $v=1+(b\%2)$. Given:*

1. $C^{\leftarrow}_{\text{streif}}(y^{\eta_1}_{s_n})$ *for* $0\leq n\leq|\text{sumweights}(\eta_1)|+1$,
2. $C^{\leftarrow}_{\text{streif}}(y^{\eta_2}_{s_n})$ *for* $0\leq n\leq|\text{sumweights}(\eta_2)|+1$,
3. *the negated reification constraint* $\neg C^{\rightarrow}_{\text{streif}}(y^\eta_{s_j})$,
4. $\bar{y}^{\eta_b}_{s_i}\geq 1$ *for* $0<i\leq|\text{sumweights}(\eta_b)|$ *such that there exists* $0\leq k\leq|\text{sumweights}(\eta_v)|$ *so that* $s^{\eta_b}_{i-1}+s^{\eta_v}_k\leq s^\eta_{j-1}$,

*there is a cutting planes derivation of $y^{\eta_v}_{s_{k+1}}\geq 1$, for $k$ the largest index such that $s^{\eta_b}_{i-1}+s^{\eta_v}_k\leq s^\eta_{j-1}$.*

**Proof.** Let $b=1$ without loss of generality. Given $\neg C^{\leftarrow}_{\text{streif}}(y^\eta_{s_j})$, which normalizes to

$$(|L|+|R|-s^\eta_{j-1})\cdot\bar{y}^\eta_{s_j}+L+R\geq|L|+|R|+1,$$

we can derive

$$L+R\geq s^\eta_{j-1}+1 \tag{1}$$

by weakening. Our assumption $\bar{y}^{\eta_1}_{s_i}$ contradicts the left hand side of $C^{\leftarrow}_{\text{streif}}(y^{\eta_1}_{s_i})$, so we add $(\bar{y}^{\eta_1}_{s_i}\geq 1)\cdot(|L|-s^{\eta_1}_{i-1})$ to $C^{\leftarrow}_{\text{streif}}(y^{\eta_1}_{s_i})$ to get

$$\bar{L}\geq|L|-s^{\eta_1}_{i-1}, \tag{2}$$

meaning $L\leq s^{\eta_1}_{i-1}$. Together with our assumption $L+R>s^\eta_{j-1}$ this implies $R>s^\eta_{j-1}-s^{\eta_1}_{i-1}\geq s^{\eta_2}_k$, so we add (2) to (1) and weaken to get

$$R\geq s^{\eta_2}_k+1, \tag{3}$$

which is the right hand side of $C^{\leftarrow}_{\text{streif}}(y^{\eta_2}_{s_{k+1}})$, to which we add (3) and saturate to get $y^{\eta_2}_{s_{k+1}}\geq 1$. ◀

▶ **Lemma 14.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$, let $1 \leq j \leq |\text{sumweights}(\eta)|$, let $b \in \{1, 2\}$ and let $v = 1 + (b\%2)$. Given:*

1. *the GTE of node $\eta$,*
2. *the negated reification constraint $\neg C^{\rightarrow}_{\text{streif}}(y^{\eta}_{s_j})$,*
3. *$y^{\eta_b}_{s_k} \geq 1$ for $0 < k \leq |\text{sumweights}(\eta_v)|$ so that there exists $0 \leq i \leq |\text{sumweights}(\eta_b)|$ for which $k-1$ is the largest index such that $s^{\eta_b}_i + s^{\eta_v}_{k-1} \leq s^{\eta}_{j-1}$,*

*there is a cutting planes derivation of $\bar{y}^{\eta_v}_{s_i} \geq 1$, for $i$ the least index for which $k-1$ is the largest index such that $s^{\eta_b}_i + s^{\eta_v}_{k-1} \leq s^{\eta}_{j-1}$.*

**Proof.** Let $b = 2$ without loss of generality. Given $\neg C^{\leftarrow}_{\text{streif}}(y^{\eta}_{s_j})$, which normalizes to

$$(|L| + |R| - s^{\eta}_j + 1) \cdot \bar{y}^{\eta}_{s_j} + L + R \geq |L| + |R| + 1,$$

we can derive

$$\bar{y}^{\eta}_{s_j} \geq 1 \tag{4}$$

by weakening and saturation. We have $s^{\eta}_j \leq s^{\eta_1}_i + s^{\eta_2}_k$ (otherwise, $s^{\eta}_j > s^{\eta_1}_i + s^{\eta_2}_k > s^{\eta}_{j-1}$ by maximality of $k-1$, and there is an element of sumweights($\eta$) between $s^{\eta}_{j-1}$ and $s^{\eta}_j$), so let $j \leq l \leq |\text{sumweights}(\eta)|$ be such that $s^{\eta_1}_i + s^{\eta_2}_k = s^{\eta}_l$. Then, we have $C^{\eta}_1(i, k, l)$, which we add to our assumption $y^{\eta_2}_{s_k} \geq 1$ to get

$$\bar{y}^{\eta_1}_{s_i} + y^{\eta}_{s_l} \geq 1. \tag{5}$$

If $l \neq j$, we have a sequence of constraints $\bar{y}^{\eta}_{s_l} + y^{\eta}_{s_{l-1}} \geq 1 \ldots \bar{y}^{\eta}_{s_{j+1}} + y^{\eta}_{s_j} \geq 1$, which we add to (5) to get

$$\bar{y}^{\eta_1}_{s_i} + y^{\eta}_{s_j} \geq 1. \tag{6}$$

And the second disjunct contradicts our assumption $\bar{y}^{\eta}_{s_j}$, so we just have to add (4) to (6) to get $\bar{y}^{\eta_1}_{s_i} \geq 1$. ◀

▶ **Proposition 15.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$ and let $1 \leq j \leq |\text{sumweights}(\eta)|$. Given:*

1. *the GTE of node $\eta$,*
2. *$C^{\leftarrow}_{\text{streif}}(y^{\eta_1}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_1)| + 1$,*
3. *$C^{\leftarrow}_{\text{streif}}(y^{\eta_2}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_2)| + 1$,*

*we can prove $C^{\leftarrow}_{\text{streif}}(y^{\eta}_{s_j})$ by contradiction using cutting planes.*

**Proof.** We assume $\neg C^{\leftarrow}_{\text{streif}}(y^{\eta}_{s_j})$ towards contradiction. As shown in the proof of Lemma 14, from that assumption we can derive

$$\bar{y}^{\eta}_{s_j} \geq 1. \tag{7}$$

Let $0 \leq a \leq |\text{sumweights}(\eta_1)|$ and $0 \leq b \leq |\text{sumweights}(\eta_2)|$ be such that $s^{\eta_1}_a + s^{\eta_2}_b = s^{\eta}_j$. We add $C^{\eta}_1(a, b, j)$ and (7), getting

$$\bar{y}^{\eta_1}_{s_a} + \bar{y}^{\eta_2}_{s_b} \geq 1. \tag{8}$$

We will show both disjuncts lead to contradiction, starting with the left one. If $a = 0$, we are done; otherwise, $\bar{y}^{\eta_1}_a$ fulfills the assumptions of Lemma 13, so we apply[4] Lemmas 13 and 14

[4] Technically, we apply a slightly modified version of those lemmas that ignores $\bar{y}^{\eta_2}_b$ throughout the proof.

iteratively to (8). If $s^{\eta}_{j} > s^{\eta_2}_{|\text{sumweights}(\eta_2)|}$, we get

$$y^{\eta_2}_{|\text{sumweights}(\eta_2)|+1} + \bar{y}^{\eta_2}_{s_b} \geq 1. \tag{9}$$

after at most $|\text{sumweights}(\eta_2)| - b$ iterations and an application of Lemma 13. Otherwise, let $b \leq c \leq |\text{sumweights}(\eta_2)|$ be the least index such that $s^{\eta_2}_{c} \geq s^{\eta}_{j-1}$: we get

$$\bar{y}^{\eta_1}_{0} + \bar{y}^{\eta_2}_{s_b} \geq 1. \tag{10}$$

after at most $c - b + 1$ iterations. The case for $\bar{y}^{\eta_2}_{s_b}$ is similar. ◀

The reasoning for the second proposition is symmetrical, albeit without the need of strengthened reification; we relegate it to the appendix due to space constraints.

▶ **Proposition 16.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$ and let $1 \leq j \leq |\text{sumweights}(\eta)|$. Given:*

1. *the GTE of node $\eta$,*
2. *$C^{\rightarrow}_{\text{reif}}(y^{\eta_1}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_1)| + 1$,*
3. *$C^{\rightarrow}_{\text{reif}}(y^{\eta_2}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_2)| + 1$,*

*we can prove $C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$ by contradiction using cutting planes.*

Finally, we have everything we need to prove *GTE to reification constraints*. To close the article, let us show how that would be done by presenting a blueprint for an implementation of the proof logging of one-to-one correspondence of the GTE in VeriPB-style: given a constraint $C$,

1. Derive both reification constraints $C^{\rightarrow}_{\text{reif}}(y)$ and $C^{\leftarrow}_{\text{reif}}(y)$ for every counting variable $y$ of $C$ by redundance-based strengthening.[5]
2. Derive the GTE of $C$ from the reification constraints by using the algorithm in [6] (suitably extended to our definition of the GTE). This certifies *reification constraints to GTE*.
3. Derive $C^{\leftarrow}_{\text{streif}}(y)$ for every counting variable $y$ of $C$ by using Proposition 15 in a bottom-to-top fashion (starting with the leaf nodes and then choosing an unvisited node of at least the same level in every iteration); we can do that because $C^{\leftarrow}_{\text{streif}}(y^{l}_{1})$ is trivially satisfied by every leaf node $l$, so that the hypotheses of Proposition 15 are satisfied at every step.
4. Delete $C$.[6] Let $j$ be the coefficient of $C$, and let $k$ be the least index such that $s^{\eta_{\text{rt}}}_{k} \geq j$: then, $C$ can be derived by adding $C^{\rightarrow}_{\text{reif}}(y^{\eta_{\text{rt}}}_{s_k})$ and $y^{\eta_{\text{rt}}}_{s_k} \geq 1 \cdot s^{\eta_{\text{rt}}}_{k}$ and rewriting.
5. Delete $C^{\leftarrow}_{\text{reif}}(y)$ for every counting variable $y$ of $C$ (we can do that because $C^{\leftarrow}_{\text{reif}}(y)$ trivially follows from $C^{\leftarrow}_{\text{streif}}(y)$).
6. Delete $C^{\rightarrow}_{\text{reif}}(y)$ and $C^{\leftarrow}_{\text{streif}}(y)$ for every counting variable $y$ in a top-to-bottom fashion (starting with the root node and then choosing an unvisited node of at most the same level in every iteration) by using Propositions 15 and 16.[7] This and the last step certify *GTE to reification constraints*.

[5] VeriPB allows to derive reification constraints 'for free' by using a rule called *redundance-based strengthening*.

[6] VeriPB allows to delete a constraint $D$ if there is a cutting-planes derivation of $D$ from the rest of the given constraints. This derivation can also be a proof by contradiction.

[7] The reification constraints of $y^{\eta}_{s_j}$ for the cases where $\eta$ is a leaf node or $j \in \{0, |\text{sumweights}(\eta)| + 1\}$ are trivially derivable.

## References


1 Olivier Bailleux and Yacine Boufkhad. Efficient CNF encoding of Boolean cardinality constraints. In *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, pages 108–122, 2003. doi:10.1007/978-3-540-45193-8_8.

2 Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified dominance and symmetry breaking for combinatorial optimisation. *J. Artif. Intell. Res.*, 77:1539–1589, 2023. doi:10.1613/jair.1.14296.

3 William J. Cook, Collette R. Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discret. Appl. Math.*, 18(1):25–38, 1987. doi:10.1016/0166-218X(87)90039-4.

4 Saurabh Joshi, Ruben Martins, and Vasco M. Manquinho. Generalized totalizer encoding for pseudo-Boolean constraints. In *Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings*, pages 200–209, 2015. doi:10.1007/978-3-319-23219-5_15.

5 Olivier Roussel and Vasco Manquinho. Pseudo-Boolean and cardinality constraints. In *Handbook of satisfiability*, pages 1087–1129. IOS Press, 2021.

6 Dieter Vandesande. Towards certified MaxSAT solving: Certified MaxSAT solving with SAT oracles and encodings of pseudo-Boolean constraints. Master's thesis, Vrije Universiteit Brussel (VUB), 2023. URL: https://researchportal.vub.be/nl/studentTheses/towards-certified-maxsat-solving.

7 Dieter Vandesande, Wolf De Wulf, and Bart Bogaerts. QMaxSATpb: A certified MaxSAT solver. In *Logic Programming and Nonmonotonic Reasoning - 16th International Conference, LPNMR 2022, Genova, Italy, September 5-9, 2022, Proceedings*, pages 429–442, 2022. doi:10.1007/978-3-031-15707-3_33.


## A Omitted proofs

As before, we first prove two auxiliary lemmas.

▶ **Lemma 17.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$, let $1 \leq j \leq |\text{sumweights}(\eta)|$, let $b \in \{1, 2\}$ and let $v = 1 + (b\%2)$. Given:*

1. $C^{\rightarrow}_{\text{reif}}(y^{\eta_1}_{s_n})$ *for* $0 \leq n \leq |\text{sumweights}(\eta_1)| + 1$,
2. $C^{\rightarrow}_{\text{reif}}(y^{\eta_2}_{s_n})$ *for* $0 \leq n \leq |\text{sumweights}(\eta_2)| + 1$,
3. *the negated reification constraint* $\neg C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$,
4. $y^{\eta_b}_{s_i} \geq 1$ *for* $0 \leq i \leq |\text{sumweights}(\eta_b)|$ *such that there exists* $0 \leq k \leq |\text{sumweights}(\eta_v)|$ *so that* $s^{\eta_b}_i + s^{\eta_v}_k \geq s^{\eta}_j$.

*there is a cutting planes derivation of* $\bar{y}^{\eta_v}_{s_k} \geq 1$, *for* $k$ *the least index such that* $s^{\eta_b}_i + s^{\eta_v}_k \geq s^{\eta}_j$.

**Proof.** Let $b = 1$ without loss of generality. Given $\neg C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$, which normalizes to

$$s^{\eta}_j \cdot y^{\eta}_{s_j} + \bar{L} + \bar{R} \geq |L| + |R| + 1,$$

we can derive

$$\bar{L} + \bar{R} \geq |L| + |R| - s^{\eta}_j + 1 \tag{11}$$

by weakening, meaning $L + R < s^{\eta}_j$. By $C^{\rightarrow}_{\text{reif}}(y^{\eta_1}_{s_i})$, we have $y^{\eta_1}_{s_i}$ implies $L \geq s^{\eta_1}_i$, so we add $(y^{\eta_1}_{s_i} \geq 1) \cdot s^{\eta_1}_i$ and $C^{\rightarrow}_{\text{reif}}(y^{\eta_1}_{s_i})$ to get

$$L \geq s^{\eta_1}_i. \tag{12}$$

Together with our assumption $L + R < s^{\eta}_{j}$, this implies $R < s^{\eta}_{j} - s^{\eta_1}_{i} \leq s^{\eta_2}_{k}$, which is what we get when we add (11) and (12) and weaken:

$$\bar{R} \geq |R| - s^{\eta_2}_{k} + 1. \tag{13}$$

But this contradicts the right hand side of $C^{\rightarrow}_{\text{reif}}(y^{\eta_2}_{s_k})$, so we add that reification constraint to (13) and saturate, getting

$$\bar{y}^{\eta_2}_{s_k} \geq 1. \tag{14}$$

◀

▶ **Lemma 18.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$, let $1 \leq j \leq |\text{sumweights}(\eta)|$, let $b \in \{1, 2\}$ and let $v = 1 + (b\%2)$. Given:*

1. *the GTE of node $\eta$,*
2. *the negated reification constraint $\neg C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$,*
3. *$\bar{y}^{\eta_b}_{s_k} \geq 1$ for $0 < k \leq |\text{sumweights}(\eta_v)|$ such that there exists $0 \leq i \leq |\text{sumweights}(\eta_b)|$ for which $k$ is the least index such that $s^{\eta_b}_{i} + s^{\eta_v}_{k} \geq s^{\eta}_{j}$,*

*there is a cutting planes derivation of $y^{\eta_v}_{s_{i+1}} \geq 1$, for $i$ the largest index for which $k$ is the least index such that $s^{\eta_b}_{i} + s^{\eta_v}_{k} \geq s^{\eta}_{j}$.*

**Proof.** Let $b = 2$ without loss of generality. Given $\neg C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$, which normalizes to

$$s^{\eta}_{j} \cdot y^{\eta}_{s_j} + \bar{L} + \bar{R} \geq |L| + |R| + 1,$$

we can derive

$$y^{\eta}_{s_j} \geq 1 \tag{15}$$

by weakening and saturation. We have $s^{\eta}_{j-1} \geq s^{\eta_1}_{i} + s^{\eta_2}_{k-1}$ (otherwise, $s^{\eta}_{j-1} < s^{\eta_1}_{i} + s^{\eta_2}_{k-1} < s^{\eta}_{j}$ by minimality of $k$, and there is an element of sumweights($\eta$) between $s^{\eta}_{j-1}$ and $s^{\eta}_{j}$), so let $0 \leq l \leq j - 1$ be such that $s^{\eta_1}_{i} + s^{\eta_2}_{k-1} = s^{\eta}_{l}$. Then, we have $C^{\eta}_{2}(i, k-1, l)$, which we add to our assumption $\bar{y}^{\eta_2}_{s_k} \geq 1$ to get

$$y^{\eta_1}_{s_{i+1}} + \bar{y}^{\eta}_{s_{l+1}} \geq 1. \tag{16}$$

If $l + 1 \neq j$, we have a sequence of constraints $\bar{y}^{\eta}_{s_j} + y^{\eta}_{s_{j-1}} \geq 1 \ldots \bar{y}^{\eta}_{s_{l+2}} + y^{\eta}_{s_{l+1}} \geq 1$, which we add to (16) to get

$$y^{\eta_1}_{s_{i+1}} + \bar{y}^{\eta}_{s_j} \geq 1. \tag{17}$$

And the second disjunct contradicts our assumption $y^{\eta}_{s_j}$, so we just have to add (15) to (17) to get

$$y^{\eta_1}_{s_{i+1}} \geq 1. \tag{18}$$

◀

▶ **Proposition 16.** *Let $T$ be a GTT of a given constraint, let $\eta$ be an internal node of $T$ and let $1 \leq j \leq |\text{sumweights}(\eta)|$. Given:*

1. *the GTE of node $\eta$,*
2. *$C^{\rightarrow}_{\text{reif}}(y^{\eta_1}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_1)| + 1$,*
3. *$C^{\rightarrow}_{\text{reif}}(y^{\eta_2}_{s_i})$ for $0 \leq i \leq |\text{sumweights}(\eta_2)| + 1$,*

*we can prove* $C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$ *by contradiction using cutting planes.*

**Proof.** We assume $\neg C^{\rightarrow}_{\text{reif}}(y^{\eta}_{s_j})$ towards contradiction. As shown in the proof of Lemma 18, from that assumption we can derive

$$y^{\eta}_{s_j} \geq 1. \tag{19}$$

Let $0 \leq a \leq |\text{sumweights}(\eta_1)|$ and $0 \leq b \leq |\text{sumweights}(\eta_2)|$ be such that $s^{\eta_1}_a + s^{\eta_2}_b = s^{\eta}_{j-1}$. We add $C^{\eta}_2(a, b, j-1)$ and (19), getting

$$y^{\eta_1}_{s_{a+1}} + y^{\eta_2}_{s_{b+1}} \geq 1. \tag{20}$$

We will show both disjuncts lead to contradiction, starting with the left one. If $a = |\text{sumweights}(\eta_1)|$, we are done; otherwise, $y^{\eta_1}_{a+1}$ fulfills the assumptions of Lemma 17, so we apply[8] Lemmas 17 and 18 iteratively to (20). If $s^{\eta}_j > s^{\eta_1}_{|\text{sumweights}(\eta_1)|}$, we get

$$y^{\eta_1}_{|\text{sumweights}(\eta_1)|+1} + y^{\eta_2}_{s_{b+1}} \geq 1. \tag{21}$$

after at most $|\text{sumweights}(\eta_1)| - a$ iterations. Otherwise, let $a+1 \leq c \leq |\text{sumweights}(\eta_1)|$ be the least index such that $s^{\eta_1}_c \geq s^{\eta}_j$: we get

$$\bar{y}^{\eta_2}_0 + y^{\eta_2}_{s_{b+1}} \geq 1. \tag{22}$$

after at most $c - (a+1)$ iterations and an application of Lemma 17. The case for $y^{\eta_2}_{s_{b+1}}$ is similar. ◀

[8] As before, technically, we apply a slightly modified version of those lemmas that ignores $y^{\eta_2}_{b+1}$ throughout the proof.