Reducing signed propositional formulas*

G. Aguilera, I.P. de Guzmán, M. Ojeda, A.Valverde

Dept. Matemática Aplicada. Universidad de Málaga.
Aptdo. de Correos 4114. E-29080 Málaga, Spain
Email: guzman@ctima.uma.es

Abstract

New strategies of reduction for finite valued propositional logics are introduced in the framework of the TAS[1] methodology developed by the authors [1]. A new data structure, the $\widehat{\Delta}$-sets, is introduced to store information about the formula being analysed, and its usefulness is shown by developing efficient strategies to decrease the size of signed propositional formulas, viz. new criteria to detect the validity or unsatisfiability of subformulas, and a strong generalisation of the pure literal rule.

1 Introduction

Any (finite) multiple-valued logic can be expressed as a classical formula over signed literals. For a wide range of finite-valued propositional logics, called regular logics [6, 7], there are uniform notation style tableau systems; all these logics can be properly treated by TAS methods and, furthermore, it is possible to weaken the structure requirements on the logic by considering ortho-regular logics [4], a superset of regular logics on which our methods preserve their good behaviour. The results introduced in this paper use the representation of finite valued logic formulas as signed formulas.

In this paper we focus on the development of a set of reduction strategies which, through the efficient determination and manipulation of lists of unitary implicant and implicates, *investigates exhaustively* the possibility of decreasing the size of the formula being analysed. The interest of such a set of reduction techniques is that the performance of a given prover for signed logic can be improved because the size of a formula can be decreased, at a polynomial cost, as much as possible *before* branching.

Lists of unitary models, so-called Δ-lists, are associated to each node in the syntactic tree of the formula and used to study whether the structure of the syntactic tree has or has not

*This work has been partially supported by CICYT project TIC97-0579-C02-02 and by COST-15 action.

[1]TAS stands for *Transformaciones de Árboles Sintácticos*, the Spanish translation of Syntactic Trees Transformations.

direct information about the validity of the formula. This way, either the method ends giving this information or, otherwise, it decreases the size of the problem before applying the next transformation. So, it is possible to decrease the number of branchings or, even, to avoid them all.

The ideas in this paper generalise, in a self-contained way, previous work of the authors [2] by explicitly extending the reduction strategy to any finite valued logic and, what is more important, by complementing the information in the Δ-lists by means of the so-called $\widehat{\Delta}$-sets. The former allow derivation of an equivalent and smaller formula; the latter also allow derivation of a smaller formula, not equivalent to the previous one, but equisatisfiable.

A variety of new reduction results are shown: these reductions are useful in order to avoid as many branchings as possible when the satisfiability of a signed propositional formula is tested. One of the new reduction techniques we introduce is a generalisation of the pure literal rule: In fact, when restricted to classical logic this new pure literal rule generalises several versions appearing in the literature.

The paper is organised as follows:

- Firstly, preliminary concepts, notation and basic definitions are introduced: specifically, it is worth to note the definition of literal and the way some of them will be denoted.
- Secondly, Δ-lists, the basic tool of the TAS methodology, are introduced. This definition extends that given in [2], and eases the reduction process by integrating some reduction into the calculation of the Δ-lists. The required theorems to show how to use the information collected in those lists are proved.
- Later, the $\widehat{\Delta}$-sets are defined and results that use the information in these sets are stated. One of these is a generalisation of the pure literal rule.
- Finally, some examples are included.

2 Preliminary Concepts and Definitions

Definition 1 *Let $\mathcal{V}$ be a numerable set of propositional variables and* $\boldsymbol{n} = \left\{0, \frac{1}{n-1}, \frac{2}{n-1}, \dots, \frac{n-2}{n-1}, 1\right\}$. *Consider the set* LIT *defined as*

$$\textsc{lit} = \{S{:}p \mid S \subseteq \boldsymbol{n}, p \in \mathcal{V}\}$$

whose elements are named ***literals*** *or* ***atoms****, and the free algebra* $(\textsc{form}, \vee, \wedge)$*, generated by* LIT *with binary operations* $\vee$ *and* $\wedge$*. The* ***n-valued signed logic for*** $\mathcal{V}$*, denoted by* $\mathcal{S}_n(\mathcal{V})$*, is determined by the language* $(\textsc{form}, \vee, \wedge)$ *and the semantics defined as*

- *The truth values are* $\textsc{bool} = \{0, 1\}$.
- *The only designated truth value is* 1
- *A* ***signed assignment*** *is any application* $I\colon \mathcal{S}_n(\mathcal{V}) \to \{0, 1\}$ *verifying:*

1. *For each* $p \in \mathcal{V}$ *there is one, and only one,* $j \in \boldsymbol{n}$ *such as* $I(\{j\}{:}p) = 1$.
2. $I(S{:}p) = 1$ *if there exists* $j \in S$ *such as* $I(\{j\}{:}p) = 1$*; otherwise,* $I(S{:}p) = 0$.
3. $I(A \wedge B) = \min\{I(A), I(B)\}$.
4. $I(A \vee B) = \max\{I(A), I(B)\}$.

Throughout the rest of the paper we assume that the set $\mathcal{V}$ is completely ordered (in this work we will use the alpha-numeric ordering); ℓ_p denotes a literal with variable p, i.e. $\ell_p = S : p$ for some $S \subseteq \boldsymbol{n}$. The set of literals with propositional variable p will be denoted $\textsc{lit}_p$. We will also use the usual notions of clause (disjunction of literals) and cube (conjunction of literals).

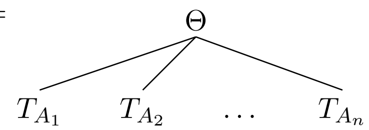
`Sub`(A) denotes the set of subformulas of A; $B \sqsubseteq A$ denotes that B is a subformula of A and $B \sqsubset A$ denotes that B is a proper subformula of A.

A formula A is said to be *satisfiable* if there exists a signed assignment I such that $I(A) = 1$, in this case I is said to be a *model* for A; formulas A and B are said to be *equisatisfiable* if A is satisfiable iff B is satisfiable; $\equiv$ denotes the semantic equality; $\models$ denotes the logical consequence. For each p and for all signed assignment I we have $I(\varnothing{:}p) = 0$ and $I(\boldsymbol{n}{:}p) = 1$; we introduce the symbols $\perp$ and $\top$ to represent these formulas, i.e. $\perp = \varnothing{:}p$ and $\top = \boldsymbol{n}{:}p$.

By using the associative laws we will consider expressions like $A_1 \vee \cdots \vee A_n$ or $A_1 \wedge \cdots \wedge A_n$ as formulas. Given a signed formula A, the *syntactic tree* of A, denoted by T_A, is defined as follows:

1. If A is a literal or a logical constant (either $\top$ or $\perp$), then T_A is A.
2. If $A = \Theta_{i=1}^{n} A_i$, where Θ is either $\bigwedge$ or $\bigvee$, then

$T_A =$



We will use the standard notion of tree and address of a node in a tree (see for instance [5]). Given a formula A, if $B \sqsubseteq A$, then η_B denotes the address of the node in T_A corresponding to B; specifically, $\eta_A = \varepsilon$. Conversely, given a formula A and an address η in T_A, the expression `Node`(η) means the subformula of A corresponding to η in T_A.

In the rest of the paper, A will denote either a formula or its syntactic tree, being the context which clarifies the meaning.

- If A, B and C are signed formulas, and $B \sqsubseteq A$, then $A[B/C]$ denotes the result of substituting in A any occurrence of B by C.
- If η is an address in A and C is another signed formula, then the expression $A[\eta/C]$ is the formula obtained after substituting in A the subtree rooted at address η by C.
- Finally, we will consider expressions like $A[*_1/*'_1, \dots, *_m/*'_m]$ where $*_i/*'_i$ is any of the substitutions introduced above.

We will also use lists with its standard notation; elements in a list are written in juxtaposition; `nil` denotes the empty list; if α and β are lists, $\ell \in \alpha$ denotes that ℓ is an element of α; $\alpha \subseteq \beta$ means that all elements of α are elements of β;

2.1 The Boole Algebras of literals

For each $p \in \mathcal{V}$ we define an ordering in $\textsc{lit}_p$, denoted $\leq$, as follows:

$$S_1{:}p \leq S_2{:}p \qquad \text{iff} \qquad S_1{:}p \models S_2{:}p \qquad \text{or, equivalently, } S_1 \subseteq S_2$$

Obviously, this ordering makes $\textsc{lit}_p$ a Boole algebra isomorphic to $(\wp(\boldsymbol{n}), {}^c, \cup, \cap)$ for all p. The operations in these algebras are denoted $^-$, $\vee$ and $\wedge$; specifically:

$$\overline{S{:}p} = S^c{:}p, \qquad S_1{:}p \vee S_2{:}p \equiv (S_1 \cup S_2){:}p, \qquad S_1{:}p \wedge S_2{:}p \equiv (S_1 \cap S_2){:}p$$

If ℓ is a literal, $\overline{\ell}$ is called the *opposite* literal to ℓ; if $\alpha = \ell_1\ell_2\dots\ell_n$ is a list of literals, then $\overline{\alpha} = \overline{\ell_1}\,\overline{\ell_2}\dots\overline{\ell_n}$.

Throughout this work, literals $\{j\}{:}p$ and their negated will play an important role;[2] so we will adopt the following simpler notation:

$$pj = \{j\}{:}p$$

By using these literals, we have a disjunctive and a conjunctive expression for each literal $S{:}p$, namely

$$S{:}p = \bigwedge_{j \in S^c} \overline{pj}, \qquad S{:}p = \bigvee_{j \in S} pj$$

Thus we introduce, for each $p \in \mathcal{V}$ the following subsets of $\textsc{lit}_p$:

$$\textsc{lit}_p^+ = \{pj \mid j \in \boldsymbol{n}\} \cup \{\top\} \qquad \textsc{lit}_p^- = \{\overline{pj} \mid j \in \boldsymbol{n}\} \cup \{\bot\}$$

and also

$$\textsc{lit}^+ = \bigcup_{p \in \mathcal{V}} \textsc{lit}_p^+ \qquad \textsc{lit}^- = \bigcup_{p \in \mathcal{V}} \textsc{lit}_p^-$$

Finally, we will also use the following sets of lists:

- $\textsc{Llit}^-$ is the set of lists of elements in $\textsc{lit}^-$, i.e. lists of negative literals.
- $\textsc{Llit}^+$ is the set of lists of elements in $\textsc{lit}^+$, i.e. lists of positive literals.
- $\textsc{Llit}_\bot^-$ is the set $\textsc{Llit}^-$ where those lists containing either $\bot$ or a set $\textsc{lit}_p^-$ for some propositional variable p are identified with $\bot$.
- $\textsc{Llit}_\top^+$ is the set $\textsc{Llit}^+$ where those lists containing either $\top$ or a set $\textsc{lit}_p^+$ for some propositional variable p are identified with $\top$. [3]

[2] In fact, single truth values can be used as signs for our purposes: Since our formulas may be arbitrarily nested, a sign can always be replaced by a disjunction of singleton signed literals and vice-versa. Anyway, the development does not get substantially improved with this simplification and complicates the presentation of the examples.

[3] Formally, the sets $\textsc{Llit}_\bot^-$ and $\textsc{Llit}_\top^+$ are quotient sets of $\textsc{Llit}^-$ and $\textsc{Llit}^+$ under the equivalence relations $\Re^-$ and $\Re^+$ defined as: if $\alpha = \beta \in \textsc{Llit}^-$ then $\alpha\Re^-\beta$; if $\bot \in \alpha$, then $\bot\Re^-\alpha$; if $\textsc{lit}_p^- \subseteq \alpha$ for some p, then $\bot\Re^-\alpha$, and if $\alpha = \beta \in \textsc{Llit}^+$ then $\alpha\Re^+\beta$; if $\top \in \alpha$, then $\top\Re^+\alpha$; if $\textsc{lit}_p^+ \subseteq \alpha$ for some p, then $\top\Re^+\alpha$.

3 Adding Information to the Tree: $\boldsymbol{\Delta}$-lists and $\widehat{\boldsymbol{\Delta}}$-sets

As in every TAS method, the underlying idea is the use of information given by partial assignments; in the classical case these were unitary assignments and in the multiple-valued case will be assignments like $I(S:p) = 1$ or $I(S:p) = 0$.

We associate to each signed formula A:

- two lists denoted $\Delta_0(A)$ and $\Delta_1(A)$, in $\mathrm{LLIT}_\perp^-$ and $\mathrm{LLIT}_\top^+$ respectively, these lists are called the associated Δ-lists of A;
- two sets of lists, denoted $\widehat{\Delta_0}(A)$ and $\widehat{\Delta_1}(A)$, whose elements are obtained out of the associated Δ-lists of the subformulas of A.

The Δ-lists and the $\widehat{\Delta}$-sets are the key tools of our method to reduce the size of the formula being analysed for satisfiability.

3.1 The $\boldsymbol{\Delta}$-lists

In a nutshell, $\Delta_0(A)$ and $\Delta_1(A)$ are either a logical constant ($\perp$ or $\top$, respectively) or lists of unitary implicates/implicants of A, respectively. The purpose of these lists is two-fold: firstly, to transform the formula A into an equivalent and smaller-sized one (Section 3.2), and secondly, by means of the $\widehat{\Delta_b}$ sets (Section 3.3), to get an equisatisfiable and smaller-sized one. Its formal definition is the following:

Definition 2 *Given a signed formula A, the lists $\Delta_0(A) \in \mathrm{LLIT}_\perp^-$ and $\Delta_1(A) \in \mathrm{LLIT}_\perp^+$ are recursively defined as:*

$$
\begin{array}{l}
\Delta_0(\perp) = \perp, \qquad \Delta_1(\perp) = \mathtt{nil} \\
\Delta_0(\top) = \mathtt{nil}, \quad \Delta_1(\top) = \top \\
\Delta_0(S{:}p) = \overline{pj_1} \dots \overline{pj_m}, \quad \textit{if } S^c = \{j_1, \dots, j_m\}, \textit{ and } \varnothing \neq S \neq \boldsymbol{n} \\
\Delta_1(S{:}p) = pj_1 \dots pj_m, \quad \textit{if } S = \{j_1, \dots, j_m\}, \textit{ and } \varnothing \neq S \neq \boldsymbol{n} \\
\Delta_0\left(\bigwedge_{i\in I} A_i\right) = \bigcup_{i\in I} \Delta_0(A_i) \\
\Delta_0\left(\bigvee_{i\in I} A_i\right) = \bigcap_{i\in I} \Delta_0(A_i) \\
\Delta_1\left(\bigwedge_{i\in I} A_i\right) = \bigcap_{i\in I} \Delta_1(A_i) \\
\Delta_1\left(\bigvee_{i\in I} A_i\right) = \bigcup_{i\in I} \Delta_1(A_i)
\end{array}
$$

As Δ_0-list is in $\mathrm{LLIT}_\perp^-$ and a Δ_1-list is in $\mathrm{LLIT}_\top^+$, elements in Δ_0-lists (resp. Δ_1-lists) can be considered as conjunctively (resp. disjunctively) connected.

Example 1: In a four-valued signed logic, we have:

1. $\Delta_0(\{{}^1\!/\!_3,1\}{:}p) = \overline{p0p\,{}^2\!/\!_3}$
   $\Delta_1(\{{}^1\!/\!_3,1\}{:}p) = p\,{}^1\!/\!_3\,p1$

2. $\Delta_0\big(\{0,{}^2\!/\!_3\}{:}p \vee \{{}^1\!/\!_3\}{:}q \vee \{{}^2\!/\!_3,1\}{:}r\big) = \texttt{nil}$
   $\Delta_1\big(\{0,{}^2\!/\!_3\}{:}p \vee \{{}^1\!/\!_3\}{:}q \vee \{{}^2\!/\!_3,1\}{:}r\big) = p0p\,{}^2\!/\!_3q\,{}^1\!/\!_3r\,{}^2\!/\!_3r1$

3. $\Delta_0\big(\{{}^2\!/\!_3,1\}{:}p \wedge \{1\}{:}r \wedge \{0,{}^2\!/\!_3\}{:}q\big) = \overline{p0p\,{}^1\!/\!_3q\,{}^1\!/\!_3q1r0r\,{}^1\!/\!_3r\,{}^2\!/\!_3}$
   $\Delta_1\big(\{{}^2\!/\!_3,1\}{:}p \wedge \{1\}{:}r \wedge \{0,{}^2\!/\!_3\}{:}q\big) = \texttt{nil}$

4. $\Delta_0\Big(\big(\{0,{}^1\!/\!_3\}{:}p \wedge \{1\}{:}q\big) \vee \big(\{{}^1\!/\!_3,1\}{:}p \wedge \{0,1\}{:}r\big)\Big) = \overline{p\,{}^2\!/\!_3}$
   $\Delta_1\Big(\big(\{0,{}^1\!/\!_3\}{:}p \wedge \{1\}{:}q\big) \vee \big(\{{}^1\!/\!_3,1\}{:}p \wedge \{0,1\}{:}r\big)\Big) = \texttt{nil}$

5. $\Delta_0\Big(\big(\{{}^1\!/\!_3,1\}{:}p \vee \{{}^1\!/\!_3,{}^2\!/\!_3\}{:}q\big) \wedge \{0,1\}{:}p\Big) = \overline{p\,{}^1\!/\!_3p\,{}^2\!/\!_3}$
   $\Delta_1\Big(\big(\{{}^1\!/\!_3,1\}{:}p \vee \{{}^1\!/\!_3,{}^2\!/\!_3\}{:}q\big) \wedge \{0,1\}{:}p\Big) = p1$

6. $\Delta_0\Big(\{0,{}^2\!/\!_3\}{:}p \wedge \big(\{1\}{:}p \vee \big(\{{}^1\!/\!_3,{}^2\!/\!_3\}{:}q \wedge \{{}^1\!/\!_3\}{:}p\big)\big)\Big) = \overline{p0p\,{}^1\!/\!_3p\,{}^2\!/\!_3p1} = \bot$
   $\Delta_1\Big(\{0,{}^2\!/\!_3\}{:}p \wedge \big(\{1\}{:}p \vee \big(\{{}^1\!/\!_3,{}^2\!/\!_3\}{:}q \wedge \{{}^1\!/\!_3\}{:}p\big)\big)\Big) = \texttt{nil}$

□

3.2 Information in the Δ-lists

In this section we study the information contained in the Δ-lists of a given formula. Our first theorem states that literals in $\Delta_0(A)$ are unitary implicates of A, and literals in $\Delta_1(A)$ are unitary implicants of A.

Theorem 1 *Let A be a signed formula and ℓ be a literal in A then:*

1. *If $\ell \in \Delta_0(A)$, then $A \models \ell'$ for all $\ell' \geq \ell$; equivalently, $A \equiv \ell' \wedge A$ for all $\ell' \geq \ell$.*
2. *If $\Delta_0(A) = \bot$, then $A \equiv \bot$.*
3. *If $\ell \in \Delta_1(A)$, then $\ell' \models A$ for all $\ell' \leq \ell$; equivalently, $A \equiv \ell' \vee A$ for all $\ell' \leq \ell$.*
4. *If $\Delta_1(A) = \top$, then $A \equiv \top$.*

Proof:
1. For literals, the result is trivial: if $A = S{:}p$ and $S^c = \{j_1, \dots, j_m\}$, then $\Delta_0(A) = \overline{pj_1} \dots \overline{pj_m}$ and $A \equiv \bigwedge_{j_i \in S^c} \overline{pj_i}$.

Assume that for all formulas X with degree $k < n$ we have:

If $\ell \in \Delta_0(X)$, then $X \models \ell$

Let A be a formula of degree n.

- If $A = A_1 \vee A_2$ and $\ell \in \Delta_0(A)$, then $\ell \in \Delta_0(A_1)$ and $\ell \in \Delta_0(A_2)$; by the induction hypothesis we have $A_1 \models \ell$ and $A_2 \models \ell$, therefore $A \models \ell$.
- If $A = A_1 \wedge A_2$ and $\ell \in \Delta_0(A)$, then $\ell \in \Delta_0(A_i)$ for some i; consequently, $A_i \models \ell$ and, by using $A \models A_i$, we get $A \models \ell$.

2. For literals, the result is trivial: we have $\Delta_0(S{:}p) = \perp$ if and only if $S = \varnothing$, i.e. $S{:}p = \perp$. Assume that for all formulas X with degree $k < n$ we have:

If $\Delta_0(X) = \perp$, then $X \equiv \perp$

If $\Delta_0(A) = \perp$ and $A = \bigvee_{i=1}^{n} A_i$, then, by definition, $\Delta_0(A_i) = \perp$ for all i; now, by the induction hypothesis $A_i \equiv \perp$ for all i and, therefore, $A \equiv \perp$.

If $\Delta_0(A) = \perp$ and $A = \bigwedge_{i=1}^{n} A_i$, then we have to consider two cases:

- $\Delta_0(A_i) = \perp$ for some i, and thus $A_i \equiv \perp$ and $A \equiv \perp$;
- There is $p \in \mathcal{V}$ such that for all $j \in \boldsymbol{n}$, there exists i such that $\overline{pj} \in \Delta_0(A_i)$; therefore, $A \models \bigwedge_{j \in \boldsymbol{n}} \overline{pj}$ and, by $\bigwedge_{j \in \boldsymbol{n}} \overline{pj} \equiv \perp$, we have $A \equiv \perp$.

The proof for the part $\mathtt{Norm}(\Delta_1(A)) \models A$ is similar.

QED

In the statement of the following corollary we use the operator `Norm` which makes explicit the fact that Δ_0-lists (Δ_1-lists) are considered conjunctively (disjunctively) connected: If $\alpha \in \mathrm{LLIT}_{\top}^{+}$, then $\mathtt{Norm}(\alpha)$ is obtained from $\bigvee_{\ell \in \alpha} \ell$ by using the simplification $S_1{:}p \vee S_2{:}p \equiv (S_1 \cup S_2){:}p$ and the 0-1 laws; if $\alpha \in \mathrm{LLIT}_{\perp}^{-}$ then $\mathtt{Norm}(\alpha)$ is obtained from $\bigwedge_{\ell \in \alpha} \ell$ by using the simplification $S_1{:}p \wedge S_2{:}p \equiv (S_1 \cap S_2){:}p$ and the 0-1 laws.

The corollary below states some properties of the formula $\mathtt{Norm}(\Delta_b(A)$ with respect to the formula A.

Corollary 2

1. $\mathtt{Norm}(\Delta_1(A)) \models A \models \mathtt{Norm}(\Delta_0(A))$
2. *If C is a cube, then* $C \equiv \mathtt{Norm}(\Delta_0(C))$
3. *If C is a clause, then* $C \equiv \mathtt{Norm}(\Delta_1(C))$

The following result on the structure of the Δ-lists shows the possibility of determining when a given formula is equivalent to a literal. Its proof is immediate from Theorem 1.

Corollary 3

1. *For every signed formula A one and only one of the following statements hold:*
   - *There is $b \in \{0, 1\}$ such as* $\Delta_b(A) = \mathtt{nil}$
   - *There exists $p \in \mathcal{V}$ such that all elements of $\Delta_0(A)$ are elements of LIT_p^{-} and all elements of $\Delta_1(A)$ are elements of LIT_p^{+}.*
2. *If $pj_1 \in \Delta_1(A)$ and $\overline{pj_0} \in \Delta_0(A)$, then $j_1 \neq j_0$.*
3. *If* $\mathtt{Norm}(\Delta_0(A)) = \mathtt{Norm}(\Delta_1(A)) = C$*, then C is a literal and $A \equiv C$.*

The following corollary states that the non-empty Δ_1-lists might directly detect the satisfiability of a formula.

Corollary 4
If $\Delta_1(A) \neq \texttt{nil}$, then A is satisfiable, moreover if $pj \in \Delta_1(A)$, then any assignment I such that $I(pj) = 1$ is a model for A.

On the other hand, the following result states conditions on the Δ-lists assuring the validity or unsatisfiability of a formula.

Corollary 5
Let A be a signed formula, then

1. *If $A = \bigwedge_{i=1}^{n} A_i$ in which a conjunct A_{i_0} with $i_0 \in \{1, \dots, n\}$ is a clause such that $\overline{\Delta_1(A_{i_0})} \subseteq \Delta_0(A)$, then $A \equiv \perp$.*
2. *If $A = \bigvee_{i=1}^{n} A_i$ in which a disjunct A_{i_0} with $i_0 \in \{1, \dots, n\}$ is a cube such that $\overline{\Delta_0(A_{i_0})} \subseteq \Delta_1(A)$, then $A \equiv \top$.*

PROOF:

1. Consider $A = \bigwedge_{i=1}^{n} A_i$, let A_{i_0} be a clause such that $\Delta_1(A_{i_0}) = \ell_1 \dots \ell_m$ then, by Corollary 2, $A_{i_0} \equiv \ell_1 \vee \dots \vee \ell_m$; let us denote C the latter disjunction. On the other hand, as $\overline{\Delta_1(A_{i_0})} \subseteq \Delta_0(A)$, by Theorem 1, we have $A \equiv A \wedge (\overline{\ell_1} \wedge \dots \wedge \overline{\ell_m}) = A \wedge \neg C$, therefore:
$$A \equiv C \wedge A \equiv C \wedge (\neg C \wedge A) \equiv \perp$$
2. It is similar to the previous one.

QED

The following definitions name those formulas for which validity or unsatisfiability can be determined directly from their Δ-lists.

Definition 3 *Let A be a signed formula; then it is said that A is:*

- ***finalizable*** *if one of the following conditions holds:*
  1. $\Delta_1(A) \neq \texttt{nil}$.
  2. $\Delta_0(A) = \perp$.
- Δ_0***-conclusive*** *if one of the following conditions holds:*
  1. $\Delta_0(A) = \perp$.
  2. *$A = \bigwedge_{i=1}^{n} A_i$ and a conjunct A_{i_0} is a clause such that $\overline{\Delta_1(A_{i_0})} \subseteq \Delta_0(A)$.*
- Δ_1***-conclusive*** *if one of the following conditions holds:*
  1. $\Delta_1(A) = \top$.
  2. *$A = \bigvee_{i=1}^{n} A_i$ and a disjunct A_{i_0} is a cube such that $\overline{\Delta_0(A_{i_0})} \subseteq \Delta_1(A)$.*

- ***ℓ-simple*** *if A is not a literal and* $\texttt{Norm}(\Delta_0(A)) = \texttt{Norm}(\Delta_1(A)) = \ell$.

The previous results characterise the information that must be in the Δ-lists to have complete information about satisfiability; when all these definitions are applied exhaustively to simplify formula, the resulting one is said to be Δ-*restricted*; its formal definition is the following:

Definition 4 *Let A be a signed formula then it is said that A is* ***Δ-restricted*** *if it satisfies the following conditions:*

- *it is not finalizable,*
- *it has no subtree which is either Δ_0-conclusive, or Δ_1-conclusive, or ℓ-simple,*
- *it has neither $\top$ nor $\bot$ leaves.*

Definition 5 *If A is a signed formula, to* ***Δ-label*** *A means to associate to each*[4] *node N in A the ordered pair $\big(\Delta_0(N), \Delta_1(N)\big)$.*

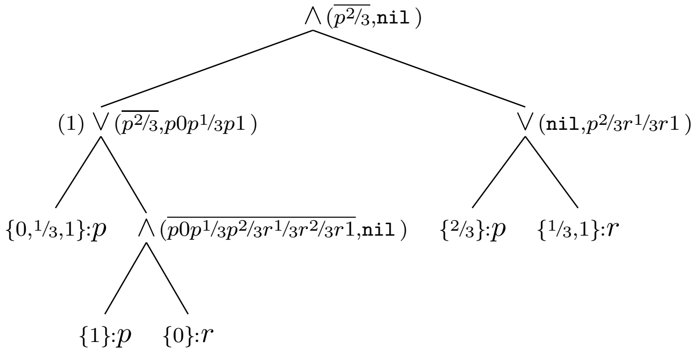
Remark 1 By the previous results we can state that if A is a signed formula, then after Δ-labelling A we get a Δ-restricted formula by applying the following steps:

- Substitute a subformula $B \sqsubseteq A$ by either $\top$ (if B is Δ_1-conclusive), or $\bot$ (if B is Δ_0-conclusive) or a literal ℓ (if B is ℓ-simple).
- Simplify a constant $\top$ or $\bot$, as soon as it is introduced by using the 0-1-laws.
- If A is finalizable, conclude the satisfiability of A (if $\Delta_1(A) \neq \texttt{nil}$) or its unsatisfiability (if $\Delta_0(A) = \bot$).

Example 2: Consider the formula

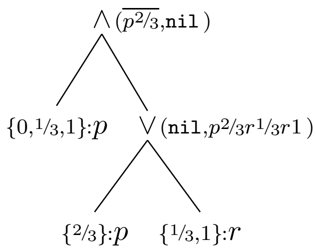
$$\big(\{0,{}^1\!/\!{}_3,1\}{:}p \vee (\{1\}{:}p \wedge \{0\}{:}r)\big) \wedge \big(\{{}^2\!/\!{}_3\}{:}p \vee \{{}^1\!/\!{}_3,1\}{:}r\big)$$

whose labelled syntactic tree is



[4]Regardless what this definition says, no leaf will be labelled in the examples.

In node 1 in the previous formula we have $\mathtt{Norm}(\overline{p\,2/3}) = \mathtt{Norm}(p0p\,1/3p1) = \{0,1/3,1\}{:}p$, i.e. it is $\{0,1/3,1\}{:}p$-simple and, therefore, the formula is equivalent to



This formula is Δ-restricted.

□

New applications of the Δ-lists to get information (up to equivalence) of a formula A is given by the following result, which is an important generalisation of the corresponding one in Classical Logic; in spite of its greater metatheoretical difficulty, it allows to extend the result in a natural manner and preserve the good computational behaviour.

Theorem 6 *Let A be a signed formula:*

1. *Assume $\overline{pj} \in \Delta_0(A)$ and let η_1 and η_2 be two arbitrary addresses in A such that $\mathtt{Node}(\eta_1) \neq \overline{pj}$ and $\mathtt{Node}(\eta_2) \neq pj$; then*

$$A \equiv \overline{pj} \wedge A[\overline{pj}/\top, pj/\bot, \eta_1/\,\mathtt{Node}(\eta_1) \wedge \overline{pj}, \eta_2/\,\mathtt{Node}(\eta_2) \vee pj]$$

2. *Assume $pj \in \Delta_1(A)$ and let η_1 and η_2 be two addresses in A such that $\mathtt{Node}(\eta_1) \neq \overline{pj}$ and $\mathtt{Node}(\eta_2) \neq pj$; then*

$$A \equiv pj \vee A[\overline{pj}/\top, pj/\bot, \eta_1/\,\mathtt{Node}(\eta_1) \wedge \overline{pj}, \eta_2/\,\mathtt{Node}(\eta_2) \vee pj]$$

PROOF:

1. By Theorem 1 we have $A \equiv \overline{pj} \wedge A$. Let I be a signed assignment:

   (a) If $I(\overline{pj}) = 0$, then $I(\overline{pj} \wedge B) = 0$ for all B, therefore:

   $$I(A) = I(\overline{pj} \wedge A) = 0 = I(\overline{pj} \wedge A[\overline{pj}/\top, pj/\bot, \eta_1/\,\mathtt{Node}(\eta_1) \wedge \overline{pj}, \eta_2/\,\mathtt{Node}(\eta_2) \vee pj])$$

   (b) If $I(\overline{pj}) = 1$, then $I(pj) = 0$, $I(\overline{pj} \wedge B) = I(B)$ and $I(pj \vee B) = I(B)$ for all B; therefore:

   $$I(A) = I(\overline{pj} \wedge A) = I(\overline{pj} \wedge A[\overline{pj}/\top, pj/\bot, \eta_1/\,\mathtt{Node}(\eta_1) \wedge \overline{pj}, \eta_2/\,\mathtt{Node}(\eta_2) \vee pj])$$

2. Similar.

QED

Remark 2 The different possibilities of substitutions that can be applied by using this theorem will be exploited in the definition of the $\widehat{\Delta}$-sets. Specifically, substitutions $A[\overline{pj}/\top, \eta_1/\,\mathtt{Node}(\eta_1)\wedge \overline{pj}]$ are encoded in the function `Filter`, to be used in the construction of the Δ_0-sets, and substitutions $A[pj/\bot, \eta_2/\,\mathtt{Node}(\eta_2) \vee pj]$ are encoded in the function `Filter` to be used in the construction of the Δ_1-sets.

The following result is a consequence of the theorem above; in its statement, the following notation will be used: If Γ is a set of literals and $\ell \notin \Gamma$ then $A[\Gamma/\Gamma \wedge \ell]$ is the formula obtained after substituting in A, every occurrence of $\ell' \in \Gamma$ by $\ell' \wedge \ell$.

Theorem 7 *If $\overline{pj} \in \Delta_0(A)$ and $\Gamma = \textsc{lit}_p \smallsetminus \{\overline{pj}\}$, then A is satisfiable if and only if $A[\overline{pj}/\top, \Gamma/\Gamma\wedge \overline{pj}]$ is satisfiable. If they are satisfiable, then there is a model I of A verifying $I(pj) = 0$.*

Proof:
From the previous theorem

$$A \equiv \overline{pj} \wedge A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}] \tag{1}$$

Now the necessary condition is immediate.

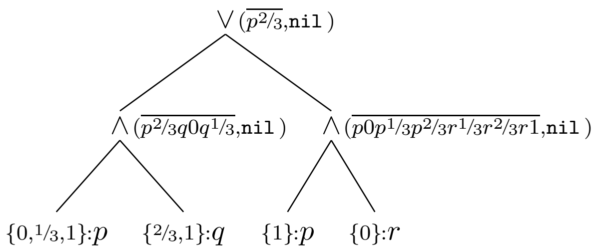
Conversely, let I be a signed assignment such that $I(A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}]) = 1$. Then:

- If $I(pj) = 0$, then $I(\overline{pj}) = 1$ and $I(A) = 1$ from equivalence (1).
- If $I(pj) = 1$, then $I(\overline{pj}) = 0$ and $I(\ell_p) = 0$ for all ℓ_p in $A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}]$, for $\ell_p \equiv \ell'_p \wedge \overline{pj}$; considering any assignment J verifying $J(qi) = I(qi)$ if $q \neq p$, $J(pi) = I(pi)$ if $i \neq j$, and $J(pj) = 0$, we have, by monotonicity of boolean conjunction and disjunction, $J(A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}]) = 1$ and therefore:

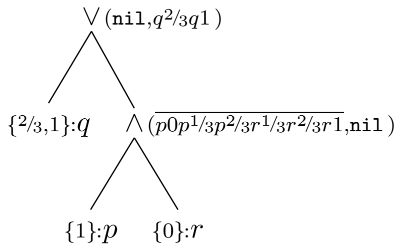
$$J(A) = J(\overline{pj} \wedge A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}]) = J(A[\overline{pj}/\top, \Gamma/\Gamma \wedge \overline{pj}]) = 1$$

QED

Example 3: Consider the formula with the syntactic tree below:



An application of the theorem above leads to the following equisatisfiable formula



$\square$

3.3 The $\widehat{\boldsymbol{\Delta}}$-sets

Given a Δ-restricted signed formula A, we define the sets $\widehat{\Delta_0}(A)$ and $\widehat{\Delta_1}(A)$, whose elements are pairs (α, η) where α is a Δ-list associated to a subformula B of A, and η is the address of B in A. The purpose of these sets is to transform the formula A into an equisatisfiable and smaller sized one.

The definition of these sets is given informally below:

Definition 6 *Let A be a Δ-restricted signed formula. For $b \in \{0, 1\}$, the set $\widehat{\Delta_b}(A)$ is recursively defined as follows:*

- *If ℓ is a literal, then $\widehat{\Delta_0}(\ell) = \widehat{\Delta_1}(\ell) = \varnothing$*
- *Otherwise, $\widehat{\Delta_b}(A)$ is defined as the set $\{(\texttt{Filter}(\Delta_b(B)), \eta_B) \mid B$ is a subformula of $A\}$*

where $\texttt{Filter}(\Delta_b(B)$ *is the result of*

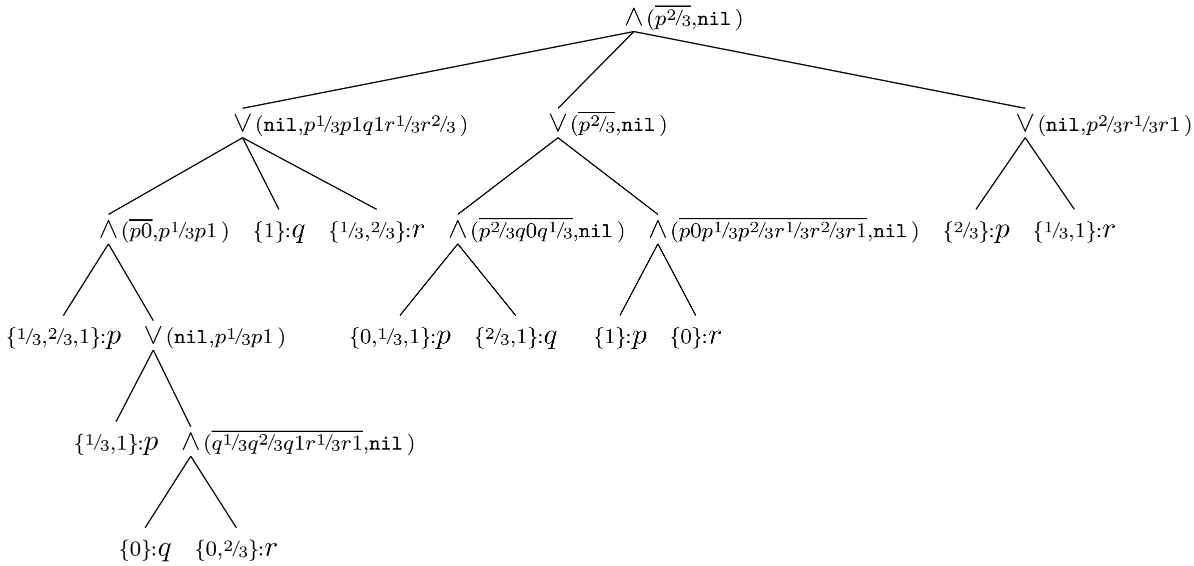
1. *Framing a literal ℓ_p if it is in $\Delta_b(B)$ and one of the following conditions hold:*
   (a) *$\ell_p \in \Delta_b(B')$ where $B \sqsubset B'$.*
   (b) *$\overline{\ell_p} \in \Delta_{\overline{b}}(B')$ where $B \sqsubset B'$.*
2. *Add $\boxed{\ell_p}$ to $\Delta_b(B)$ if $\ell_p \notin \Delta_b(B)$ but $\text{LIT}_p \cap \Delta_b(B) \neq \varnothing$ and either $\ell_p \in \Delta_b(B')$ or $\overline{\ell_p} \in \Delta_{\overline{b}}(B')$ where $B \sqsubset B'$.*
3. *Determining the subformulas which can be substituted by either a constant or a literal, by using the simplifications below:*
   - *If $b = 0$, identify the list with $\perp$ if it includes all the literals in LIT_p^- (either framed or unframed) for some p.*
   - *If $b = 1$, identify the list with $\top$ if it includes all the literals in LIT_p^+ (either framed or unframed) for some p.*

In the following example, the calculation in great detail of the $\widehat{\Delta}$-sets for a four-valued signed formula is introduced.

Example 4: Let us calculate the $\widehat{\Delta}$-sets for the following signed formula in $\mathcal{S}_4$

$$A = \big(\big(\{1/3,2/3,1\}{:}p \wedge \big(\{1/3,1\}{:}p \vee \big(\{0\}{:}q \wedge \{0,2/3\}{:}r\big)\big)\big) \vee \{1\}{:}q \vee \{1/3,2/3\}{:}r\big) \\ \wedge \big(\big(\{0,1/3,1\}{:}p \wedge \{2/3,1\}{:}q\big) \vee \big(\{1\}{:}p \wedge \{0\}{:}r\big)\big) \wedge \big(\{2/3\}{:}p \vee \{1/3,1\}{:}r\big)$$

The Δ-labelled syntactic tree of the formula is:



The Δ_0-lists for this formula are the following:

- $\Delta_0(\texttt{Node}(1122)) = \overline{q\,1/3}\,\overline{q\,2/3}\,\overline{q1}\,\overline{r\,1/3}\,\overline{r1}$.

  Literals $\overline{q1}$ and $\overline{r\,1/3}$ get framed because they are dominated (by literals $q1$ and $r\,1/3$ in node 1). In addition, since $r\,2/3$ also dominates this node, therefore $\boxed{\overline{r\,2/3}}$ is added to this address.[5] The output of `Filter` for this node is $\overline{q\,1/3}\,\overline{q\,2/3}\,\boxed{\overline{q1}}\,\boxed{\overline{r\,1/3}}\,\boxed{\overline{r\,2/3}}\,\overline{r1}$.

- $\Delta_0(\texttt{Node}(11)) = \overline{p0}$.

  Literals $p\,1/3$ and $p1$ affect this node from address 1, and literal $\overline{p\,2/3}$ affects this node from the root. The filtering of this Δ-list would be $\overline{p0}\boxed{\overline{p\,1/3}}\boxed{\overline{p\,2/3}}\boxed{\overline{p1}}$. In addition, as we are in $\mathcal{S}_4$, these are all the possibilities for p, so the final result is $(\bot, 11)$.

- $\Delta_0(\texttt{Node}(21)) = \overline{p\,2/3}\,\overline{q0}\,\overline{q\,1/3}$.

  In this case, only $\overline{p\,2/3}$ gets affected by the filtering process, the final result is $\boxed{\overline{p\,2/3}}\,\overline{q0}\,\overline{q\,1/3}$.

- $\Delta_0(\texttt{Node}(22)) = \overline{p0}\,\overline{p\,1/3}\,\overline{p\,2/3}\,\overline{r\,1/3}\,\overline{r\,2/3}\,\overline{r1}$.

  Only $\overline{p\,2/3}$ gets affected by the filtering process, the final result is $\overline{p0}\,\overline{p\,1/3}\,\boxed{\overline{p\,2/3}}\,\overline{r\,1/3}\,\overline{r\,2/3}\,\overline{r1}$.

- $\Delta_0(\texttt{Node}(2)) = \overline{p\,2/3}$.

  Once again, only $\overline{p\,2/3}$ gets affected, and the final result is $\boxed{\overline{p\,2/3}}$.

[5]Note that there are literals on the variable p affecting this node, but we don't bother about those literals because there are no occurrences of p in this Δ-list.

- $\Delta_0(A) = \overline{p\,{}^{2}\!/_{3}}$. As no node dominates the root, no filtering applies.

After the filtering process, we have that

$$\begin{aligned}\widehat{\Delta_0}(A) \quad = \quad \{ & (\overline{q\,{}^{1}\!/_{3}}\,\overline{q\,{}^{2}\!/_{3}}\,\boxed{\overline{q1}}\,\boxed{\overline{r\,{}^{1}\!/_{3}}}\,\boxed{\overline{r\,{}^{2}\!/_{3}}}\,\overline{r1}, 1122), (\bot, 11), (\overline{p\,{}^{2}\!/_{3}}, \varepsilon), (\boxed{\overline{p\,{}^{2}\!/_{3}}}\,\overline{q0}\,\overline{q\,{}^{1}\!/_{3}}, 21), \\ & (\overline{p0}\,\overline{p\,{}^{1}\!/_{3}}\,\boxed{\overline{p\,{}^{2}\!/_{3}}}\,\overline{r\,{}^{1}\!/_{3}}\,\overline{r\,{}^{2}\!/_{3}}\,\overline{r1}, 22), (\boxed{\overline{p\,{}^{2}\!/_{3}}}, 2)\}\end{aligned}$$

The Δ_1-lists for this formula are the following:

- $\Delta_1(\texttt{Node}(112)) = p\,{}^{1}\!/_{3}p1$.

  This is dominated by $\overline{p0}$ at address 11 and by $\overline{p\,{}^{2}\!/_{3}}$ at the root. Therefore, the result would be $\boxed{p0}\boxed{p\,{}^{1}\!/_{3}}\boxed{p\,{}^{2}\!/_{3}}\boxed{p1}$, but as we are in $\mathcal{S}_4$, a simplification applies and the final output is $(\top, 112)$.

- $\Delta_1(\texttt{Node}(11)) = p\,{}^{1}\!/_{3}p1$.

  Here, literals $p\,{}^{1}\!/_{3}$ and $p1$ get affected from node 1, in addition $\overline{p\,{}^{2}\!/_{3}}$ dominates this node from the root. The final result after filtering is $\boxed{p\,{}^{1}\!/_{3}}\,\boxed{p\,{}^{2}\!/_{3}}\,\boxed{p1}$.

- $\Delta_1(\texttt{Node}(1)) = p\,{}^{1}\!/_{3}p1q1r\,{}^{1}\!/_{3}r\,{}^{2}\!/_{3}$.

  In this case, $p\,{}^{2}\!/_{3}$ dominates this node. The result is $p\,{}^{1}\!/_{3}\,\boxed{p\,{}^{2}\!/_{3}}\,p1\,q1\,r\,{}^{1}\!/_{3}\,r\,{}^{2}\!/_{3}$.

- $\Delta_1(\texttt{Node}(3)) = p\,{}^{2}\!/_{3}r\,{}^{1}\!/_{3}r1$.

  Here, only $p\,{}^{2}\!/_{3}$ gets dominated. The result is $\boxed{p\,{}^{2}\!/_{3}}\,r\,{}^{1}\!/_{3}\,r1$.

After the filtering process, we have that

$$\widehat{\Delta_1}(A) \quad = \quad \{(\top, 112), (\boxed{p\,{}^{1}\!/_{3}}\boxed{p\,{}^{2}\!/_{3}}\boxed{p1}, 11), (p\,{}^{1}\!/_{3}\boxed{p\,{}^{2}\!/_{3}}p1q1r\,{}^{1}\!/_{3}r\,{}^{2}\!/_{3}, 1), (\boxed{p\,{}^{2}\!/_{3}}r\,{}^{1}\!/_{3}r1, 3)\}$$

□

Note the following consequences from the definition of the $\widehat{\Delta_b}$-sets for a given Δ-restricted signed formula A and $b \in \{0, 1\}$:

1. If $\alpha = \Delta_0(A) \neq \texttt{nil}$, then $(\alpha, \varepsilon) \in \widehat{\Delta_0}(A)$, $\alpha \neq \bot$ and α does not have framed literals. (Just note that a literal $\ell \in \alpha$ is framed in (α, η) from the information in the Δ-lists of its ancestors).
2. For every literal in $\widehat{\Delta_0}(A) \cup \widehat{\Delta_1}(A)$, at least one of its occurrences is not framed.
3. As A is a Δ-restricted signed formula then no element in $\widehat{\Delta_1}(A)$ is (α, ε).
4. If $(\alpha, \eta) \in \widehat{\Delta_b}(A)$, then η is not the address of a leaf of T_A (since $\widehat{\Delta_0}(\ell) = \widehat{\Delta_1}(\ell) = \varnothing$ for all literal ℓ).

The following theorem states that, as the Δ-labels, the $\widehat{\Delta}$-labels also allow substitution of subformulas in A by either $\top$, or $\bot$, or a literal.

Theorem 8 *Let A be a Δ-restricted signed formula then*

1. *If $(\bot, \eta) \in \widehat{\Delta_0}(A)$, then*

$$A \equiv \begin{cases} A[\eta/\bot] & \text{if } \Delta_1(\texttt{Node}(\eta)) = \texttt{nil} \\ A[\eta/\,\texttt{Norm}(\Delta_1(\texttt{Node}(\eta)))] & \text{otherwise} \end{cases}$$

2. *If $(\top, \eta) \in \widehat{\Delta_1}(A)$, then*

$$A \equiv \begin{cases} A[\eta/\top] & \text{if } \Delta_0(\texttt{Node}(\eta)) = \texttt{nil} \\ A[\eta/\,\texttt{Norm}(\Delta_0(\texttt{Node}(\eta)))] & \text{otherwise} \end{cases}$$

We will use the following technical lemma to prove the theorem.

Lemma 9 *Let A be a signed formula, $\eta \neq \varepsilon$ an address in A and $C = \texttt{Node}(\eta)$; then:*

1. *If $\overline{pj} \in \Delta_0(A)$, then $A \equiv A[\eta/\overline{pj} \wedge C]$.*
2. *If $\overline{pj} \in \Delta_0(A)$, then $A \equiv \overline{pj} \wedge A[\eta/pj \vee C]$.*
3. *If $pj \in \Delta_1(A)$, then $A \equiv pj \vee A[\eta/\overline{pj} \wedge C]$.*
4. *If $pj \in \Delta_1(A)$, then $A \equiv A[\eta/pj \vee C]$.*

PROOF:

1. Assume that $\overline{pj} \in \Delta_0(A)$ and consider an assignment I.
   - If $I(pj) = 0$, then $I(C) = I(\overline{pj} \wedge C)$ and therefore $I(A) = I(A[\eta/\overline{pj} \wedge C])$.
   - If $I(pj) = 1$, then $I(A) = 0$ since $A \equiv \overline{pj} \wedge A$; on the other hand, $I(\overline{pj} \wedge C) = 0 \leq I(C)$; then, by monotonicity of boolean conjunction and disjunction, $I(A[\eta/\overline{pj} \wedge C]) \leq I(A) = 0$ and therefore $I(A[\eta/\overline{pj} \wedge C]) = 0$.
2. 3. and 4. The proof is similar.

QED

PROOF OF THEOREM 8:

1. Suppose that $(\bot, \eta) \in \widehat{\Delta_0}(A)$ and consider $C = \texttt{Node}(\eta)$. By the definition of $\widehat{\Delta_0}(A)$ there exist subformulas $B_1, \dots B_m$, with[6] $1 \leq m < n$ and $p \in \mathcal{V}$ such that:
   - $C \sqsubset B_1 \sqsubset \cdots \sqsubset B_m \sqsubseteq A$.
   - $\Delta_0(C) \cap \text{LIT}_p^- \neq \varnothing$ and for all $\ell \in \text{LIT}_p^-$ either $\ell \in \Delta_0(C)$ or there exists B_i such that $\ell \in \Delta_0(B_i) \cup \overline{\Delta_1(B_i)}$.

[6] Recall that n denotes the number of truth values.

By Corollary 2 we have $C \equiv X \vee (\mathtt{Norm}(\Delta_0(C)) \wedge C)$, where the formula X is defined as follow:

$$X = \begin{cases} \bot & \text{if } \Delta_1(\mathtt{Node}(\eta)) = \mathtt{nil} \\ \mathtt{Norm}(\Delta_1(\mathtt{Node}(\eta))) & \text{otherwise} \end{cases}$$

Therefore, $A \equiv A[\eta/X \vee (\mathtt{Norm}(\Delta_0(C)) \wedge C)]$ and this substitution does not change the Δ-lists associated to the ascendant nodes of η.[7]

Let Γ be the list obtained by ordering the elements in $\textsc{lit}_p^- \setminus \Delta_0(C)$. For all $\ell \in \Gamma$, there exists i with $1 \leq i \leq m$ such that $\ell \in \Delta_0(B_i) \cup \overline{\Delta_1(B_i)}$; now, by items 1 and 3 of Lemma 9, we have either $B_i \equiv B_i[\eta/\ell \wedge C]$ (if $\ell \in \Delta_0(B_i)$) or $B_i \equiv \overline{\ell} \vee B_i[\eta/\ell \wedge C]$ (if $\overline{\ell} \in \Delta_1(B_i)$); as the Δ_1-lists are invariant under those substitutions (only the Δ_0-lists of the ascendant nodes can be increased), all the substitutions can be applied one after the other and we get:

$$A \equiv A\big[\eta/X \vee \big(\mathtt{Norm}(\Delta_0(C)) \wedge \mathtt{Norm}(\Gamma) \wedge C\big)\big]$$

finally, as $\mathtt{Norm}(\Delta_0(C)) \wedge \mathtt{Norm}(\Gamma) \equiv \bot$ we have that

$$A \equiv A[\eta/X]$$

2. Similar, by using items 2 and 4 in Theorem 8.

QED

Remark 3 Note that, as with the Δ-labels, the formula obtained in the previous theorem after substituting is equivalent to the initial formula, but there is a substantial difference: the information given by the Δ-lists substitutes subformulas which are equivalent to either $\top$ or $\bot$ or a literal; however, under the hypotheses of this theorem, it needn't be true that $\mathtt{Node}(\eta)$ is equivalent to either $\top$ or $\bot$ or a literal.

Definition 7 *Let A be an signed formula; then it is said that A is* ***restricted*** *if it is Δ-restricted and satisfies the following:*

- *There are not elements $(\bot, \eta)$ in $\widehat{\Delta_0}(A)$.*
- *There are not elements $(\top, \eta)$ in $\widehat{\Delta_1}(A)$.*

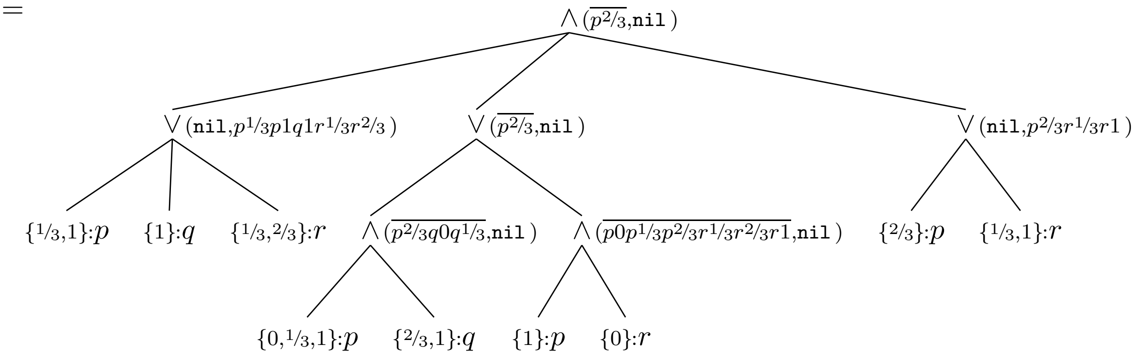
Definition 8 *If A is a signed formula, to* ***label*** *T_A means Δ-label T_A and to associate to the root of T_A the ordered pair $\big(\widehat{\Delta_0}(A), \widehat{\Delta_1}(A)\big)$.*

Remark 4 Note that given a Δ-restricted signed formula, A, after calculating $\big(\widehat{\Delta_0}(A), \widehat{\Delta_1}(A)\big)$ we get either the (un)satisfiability of A or an equivalent and restricted signed formula by means of the substitutions determined by Theorem 8, and the 0-1-laws.

[7] Note that if $(\bot, \eta) \in \widehat{\Delta_0}(A)$, then $\Delta_0(\mathtt{Node}(\eta)) \neq \mathtt{nil}$; thus, either $\Delta_1(\mathtt{Node}(\eta)) = \mathtt{nil}$ or $\mathtt{Norm}(\Delta_1(\mathtt{Node}(\eta)))$ is a literal.

Example 5: Following with the formula in the previous example; as $(\bot, 11) \in \widehat{\Delta_0}(A)$ and $\Delta_1(\texttt{Node}(11)) = p\,1/3\,p1$, node 11 is substituted by the literal $\{1/3,1\}{:}p$ and we obtain the following formula B which is equivalent to A:

$B =$



The $\widehat{\Delta}$-sets for B are:

$$\widehat{\Delta_0}(B) = \{(\overline{p\,2/3}, \varepsilon), (\boxed{\overline{p\,2/3}}\,\overline{q0}\,\overline{q\,1/3}, 21), (\overline{p0}\,\overline{p\,1/3}\,\boxed{\overline{p\,2/3}}\,\overline{r\,1/3}\,\overline{r\,2/3}\,\overline{r1}, 22), (\boxed{\overline{p\,2/3}}, 2)\}$$
$$\widehat{\Delta_1}(B) = \{(p\,1/3\boxed{p\,2/3}p1q1r\,1/3r\,2/3, 1), (\boxed{p\,2/3}r\,1/3r1, 3)$$

Therefore, B is restricted.

□

3.4 Some satisfiability-preserving results

The following result is the global counterpart of Theorem 7, where we state the way in which the the information of the $\widehat{\Delta}$-sets is used to translate a signed formula A into an equisatisfiable and smaller sized A'. In addition, we give information about a possible model by restricting the set of truth-values for some variables in A.

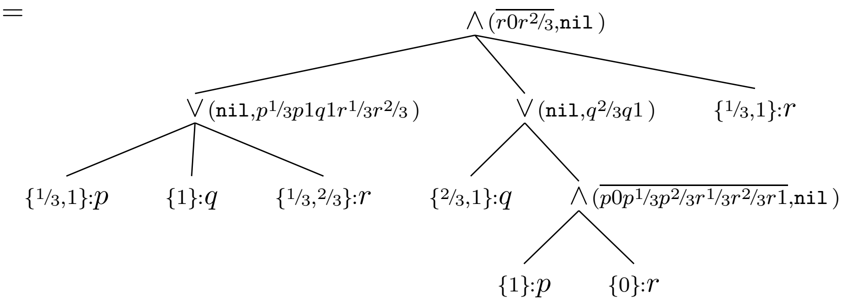
Corollary 10 (Complete Reduction) *Let A be a signed formula such that $(\alpha, \varepsilon) \in \widehat{\Delta_0}(A)$, $\alpha = \overline{p_1 j_1} \dots \overline{p_m j_m}$; for all i with $1 \le i \le m$ consider $\Gamma_i = \textsc{lit}^-_{p_i} \smallsetminus \{\overline{p_i j_i}\}$; and consider also the formulas A_i for all i, recursively defined as:*

$$A_1 = A[\overline{p_1 j_1}/\top, \Gamma_1/\Gamma_1 \wedge \overline{p_1 j_1}], \qquad A_k = A_{k-1}[\overline{p_k j_k}/\top, \Gamma_k/\Gamma_k \wedge \overline{p_k j_k}] \textit{ for } 2 \le k \le m$$

Then, A is satisfiable if and only if A_m is satisfiable; in this case there is a model verifying $I(p_i j_i) = 0$ for all i.

Example 6: From the previous example; as $\overline{p\,2/3} \in \Delta_0(B)$, with an application of Theorem 7, making the substitutions $[\overline{p\,2/3}/\top]$, $[p\,2/3/\bot]$, $[\ell_p/\overline{p\,2/3} \wedge \ell_p]$, we obtain a formula C equisatisfiable with B, and the information $I(p\,2/3) = 0$ is stored for a possible model:

$C =$

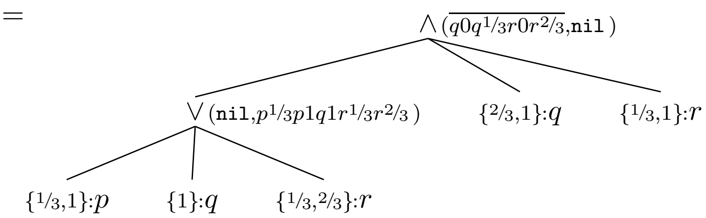


The $\widehat{\Delta}$-sets for C are:

$$\widehat{\Delta_0}(C) = \{(\overline{r0}\,\overline{r\,{}^2\!/\!_3}, \varepsilon), (\bot, 22)\}$$
$$\widehat{\Delta_1}(C) = \{(p\,{}^1\!/\!_3p1q1\boxed{r0}r\,{}^1\!/\!_3\boxed{r\,{}^2\!/\!_3}, 1), (q\,{}^2\!/\!_3q1, 2)\}$$

After substituting node 22 by $\bot$ we get the following formula

$D =$

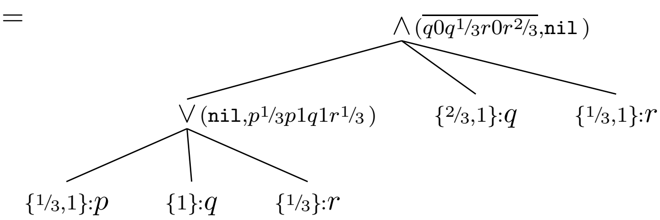


The $\widehat{\Delta}$-sets for D are:

$$\widehat{\Delta_0}(D) = \{(\overline{q0q\,{}^1\!/\!_3r0r\,{}^2\!/\!_3}, \varepsilon)$$
$$\widehat{\Delta_1}(D) = \{(p\,{}^1\!/\!_3p1\boxed{q0}\boxed{q\,{}^1\!/\!_3}q1\boxed{r0}r\,{}^1\!/\!_3\boxed{r\,{}^2\!/\!_3}, 1)\}$$

A final application of complete reduction leads to

$E =$



The $\widehat{\Delta}$-sets for E are:

$$\widehat{\Delta_0}(E) = \{(\overline{q0q\,{}^1\!/\!_3r0r\,{}^2\!/\!_3}, \varepsilon)$$
$$\widehat{\Delta_1}(E) = \{(p\,{}^1\!/\!_3p1\boxed{q0}\boxed{q\,{}^1\!/\!_3}q1\boxed{r0}r\,{}^1\!/\!_3\boxed{r\,{}^2\!/\!_3}, 1)\}$$

□

In [3], the following generalisation of the pure literal rule was introduced in the framework of Signed Logic:

Definition 9 *Let A be a signed formula and $p \in \mathcal{V}$. A literal pj is said to be* ***pure in A*** *if for all leaf $S : p$ in A we have $j \in S$*

Theorem 11 (Pure literal rule) *Let A be a signed formula and assume pj is pure in A. Then, A is satisfiable if and only if $A[\text{LIT}_p/\top]$ is satisfiable; equivalently, if A has a model I such that $I(pj) = 1$.*

PROOF:
The sufficiency is trivial. For the converse, assume that $I(A) = 1$ and consider the assignment J defined as $J(qi) = I(qi)$ if $q \neq p$, and $J(pj) = 1$; since pj is pure in A, if $S : p$ is a leaf in A, then $j \in S$ and, therefore, $J(S : p) = 1$; now, by monotonicity of boolean conjunction and disjunction we have $J(A) \geq I(A)$ and

$$1 = J(A) = J(A[\text{LIT}_p/\top]) = I(A[\text{LIT}_p/\top])$$

QED

Example 7: In the formula in the previous example, literals $p\,{}^1\!/\!_3$, $q1$ and $r\,{}^1\!/\!_3$ are pure. By the pure literal rule, formula E is transformed into $\top$, therefore E is satisfiable, being $I(p\,{}^1\!/\!_3) = 1$, $I(q1) = 1$ and $I(r\,{}^1\!/\!_3) = 1$ a model for A.

□

We introduce below an important generalisation of the previous theorem. Firstly, we generalise the definition of pure literal by using our $\widehat{\Delta}$-sets.

Definition 10 *Let A be a signed formula. Literal pj is said to be* ***$\widehat{\boldsymbol{\Delta}}$-pure*** *in A if the following conditions hold:*

1. *$\overline{pj}$ does not occur unframed in $\widehat{\Delta_0}(A)$.*
2. *If $(\alpha, \eta) \in \widehat{\Delta_1}(A)$ and $\alpha \cap \text{LIT}_p^+ ne\varnothing$, then $pj \in \alpha$ (possibly framed).*

Theorem 12 (Extended pure literal rule) *Let A be a signed formula and pj a $\widehat{\Delta}$-pure literal in A. Then, A is satisfiable if and only if the formula $A[\text{LIT}_p/\top]$; equivalently, A is satisfiable if and only if it has a model I such that $I(pj) = 1$.*

PROOF:
Let $S{:}p$ be a literal in A such that $j \notin S$. As pj is $\widehat{\Delta}$-pure we have two cases:

- If $S{:}p$ is a child of a conjunction $B = \texttt{Node}(\eta)$, then $(\alpha, \eta) \in \widehat{\Delta_0}(A)$ and $\boxed{\overline{pj}} \in \alpha$.
- If $S{:}p$ is a child of a disjunction $B = \texttt{Node}(\eta)$, then $(\alpha, \eta) \in \widehat{\Delta_1}(A)$ and $\boxed{pj} \in \alpha$.

In both cases, by the definition of the $\widehat{\Delta}$-sets and $\widehat{\Delta}$-pure literal, we have that $S{:}p$ has an ancestor η such that $(\alpha, \eta) \in \widehat{\Delta_1}(A)$ and $pj \in \alpha$.

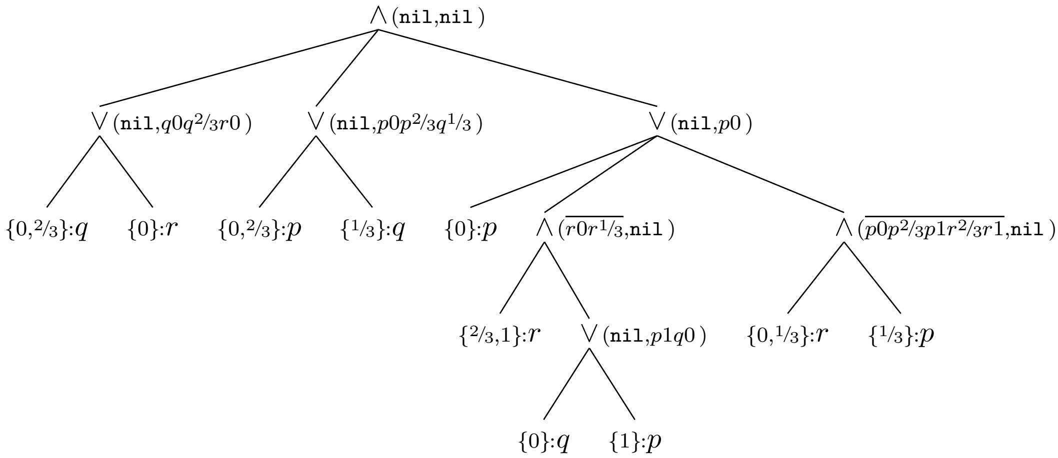
Let $\eta_1, \dots, \eta_m$ the addresses such that $\{(\alpha_{1,}, \eta_1), \dots, (\alpha_{1,}, \eta_1)\} \subseteq \widehat{\Delta_1}(A)$ and $pj \in \alpha_i$ for all i; and let B the formula obtained from A by substituting each node η_i by $\texttt{Node}(\eta_i)[\text{LIT}_p/\text{LIT}_p \vee pj]$. By Lemma 9, we have that $A \equiv B$ and, by the choosing of η_i, pj is pure in B (in the sense of Definition 9). Therefore, B is satisfiable iff it has a model verifying $I(pj) = 1$.

QED

Example 8: Consider the following Δ-labelled signed formula in $\mathcal{S}_4$:

$$A = (\{0,2/3\}{:}q \vee \{0\}{:}r) \wedge (\{0,2/3\}{:}p \vee \{1/3\}{:}q) \wedge (\{0\}{:}p \vee (\{2/3,1\}{:}r \wedge (\{0\}{:}q \vee \{1\}{:}p)) \vee (\{0,1/3\}{:}r \wedge \{1/3\}{:}p))$$

After Δ-labelling we obtain

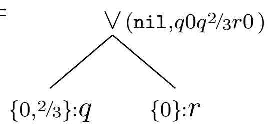


The $\widehat{\Delta}$-sets for A are

$$\widehat{\Delta_0}(A) = \{(\overline{r0}\,\overline{r\,1/3}, 32), (\boxed{p0}\,\overline{p\,2/3}\,\overline{p1}\,\overline{r\,2/3}\,\overline{r1}, 33)\}$$
$$\widehat{\Delta_1}(A) = \{(q0q\,2/3r0, 1), (p0p\,2/3q\,1/3, 2), (\boxed{p0}p1q0, 322), (p0, 3)\}$$

Now, $p0$ is $\widehat{\Delta}$-pure[8], therefore A is satisfiable iff $B = A[\text{LIT}_p/\top]$ is:

$B =$



and information $I(p0) = 1$ is stored. As $q0 \in \Delta_1(B)$, B is finalizable and, consequently, satisfiable by any assignment verifying $I(q0) = 1$. Therefore, A is satisfiable by any assignment such that $I(p0) = 1$ and $I(q0) = 1$.

□

[8] Note that it is not pure in A.

4 Conclusions

The basic idea of the TAS methods is to decrease the size of a formula by extracting information from its syntactic tree in order to avoid as much branching as possible.

New reduction techniques for signed propositional logics are introduced in the framework of the TAS methodology. All these reductions have at most quadratic complexity wrt the size of the formula; specifically, on the one hand, the tests to check the applicability are quadratic in the worst case, on the other hand, the complexity of applying any reduction is linear. This way, the exponential complexity in the worst case of a given prover, due to the branching procedure, can be decreased by using these reduction strategies before branching.

The improvements introduced wrt previous works are the explicit extension to any finite-valued logic; a new definition of the Δ-lists which now facilitates certain reductions by integrating them in the calculation of the lists; the $\widehat{\Delta}$-sets are introduced in the multiple-valued framework and as a result new reduction strategies arise. We have introduced techniques of manipulating lists of unitary implicants/implicates (essentially, inserting and/or framing elements) which can improve the performance of a given prover for signed logic by decreasing the size of the formulas to be branched.

Acknowledgements

We would like to thank two anonymous referees for helpful comments and suggestions on an earlier version of this paper

References

[1] G. Aguilera, I. P. de Guzmán, and M. Ojeda. Increasing the efficiency of automated theorem proving. *Journal of Applied Non-Classical Logics*, 5(1):9–29, 1995.

[2] G. Aguilera, I. P. de Guzmán, and M. Ojeda. A reduction-based theorem prover for 3-valued logic. *Mathware & Soft Computing* IV(2):99-127, 1997.

[3] G. Aguilera, I. P. de Guzmán, M. Ojeda, A. Valverde. On reductions of signed formulas. In *Proceedings of the WG2 meeting of COST Action 15 "Many-Valued Logics for Computer Science Applications", Málaga*, September 1997.

[4] I. P. de Guzmán, M. Ojeda, A. Valverde. Ortho-regular logics and bi-implications in ortho-regular logics. Tech. Report 97-10. Dept. Matemática Aplicada, Univ. Málaga. Submitted for publication, 1997

[5] J. Gallier. *Logic for Computer Science.* John Wiley & sons, 1988.

[6] R. Hähnle. *Automated Deduction in Multiple-valued Logics.* Oxford University Press, 1993.

[7] N. Murray and E. Rosenthal. Adapting classical inference techniques to multiple-valued logics using sign formulas. *Fundamenta Informaticae*, 21(3):237–253, 1994.