

# A Polynomial Algorithm for **3-sat**

M Angela Weiss

weiss@ime.usp.br, Universidade de São Paulo, São Paulo - Brasil

May 30, 2011

## Abstract

We present here a **KE**-tableaux based method we use as an algorithm for solving **3-sat**. We show that using this method we polynomially decide if a given formula **3-sat** is satisfiable or not. We solve, in this way, the classic question whether  $\mathbf{P} = \mathbf{NP}$ .

## 1 Introduction

In this paper, we present a polynomial (in time and space) algorithm to decide if a given **3-sat** formula is satisfiable or not. We call the algorithm a *macrotableau for 3-sat*. The name *macro* is due to the fact that we use classical **KE**-tableaux methods to build a macro where we store the relevant data (all closed branches) for a tableau for a given **3-sat** formula  $\Psi$ .

The class of **sat** problems was shown to be **NP-complete**. See Cook, in [2] and Cook and Reckhow in [3] and seminal papers of L. Levin, in [6] and [5]. The goals are to establish lower bounds in complexity, see Meyer and Sotckmeyer in [12] and Meyer in [11]. The literature in this area is rich of very nice surveys, like [4] and [13].

Our approach for deciding satisfiability of formulas in **3-sat** is via tableaux. The bound in space and time, discussed in Section 6, is a polynomial function over the number of clauses of a given **3-sat** formula. The method we introduce here is based on **KE** tableaux (see Mondatori in [7], [8], [9] and [10]). In this paper, we make use of labeled tableaux (the label ranges over the set of values **false**,  $F$  or **true**  $T$ ) for **3-sat** formulas.

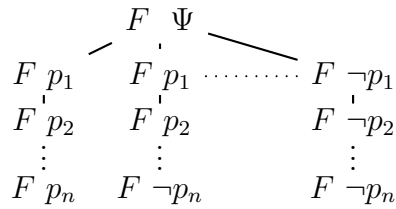
The decision method we propose here has its structure based on the rules of **KE** tableaux. However, we strongly use the restriction of number of clauses (the input) of a given formula  $\Psi$ , written in CNF, and explore repetition of patterns of clauses. Spite the complexity of an Analytic Tableaux can grow exponentially (see [1]), grouping clauses in similar groups prevents the exponential growth. We show that the number of branches, under this particular grouping, is polynomially bounded by the number of clauses of  $\Psi$ .

The novelty introduced here is in that we deal with similar groups of clauses, saving time and space. Decision methods using **KE** tableaux is a well trodden path. We could have chose equivalent decision methods, since we do not write repeated deductions, but store similar bits of deduction in a unique storage space, bounded by the input given by the number of clauses.

A preliminary version of this paper was published a technical report ([14]).

## 2 Summary

Given a **3-sat** formula  $\Psi$ , we choose a set, called labels, of conjugated pairs of literals in  $\Psi$  and suppose that a tableaux for  $F\Psi$  where the nodes are ranging over all consistent choices over  $\{p_1, \neg p_1, \dots, p_n, \neg p_n\}$ , labeled as **false**.



The tableau above has  $2^n$  branches. Call it the *Combinatorics Choice Tableau*. Its branches are the result of  $2^n$  (consistent) combinations over the set of labels. Our goals are to show that further development of the above (expsize) choices can be stored using space and time bounded by a polynomial function of the output (the number of literals in the formula). The storage tools are developed in Section 3, where we search for closed branches of the Combinatorics Choice Tableau. In Section 5, we perform a fast verification if the Combinatorics Choice Tableau is closed. In Section 4 we show that we store only the closed branches of the Combinatorics Choice Tableau.

### 3 Basic Definitions

In this Section, we deal with the basic definitions. We built here the machinery for both, in Section 4 polynomially write all the groups of closed branches in the Combinatorics Choice Tableau for  $\Psi$  and, in Section 5, we decide if the closed branches we wrote are all branches of the Combinatorics Choice Tableau or no.

A **3-sat** formula  $\Psi$  is a conjunctions of a number, say  $\mathbf{n}$ , of formulas where each of them is a disjunction of three literals. We write

$$\Psi \equiv (l_1^1 \vee l_1^2 \vee l_1^3) \wedge \cdots \wedge (l_{\mathbf{n}}^1 \vee l_{\mathbf{n}}^2 \vee l_{\mathbf{n}}^3) \equiv C_1 \wedge \cdots \wedge C_{\mathbf{n}}$$

A subformula  $C_k \equiv l_k^1 \vee l_k^2 \vee l_k^3$ ,  $1 \leq k \leq n$  of  $\Psi$  is called a *clause*.

**Definition 3.1** *The set of literals of a formula  $\Psi$  is denoted by  $\text{Letter}(\Psi)$ .*

*A pair of a literal and its negation is called a conjugated pair.*

**Notation 3.2** *Consider the 3-sat formula*

$$\psi \equiv (p \vee q_{11} \vee q_{12}) \wedge (p \vee q_{21} \vee q_{22}) \wedge \cdots \wedge (p \vee q_{t1} \vee q_{t2})$$

*The factorization of  $\psi$  by  $p$ , denoted  $p \vee S_p$ , is*

$$p \vee ((q_{11} \vee q_{12}) \wedge (q_{21} \vee q_{22}) \wedge \cdots \wedge (q_{t1} \vee q_{t2})) \quad (1)$$

*The literal  $p$  is called the factor of  $S_p$ .*

**Definition 3.3** *Given a 3-sat formula  $\Psi$ , a set of labels is a set of literals in  $\Psi$ ,  $\text{Label} = \{p_{01}, \neg p_{01}, \dots, p_{0k_0}, \neg p_{0k_0}, \dots, p_{h1}, \neg p_{h1}, \dots, p_{hk_h}, \neg p_{hk_h}\}$  so that*

$$\Psi \equiv S_3^1 \wedge S_3^2 \wedge \cdots \wedge S_3^h \wedge S_{\top}$$

*where each  $S_3^i$  are factorizations of  $\Psi$  ( $\Psi$  is called a partitioned formula).*

$$\begin{aligned} S_3^0 &\equiv (l_{01} \vee S_{l_{01}}) \wedge \cdots \wedge (l_{k_0} \vee S_{l_{k_0}}) \\ S_3^1 &\equiv (l_{11} \vee S_{l_{11}}) \wedge \cdots \wedge (l_{1k_1} \vee S_{l_{1k_1}}) \\ &\vdots \\ S_3^h &\equiv (l_{h1} \vee S_{l_{h1}}) \wedge \cdots \wedge (l_{hk_h} \vee S_{l_{hk_h}}) \end{aligned}$$

*each  $l_{is}$  contained in  $\{p_{01}, \neg p_{01}, \dots, p_{0k_0}, \neg p_{0k_0}, \dots, p_{h1}, \neg p_{i1}, \dots, p_{ik_i}, \neg p_{ik_i}\}$ ,  $S_{\top}$  is at most a 2-sat formula and if a literal  $l$  is a factor of  $S_3^i$ , then it cannot be subformula of any  $S_3^{i'}$  for any  $i' \geq i$  or of  $S_{\top}$ .*

In Example 7.6, Part 1, we show a partition.

As a practical result, solving a branch whose nodes are  $F l_1, \dots, F l_n$  means to decide if the conjunction of each  $S l_1, \dots, S l_n$  is consistent. In Section 4 we write, polynomially, the inconsistent disjunctions (i.e., the closed branches of the Combinatorics Choice Tableau).

**Definition 3.4** *The Combinatorics Choice Tableau is the tableaux for  $F\psi$  whose branches are formed by all (consistent) choices*

$$\begin{array}{l} F l_{01} \\ \vdots \\ F l_{hk_h} \\ T S_{l_{01}} \\ \vdots \\ T S_{l_{hk_h}} \\ T S_{\top} \end{array}$$

$l_{ij} \in \{p_{ij}, \neg p_{ij}\}$  ranging over the labels (see Example 7.6, Part 3).

**Observe:** If there is a literal  $r$  so that  $r \in \text{Letter}(\Psi)$  and  $\neg r \notin \text{Letter}(\Psi)$ , then  $\Psi$  is false iff the formula

$$\Psi' \equiv \{C_i \mid 1 \leq i \leq n \wedge r \notin \text{Letter}(C_i)\}$$

is false since a valuation  $r = \top$  trivializes our search. We work, thus, with a *complete set of labels*, that is, we assume that if a literal  $q \in \text{Letter}$ , then its negation  $\neg q$  belongs of  $\text{Letter}$ . If  $p_{rs_r}, \neg p_{rs_r} \in \mathbb{L}abel$ ,  $r > 0$ ,  $1 \leq s_r \leq k_r$ , then at least one literal in  $\{p_{rs_r}, \neg p_{rs_r}\}$  belongs to a  $S_l$ , for  $l \in \{p_{tut}, \neg p_{tut}\}$ ,  $t < r$ ,  $1 \leq ut < k_t$  as we see in Example 7.6.

**Definition 3.5** *Given a partitioned 3-sat formula  $\Psi$ , define the cylindrical digraph  $\mathbb{C}lndr$ -graph generated by  $\Psi$ ,  $(L_{01}, \Rightarrow, \rightarrow)$  as*

- $L_{01} \subseteq (\text{Letter}(\Psi) \times \{0, 1\}) \cup \{\perp, \top\}$ ;
- If  $a \vee b$  is a clause of  $S_{l_1}, \dots, S_{l_k}$ , then  $a_0 \Rightarrow b_1$  and  $b_0 \Rightarrow a_1$  are edges and both have as label the string  $l_1 \dots l_k$  of literals in  $\mathbb{L}abel$ ;
- If  $a \equiv a \vee \perp$  is a clause of  $S_{l_1}, \dots, S_{l_k}$ , then  $a_0 \Rightarrow \perp$  and  $\perp \Rightarrow a_1$  are edges in  $\mathbb{C}lndr$ -graph both with labels  $l_1 \dots l_k$ ;
- If  $a$  is a literal of some  $S_l$  and  $a \in \mathbb{L}abel$ , then  $\neg a_0 \Rightarrow \top$  and  $\top \Rightarrow \neg a_1$  are edges of  $\mathbb{C}lndr$ -graph with label  $a$ ;

- For all conjugated pair,  $a$  and  $\neg a$  in some  $S_l$   $a_1 \rightarrow \neg a_0$  and  $\neg a_1 \rightarrow a_0$ .

cylindrical digraphs can be seen in Examples 7.1 and 7.6, Part 2.

**Definition 3.6** Let  $a_0$  and  $b_1$  be two vertexes of the cylindrical digraph.

1. An alternated sequence from  $a_0$  to  $b_1$  is a subdigraph of the form

$$a_0 \Rightarrow c_1^1 \rightarrow \neg c_0^1 \dots c_1^k \rightarrow \neg c_0^k \Rightarrow b_1$$

2. An interval between  $a_0$  and  $b_1$  is the (non empty) subdigraph that contains all alternated sequences from  $a_0$  to  $b_1$ . Denote the interval  $[a_0, b_1]$ ;
3. Given an interval  $[a_0, b_1]$  a sequence in the interval  $[a_0, b_1]$  is any alternated sequence with starting point  $a_0$  and ending  $b_1$ .

Notice: If there is no alternated sequence between  $a_0$  and  $b_1$ , we define that there is no interval between the vertexes.

The search for intervals is the part of our algorithm of search for satisfiability in a **3 – sat** formula. In short, this search is performed as follows: Let  $C = \{V, \Rightarrow, \rightarrow\}$  be a subdigraph of  $\mathcal{C}lndr$ . Enumerate the vertexes and edges of  $C$ .

First, select polynomially the subdigraph  $C_{a_0}$  of all alternated sequences of the form  $a_0 \xRightarrow{l_1} c_1^1 \rightarrow \neg c_0^1 \xRightarrow{l_2} c_1^2 \rightarrow \neg c_0^2 \dots c_1^m \rightarrow \neg c_0^m \xRightarrow{l_m} \dots$ .

Still polynomially, write the digraph that contains all alternated sequences in  $C_{a_0}$  ending at  $b_1$ ,  $\dots \xRightarrow{l_1} c_1^1 \rightarrow \neg c_0^1 \xRightarrow{l_2} c_1^2 \rightarrow \neg c_0^2 \dots c_1^m \rightarrow \neg c_0^m \xRightarrow{l_m} b_1$ .

As a result, we have the set of all alternated sequences in between  $a_0$  and  $b_1$ , that is, the digraph  $[a_0, b_1]$ .

We show how polynomially write the subdigraph  $C_{a_0}$ . Writing  $C_{b_1}$  is a similar procedure. Let us describe the algorithm to obtain  $C_{a_0}$ .

**Algorithm 3.7 (Writing an Interval)** *Input:*  $(L_{01}, \Rightarrow, \rightarrow)$  (the cylindrical digraph)

$$\begin{aligned} L_1 &= \emptyset \\ \Rightarrow_{a_0} &= \{a_0 \xRightarrow{l} b_1 \in I\} \\ \overline{Set}_0 &= \{a_0\} \\ Set_0 &= \emptyset \\ Set_1 &= \{b_1 \in V | (a_0 \xRightarrow{l} b_1)\} \\ \overline{Set}_1 &= \emptyset \\ \rightarrow_{a_0} &= \emptyset \end{aligned}$$

While  $L_{01} \neq \emptyset$ , do

$L_1 \leftarrow L_1 \cup Set_1$ $Set_0 \leftarrow \{\neg b_0 \in V \mid b_1 \in Set_1\}$ $\rightarrow_{a_0} \leftarrow \rightarrow_{a_0} \cup \{b_1 \rightarrow \neg b_0 \mid b_1 \in Set_1\}$ $\overline{Set_1} \leftarrow \{d_1 \in L_{01} \mid \exists c_0 \in Set_0 (c_0 \xrightarrow{l_i} d_1)\}$ $\overline{Set_0} \leftarrow \overline{Set_0} \cup Set_0$ $\overline{Set_1} \leftarrow \overline{Set_1} \cup Set_1$ $\Rightarrow_{a_0} \leftarrow \Rightarrow_{a_0} \cup \{c_0 \xrightarrow{l_i} d_1 \mid \exists c_0 \in Set_0 (c_0 \xrightarrow{l_i} d_1)\}$ <p style="margin: 0;"><i>For all <math>c_0</math> and <math>d_1</math> if <math>(c_0 \Rightarrow d_1 \in L_{01}) \wedge (c_0 \notin \overline{Set_0})</math>,</i></p> $\Rightarrow \leftarrow (\Rightarrow \setminus \{c_0 \Rightarrow d_1\})$ <p style="margin: 0;"><i>% We are, in the previous line, pruning the loops</i></p> $\Rightarrow \leftarrow \{(d_0 \xrightarrow{l_i} e_1)_i\}_{i \in I} \setminus \{\Rightarrow_{a_0}\}$ $L_{01} \leftarrow L_{01} \setminus (\overline{Set_0} \cup \overline{Set_1})$
--

Define  $C_{a_0} = \{\overline{Set_0} \cup \overline{Set_1}, \Rightarrow_{a_0}, \rightarrow_{a_0}\}$ .

If  $b_1 \in C_{a_0}$ , there is an interval between  $a_0$  and  $b_1$  and we proceed with a similar algorithm to search all sequences ending in  $b_1$  contained in the digraph  $C_{a_0}$ , else, the algorithm for writing the interval  $[a_0, b_1]$  ends with the output **there is no interval between  $a_0$  and  $b_1$** .

Observe: A loop in an interval  $[a_0, b_1]$  is an alternated sequence

$$c_0 \Rightarrow d_1^1 \rightarrow \neg d_0^1 \dots d_1^k \rightarrow \neg d_0^k \Rightarrow \neg c_1$$

If  $d_1^1$  belongs to the  $C_{a_0}$  and  $c_0$  does not, the simple act of erasing the edge  $c_0 \Rightarrow d_1^1$  from the set of edges prevents us of writing the loop. See Lemma 4.10 about cutting loops.

**Definition 3.8** A literal  $r \in \mathbb{L}abel \cap (\cup Letter(\Psi))$  is called necessarily false literal. The set of necessarily false literals is denoted by *NecFls*.

If an interval  $[a_0, a_1]$  is non-empty digraph, then  $a$  is called a necessarily true literal. The set of necessarily true literals is denoted by *Nec*.

Notice that if  $b \in NecFls$ , there exists an interval  $[-b_0, \top]$  generated by the alternated sequence  $\neg b_0 \Rightarrow \top$  whose only label is  $b$ .

**Definition 3.9** Given a cylindrical digraph, its set of closed digraphs, *CSD* is defined as

- 1  $[a_0, a_1] \rightarrow [\neg a_0, \neg b_1] \rightarrow [b_0, b_1]$
- 2  $[a_0, a_1] \rightarrow [\neg a_0, c_1] \rightarrow [\neg c_0, \top]$
- 3  $[\top, \neg d_1] \rightarrow [d_0, c_1] \rightarrow [\neg c_0, \top]$

where 1 to 3 denotes the smallest subdigraph of the cylindrical digraph that contains, respectively,

- 1  $[a_0, a_1]$ ,  $[\neg a_0, \neg b_1]$  and  $[b_0, b_1]$ , for all  $a, b \in \mathbf{Nec}$ ;
- 2  $[a_0, a_1]$ ,  $[\neg a_0, c_1]$  and  $[\neg c_0, \top]$ , for all  $a \in \mathbf{Nec}$ ,  $c \in \mathbf{NecFls}$ ;
- 3  $[\neg d_0, \top]$ ,  $[d_0, c_1]$  and  $[\neg c_0, \top]$ , for all  $d, c \in \mathbf{NecFls}$ .

In particular if  $a \in \mathbf{Nec} \cap \mathbf{NecFls}$ , in case 2, denote the intervals

$$0 \ [a_0, a_1] \rightarrow [\neg a_0, \top]$$

We spell out all the intervals in a cylindrical digraph in Example 7.6, Part 4.

In order to illustrate our reasoning, consider the above formula  $\phi$

$$(a \vee r) \wedge (\neg r \vee a) \wedge (\neg a \vee \neg b) \wedge (b \vee s) \wedge (\neg s \vee b) \wedge (\neg a \vee c) \wedge (d \vee c)$$

In a tableau for a formula that has  $T \phi$  as a node, we develop the branches,

$$\begin{array}{l} T \ a \vee r \quad \mathbf{r} \\ T \ \neg r \vee a \quad \mathbf{s} \\ T \ \neg a \vee \neg b \quad \mathbf{t} \\ T \ b \vee s \quad \mathbf{u} \\ T \ \neg s \vee b \quad \mathbf{w} \\ T \ \neg a \vee c \quad \mathbf{v} \\ T \ d \vee c \quad \mathbf{x} \end{array}$$

If the literals of  $\phi$  form the edges of a cylindrical digraph for  $\Psi$  and, moreover, if we suppose that  $c, d \in \mathbf{NecFls}$ , then, clearly  $a, b \in \mathbf{Nec}$ , since both intervals  $[a_0, b_1]$  and  $[b_0, b_1]$  contain the sequences

$$\begin{array}{l} a_0 \Rightarrow r_1 \rightarrow \neg r_0 \Rightarrow a_1 \\ a_0 \Rightarrow \neg r_1 \rightarrow r_0 \Rightarrow a_1 \\ b_0 \Rightarrow s_1 \rightarrow \neg s_0 \Rightarrow s_1 \\ b_0 \Rightarrow \neg s_1 \rightarrow s_0 \Rightarrow s_1 \end{array}$$

The literal  $a$  causes a branching,  $Fa$  or  $F\neg a$ , in the Combinatorics Choice Tableaux. Using  $\mathbf{r}$  and  $\mathbf{s}$ , we have that the branch  $F a$  closes. An analogous reasoning as used in  $\mathbf{r}$  and  $\mathbf{s}$  applied in  $\mathbf{u}$  and  $\mathbf{w}$  implies that branches with  $F b$  closes. Thus node  $\mathbf{t}$  together with the hypothesis  $a, b \in \mathbf{Nec}$  (nodes  $\mathbf{r}$ ,  $\mathbf{s}$ ,  $\mathbf{u}$  and  $\mathbf{w}$ ) are closed.

If  $a \in \mathbf{Nec}$  (nodes  $\mathbf{r}$  and  $\mathbf{s}$ ) and  $c \in \mathbf{NecFls}$ , the node  $F c$  and node  $\mathbf{v}$  give us closed branches under the hypothesis  $F a$  and  $F c$ .

If  $c, d \in \mathbf{NecFls}$ , branches that contains  $F c$  and  $F d$  and node  $\mathbf{x}$  are closed.

In a broader sense, we search for closed branches in the Combinatorics Choice Tableaux that entails similar contradictions.

Observe that there is a symmetric pattern over cylindrical digraphs and intervals. If there is a sequence

$$a_0 \Rightarrow b_1^1 \rightarrow \neg b_0^1 \Rightarrow b_1^2 \rightarrow \neg b_0^2 \dots b_1^k \rightarrow \neg b_0^k \Rightarrow b_1$$

there is a symmetric alternated sequence

$$b_0 \Rightarrow \neg b_1^k \rightarrow b_0^k \dots b_0^2 \rightarrow \neg b_1^2 \Rightarrow \neg b_1^1 \rightarrow b_0^1 \Rightarrow a_1$$

this will not spoil our, still polysize, counting. The symmetric alternated sequences result only on writing lines twice, an armless action that won't affect complexity.

## 4 Algorithm I - Closed Branches in the Combinatorics Choice Tableau

In this section, we define the *macrotableau* generated by the cylindrical digraph. The macrotableau is the storing place, written in a very economic way, of all closed branches for a tableau for a given **3-sat** formula  $\Psi$ .

**Definition 4.1** *Given an alternated sequence (in CSD),*

$$S = a_0 \xRightarrow{\mathbb{C}_1} b_1^1 \rightarrow \neg b_0^1 \xRightarrow{\mathbb{C}_2} b_1^2 \rightarrow \neg b_0^2 \dots b_1^k \rightarrow \neg b_0^k \xRightarrow{\mathbb{C}_k} c_1$$

*call a linear macrobranch generated by the above sequence to all the branches in the Combinatorics Choice Tableau formed by all choices*

$$\begin{array}{l} Fl_1 \\ \vdots \\ Fl_k \end{array}$$

*where  $l_i$  is a choice of a literal of  $\mathbb{L}abel$  in  $\mathbb{C}_r$ ,  $1 \leq r \leq k$  in a non contradictory way (no two chosen literals are conjugated) or, if no choice is possible, we have the empty branch (see Observation 5.15).*

Denote linear macrobranches as  
 $FC_1$   
 $\vdots$   
 $FC_k$   
and call each  $C_i$  a macronode.

Built macrobranches by using the operations  $\wedge$  and  $\vee$  conjunction and disjunction, below defined.

**Definition 4.2** *Macrobranches are inductively defined as,*

1. *A macronode is a macrobranch;*
2. *Given two macrobranches  $MBr_1$  and  $MBr_2$ , their conjunction  $MBr_1 \wedge MBr_2$  is the set of all branches given by the conjunctions of the branches of  $MBr_1$  with the branches of  $MBr_2$ ;*
3. *Given two macrobranches  $MBr_1$  and  $MBr_2$ , their disjunction  $MBr_1 \vee MBr_2$  is the set of all branches given by the disjunction of the branches of  $MBr_1$  with the branches of  $MBr_2$ .*

A *macronode* was taken as an atom. Linear macrobranches are, indeed, conjunction of macronodes, although we used linear macrobranches as the basis of our construction.

**Definition 4.3** *The Mirror Tableau of formulas  $\psi$  is the subtableau of the Combinatorics Tableau for  $\psi$  whose branches are generated by disjunction of all linear macrobranches generated by the alternated sequences in  $\mathcal{CSD}$ .*

In other words, modulo repeated modes  $Fl$ , for  $l \in \text{Label}$ , the Mirror Tableau is the subtableau of the Combinatorics Choice Tableau whose branches are generated by all disjunctions of the macrobranches generated by the alternated sequences in  $\mathcal{CSD}$ . We do not explicitly write all the sequences, which can cause an exponential growth of the branches. Rather, we develop strategies, in  $\mathcal{CSD}$  (Section 5) in order to analyze the Mirror Tableau. We close this Section enlightening the role of the Mirror Tableau in the answer of our question.

**Definition 4.4** *Given a macrobranch  $MBr$ , define its transposed,  $MBr^t$  as*

1. The transposed of a macronode  $F r_1 r_2 \dots r_n$  is the linear macrobranch
 
$$\begin{array}{l} F \neg r_1 \\ F \neg r_2 \\ \vdots \\ F \neg r_n \end{array}$$
2. The transposed the a macronode containing  $S_\top$  is the (closed) macronode  $\perp$ ;
3.  $(\text{MBr}_1 \vee \text{MBr}_2)^t = \text{MBr}_1^t \wedge \text{MBr}_2^t$  and  $(\text{MBr}_1 \wedge \text{MBr}_2)^t = \text{MBr}_1^t \vee \text{MBr}_2^t$ .

We consider the tableau generated by the disjunction of all branches generated by the alternated sequences in a given  $\mathcal{CSD}$ , the Mirror Tableau. The set of all (maximal) alternated sequences in the  $\mathcal{CSD}$  encodes all closed branches in the Choice Tableau, so all unsatisfiable sequences. We do not open the  $\mathcal{CSD}$  and write all possible alternated sequences, an expsize task.

**Definition 4.5** An antichain in a macrobranch  $\text{MBr}$ ,  $a$  is a set of macronodes  $\text{AntCh} = \{Nd_1, \dots, Nd_n\}$  so that

1. If  $\text{MBr}$  is a linear macrobranch,  $\text{AntCh}$  is a unitary set and its only element is a macronode of  $\text{MBr}$ ;
2. If  $\text{MBr} = \bigvee_{1 \leq j \leq m} \text{MBr}_j$ , then for all  $1 \leq i \leq m$ , there is  $\text{Anti}_i$ , a non-empty subset of  $\{Nd_1, \dots, Nd_n\}$  so that  $\text{Anti}_i$  is an antichain of  $\text{MBr}_i$ ;
3. If  $\text{MBr} = \bigwedge_{1 \leq j \leq m} \text{MBr}_j$ , there is one  $1 \leq j \leq m$  so that  $\{Nd_1, \dots, Nd_n\}$  is an antichain of  $\text{MBr}_j$ .

Given an antichain  $\{Nd_1, \dots, Nd_n\}$ ,  $\{Nd_1^t, \dots, Nd_n^t\}$  is called a chain.

As an example, if  $\text{MBr}_1, \text{MBr}_2, \text{MBr}_3, \text{MBr}_4, \text{MBr}_5$  are linear macrobranches and  $\text{MBr} = \text{MBr}_1 \vee ((\text{MBr}_2 \wedge \text{MBr}_3) \wedge (\text{MBr}_4 \vee \text{MBr}_5))$ , the chains in  $\text{MBr}$  are contained either in  $\{\text{MBr}_1\}$ ,  $\{\text{MBr}_2, \text{MBr}_3, \text{MBr}_4\}$  or  $\{\text{MBr}_2, \text{MBr}_3, \text{MBr}_5\}$ . The antichains are contained in  $\{\text{MBr}_1, \text{MBr}_2\}$ ,  $\{\text{MBr}_1, \text{MBr}_3\}$  and  $\{\text{MBr}_1, \text{MBr}_4, \text{MBr}_5\}$ .

**Lemma 4.6** Given an antichain  $\{Nd_1, \dots, Nd_m\}$  in the Mirror Tableau,  $\overline{\mathcal{CSD}}$ ,  $\bigwedge_{1 \leq i \leq m} Nd_i^t$  is a macrobranch in a  $\mathcal{CSD}^t$ . Reversely, if the conjunction of  $\{Nd_1, \dots, Nd_m\}$  forms a macrobranch in  $\overline{\mathcal{CSD}}$ , then  $\{Nd_1^t, \dots, Nd_m^t\}$  is a antichain in  $\mathcal{CSD}^t$ .

**Definition 4.7** Call closed the linear macrobranch

$$\begin{array}{l}
 F\mathbb{C}_1 \\
 \vdots \\
 F\mathbb{C}_k \\
 \text{so that} \\
 F r_1 \\
 \vdots \\
 F r_n \\
 T S_{r_1} \\
 \vdots \\
 T S_{r_n}
 \end{array}$$

is a closed subbranch of the choice tableau for all choices of literals  $r_i$  in the label  $\mathbb{C}_i$ ,  $1 \leq i \leq n$ .

An interval is closed iff the disjunction of all macrobranches generated by the all the maximal alternated sequence in it are closed.

The set of closed branches in the Combinatorics Choice Tableau is the set of branches in the Mirror Tableau  $\overline{\mathcal{CSD}}$ .

**Theorem 4.8** The macrobranches in a  $\overline{\mathcal{CSD}}$  are closed.

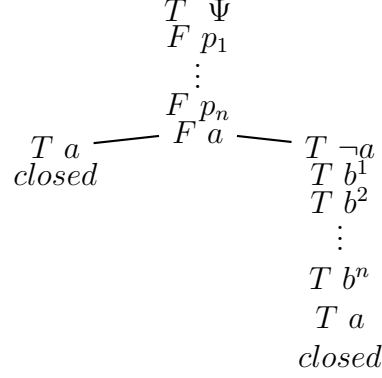
**Proof:** Case 0) The interval  $I \in \mathcal{CSD}$  is generated by a literal  $a$  that is both necessarily false and necessarily true. Thus, maximal alternated sequences in  $I$  have the form

$$a_0 \xrightarrow{l_1} b_1^1 \rightarrow \neg b_0^1 \dots \neg b_0^n \xrightarrow{l_n} a_1 \rightarrow \neg a_0 \xrightarrow{a} \top$$

and its associated linear macrobranch,

$$\begin{array}{l}
 F l_1 \\
 \vdots \\
 F l_n \\
 F a
 \end{array}$$

Let  $p_i \in l_i$ ,  $1 \leq i \leq n$ . Develop the below branch as



Hence, the interval  $[a_0, a_1] \rightarrow [\neg a_0, \top]$  is closed.

Case 1) The interval  $I$  is generated by  $a^i$  and  $a^j$  that are necessarily true. Thus, the intervals  $[a_0^i, a_1^i]$  and  $[a_0^j, a_1^j]$  generate two macrobranches,  $\text{MBr}_{a^i}$  and  $\text{MBr}_{a^j}$  that close, respectively, under  $F a^i$  and  $F a^j$ .

If  $a^i$  and  $a^j$  are conjugated literals, our claim follows. Else, the branches generated by the interval  $[\neg a_0^i, \neg a_1^j]$  close under  $F \neg a_0^i$  and  $F \neg a_1^j$ . In fact, any possible branch of  $[\neg a_0^i, \neg a_1^j]$  is of the form

$$\begin{array}{l}
 T \neg a^i \vee d^1 \\
 T \neg d^1 \vee d^2 \\
 \vdots \\
 T \neg d^n \vee \neg a^j
 \end{array}$$

that closes under the hypothesis  $F \neg a^i$  and  $F \neg a^j$ .

So,  $\text{MBr}$  closes under the branching  $F a^i$ ,  $F a^j$  and  $F(\neg a^i \wedge \neg a^j)$ .

Case 2) The interval  $I$  has two distinct necessarily true and necessarily false literals  $a$  and  $c$ . The macrobranches has as a macronode  $F c$ . The macrobranch generated by  $[a_0, a_1]$  closes with the addition of the macronode  $F a$ . The macrobranch generated by  $[\neg a_0, c_1]$  closes added to the macronodes  $F \neg a$ . Hence,

$$[a_0, a_1]a_1 \rightarrow \neg a_0[\neg a_0, c_1]b_1 \rightarrow \neg c_0[\neg c_0, \top]$$

is closed.

Case 3) Consider the interval  $I = [\top, \neg d_1] \rightarrow [d_0, c_1] \rightarrow [\neg c_0, \top]$ ,  $c, d$  necessarily false. Alternated sequences in  $I$  have the form,

$$\neg d_1 \xrightarrow{d} d_0 \xrightarrow{Nd_1} e_1^1 \rightarrow \neg e_1^1 \dots \neg e_0^n \xrightarrow{Nd_n} c_1 \xrightarrow{c} \neg c_0$$

whose macrobranch close due to the macronodes

$$\begin{array}{l}
T \ d \vee e^1 \\
T \ \neg e^1 \vee e^2 \\
\vdots \\
T \ \neg e^n \vee c
\end{array}
\quad \dashv$$

**Theorem 4.9** *Any closed branch MBr in the combinatorial choice tableau belongs to the mirror tableau.*

**Proof:** We must show that if a branch  $Br$  in the Combinatorics Choice Tableau has not a subbranch of the Mirror Tableau, then  $Br$  is open.

Let  $Br$  be

$$\begin{array}{l}
F \ l_1 \\
\vdots \\
F \ l_t
\end{array}$$

$l_i \in \mathbb{L}abel$ ,  $1 \leq i \leq t$ . By hypothesis, for any subset of  $A$  of  $\{l_1, \dots, l_t\}$ , there is no alternated sequence in  $\mathcal{CSD}$ ,

$$a_0 \xrightarrow{Nd_1} b_1^1 \rightarrow \neg b_0^1 \dots \neg b_0^n \xrightarrow{Nd_r} c_1$$

with each  $Nd_i$ ,  $1 \leq i \leq t$  containing one literal of  $A$ . Thus, there is no branches

$$\begin{array}{l}
T \ a \vee b^1 \\
T \ \neg b^1 \vee b^2 \\
\vdots \\
T \ \neg b^n \vee c
\end{array}$$

with  $a$  and  $c$  meeting the requirements of Definition 4.7, which leaves some of the possible branches open.  $\dashv$

**Lemma 4.10** *Cutting loops (see Algorithm3.7) does not change the set of closed branches in the Mirror Tableau.*

**Proof:** Observe that if a branch

$$\begin{array}{l}
F \ l_1 \\
\vdots \\
F \ l_t
\end{array}$$

closes, then

$$\begin{array}{l}
F \ l_1 \\
\vdots \\
F \ l_t \\
F \ k_1 \\
\vdots \\
F \ k_r
\end{array}$$

closes for all consistent choice  $\{l_1, \dots, l_t, k_1, \dots, k_r\}$  in the set of labels,  $\mathbb{L}abel$ .  
 $\dashv$

**Definition 4.11**  $\mathcal{CSD}$  represents the Combinatorics Choice Tableau *iff all branches in the Combinatorics Choice Tableau (the branches whose nodes are labeled as **False** for a literal in  $\mathbb{L}abel$ ) have as subbranch a branch in the Mirror Tableau.*

## 5 Algorithm II - Polynomial Decision

We study, in this Section, the conditions to check whether  $\mathcal{CSD}$  represents the Combinatorics Choice Tableau and therefore,  $\Psi$  is non satisfiable. We give a decision procedure in order to decide if  $\mathcal{CSD}$  represents the Combinatorics Choice Tableau.

We work with a non empty  $\mathcal{CSD}$  for otherwise the  $\mathcal{CSD}$  has the obvious answer that all branches in the combinatorial tableau are open.

**Definition 5.1** A set of macronodes,  $\mathbf{C} = \{Nd_1, \dots, Nd_n\}$  is incompatible if there is a pair  $l$  and  $\neg l$  in  $\mathbf{CS}$  so that for some  $i$  and  $j$  (not necessarily  $i \neq j$ )  $l$  belongs to  $Nd_i$  and  $\neg l$  belongs to  $Nd_j$ . If otherwise, we say that  $\mathbf{C}$  is compatible and denote  $comp(Nd_1, \dots, Nd_n)$ .

Observe that the macronode  $S_{\top}$  is incompatible with any other macronode and plays the following rules on our game,

1. If  $Nd = l_1 l_2 \dots l_n$  is a macronode and  $S_{\top}$  is one of its string then  $Nd$  can be replaced by  $Nd' = S_{\top}$ , called a *universal macronode*;
2. The universal macronode can be erased in any column of a linear macrobranch that contains more than universal macronodes in its set of labels;

3. A column of a linear macrobranch that contains only universal macronodes is incompatible with all other columns.

A macronode in  $S_{\top}$  is equivalent a one macronode where all literals label occurs and it is incompatible with any macronode.

**Proposition 5.2** *For any two branches  $\mathbf{MBr}$  and  $\mathbf{MBr}'$ , in the Mirror Tableau,  $\overline{\mathcal{CSD}}$  and in its transposed, respectively,  $\overline{\mathcal{CSD}}^t$ , there is a pair of conjugated literals  $l$  and  $\neg l$  so that  $l$  belongs to  $\mathbf{MBr}$  and  $\neg l$  belongs to  $\mathbf{MBr}'$ .*

**Proof:** Inductively over the structure of the macrotableau. If  $\mathbf{MBr}$  has the form

$$\begin{array}{l} F Nd_1 \\ \vdots \end{array}$$

$$F Nd_k$$

where for each  $1 \leq i \leq k$ ,  $Nd_i = \{p_{1i}, \dots, p_{m_i i}\}$ , then  $\mathbf{MBr}^t$  is of the form  $\bigvee_{1 \leq i \leq k} Nd_i^t$ , where each  $Nd_i^t$ , is

$$\begin{array}{l} F \neg p_{1i} \\ \vdots \end{array}$$

$$F \neg p_{m_i i}$$

A branch in  $\mathbf{MBr}^t$  is a macrobranch  $Nd_i^t$ , for some  $1 \leq i \leq k$  and a branch in  $\mathbf{MBr}$  is

$$\begin{array}{l} F p_{j_1 1} \\ \vdots \end{array}$$

$$F p_{j_k k}$$

where each  $p_{j_r r}$  is a choice in  $Nd_r$ ,  $1 \leq r \leq k$ . Thus, for  $r = i$  the literals  $p_{j_i i}$  and  $\neg p_{j_i i}$  satisfy our claim.

If  $\mathbf{MBr}$  is a conjunction,  $\mathbf{MBr}_1 \wedge \dots \wedge \mathbf{MBr}_n$ , a branch in  $\mathbf{MBr}$  is the conjunction of the branches in *each*  $Br_i$ ,  $1 \leq i \leq n$ . A branch in  $\mathbf{MBr}^t$ , is a branch in *some*  $Br_i^t$ .

If  $\mathbf{MBr}$  is a disjunction  $Br_1 \vee \dots \vee Br_n$ , a branch in  $\mathbf{MBr}$  is a branch in *one*  $\mathbf{MBr}_i$ ,  $1 \leq i \leq n$ . A branch in  $\mathbf{MBr}^t$  is a the conjunctions of all branches in *any*  $Br_i^t$ . In both cases, our claim follows by reduction over the complexity of  $\mathbf{MBr}$ .  $\dashv$

A very important piece of our construction is the following theorem that allow us to decide if the formula  $\Psi$  is valid searching its set of closed digraphs.

**Theorem 5.3** *A given  $\mathcal{CSD}$  represents the mirror tableau iff there is no compatible antichain in  $\overline{\mathcal{CSD}}$ .*

**Proof:** Recall that the conjunction of transposed macronodes of an antichain in  $\overline{\mathcal{CSD}}$  is a branch in  $\mathcal{CSD}^t$ . Besides, observe that the conjunction of branches generated by  $\mathcal{CSD}$  and  $\mathcal{CSD}^t$  is the Combinatorics Choice Tableau.

There is no compatible antichain in the Mirror Tableau iff any branch generated by  $\mathcal{CSD}^t$  contains a pair of conjugated literals. So, by Proposition 5.2, any combination that contains no conjugated literals belongs to the Mirror Tableau.  $\dashv$

We will give the algorithm to decide if a given  $\mathcal{CSD}$  has antichains or no. Our algorithm is based on a partial order naturally given over the (maximal) intervals in  $\mathcal{CSD}$ .

**Definition 5.4** Given a  $\mathcal{CSD} = (V, E, \Rightarrow, \rightarrow)$  its sets of source, spillway and roots,  $\mathbb{F}$ ,  $\mathbb{S}$  and  $\mathbb{R}$  is given, respectively, by

$$\begin{aligned}\mathbb{F} &= \{a_0 \mid (\exists c_1 \in V \exists c'_1 \in V (c_1 \neq c'_1) \wedge (a_0 \Rightarrow c_1 \in E) \wedge (a_0 \Rightarrow c'_1 \in E)) \\ &\quad \vee (\neg a_1 \rightarrow a_0 \notin E)\} \\ \mathbb{S} &= \{b_1 \mid (\exists c_0 \in V \exists c'_0 \in V (c_0 \neq c'_0) \wedge (c_1 \rightarrow b_0 \in E) \wedge (c'_1 \rightarrow b_0 \in E)) \\ &\quad \vee (b_1 \rightarrow \neg b_0 \notin E)\} \\ \mathbb{R} &= \{a_0 \in \mathbb{F} \mid \neg a_1 \rightarrow a_0 \notin E\} \subseteq \mathbb{F}\end{aligned}$$

**Definition 5.5** Call an interval  $[a_0, b_1]$  contained in  $\mathcal{CSD}$  a maximal tabular array, MAXLIN, if  $b_1$  is a spillway or  $\neg b_0$  is a source,  $a_0$  is a source or  $\neg a_1$  a spillway, and it has  $j$  alternated sequences of the form

$$S^i = a_0 \Rightarrow c_1^{i1} \rightarrow \neg c_0^{i1} \Rightarrow c_1^{i2} \rightarrow \neg c_0^{i2} \dots c_1^{ik} \rightarrow \neg c_0^{ik} \Rightarrow b_1$$

$1 \leq i \leq j$ , where  $\neg c_0^{is} \notin \mathbb{F}$  or  $c_1^{is} \notin \mathbb{S}$ .

**Definition 5.6** A maximal tabular array MAXLIN generates a macrobranch LINMCR (contained in the Mirror Tableau) of the form  $\bigvee_{1 \leq i \leq j} L_i$ , denoted as

$$\begin{array}{cccc} Nd_1^1 & Nd_1^2 & & Nd_1^j \\ \vdots & \vdots & & \vdots \\ Nd_{k_1}^1 & Nd_{k_2}^2 & \dots & Nd_{k_j}^j \end{array}$$

We give next the algorithm that search and give a partial order to the intervals MAXLIN in a given  $\mathcal{CSD}$ .

**Notation 5.7** Each alternated sequence in a closed digraph  $\mathcal{CSD}$  is associated to a Capital letter called its alternated sequence label.

**Definition 5.8** Define the ordered set of labels,  $\mathbb{U}$ , as  $r \in \mathbb{U}$  if

1.  $r$  is a alternated sequence label;
2.  $\{A_1, \dots, A_j\} \subseteq \mathbb{U}$ ,  $B$  is an alternated sequence label,  $r = (A_1 \bullet \dots \bullet A_j)B$ .

The set  $\mathbb{U}$  is partially ordered with the order given by the closure of the relation  $r < s$  if

$$(s = (A_1 \bullet \dots \bullet A_j)B) \wedge (\exists 1 \leq i \leq j (r = A_i))$$

Each  $A_i$  is called a sequent of  $s$  and if  $C$  is a sequent of some  $A_i$ , we define that  $C$  is a sequent of  $s$ .

**Definition 5.9** The ordered set of labels associated to a closed digraph is the subset of  $\mathbb{U}$  so that,

1. If  $\mathbf{seq}$  is a sequence between  $a_0$  and  $b_1$ ,  $a_0$  a root, its ordered label is its alternated sequence label;
2. If  $a_0$  is a source or  $\neg a_1$  is a spillway, the ordered labels of the alternated sequences that converge to  $\neg a_1$  are  $A_1, \dots, A_j$ ,  $j \geq 1$  and each sequence from  $a_0$  has alternated sequence label  $B_1, \dots, B_i$ ,  $i \geq 1$ , then the sequences from  $a_0$  have ordered label, respectively,  $(A_1 \bullet \dots \bullet A_j)B_1, \dots, (A_1 \bullet \dots \bullet A_j)B_k$ .

An algorithm to write ordered and alternated sequence labels must do:

**Algorithm 5.10** Let  $E$  be a set of edges and  $V$  a set of vertexes in a CSD.

1. Search for source, spillway and roots;
2. Mark the set of alternated sequences

$$S = a_0 \Rightarrow c_1^1 \rightarrow \neg c_0^1 \Rightarrow c_1^2 \rightarrow \neg c_0^2 \dots c_1^k \rightarrow \neg c_0^k \Rightarrow b_1$$

so that  $b_1$  is a spillway or  $\neg b_0$  is a source,  $a_0$  is a source or  $\neg a_1$  is a spillway,  $\neg c_0^s \notin \mathbb{F}$  and  $c_1^s \notin \mathbb{S}$  (see Definition 5.5) with Capital letter

3. Interactively, if the label of a sequence  $\mathbf{seq}$  from  $a_0$  to  $b_1$  is  $B$ , the alternated sequences that end at  $\neg a_1$  have labels  $A_1, \dots, A_j$ , then the label associated to  $\mathbf{seq}$  is  $(A_1 \bullet \dots \bullet A_j)B_1$ . If  $a_0$  is a root, the label of  $\mathbf{seq}$  is  $B_1$ .

See Example 7.4 for ordered set of labels associated to a closed digraph.

The alternated sequences in an interval in MAXLIN are ordered by the order inherited from the order over  $\mathbb{U}$ . The maximal tabular arrays associated to each interval have their linear macrobranches ordered as well. Moreover, two linear macrobranches  $\text{LINMCR}_1$  and  $\text{LINMCR}_2$  associated to intervals of the form  $[a_0, b_1]$  and  $[-b_0, c_1]$  have linear macrobranches  $\{Br_1, \dots, Br_i\}$  and  $\{Br'_1, \dots, Br'_j\}$  form disjunction and conjunction of macrobranches as follows:

$$\left( \bigvee_{1 \leq l \leq i} Br_l \right) \bigwedge \left( \bigvee_{1 \leq kl \leq j} Br'_k \right)$$

**Definition 5.11** *Define the order over the set of labels in a CSD by naming a node  $(r, i)$ , where  $r \in \mathbb{U}$  and  $i$  is a numeration of the place the node occupies in the column,  $1 \leq i \leq k_i$ . Let by  $\prec$  the lexicographic order over the Cartesian product of  $\mathbb{U}$  and the natural numbers.*

The labels in a closed digraph generate the nodes  $Nd_s$  in a Mirror Tableau. We will define a set of nodes called **Nest** in such way that for each node  $Nd_r$  in the column  $r$ , its associated set **Nest** $(Nd_r)$ . The dual of **Nest** $(Nd_r)$  is **setarray** $(Nd_r)$  in the sense that

$$\begin{aligned} \mathbf{Nest}(Nd_r) &= \{Nd_s \mid Nd_r \in \mathbf{setarray}(Nd_s)\} \\ \mathbf{setarray}(Nd_r) &= \{Nd_s \mid Nd_r \in \mathbf{Nest}(Nd_s)\} \end{aligned}$$

**Definition 5.12 (Complete)** *A set of nodes  $\mathbf{S}$  is complete if it contains an antichain.*

Recall the definition of antichain in the poset  $\mathbb{U}$ , given below.

**Definition 5.13** *A subset  $V$  of  $\mathbb{U}$  contains an antichain if for all  $r \in \mathbb{U}$ , there is  $s \in V$  so that  $r$  and  $s$  are comparable (either  $r < s$  or  $s < r$ ).*

Both definitions, antichain over nodes and in the poset  $\mathbb{U}$  match, as we see below.

**Proposition 5.14** *A set of nodes  $Nds = \{Nd_{(r_1, i_1)}, \dots, Nd_{(r_k, i_k)}\}$  contains an antichain (see Definition 4.5) iff  $\{r_i \mid \exists Nd_{(r_1, i_1)} \in Nds \mid (r_1, i_1) \text{ is a subindex of } Nd_{(r_1, i_1)}\}$  contains an antichain (see Definition 5.13).*

**Proof:** Case tabular array, we have the the disjunction of a conjunction of macronodes,

$$\begin{array}{ccc} Nd_1^1 & Nd_1^2 & Nd_1^j \\ \vdots & \vdots & \vdots \\ Nd_{k_1}^1 & Nd_{k_2}^2 & \cdots Nd_{k_j}^j \end{array}$$

by definition, an antichain is a choice of a macronode  $Nd_{1t}^t$ ,  $1 \leq t \leq j$  in each  $1 \leq i \leq j$ , thus, the set  $\{1, 2, \dots, j\}$  is an antichain in the partially ordered set  $\{1, 2, \dots, j\}$  endowed with the order  $a \leq b$  iff  $a = b$ .

Else, if  $Lin$  is the set of all linear macrobranches, subindexed over the elements of  $\mathbb{U}$ , the Mirror Tableau (generated by  $\mathcal{CSD}$ ) is

$$\mathbb{I} = \bigvee_{s \subseteq \mathbb{U}} \bigwedge_{i \in s} \{\mathbb{I}_i \in Lin \mid (i \in s) \wedge (s \text{ is a chain in } \mathbb{U})\}$$

Using the definition of antichains, our assertion follows.  $\dashv$

**Observation 5.15** *It is possible that a macrobranch cannot be expressed as a branch of the Mirror Tableau, as we see above,*

$$F p_1$$

$$F p_2$$

$$F \neg p_1 \neg p_2$$

but once we write the  $\mathcal{CSD}$ , develop it as disjunction of macrobranches, partitioned into contradictory, **Contr**, and non contradictory, **Poss**, macrobranches,

$$\bigvee_{i \in \text{Contr}} \{\text{Macrobr}_i\} \vee \bigvee_{i \in \text{Poss}} \{\text{Macrobr}_i\}$$

and its transposed,

$$\bigwedge_{i \in \text{Contr}} \{\text{Macrobr}_i^t\} \wedge \bigwedge_{i \in \text{Poss}} \{\text{Macrobr}_i^t\}$$

which is equivalent to

$$\bigvee_{i \in \text{Poss}} \{\text{Macrobr}_i\}$$

and

$$\bigwedge_{i \in \text{Poss}} \{\text{Macrobr}_i^t\}$$

so, an algorithm that searches for contradictory macrobranches is not necessary.

**Lemma 5.16** *There is a polynomial algorithm to decide whether there is an antichain in a set  $V$  in  $\mathbb{U}$ .*

**Proof:** Comparison between two elements of  $\mathbb{U}$  and  $V$  demands a  $|V| \times |\mathbb{U}| \leq |\mathbb{U}|^2$  operations, where  $|S|$  denotes the size of a set  $S$ .  $\dashv$

Let  $\trianglelefteq$  be an arbitrary linear order over the label of macronodes. The following Algorithm, *Solving a CSD*, is the decision algorithm to decide whether there are antichains in the *CSD*.

**Algorithm 5.17 (SOLVE)** *Consider the intervals in a CSD.*

1. For all  $(r \trianglelefteq s)$  so that  $r$  and  $s$  are not comparable in the order  $<$ ,  $1 \leq i \leq k_r$  and  $1 \leq j \leq k_s$ , add  $Nd_{(r,i)}$  to **setarray** $(Nd_{(s,j)})$  if both are compatible. Recall that  $Nd_{(r,i)}$  is comparable with  $Nd_{(r,i)}$  and therefore,  $Nd_{(r,i)}$  belongs to its setarray and nest;
2. Until no operation can be performed, do
  - (a) Erase all  $Nd_{(s,j)}$  from **setarray** $(Nd_{(r,i)})$  if the intersection

$$\begin{aligned} & (\mathbf{setarray}(Nd_{(r,i)}) \cup \mathbf{Nest}(Nd_{(r,i)})) \cap \\ & (\mathbf{setarray}(Nd_{(s,j)}) \cup \mathbf{Nest}(Nd_{(s,j)})) \end{aligned}$$

*is not complete.*

*If all macronodes erased, stop the computation with the output **There is no compatible antichain**. Else, the computation stops because the set of macronodes was kept unchanged and the output is **There are compatible antichains**. Obtain the output **Solve**, a list with **setarray** of all macronodes.*

Examples 7.4 and 7.5 illustrate the previous algorithm. Notice that in 7.5 we deal with the simplest case, where the closed digraph is an interval MAXLIN.

**Definition 5.18** *If the solution of a CSD (Algorithm 5.17) is non empty, its solution set, **Sol**, is the set of all sets of macronodes  $\{Nd_{(r_1,i_1)}, \dots, Nd_{(r_s,i_s)}\}$  so that each macronode  $Nd_{(r_m,i_m)} \in \mathbf{Nest}(Nd_{(r_n,i_n)})$ . Besides, each set in **Sol** is maximal regarding to subsets, that is, if to set  $I_1$  and  $I_2$  satisfy the condition of **Sol** and  $I_1 \subsetneq I_2$ , then  $I_1 \notin \mathbf{Sol}$ .*

**Theorem 5.19** *The solution set coincides with the set of compatible antichains in a given tabular macrobranch.*

**Proof:** Suppose that  $S = \{Nd_{(i_1, r_1)}, Nd_{(i_2, r_2)}, \dots, Nd_{(i_n, r_n)}\}$  is a set of pairwise compatible macronodes so that  $\{r_1, r_2, \dots, r_n\}$  is an antichain. Clearly  $S$  belongs to the solution set. For all  $(i_k, r_k) < (i_l, r_l)$ ,  $Nd_{(i_k, r_k)} \in \mathbf{Nest}(Nd_{(i_l, r_l)})$ .

Reciprocally, we must show that a non empty output generates only antichains of compatible macronodes. Once we made a choice of compatible (non complete) macronodes, say  $T = \{Nd_{(i_1, r_1)}, \dots, Nd_{(i_n, r_n)}\}$ , if we cannot extend our choice by adding to  $T$  a macronode  $Nd_{(i_{n+1}, r_{n+1})}$  compatible with any node in  $T$ , then all macronodes in  $T$  would be erased.  $\dashv$

Henceforth, we can take for granted that an output  $\emptyset$  means that there is no compatible antichain and conversely. Naming all antichains is an (unnecessary) expsize task, as we see in Example 7.5.

## 6 Bounds in Computation

**Lemma 6.1** *Writing an interval is polynomial in space and time.*

**Proof:** We perform our search over the edges  $E$  which, recursively, is exhausted according we enlarge the set  $S_1$ , which leads, in the worse case, on a progression  $|E| + (|E| - 1) + \dots + 1$ . So, the search is bounded by the square of the elements of the set  $E$  and, thus, by the 4<sup>th</sup> power of the set of literals in  $\Psi$ .

Notice also that the number of interval is bounded by the number of necessarily true plus the number of necessarily false literal, that is, at most the square of the literals of  $\Psi$ .  $\dashv$

**Lemma 6.2** *Writing intervals MAXLIN is a polynomial task.*

**Proof:** Let us analyze the steps of Algorithm 5.10 Given a  $\mathcal{CSD}$ , let  $V$  be its set of edges and  $V$  be its set of vertexes.

1. The search for source and spillway requires a search over the edges, a linear search;
2. The search for the roots is a linear search over the set of sources;
3. Marking each alternated sequence with *alternated sequence label* depends on writing intervals, a polysize task;

4. Writing the *ordered set of labels*,  $\mathbb{U}$ , depends on writing intervals, a polysize operation. Moreover, item 3 of Algorithm 5.10 is an iterative search over the number of alternated sequences, in the worse case, the repetition is bounded by the square of the number of alternated sequences. The number of alternated sequences is bounded by the number of edges in a closed digraph.

–

**Lemma 6.3** *Writing the alternated sequence labels and the ordered set of labels associated to a closed digraph is a polynomial task.*

**Proof:** The number of alternated sequences is bounded by the number of edges in the closed digraph. Writing the ordered labels depends on a search performed on, at most  $n + (n - 1) + \dots + 2 + 1$  over the number of edges. –

**Proposition 6.4** *The search for antichains in a set of macronodes  $V$  is polynomially bounded.*

**Proof:** The search depends on comparison between the set  $\mathbb{U}$  and  $V$ , again bounded by the square of  $|\mathbb{U}|$ . –

**Proposition 6.5** *Algorithm 5.17 is polynomially bounded.*

**Proof:** Perform a polysize search over the Cartesian product of the set of macronodes (compare a macronode with other macronode).

In step 2, we compare sets of the size of the set of macronodes while performing the operations

$$\left( \mathbf{setarray}(Nd_{(r,i)}) \cup \mathbf{Nest}(Nd_{(r,i)}) \right) \cap \left( \mathbf{setarray}(Nd_{(s,j)}) \cup \mathbf{Nest}(Nd_{(s,j)}) \right)$$

In the worse case, we erase macronodes in macrobranches, in a **setarray** or in **Nest**, which can demand a search over space. If  $|Nd|$  is the number of macronodes, the space is  $|Nd|^2$ , the number of macronodes plus its set **setarray**  $\cup$  **Nest**. Thus, erasing macronode by macronode, we spend a number square of the space, thus  $|Nd|^4$ .

As the number of macronodes is bounded by the number edges in each interval in the set of closed digraphs  $\mathcal{CSD}$ . The number of closed digraphs,

given in Definition 3.9 is, at most, the order square of the literals in  $\Psi$  and the number of edges in each interval is bounded by the number of edges in a cylindrical digraph, that is, the square of the number of literals. So, at most  $|\text{Letter}(\Psi)|^2$  edges in each interval, whose number bounded by  $|\text{Letter}(\Psi)|^2$

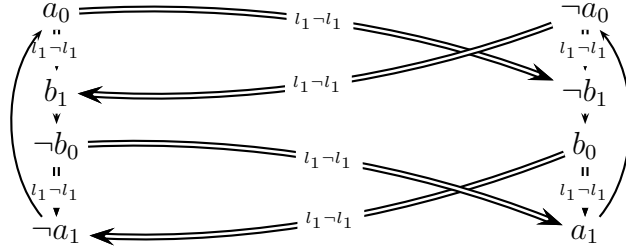
+

## 7 Examples

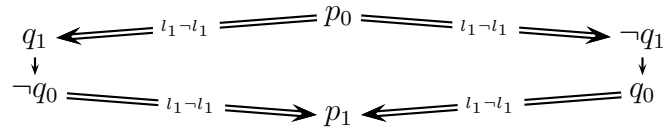
**Example 7.1** Let  $\psi$  be

$$l_1 \wedge ((a \vee b) \wedge (a \vee \neg b) \wedge (\neg a \vee b) \wedge (\neg a \vee \neg b)) \wedge \\ \neg l_1 \wedge ((a \vee b) \wedge (a \vee \neg b) \wedge (\neg a \vee b) \wedge (\neg a \vee \neg b))$$

its corresponding cylindrical digraph is depicted in the above figure

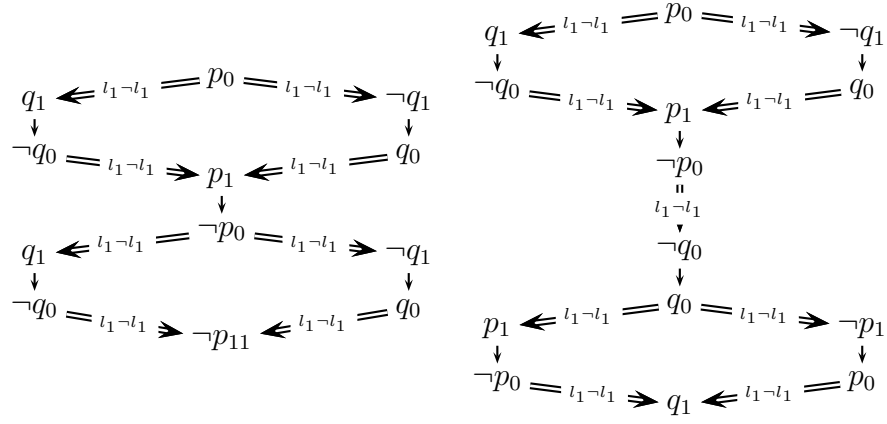


The set of necessarily true literals  $Nec$  is  $\{a, \neg a, b, \neg b\}$ ,  $NecFls$  is empty. An interval  $[p_0, p_1]$ ,  $p \in Nec$  is given by



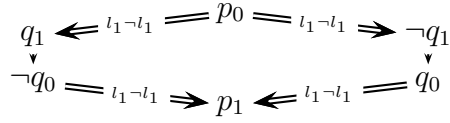
where  $q$  equals to  $b$  if  $p$  is  $a$  or  $\neg a$  and  $q$  equals to  $a$  if  $p$  is  $b$  or  $\neg b$ . The four intervals  $[a_0, a_1]$ ,  $[\neg a_0, \neg a_1]$ ,  $[b_0, b_1]$  and  $[\neg b_0, \neg b_1]$  generates the CSD

*CSD* is given by the digraphs



where in the leftmost side if  $p = a$ , then  $b = q$  and if  $p = b$ ,  $q = a$ . In the rightmost digraph  $p$  ranges over  $\{a, \neg a\}$  and  $q$  ranges over  $\{b, \neg b\}$ , that is,  $3 + 2 + 1$  digraphs.

The intervals MAXLIN of the form



associated to the macrobranch

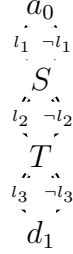
$l_1 \neg l_1$        $l_1 \neg l_1$   
 $l_1 \neg l_1$        $l_1 \neg l_1$

with incompatible the macronodes  $l_1 \neg l_1$ . An analogous analysis for

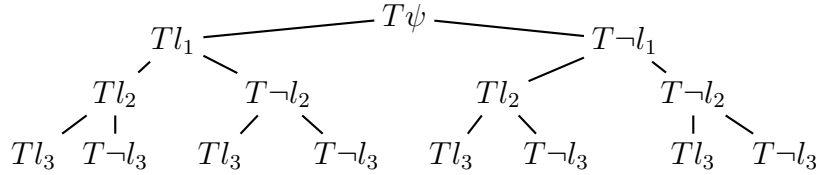


since the single node  $l_1 \neg l_1$  is not compatible. So, we have as output **there is no compatible chain**, that is, all combinations are not satisfiable.

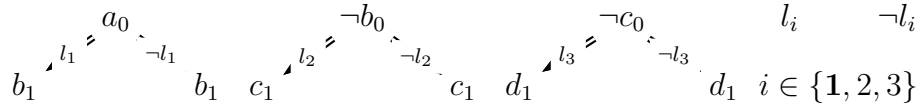
**Example 7.2** Consider the CSD



where  $S$  is an edge, say  $b_1 \rightarrow \neg b_0$ ,  $T$  is an edge  $c_1 \rightarrow \neg c_0$ . Of course, we have  $2^3$  lines in the Mirror Tableaux, that is, the combination of the two nodes in between 0 and 1, two nodes in between 1 and 2 and the two nodes in between 2 and 3:



Intervals  $\alpha_1 > \alpha_2 > \alpha_3$ , respectively, with columns ordered left to right



We have,  $\{l_1, \neg l_1\} > \{l_2, \neg l_2\} > \{l_3, \neg l_3\}$ , each  $l_i, \neg l_i$  is not comparable, so each interval is erased and the output is **no chains**.

As an example of ordered labels, suppose that the sequences, top to bottom and left to right, have alternated sequence labels  $A_1, A_2, B_1, B_2, C_1, C_2$  and Ord

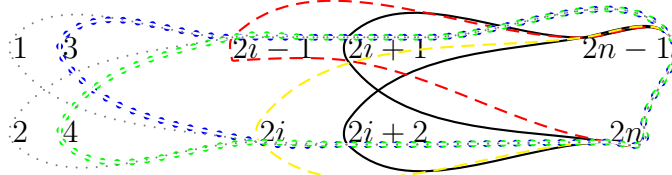
$$\begin{array}{l}
 A_1, A_2, (A_1 \bullet A_2)B_1, (A_1 \bullet A_2)B_2, \\
 ((A_1 \bullet A_2)B_1 \bullet (A_1 \bullet A_2)B_2)C_1, ((A_1 \bullet A_2)B_1 \bullet (A_1 \bullet A_2)B_2)C_2
 \end{array}$$

**Example 7.3** Consider the tabular array given by the disjunction of  $n$  linear macrobranches

$$\begin{array}{cccc}
 1 & 3 & & 2n - 1 \\
 2 & 4 & \cdots & 2n
 \end{array}$$

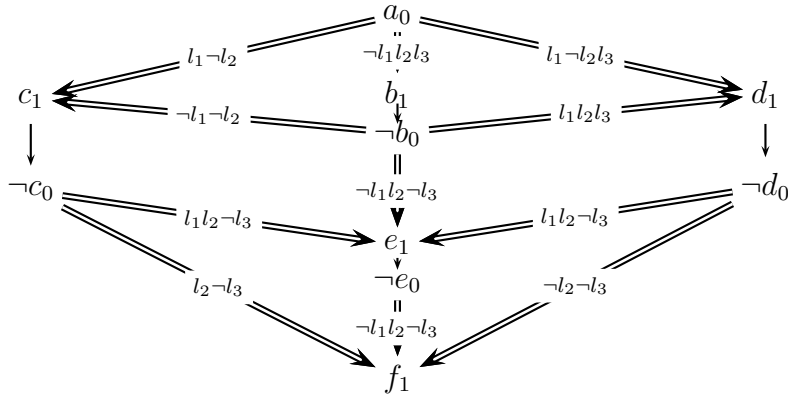
Suppose that all macronodes are compatible, except, maybe, macronodes  $2j$  and  $2j - 1$ ,  $1 \leq j \leq n$ .

Obtain

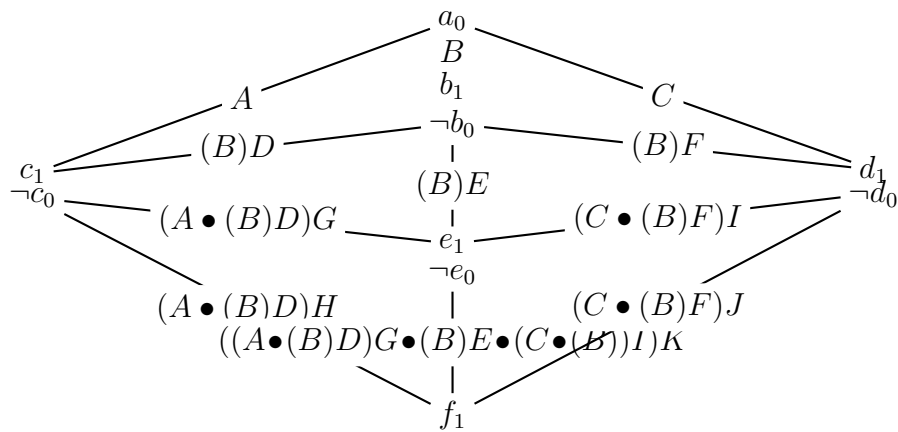


where the circling is the **Nest** of the lower number. That is, 1 and 2 are in the **setarray**( $k$ ) of all node  $k$ , for  $k \geq 3$ , 3 and 4 are in the **setarray**( $k$ ) of all macronode  $k$ , for  $k \geq 5$  and so on. The output is **there are compatible chains**. Of course, naming **all** of them is the exponential task of writing all the sets  $\{l_1, l_2, \dots, l_n\}$ , where each  $l_i$  ranges over  $\{2i - 1, 2i\}$ .

**Example 7.4** Consider the CSD,



Source is the set  $\{v_0 \in V\}$ , spillway is the set  $\{v_1 | v_1 \in V\}$  and root is  $a_0$ .



Let the order  $\prec$  be the alphabetic order over the alternated sequences label, the group of nodes is incomparable

Table for **setarray**(Nd) and **Nest**(Nd) (omit the node in its table)

A	C, (C • (B))J	$\emptyset$	
B	$\emptyset$	$\emptyset$	
C	$\emptyset$	A	
(B)D	((C • (B)F)J		
(B)E	(A • (B)D)H		
(B)F	$\emptyset$		
(A • (B)D)G	(A • (B)D)H, (C • (B)F)I		
(A • (B)D)H	((A • (B)D)G • (B)E • (C • (B))I)K	(B)E, (A • (B)D)G	
(C • (B)F)I	$\emptyset$	(A • (B)D)G	
((C • (B)F)J	$\emptyset$	A, (B)D	
((A • (B)D)G • (B)E • (C • (B))I)K	$\emptyset$	(A • (B)D)H	

As no **setarray**(Nd) and **Nest**(Nd) are complete, the output is **there are no compatible chains**.

**Example 7.5** Consider the disjunction of the columns,

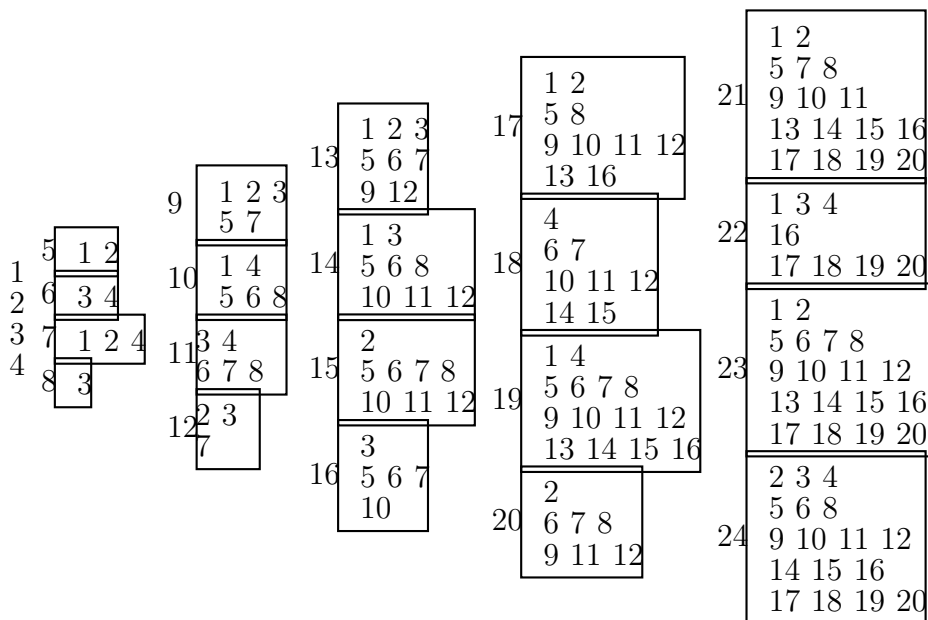
1	5	9	13	17	21
2	6	10	14	18	22
3	7	11	15	19	23
4	8	12	16	20	24

and the compatibility relation generated by

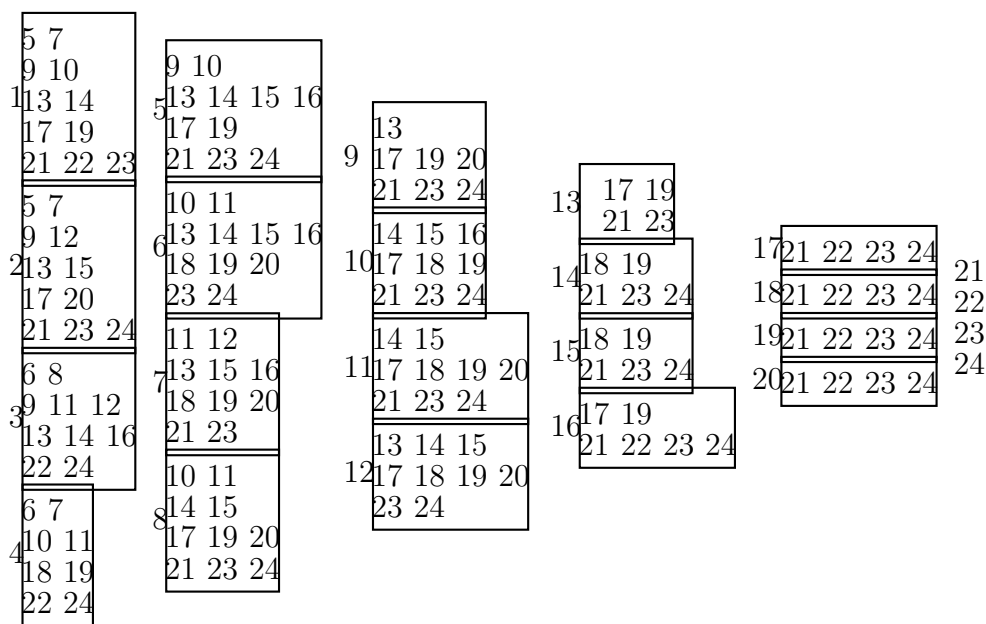
1	5, 7, 9, 10, 13, 14, 17, 19, 21, 22, 23	11	14, 15, 17, 18, 19, 20, 21, 23, 24
2	5, 7, 9, 12, 13, 15, 17, 20, 21, 23, 24	12	13, 14, 15, 17, 18, 19, 20, 23, 24
3	6, 8, 9, 11, 12, 13, 14, 16, 22, 24	13	17, 19, 21, 23
4	6, 7, 10, 11, 18, 19, 22, 24	14	18, 19, 21, 23, 24
5	9, 10, 13, 14, 15, 16, 17, 19, 21, 23, 24	15	18, 19, 21, 23, 24
6	10, 11, 13, 14, 15, 16, 18, 19, 20, 23, 24	16	17, 19, 21, 22, 23, 24
7	11, 12, 13, 15, 16, 18, 19, 20, 21, 23	17	21, 22, 23, 24
8	10, 11, 14, 15, 17, 19, 20, 21, 23, 24	18	21, 22, 23, 24
9	13, 16, 17, 19, 20, 21, 23, 24	19	21, 22, 23, 24
10	14, 15, 16, 17, 18, 19, 21, 23, 24	20	21, 22, 23, 24

where the node in the first column is compatible with all nodes in the second column.

Write  $\text{setarray}(Nd_j^r)$



$\text{Nest}(Nd_j^r)$



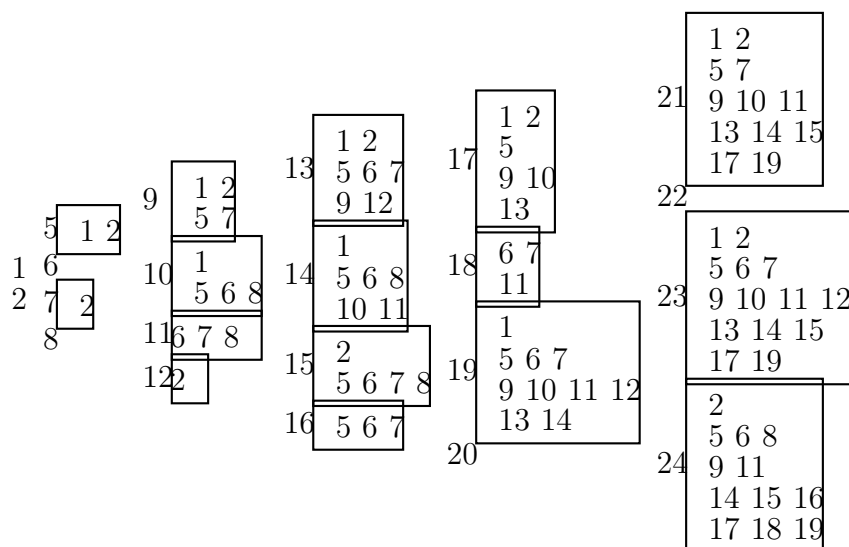
Erase, 1 in  $\text{setarray}(7)$  and  $\text{setarray}(22)$ . Erase 2 in  $\text{setarray}(20)$ . Erase 3 and 4. Erase 6 in  $\text{setarray}(20)$ . Erase 7 in  $\text{setarray}(12)$  and

**setarray(20).** Erase 8 in **setarray(17)**, **setarray(19)**, **setarray(20)**, **setarray(21)** and in **setarray(23)**. Erase 9 in **setarray(20)**.

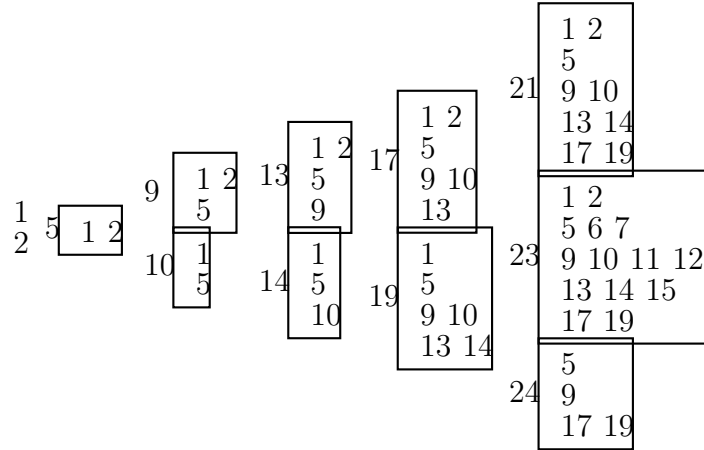
Erase 10 in **setarray(15)**, **setarray(16)**, **setarray(17)**, **setarray(18)** and in **setarray(24)**. Erase 11 in **setarray(15)**, **setarray(17)**, **setarray(20)**. Erase 12 in **setarray(14)**, **setarray(15)**, **setarray(17)**, **setarray(18)**, **setarray(20)**, **setarray(24)**. Erase 14 in **setarray(18)**. Erase 15 in **setarray(18)** and in **setarray(19)**.

Erase 16 in **setarray(17)**, **setarray(19)**, **setarray(21)** and in **setarray(23)**. Erase 17 in **setarray(22)**. Erase 18 in **setarray(21)**, **setarray(22)** and in **setarray(23)**. Erase 19 in **setarray(22)**.

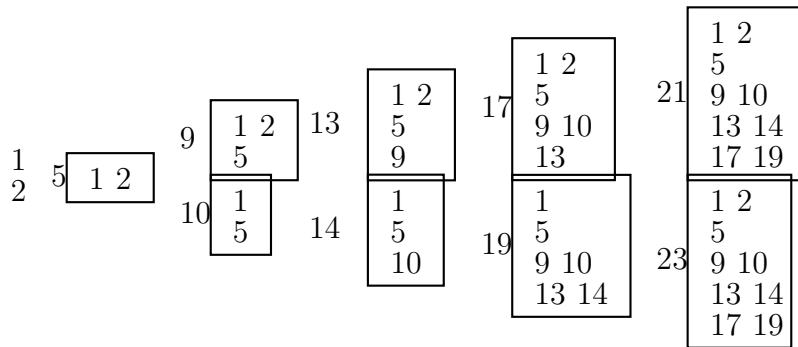
Erase 20 in **setarray(21)**, **setarray(22)**, **setarray(23)** and in **setarray(24)**. Erase 22,  
Obtain,



Erase 6, 7, 8, 11, 12, 15, 16, 18, 20, 22, 2 and 14 in `setarray(24)`.



Finally, erase 24,



We cannot perform any operation already have the answer **There are compatible antichains**. The (expensive unnecessary) task of naming all the antichains (in the solution set) give us the sets:

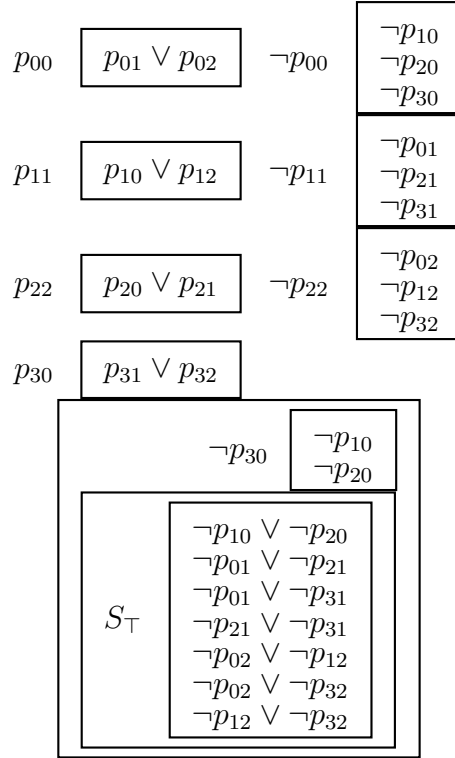
$$\begin{array}{l|l|l}
 1, 5, 9, 13, 17, 21 & 1, 5, 9, 13, 19, 21 & 1, 5, 9, 13, 17, 23 \\
 1, 5, 9, 13, 19, 23 & 1, 5, 10, 14, 19, 21 & 1, 5, 10, 14, 19, 23 \\
 2, 5, 9, 13, 17, 21 & 2, 5, 9, 13, 17, 23 & 
 \end{array}$$

**Example 7.6** We show a very simple pigeon hole problem. Let  $PH_4^3$  be the problem of four pigeons and three houses,

## Part 1: Partition

$$\begin{aligned}
& (p_{00} \vee p_{01} \vee p_{02}) \wedge (p_{10} \vee p_{11} \vee p_{12}) \wedge (p_{20} \vee p_{21} \vee p_{22}) \wedge (p_{30} \vee p_{31} \vee p_{32}) \wedge \\
& (\neg p_{00} \vee \neg p_{10}) \wedge (\neg p_{00} \vee \neg p_{20}) \wedge (\neg p_{00} \vee \neg p_{30}) \wedge \\
& (\neg p_{10} \vee \neg p_{20}) \wedge (\neg p_{10} \vee \neg p_{30}) \wedge (\neg p_{20} \vee \neg p_{30}) \wedge \\
& (\neg p_{01} \vee \neg p_{11}) \wedge (\neg p_{01} \vee \neg p_{21}) \wedge (\neg p_{01} \vee \neg p_{31}) \wedge \\
& (\neg p_{11} \vee \neg p_{21}) \wedge (\neg p_{11} \vee \neg p_{31}) \wedge (\neg p_{21} \vee \neg p_{31}) \wedge \\
& (\neg p_{02} \vee \neg p_{12}) \wedge (\neg p_{02} \vee \neg p_{22}) \wedge (\neg p_{02} \vee \neg p_{32}) \wedge \\
& (\neg p_{12} \vee \neg p_{22}) \wedge (\neg p_{12} \vee \neg p_{32}) \wedge (\neg p_{22} \vee \neg p_{32})
\end{aligned}$$

One possible partition is shown above, where the boxes are used to denote disjunctions and the separation by lines, in formulas inside the box, denotes conjunction.



End Par 1

## Part 2: Cylindrical Digraph

Write a sketch of the cylindrical digraph associated to  $PH_4^3$ . The only necessarily false literal is  $\neg p_{30}$ , that has an edge of the form  $p_{30} \xrightarrow{\neg p_{30}} \top$ . The

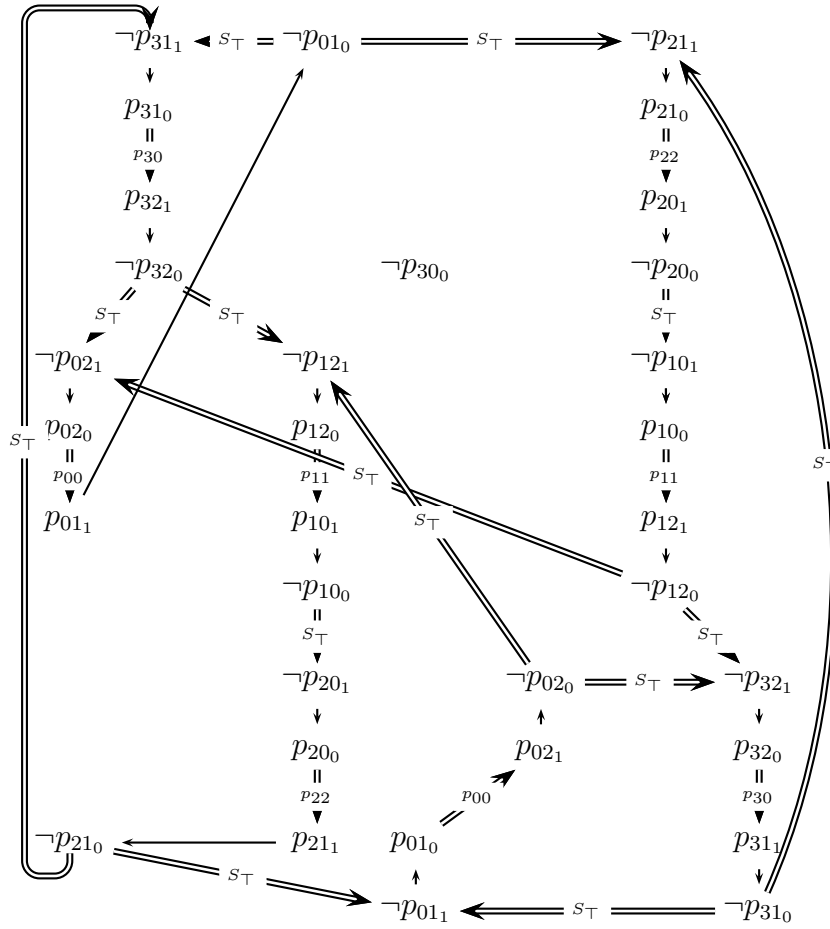
*literals*

$$P = \{p_{01}, p_{02}, p_{10}, p_{12}, p_{20}, p_{21}, p_{31}, p_{32}\}$$

$$N = \{\neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\}$$

are necessarily true. Each literal  $\neg p_{ij} \in N$  has an edge  $l_0 \Rightarrow \perp \Rightarrow l_1$  whose label is  $\neg p_{jj}$ , forming the sequence

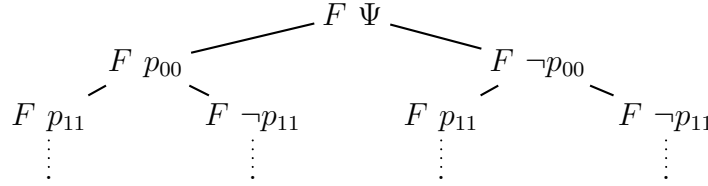
$$\neg p_{ij_0} \xrightarrow{\neg p_{jj}} \perp \xrightarrow{\neg p_{jj}} \neg p_{ij_1}$$



## End Part 2

**Part 3: The Combinatorics Choice Tableau** *In our algorithm, we never write explicitly the Combinatorics Choice Tableau, a task that demands an exponential use (over the set of labels) of the space.*

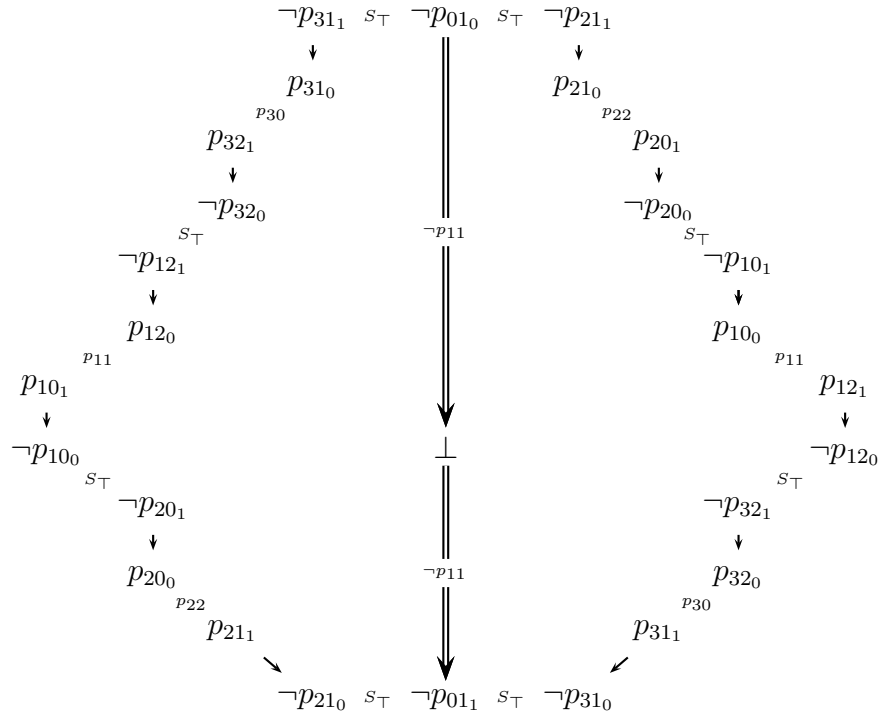
The set of labels is  $\{p_{00}, p_{11}, p_{22}, p_{30}, \neg p_{00}, \neg p_{11}, \neg p_{22}, \neg p_{30}\}$  and, therefore, the Combinatorics Choice Tableau is



Each branches ending in  $F p_{11}$  and  $F \neg p_{11}$  have a branching in  $F p_{22}$  and  $F \neg p_{22}$ . Branches ending in  $F p_{22}$  and  $F \neg p_{22}$  have a branching in  $F p_{30}$  and  $F \neg p_{30}$ , so, we write  $2^4$  branches. Our proposal is storing that exponentially big bit of information on a polynomially bounded storage place.

**End Part 3**

To illustrate a typical interval in  $N$ , case  $\neg p_{01}$  we have the digraph:



cut the unnecessary  $S_{\top}$  and obtain the macrotableau

$$\begin{array}{ccc} p_{30} & \neg p_{11} & p_{22} \\ p_{11} & \neg p_{11} & p_{11} \\ p_{22} & & p_{30} \end{array}$$

**Part 4:**

Applying Definition 3.9, we have the closed digraphs,

0  $[a_0, a_1] \rightarrow [\neg a_0, \top]$ , for  $a = \neg p_{30} \in \mathbf{Nec} \cap \mathbf{NecFls}$ . Obtain only one sequence,

$$\neg p_{30_0} \xrightarrow{\neg p_{00}} \perp \xrightarrow{\neg p_{00}} \neg p_{30_1} \rightarrow p_{30} \xrightarrow{\neg p_{30}} \top$$

1  $[a_0, a_1] \rightarrow [\neg a_0, \neg b_1] \rightarrow [b_0, b_1]$ , for all  $a, b \in \mathbf{Nec}$  deals with all combinations of

$$\begin{aligned} P &= \{p_{01}, p_{02}, p_{10}, p_{12}, p_{20}, p_{21}, p_{31}, p_{32}\} \\ N &= \{\neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \end{aligned}$$

we have  $\frac{16 \times 17}{2} = 136$  digraphs.

$$\begin{aligned}
& [p_{01_0}, p_{01_1}] \rightarrow [\neg p_{01_0}, \neg p_{01_1}] \\
& [p_{01_0}, p_{01_1}] \rightarrow [\neg p_{01_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{02}, p_{10}, p_{12}, p_{20}, p_{21}, p_{31}, p_{32}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{02_0}, p_{02_1}] \rightarrow [\neg p_{02_0}, \neg p_{02_1}] \\
& [p_{02_0}, p_{02_1}] \rightarrow [\neg p_{02_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{10}, p_{12}, p_{20}, p_{21}, p_{31}, p_{32}, \neg p_{01}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{10_0}, p_{10_1}] \rightarrow [\neg p_{10_0}, p_{10_1}] \\
& [p_{10_0}, p_{10_1}] \rightarrow [\neg p_{10_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{12}, p_{20}, p_{21}, p_{31}, p_{32}, \neg p_{01}, \neg p_{02}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{12_0}, p_{12_1}] \rightarrow [\neg p_{12_0}, p_{12_1}] \\
& [p_{12_0}, p_{12_1}] \rightarrow [\neg p_{12_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{20}, p_{21}, p_{31}, p_{32}, \neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{20_0}, p_{20_1}] \rightarrow [\neg p_{20_0}, \neg p_{20_1}] \\
& [p_{20_0}, p_{20_1}] \rightarrow [\neg p_{20_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{21}, p_{31}, p_{32}, \neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{21_0}, p_{21_1}] \rightarrow [\neg p_{21_0}, \neg p_{21_1}] \\
& [p_{21_0}, p_{21_1}] \rightarrow [\neg p_{21_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{31}, p_{32}, \neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [p_{31_0}, p_{31_1}] \rightarrow [\neg p_{31_0}, \neg p_{31_1}] \\
& [p_{31_0}, p_{31_1}] \rightarrow [\neg p_{31_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{p_{32}, \neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{32}, \neg p_{30}\} \\
& [p_{32_0}, p_{32_1}] \rightarrow [\neg p_{32_0}, \neg p_{32_1}] \\
& [p_{32_0}, p_{32_1}] \rightarrow [\neg p_{32_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{01}, \neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{30}\} \\
& [\neg p_{01_0}, \neg p_{01_1}] \rightarrow [p_{01_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{02_0}, \neg p_{02_1}] \rightarrow [p_{02_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{10_0}, \neg p_{10_1}] \rightarrow [p_{10_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{12_0}, \neg p_{12_1}] \rightarrow [p_{12_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{20_0}, \neg p_{20_1}] \rightarrow [p_{20_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{21_0}, \neg p_{21_1}] \rightarrow [p_{21_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{31}, \neg p_{32}, \neg p_{30}\} \\
& [\neg p_{31_0}, \neg p_{31_1}] \rightarrow [p_{31_0}, \neg a_1] \rightarrow [a_0, a_1], \\
& a \in \{\neg p_{32}, \neg p_{30}\} \\
& [\neg p_{32_0}, \neg p_{32_1}] \rightarrow [p_{32_0}, p_{30_1}] \rightarrow [\neg p_{30_0} \neg p_{30_1}]
\end{aligned}$$

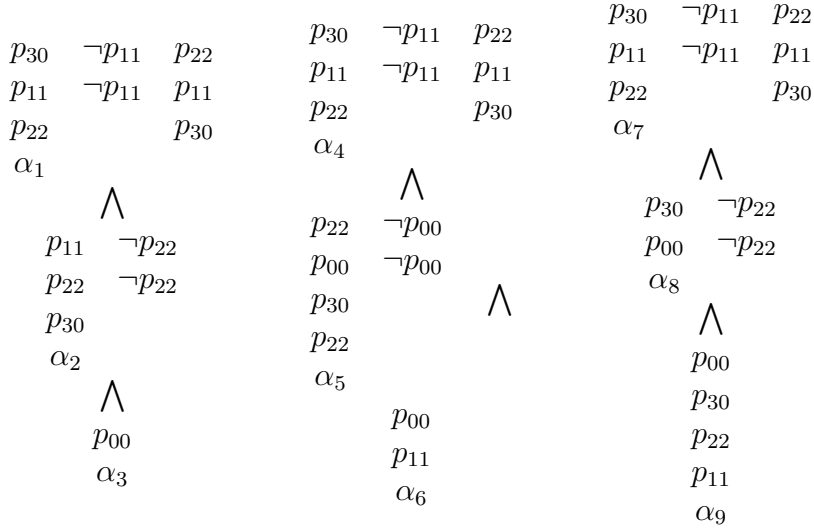
- 2  $[a_0, a_1] \rightarrow [\neg a_0, \neg p_{30_1}] \rightarrow [p_{30_1}, \top]$  all a ranging over  $P \cup N$ .  
 3  $[\top, \neg d_1] \rightarrow [d_0, c_1] \rightarrow [\neg c_0, \top]$ , for  $d, c \in \text{NecFls}$  is

$$\top \xrightarrow{\neg p_{30}} p_{31} \rightarrow \neg p_{30_0} \xrightarrow{\neg p_{00}} \perp \xrightarrow{\neg p_{00}} \neg p_{30_1} \rightarrow p_{30} \xrightarrow{\neg p_{30}} \top$$

#### End Part 4

For the sake of clearness, we search for antichains in a small piece of  $\mathcal{CSD}$ . We will conclude that there is no compatible antichain in this set and, therefore, there is no compatible antichains in the whole  $\mathcal{CSD}$ .

Write the closed digraphs for  $[\neg p_{01_0}, \neg p_{01_1}] \rightarrow [p_{01_0}, \neg a_1] \rightarrow [a_0, a_1]$ , a ranging, respectively, over the set  $\{\neg p_{02}, \neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}, \neg p_{30}\}$



$$\begin{array}{l}
p_{30} \quad \neg p_{11} \quad p_{22} \\
p_{11} \quad \neg p_{11} \quad p_{11} \\
p_{22} \quad \quad \quad p_{30} \\
\alpha_{10} \\
\bigwedge \\
p_{11} \quad \neg p_{00} \\
p_{30} \quad \neg p_{00} \\
p_{00} \\
p_{11} \\
\alpha_{11} \\
\bigwedge \\
p_{00} \\
p_{30} \\
p_{22} \\
\alpha_{12} \\
p_{30} \quad \neg p_{11} \quad p_{22} \\
p_{11} \quad \neg p_{11} \quad p_{11} \\
p_{22} \quad \quad \quad p_{30} \\
\alpha_{19} \\
\bigwedge \\
p_{00} \quad p_{11} \quad \neg p_{22} \\
p_{22} \quad p_{22} \quad \neg p_{22} \\
p_{11} \quad p_{00} \\
\alpha_{20} \\
\bigwedge \\
p_{00} \\
p_{11} \\
p_{22} \\
p_{30} \\
\alpha_{21}
\end{array}
\quad
\begin{array}{l}
p_{30} \quad \neg p_{11} \quad p_{22} \\
p_{11} \quad \neg p_{11} \quad p_{11} \\
p_{22} \quad \quad \quad p_{30} \\
\alpha_{13} \\
\bigwedge \\
p_{30} \quad \neg p_{11} \\
p_{00} \quad \neg p_{11} \\
\alpha_{14} \\
\bigwedge \\
p_{00} \\
p_{30} \\
\alpha_{15}
\end{array}
\quad
\begin{array}{l}
p_{30} \quad \neg p_{11} \quad p_{22} \\
p_{11} \quad \neg p_{11} \quad p_{11} \\
p_{22} \quad \quad \quad p_{30} \\
\alpha_{16} \\
\bigwedge \\
p_{22} \quad p_{00} \quad \neg p_{11} \\
p_{11} \quad p_{11} \quad \neg p_{11} \\
p_{00} \quad p_{22} \\
\alpha_{17} \\
\bigwedge \\
p_{00} \\
p_{30} \\
\alpha_{18}
\end{array}$$

The last macrobranch came from an empty digraph, so, it is empty and not considered in our computation.

Write:

$$[\neg p_{02_0}, \neg p_{02_1}] \rightarrow [p_{02_0}, \neg a_1] \rightarrow [a_0, a_1], \quad a \in \{\neg p_{10}, \neg p_{12}, \neg p_{20}, \neg p_{21}, \neg p_{31}, \neg p_{32}\}$$

$$\begin{array}{l}
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{22} \\
\bigwedge \\
p_{00} \\
p_{22} \\
p_{11} \\
p_{30} \\
p_{00} \\
p_{11} \\
\alpha_{23} \\
\bigwedge \\
p_{00} \quad \neg p_{00} \\
p_{00} \quad \neg p_{00} \\
p_{30} \\
p_{22} \\
\alpha_{24} \\
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{31} \\
\bigwedge \\
p_{00} \\
p_{22} \\
p_{11} \\
p_{30} \\
p_{00} \\
p_{11} \\
p_{22} \\
\alpha_{32} \\
\bigwedge \\
p_{30} \quad \neg p_{11} \\
p_{00} \quad \neg p_{11} \\
\alpha_{33} \\
[-p_{21_0}, \neg p_{21_1}] \rightarrow [p_{21_0}, \neg a_1] \rightarrow [a_0, a_1], a \in \{\neg p_{31}, \neg p_{32}\}
\end{array}$$

$$\begin{array}{l}
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{25} \\
\bigwedge \\
p_{00} \\
p_{22} \\
p_{11} \\
\alpha_{26} \\
\bigwedge \\
p_{30} \quad \neg p_{22} \\
p_{00} \quad \neg p_{22} \\
\alpha_{27}
\end{array}$$

$$\begin{array}{l}
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{28} \\
\bigwedge \\
p_{00} \\
p_{22} \\
\alpha_{29} \\
\bigwedge \\
p_{11} \quad \neg p_{00} \\
p_{30} \quad \neg p_{00} \\
p_{00} \\
p_{11} \\
\alpha_{30}
\end{array}$$

$$\begin{array}{l}
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{34} \\
\bigwedge \\
p_{00} \\
p_{22} \\
p_{11} \\
p_{30} \\
\alpha_{35} \\
\bigwedge \\
p_{00} \quad \neg p_{11} \\
p_{11} \quad \neg p_{11} \\
p_{22} \\
\alpha_{36}
\end{array}$$

$$\begin{array}{l}
p_{11} \quad \neg p_{22} \\
p_{22} \quad \neg p_{22} \\
p_{30} \\
\alpha_{37} \\
\bigwedge \\
p_{00} \\
p_{30} \\
\alpha_{38} \\
\bigwedge \\
p_{11} \quad p_{00} \quad \neg p_{22} \\
p_{22} \quad p_{22} \quad \neg p_{22} \\
p_{00} \quad p_{11} \\
\alpha_{39}
\end{array}$$

$p_{00} \quad \neg p_{11}$	$p_{00} \quad \neg p_{11}$
$p_{30} \quad \neg p_{11}$	$p_{30} \quad \neg p_{11}$
$\alpha_{40}$	$\alpha_{43}$
$\bigwedge$	$\bigwedge$
$p_{22}$	$p_{22}$
$p_{11}$	$p_{11}$
$p_{30}$	$p_{30}$
$\alpha_{41}$	$\alpha_{44}$
$\bigwedge$	$\bigwedge$
$p_{00} \quad p_{22} \quad \neg p_{11}$	$p_{00} \quad p_{11} \quad \neg p_{22}$
$p_{11} \quad p_{11} \quad \neg p_{11}$	$p_{22} \quad p_{22} \quad \neg p_{22}$
$p_{22} \quad p_{00}$	$p_{11} \quad p_{00}$
$\alpha_{42}$	$\alpha_{45}$

*Intersection of necessarily false and true, for  $\neg p_{30}$*

$\neg p_{00}$	
$\neg p_{00}$	
$\neg p_{30}$	
$\alpha_{46}$	
$[\neg p_{21_0}, \neg p_{21_1}] \rightarrow [p_{21_0}, \neg a_1] \rightarrow [a_0, a_1], a \in \{\neg p_{31}, \neg p_{32}\}$	
$p_{00} \quad \neg p_{11}$	$p_{00} \quad \neg p_{11}$
$p_{30} \quad \neg p_{11}$	$p_{30} \quad \neg p_{11}$
$\alpha_{47}$	$\alpha_{50}$
$\bigwedge$	$\bigwedge$
$p_{22}$	$p_{22}$
$p_{11}$	$p_{11}$
$p_{30}$	$p_{30}$
$\alpha_{48}$	$\alpha_{51}$
$\bigwedge$	$\bigwedge$
$p_{00} \quad p_{22} \quad \neg p_{11}$	$p_{00} \quad p_{11} \quad \neg p_{22}$
$p_{11} \quad p_{11} \quad \neg p_{11}$	$p_{22} \quad p_{22} \quad \neg p_{22}$
$p_{22} \quad p_{00}$	$p_{11} \quad p_{00}$
$\alpha_{49}$	$\alpha_{52}$

*Solve the disjunction of the above macrobranches. Obtain an answer that there are no compatible chains and, therefore, conclude that there are no compatible chains over the disjunctions of all macrobranches.*

*Observe that  $\alpha_1, \alpha_4, \dots, \alpha_{3K+1}, \dots, \alpha_{19}$  do have three columns and, after adding compatible nodes, we erase  $p_{11}$  and obtain*



$$\begin{array}{ccccccc}
p_{11} & \neg p_{22} & & & & & p_{11} & \neg p_{22} \\
& \neg p_{22} & & & & & & \neg p_{22} \\
\alpha_{28} & \wedge & & p_{11} & \neg p_{22} & & \alpha_{34} & \wedge \\
p_{11} & \neg p_{00} & & \neg p_{22} & & & p_{22} & \neg p_{11} \\
& \neg p_{00} & & \alpha_{31} & & & & \neg p_{11} \\
p_{11} & & & & & & \alpha_{36} & \\
\alpha_{30} & & & & & & & \\
p_{11} & \neg p_{22} & & & & & & \\
& \neg p_{22} & & & & & & \\
\alpha_{37} & \wedge & & p_{22} & p_{22} & \neg p_{11} & p_{11} & p_{11} & \neg p_{22} \\
& & & & & \neg p_{11} & & & \neg p_{22} \\
p_{11} & p_{11} & \neg p_{22} & \alpha_{42} & & & \alpha_{45} & & \\
& & \neg p_{22} & & & & & & \\
\alpha_{39} & & & & & & & & \\
& & & & & & & & \\
\neg p_{00} & & p_{22} & p_{22} & \neg p_{11} & p_{11} & p_{11} & \neg p_{22} \\
\neg p_{00} & & & & \neg p_{11} & & & \neg p_{22} \\
\neg p_{30} & & \alpha_{49} & & & \alpha_{52} & & \\
\alpha_{46} & & & & & & & 
\end{array}$$

as we have  $p_{22}$  and  $p_{11}$  together with their negations in single columns, we erase everything and the output is **there are no compatible chains.**

## Index

- $\vee$ , 9
- $\wedge$ , 9
- MAXLIN, 16
  
- alternated sequence, 5
- antichain, 10
  
- chain, 10
- closed, 11
- closed digraphs, 6
- closed interval, 11
- Combinatorics Choice Tableau, 4
- Combinatorics Choice Tableau, 2
- cylindrical digraph, 4
  
- factor, 3
  
- incompatible, 14
- interval, 5
  
- linear macrobranch, 8
  
- macrobranches, 9
- macronode, 9
- maximal tabular array, 16
- Mirror Tableau, 9
  
- order over set of labels in  $\mathcal{CSD}$ , 18
- ordered set of labels associated to a  
    closed digraph, 17
- ordered set of labels, 17
  
- partitioned formula, 3
  
- represents the Combinatorics Choice  
    Tableau, 14
- roots, 16
  
- solution set, 20
- source, 16
- spillway, 16

## References

- [1] Pitassi T. Arai N. and Urquhart A. The complexity of analytic tableaux. In *Proceedings of ACM Symposium on the Theory of Computing*, 2001.
- [2] Cook S.A. The complexity of theorem proving procedures. *Proc 3rd STOC*, pages 151–158, 1971.
- [3] Cook S.A. Reckhow A.R. The relative efficiency of propositional proof systems. *J. of Symbolic Logic*, 44(1):36–50, 1979.
- [4] Stockmeyer L. J. Classifying the computational complexity problems. *The Journal of Symbolic Logic*, 52 No 1:1–43, 1987.
- [5] Levin, Leonid. Universal sorting problems. problems of information transmission. *Problemy Peredachi Informatsii*, 9:265–266, 1973.
- [6] Levin, Leonid. "universal search problems". *Problemy Peredachi Informatsii*, 6(4):265–266, 1984.
- [7] Mondatori M. Classical analytical deduction. *Annali dell'Università di Ferrara*, Sez III, 1988. Discussion paper 1, Università di Ferrara.
- [8] Mondatori M. Classical analytical deduction, aprt ii. *Annali dell'Università di Ferrara*, Sez III, 1988. Discussion paper 5, Università di Ferrara.
- [9] Mondatori M. On the notion of classical proof. *Tem e prospettive della logica e della filosofia della scienza contemporanea*, I:211–224, 1988.
- [10] Mondatori M. Sulla nozione di dimostrazione classica. *Annali dell'Università di Ferrara*, Sez III, 1988. Discussion paper 3, Università di Ferrara.
- [11] Meyer A.R. weak monadic second-order theory of successor is not elementary recursive. In R Parikh, editor, *Lecture Notes in Mathematics*, volume vol 453, pages 132–154. Springer-Verlag, New York, 1972-73.
- [12] Meyer A.R. Stockmeyer L. J. The equivalence problem for regular expressions with squaring requires exponential space. *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

- [13] Pudlak P. Satisfiability - algorithms and logic. In *Mathematical Foundations of Computer Science*, pages 129–141. Springer, 1998.
- [14] M. A. Weiss. A polynomial algorithm for deciding 3-sat. Technical report, IME-USP. Mathematics Department, Universidade de Sao Paulo, 2007. weiss@ime.usp.br.