# Leafy-ROBDD: A Formalism to Represent and Decide Mixed Formula

# CEA Technical Report (2005/08/07): DRT/LIST/DTSI/SOL/05-211

N. Rapin* , e-mail: nicolas.rapin@cea.fr

CEA/DRT/LIST/DTSI/SLA Saclay
F-91191 Gif sur Yvette Cedex
Phone : +33 1 69 08 25 87

**Abstract.** In this paper we present a structure which can be used to represent and decide formula combining pure boolean variables and constraints defined within an arbitrary decidable theory. This structure is strongly inspired from *ordered binary decision diagrams* and is called leafy-ROBDD.

**keywords**: Hybrid formula, Leafy-ROBDD.

## 1 Introduction

Formal methods, like verification, tests generation, takes models of systems as inputs and explore their semantics in order to discover their states or their behaviors. This exploration may be explosive as the number of states and possible behaviors may be huge. Symbolic formal techniques are commonly used in order to tackle this problem. They consist in a modeling systems with succint symbolic formalisms like the *symbolic transitions systems* formalism and in exploring the semantics of those models in a symbolic way. The symbolic execution technique, which computes symbolic execution graphs( or trees), is such a symbolic exploration technique. For the data part of all those symbolic structures (input models as well as their symbolic executions), it is referred to an underlying theory, generally a first order decidable theory like, for example, the Presburger Arithmetics. Indeed the work presented here has been developed as a component of the AGATHA toolset which treats models of reactive systems specified in the symbolic transition graphs formalism [1,2,3]. The AGATHA approach promotes and supports a methodology for systems development admitting two main phases. The first is to debug specifications, the second is to generate tests

* AGATHA project member

(from the debugged specifications) to be experimented on the future realized system. The main technique used by AGATHA for those two purposes is based on *symbolic execution* in the sens defined for example by King in [4]. *Symbolic execution* is an execution in which inputs are replaced by symbols and numerical values by terms. Data admitted in models treated by AGATHA are boolean and integers. Operations allowed in transitions must be linear. A guard of a transition could for example the formula $b \wedge (x > 0)$ where $b$ is a boolean variable and $x$ an integer variable. An allowed transformation could be, written in Pascal style, $x := 2.y + 1$ ($2.y$ abbreviates $y + y$).

For debugging activity, a feature of AGATHA is to execute the model until a criterion is satisfied which can be interpreted as the fact that the most significant behaviors of the models have been computed. This criterion is called *redundancy detection* and has been explained by Rapin and al. in [5]. In practice this criterion requires the decision of a formula of the form $\forall x_1, \ldots, x_n (A \Rightarrow B)$ where $x_1, \ldots, x_n$ are free in $A$ and $B$. This needs led us to develop the $\mathbb{U}^n$-*robdd* structure presented here. This structure is able to represent and decide mixed formulae combining boolean variables and linear constraints over integer variables. A formula of that kind is for example

$$(a \wedge (x - y + 1 \leq 0)) \Rightarrow (b \vee (x - y \leq 0))$$

where $x, y$ are integer variables and $a, b$ are boolean variables. Our approach consists in combining *binary decision diagrams* [6] with decision procedures for Presburger arithmetic. The structure allowing this combination is inspired from *reduced ordered binary decision diagrams* and has the same kind of properties. It provides compact representations of mixed formulae. We can also prove that semantically equivalent formulae have the same representation in our structure. This implies that any tautology (resp. antilogy) has the same representation than $\top$ (resp. $\bot$). The immediate consequence is that tautologies and antilogies are detected by construction of their representations in the structure we propose.

The paper is organised as follow. First we define the $\mathbb{U}^n$-*robdd* structure. This structure can be seen as a syntax based on graphs (instead of expressions). Then we present models of $\mathbb{U}^n$-*robdd*. The main interest of $\mathbb{U}^n$-*robdd* is given through a lemma which states that there is a one-to-one correspondance between a $\mathbb{U}^n$-*robdd* and sets of models (this is generally not the case for expressions since, for example, $\top$ and $(a \vee \neg a)$ have the same models). Then we apply our results to a more usual syntax, based on expressions, very close to Presburger arithmetics.

In the whole paper we refer to a structure $(\mathscr{B}, \prec)$ where $\mathscr{B}$ is a countable set and $\prec$ a total strict order over $\mathscr{B}$. The reader should consider $\mathscr{B}$ as a set of symbols of boolean variables for which an order is given.

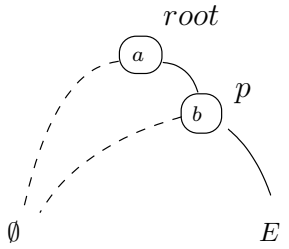
## 2 The $\mathbb{U}^n$-*robdd* structure

**Definition 1 ($\mathbb{U}^n$-*robdd*).** *Let $n \in \mathbb{N}^*$. Let $\mathbb{U}$ be a set. A* $\mathbb{U}^n$-robdd *is a 5-uple* $(N, root, Var, Hi, Lo, T)$ *where*

- $T \subseteq \mathscr{P}(\mathbb{U}^n)$ *is called the set of* terminal nodes.
- $N$ *is a finite set, called the set of* non terminal nodes.
- $Var$ *is a function from* $N$ *to* $\mathscr{B}$.
- $Hi$ *and* $Lo$ *are two functions from* $N$ *onto* $N \cup T$.
- $root \in N \cup T$ *is the unique node such that there is no* $u \in N \cup T$ *such that* $Hi(u) = root$ *or* $Lo(p) = root$.

*and such that*

- $\forall u \in N,\ Hi(u) \in N \Rightarrow Var(u) \prec Var(Hi(u))$
- $\forall u \in N,\ Lo(u) \in N \Rightarrow Var(u) \prec Var(Lo(u))$
- $\forall u, v \in N\ [\ Var(u) = Var(v) \wedge Hi(u) = Hi(v) \wedge Lo(u) = Lo(v)\ ] \Rightarrow u = v$
- $\forall u \in N\ Hi(u) \neq Lo(u)$




$$E = \{(x, y) \in \mathbb{Z}^2 \ / \ x < y \ \}$$

**Fig. 1.** A $\mathbb{Z}^2$-*robdd*

Figure 1 shows an example of a $\mathbb{Z}^2$-*robdd*. A node is represented by a circle. The symbol within the circle, called a label, represents the image of the node by $Var$. The label of the node *root* (i.e. $Var(root)$) is $a$. Node $p$ is labelled by $b$. Dashed lines denote the $Lo$ map, solid lines the $Hi$ map. For example $Hi(root) = p$ and $Lo(p) = \emptyset$.

Excepted the last one, properties of $\mathbb{U}^n$-*robdd* are illustrated by the Figure 2. Now we introduce the definition of complete path which will be a useful notion in next section to define the semantic of $\mathbb{U}^n$-*robdd*.

**Definition 2.** *Let $G$ be a* $\mathbb{U}^n$-robdd. *A* complete path *of $G$ is a finite sequence $u_1, ..., u_w$ of nodes of $G$ such that for all $s$ with $1 \leq s < w$ we have*

- $u_1$ *is the* root *node*
- $u_{s+1} = Hi(u_s)$ *or* $u_{s+1} = Lo(u_s)$
- $u_w$ *is a terminal node*

### 2.1 Models of $\mathbb{U}^n$-*robdd*

**Definition 3.** *A model for a* $\mathbb{U}^n$-robdd *structure is a couple of* $\mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$. *Let* $(\delta, \gamma) \in \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$ *be a model and* $G$ *a* $\mathbb{U}^n$-robdd. *We note* $(\delta, \gamma) \vDash G$ *if there exists a complete path* $u_1, \ldots, u_w$ *of* $G$ *such that we have for all* $s$ *with* $1 \leq s < w$

1. $\delta(Var(u_s)) = 1 \Rightarrow u_{s+1} = Hi(u_s)$
2. $\delta(Var(u_s)) = 0 \Rightarrow u_{s+1} = Lo(u_s)$
3. $\gamma \in u_w$

For example the couple $(\delta, \gamma)$ such that $\delta(a) = 1$ and $\delta(b) = 1$ and $\gamma = \{(3, 5)\}$ is a model of the $\mathbb{Z}^2$-*robdd* of Figure 1. This because there is a complete path that satisfies the Definition 3. This complete path is *root*, $p$, $E$. Let us check it

1. $\delta(Var(root)) = 1 \Rightarrow p = Hi(root)$ is true
2. $\delta(p) = 1 \Rightarrow E = Hi(p)$ is true
3. $(3, 5) \in E$ is also true

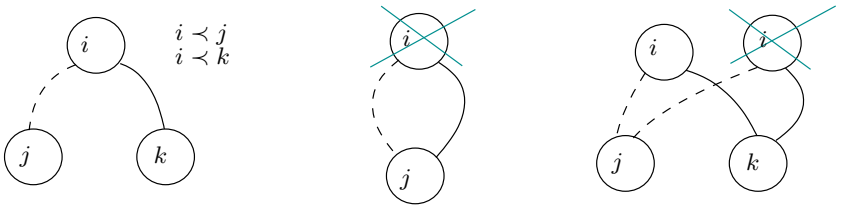



**Fig. 2.** Properties of $\mathbb{U}^n$-*robdd*

Now we introduce some features to manipulates set of models. Those features will be useful to state the key lemma below.

**Definition 4.** *Let* $\mathscr{M} \subseteq \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$ *be a set of models. Let* $a \in \mathscr{B}$, *then we introduce two notations*

$$\mathscr{M}_{(a,i)} =_{def} \{(\sigma, \gamma) \in \mathbb{B}^{\mathscr{B}-\{a\}} \times \mathbb{U}^n \text{ such that } (\sigma \cup \{(a, i)\}, \gamma) \in \mathscr{M}\}$$

$$\mathscr{M}_{(a,i,*)} =_{def} \bigcup_{s \in \{0,1\}} \{(\sigma \cup \{(a, s)\}, \gamma) \in \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n \text{ such that } (\sigma, \gamma) \in \mathscr{M}_{(a,i)}\}$$

*We say that* $a$ *is neutral in* $\mathscr{M}$ *if* $\mathscr{M}_{(a,0)} = \mathscr{M}_{(a,1)}$. *We say that* $a$ *is discriminating if* $a$ *is non neutral.*

Now we can state the key lemma.

**Lemma 1.** *Let* $\mathscr{M} \subseteq \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$. *Then there is exactly one* $\mathbb{U}^n$-robdd $G$ *such that* $\mathscr{M}$ *is the set of models of* $G$.

We say that $G$ is the unique $\mathbb{U}^n$-*robdd* which represents $\mathscr{M}$.

*Proof.* The proof rely on an induction on the number $k$ of discriminating variables of $\mathscr{M}$.

- $k = 0$. Let $\mathscr{M} \subseteq \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$ be such that it has no discriminating variables. First we prove the existence of a $\mathbb{U}^n$-*robdd* representing $\mathscr{M}$. Secondly we prove the unicity.

  1. Existence. Let us consider the $\mathbb{U}^n$-*robdd* consisting of one single node, this node being $\{x \in \mathbb{U}^n \text{ such that } \exists\delta\ (\delta, x) \in \mathscr{M}\}$. Then the set of models of this $\mathbb{U}^n$-*robdd* is $\mathscr{M}$.

  2. Unicity. We suppose that there is $G$ a $\mathbb{U}^n$-*robdd* having $\mathscr{M}$ as set of models and being different of the $\mathbb{U}^n$-*robdd* described just above in the existence proof. $G$ can't have non terminal nodes otherwise we could find an element $a \in Im(Var_G)$ non neutral in the set of models of $G$ which is supposed to be $\mathscr{M}$. So $G$ have only terminal nodes. But $G$ can't have several terminal nodes since all would satisfy the property of being root nodes. Yet the root must be unique. So $G$ consists in one single terminal node $u = \{x \in \mathbb{U}^n \text{ such that } \exists\delta\ (\delta, x) \in \mathscr{M}\}$. So $G$ is identical to the $\mathbb{U}^n$-*robdd* found in the existence proof.

- $k \geq 0$. We Suppose that the lemma is true for $k$. We show it is true for $k+1$. Let $\mathscr{M} \subseteq \mathbb{B}^{\mathscr{B}} \times \mathbb{U}^n$ be such that it has $k+1$ discriminating variables. Let $\breve{a}$ be the smallest discriminating variable of $\mathscr{M}$ with respect to $\prec$. We have that $\mathscr{M}_{(\breve{a},0,*)}$ and $\mathscr{M}_{(\breve{a},1,*)}$ are to sets of models with $k$ discriminating variables. Moreover $\mathscr{M}_{\breve{a}\mapsto 0} \neq \mathscr{M}_{\breve{a}\mapsto 1}$ otherwise $\breve{a}$ would'nt be discriminating. By induction $\mathscr{M}_{\breve{a}\mapsto 0}$ (resp. $\mathscr{M}_{\breve{a}\mapsto 1}$) is represented by an unique $\mathbb{U}^n$-*robdd* $G_0 : (N_0, root_0, Var_0, Hi_0, Lo_0, T_0)$ (resp. $G_1 : (N_1, root_1, Var_1, Hi_1, Lo_1, T_1)$). Since $\mathscr{M}_{\breve{a}\mapsto 0} \neq \mathscr{M}_{\breve{a}\mapsto 1}$ we have also $G_0 \neq G_1$. Let $u \in \mathbb{N}$ such that $u \notin N_0 \cup N_1$. Now let $N = N_0 \cup N_1 \cup \{u\}$. Let $Var = Var_0 \cup Var_1 \cup \{(u, \breve{a})\}$. Let $Hi = Hi_0 \cup Hi_1 \cup \{(u, r_1)\}$. Let $Lo = Lo_0 \cup Lo_1 \cup \{(u, r_0)\}$. Let $T = T_0 \cup T_1$. Then $G = (N, u, Var, Hi, Lo, T)$ is the only $\mathbb{U}^n$-*robdd* representing $\mathscr{M}$.

## 3 Application to mixed formulae

### 3.1 Mixed formulae

**Definition 5.** *Let $\mathscr{C} = \{\underline{c} \ / \ c \in \mathbb{Z}\}$ an infinite set of symbols of integer constants. An expression is a* mixed formula *if for some $n \in \mathbb{N}^*$ this expression satisfies the $\Phi_n$ grammar defined like this*

$$\Phi_n ::= \bot \mid \top \mid \mathscr{B} \mid \Psi_n \mid \Phi_n \wedge \Phi_n \mid \Phi_n \vee \Phi_n \mid \neg\Phi_n$$

$\Psi_n ::= (\mathscr{C}, \ldots, \mathscr{C})$ *(a vector of $n+1$ symbols of $\mathscr{C}$)*

*For $n \in \mathbb{N}^*$ we define $\Omega_n$ as a sub-grammar of $\Phi_n$*

$\Omega_n ::= \bot \mid \top \mid \Psi_n \mid \Omega_n \wedge \Omega_n \mid \Omega_n \vee \Omega_n \mid \neg\Omega_n$

The presence of sets $\mathscr{B}, \mathscr{C}$ and in the rules above means that while constructing an expression one must write a symbol belonging to the mentioned set at the location of this set. Suppose and $a, b \in \mathscr{B}$. Then $a \wedge b \wedge (\underline{1}\ ,\ \underline{-1}\ ,\ \underline{1})$ is a mixed formula. For readability we may write it $a \wedge b \wedge (x - y + 1 \leq 0)$ in the following. As one can see an expression satisfying $\Omega_n$ for some $n \in \mathbb{N}^*$ is a mixed formulae with no boolean variables.

### 3.2 Models of mixed formula

**Definition 6.** *Let $n \in \mathbb{N}^*$. $\vDash \subseteq (\mathbb{B}^{\mathscr{B}} \times \mathbb{Z}^n) \times \Omega_n$ is the smallest relation satisfying the conditions below (for readability we note $(\delta, \gamma) \vDash f$ for $((\delta, \gamma), f) \in \vDash$ and $(\delta, \gamma) \nvDash f$ for $((\delta, \gamma), f) \notin \vDash$)*

*For all $(\delta, \gamma) \in (\mathbb{B}^{\mathscr{B}} \times \mathbb{Z}^n)$*

- $(\delta, \gamma) \nvDash \bot$
- $(\delta, \gamma) \vDash \top$
- $(\delta, \gamma) \vDash b$, $b \in \mathscr{B}$, *if* $\delta(b) = 1$
- $(\delta, \gamma) \nvDash b$, $b \in \mathscr{B}$, *if* $\delta(b) = 0$
- $(\delta, \gamma) \vDash (\underline{c}_{\ 1},\ \ldots,\ \underline{c}_{\ n+1})$ *if we have* $c_1.\gamma(1) + \ldots + c_n.\gamma(n) + c_{n+1} \leq 0$
- $(\delta, \gamma) \vDash \neg g$ *if* $(\delta, \gamma) \nvDash g$
- $(\delta, \gamma) \vDash g \wedge h$ *if* $(\delta, \gamma) \vDash g$ *and* $(\delta, \gamma) \vDash h$
- $(\delta, \gamma) \vDash g \vee h$ *if* $(\delta, \gamma) \vDash g$ *or* $(\delta, \gamma) \vDash h$

For example $(\delta, \gamma) \in (\mathbb{B}^{\mathscr{B}} \times \mathbb{Z}^n)$ such that $\delta(a) = 1$, $\delta(b) = 1$, $\gamma = \{(3, 5)\}$ is a model of $a \wedge b \wedge (\underline{1}\ ,\ \underline{-1}\ ,\ \underline{1})$.

As one can deduce from Definitions 3 and 6 models of $\mathbb{Z}^n$-*robdd* and mixed formulae have exactly the same form. We say that a $\mathbb{Z}^n$-*robdd* represents a mixed formula if they both have the same models. This leads to this corollary :

**Corollary 1.** *For any mixed formula there is only one* $\mathbb{Z}^n$-robdd *representing it.*

This derives directly from Lemma 1.

## 4 Construction of $\mathbb{B}^k \times \mathbb{Z}^n$-*robdd*

### 4.1 From mixed formulae to $\mathbb{B}^k \times \mathbb{Z}^n$-*robdd*

Algorithms presented below allow to construct $\mathbb{B}^k \times \mathbb{Z}^n$-*robdd* from mixed formulae. A node has to be considered as a structure composed of four elements. An identifier $Id$ which is a positive integer. A label $Lab$ being either a boolean variable or a polyhedra of $\mathbb{Z}^n$ ; two integers $Hi$ and $Low$. The nodes are stored in a set $H$ which is supposed to contain $(0, \emptyset, 0, 0)$ and $(1, \mathbb{Z}^n, 1, 1)$. Given a node $(Id, Lab, Hi, Low)$

- $exists(H, Lab, Hi, Low)$ return $Id$ if $(Id, Lab, Hi, Low) \in H$ and $-1$ if not.
- $insert(H, Lab, Hi, Low)$ chose an identifier $i$ not already used, add $(i, Lab, Hi, Low)$ to $H$ and return $i$
- $Hi$ is the identifier of the $\ll Hi \gg$ part of node $Id$ in the sens of Definition 1.
- $Low$ is the identifier of the $\ll Low \gg$ part of node $Id$ in the sens of Definition 1.

$Omega(F)$ where $F$ is an $\Omega_n$ formula is the procedure that associates a convex polyhedra of $\mathbb{Z}^n$ to $F$. In our tool this conversion is done thanks to the Omega Library [7]. Now the algorithms.

$MakeNode(B, v, u)$ :

1. if $v == u$ then return $v$ endif
2. if $((i := exists(H, B, v, u)) \neq -1)$ then return $i$ endif
3. return $insert(H, B, v, u)$

Test of line 1. eliminates redundant nodes.

$Form2Robdd(F)$ :

```
1.  Compute β an enumeration of the k boolean variables of F
2.
3.  function Build(F,i)
4.      if (i > k) then
5.          if Omega(F) == ∅ return 0
6.          if Omega(F) == Z^n return 1
7.          return MakeNode(Omega(F), 1, 0)
8.      else
9.          u_0 = Build(F[⊥_n / β_i], i + 1)
10.         u_1 = Build(F[⊤_n / β_i], i + 1)
11.         return MakeNode(β_i, u_1, u_0)
12.     endif
13.
14. return Build(F,1)
```

*Build* is a recursive function which executes the so called Shannon expansion of $F$ with respect to $\beta$. The test $(i > k)$ line 4. ends the recursion calls. When this test is satisfied the formula passed to *Build* is necessary an $\Omega_n$ formula since all boolean variables have been replace by $\top_n$ or $\bot_n$ ; the polyhedra denoted by this formula is then computed.

### 4.2 Algebra of $\mathbb{B}^k \times \mathbb{Z}^n$-*robdd*

Classical boolean binary operations can be achieved upon $\mathbb{B}^k \times \mathbb{Z}^n$-*robdd*.

$Apply(op, u, v)$ :

1. if $Id(u) \subseteq \mathbb{Z}^n$ and $Id(v) \subseteq \mathbb{Z}^n$ then return $MakeNode(Omega(op, u, v), 1, 0)$ endif
2. if $Id(u)$ is a boolean variable and $Id(v) \subseteq \mathbb{Z}^n$ then return $MakeNode(Id(u),\ Apply(op, Hi(u), v),\ Apply(op, Lo(u), v))$ endif
3. if $Id(u) \subseteq \mathbb{Z}^n$ and $Id(v)$ is a boolean variable then return $MakeNode(Id(v),\ Apply(op, u, Hi(v)),\ Apply(op, u, Lo(v)))$ endif
4. if $Id(u)$ and $Id(v)$ are boolean variables then
5. if $Id(u) == Id(v)$ then return $MakeNode(Id(u), Apply(op, Hi(u), Hi(v)), Apply(op, Lo(u), Lo(v)))$ endif
6. if $Id(u) < Id(v)$ then return $MakeNode(Id(u),\ Apply(op, Hi(u), v),\ Apply(op, Lo(u), v))$ endif
7. if $Id(u) > Id(v)$ then return $MakeNode(Id(v),\ Apply(op, u, Hi(v)),\ Apply(op, u, Lo(v)))$ endif
8. endif

Now let us return on the example given in the introduction. It was the formulae $(a \wedge (x - y + 1 \leq 0)) \Rightarrow (b \vee (x - y \geq 0))$.

We detail in Figure 3 the calculus of its $\mathbb{U}^n$-*robdd* representation.

### 4.3 A small library

Algorithms presented in the previous section have been implemented in a C++ library available on demand addressed to the first author. Operations on leaves are achieved thanks to the Omega library [7].

## References


1. Huimin Lin. Symbolic transition graph with assignment. In *International Conference on Concurrency Theory*, pages 50–65, 1996.
2. Hubert Garavel Frédéric Lang. Ntif: A general symbolic model for communicating sequential processes with data. *Proceedings of the 22nd IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems FORTE'2002 (Houston, Texas, USA)*, November 2002.

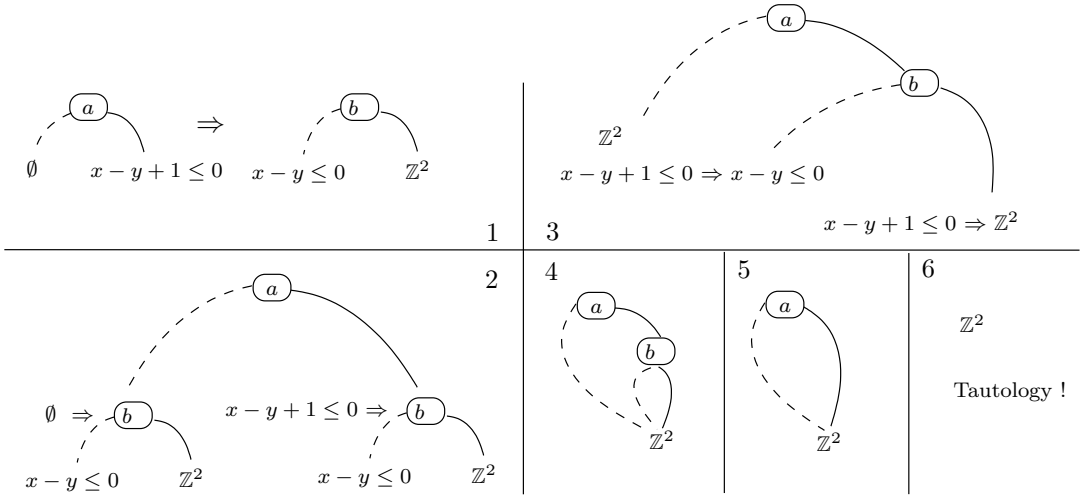



**Fig. 3.** Construction of the $\mathbb{U}^n$-*robdd* of $(a \wedge (x - y + 1 \leq 0)) \Rightarrow (b \vee (x - y \geq 0))$. Decomposition in six steps.


3. V. Rusu, L. du Bousquet, and T. Jéron. An approach to symbolic test generation. In *2nd International Workshop on Integrated Formal Method (IFM'00)*, number 1945, pages 338–357, Dagstuhl, Germany, 2000. Springer-Verlag.
4. J.-C. King. A new approach to program testing. *Proceedings of the international conference on Reliable software, Los Angeles, California*, 21-23:228–233, April 1975.
5. N.Rapin C.Gaston A.Lapitre J.P.Gallois. Behavioral unfolding of formal specifications based on communicating extended automata. *Proceedings of ATVA 2003. First Workshop on Automated Technology for Verification and Analysis Dept. of Electrical Engineering, National Taiwan University*, 2003.
6. Randal E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.
7. Omega 1.2. The Omega Project : Algorithms and Frameworks for Analyzing and Transforming Scientific Programs. 1994.