Bull. Nov. Comp. Center, Comp. Science, 24 (2006), 105–117

SAT vs. SMV for automatic validation of tabular property of superintuitionistic logics

P.A. Schreiner,* N.V. Shilov,† J.V. Grebeneva, S. O

Abstract. This paper considers theoretical background and experimental comparison of two approaches to automatic recognition of tabular property of superintuitionistic logics. A principle opportunity for automatization is based on theoretical results of L.L. Maksimova that were obtained in 1973–2003 and their algorithmic interpretation that was developed recently by P.A. Schreiner. The experimental approaches under study are: (1) Computation Tree Logic (CTL) symbolic model checking, and (2) Boolean satisfiability (SAT) decision procedure. Efficiency of SAT-based approach to tabularity is demonstrated by a number of experiments.

1. Introduction

We present some experience with automatic verification of the tabular (and pretabular) property of superintuitionistic logics. The corresponding algorithmic criterion has been developed by P.A. Schreiner [5] on base of earlier theoretical results of A.V. Kuznetsov and L.L. Maksimova [3, 4] (that have proved a principle decidability of the problem).

Initially the algorithmic criterion from [5] has been implemented on PROLOG and has passed a number of basic tests successfully. This PROLOG implementation is highly reliable, since it follows the mathematical description of the criterion (due to the declarative nature of the implementation language). Unfortunately, this implementation is extremely inefficient: for some tests (with 2–3 variables and 7–8 connectives) it runs several hours.

This negative experience with PROLOG implementation of the algorithmic criterion of tabularity has driven our research of "right" data representation and processing for efficient implementation of the criterion. Since the criterion is formulated in terms of formula refutation at finite frames, it was natural to try data structures that are in use in computer science for verification of properties (presented by logical formulas) of finite models (of computer hardware and software), i.e. in model checking.

The topic of our study is to develop and examine model checking and SAT-based approaches for automatization of checking tabularity of superintuitionistic logics. We have designed, proved correctness, implemented, and compared two reductions of tabularity to

*Supported by INTAS grant 04-77-7080 and RFBR 06-01-00358-a.
†Supported by RFBR grant 06-01-00464-a and Integration Grant N 14 of SB RAS.

- symbolic model verifier SMV [1],
- Boolean satisfiability checking SAT [2].

(We call these reductions SMV-based and SAT-based approaches, respectively.) The main conclusion that we can make on the basis of our comparative testing of these two approaches is acceptable efficiency of SMV-based automatization of tabularity checking and a very good efficiency of SAT-based approach to automatic validation of tabularity property for superintuitionistic logics.

We hope that our comparative study of SAT-based and SMV-based approaches can lead to a new series of test-suits for SATLIB, since tabularity checking is a non-trivial property, it resides in SAT-based as well as an alternative method of validation (SMV-based, in particular). At the same time, positive experience with application of SAT to tabularity encourages us to extend SAT-based approach to pretabularity and other model-theoretic and proof-theoretic properties of superintuitionistic logics that have been discussed in literature [6, 7] (interpolation property, for example).

2. Background theory

Definition 1. *Let C and I be disjoint alphabets of classical and intuitionistic propositional variables. Syntax of classical propositional logic and syntax of intuitionistic propositional logic consist of formulae constructed from classical and intuitionistic propositional variables, respectively, with the help of standard constructs (connectives) for negation ('$\neg$'), conjunction ('$\wedge$'), disjunction ('$\vee$'), and implication ('$\rightarrow$') in accordance with the standard rules.*

We assume that the standard Boolean semantics, the notion of satisfiability and the conjunctive normal form for classical propositional logic are a common knowledge. We refer to [1] for the definition of Computation Tree Logic (CTL), its syntax and semantics in terms of the labeled transition systems. In contrast, let us briefly define Kripke semantics for intuitionistic propositional logic below.

Definition 2. *Kripke frame is a partial order $(W, \leq)$, where 'universe' W is a non-empty set of 'worlds'. Kripke model is a triple $(W, \leq, \models)$, where $(W, \leq)$ is a frame, and $\models$ is a truth-relation between worlds and intuitionistic variables that enjoys the following monotonicity condition: for every $q \in I$ and all $u \leq w \in W$, if $u \models q$ then $w \models q$.*

Definition 3. *Let $(W, \leq, \models)$ be a model. The relation $\models$ can be extended to all worlds and intuitionistic formulae as follows:*

- $u \models \varphi \wedge \psi \;\Leftrightarrow\; u \models \varphi$ *and* $u \models \psi$,

- $u \models \varphi \vee \psi \;\Leftrightarrow\; u \models \varphi$ *or* $u \models \psi$,
- $u \models \varphi \rightarrow \psi \;\Leftrightarrow$ *for every* $w \geq u$, *if* $w \models \varphi$ *then* $w \models \psi$,
- $u \models \neg\varphi \;\Leftrightarrow\; w \not\models \varphi$ *for every* $w \geq u$.

Definition 4. *Let* $(W, \leq)$ *be a frame and* φ *be an intuitionistic formula. The formula* φ *is said to be* valid *in the frame* $(W, \leq)$, *if* $w \models \varphi$ *for every model* $(W, \leq, \models)$ *and every world* $w \in W$*; otherwise let us say that the frame* $(W, \leq)$ *refutes the formula* φ.

Definition 5.
Intuitionistic propositional logic Int *consists of all intuitionistic formulae that are valid in all frames.* Superintuitionistic logic *is a set of intuitionistic formulae containing* Int *that is closed under*

- *Modus Ponens (MP):* $\frac{\varphi\ ,\ \varphi \rightarrow \psi}{\psi}$ *and*
- *Substitution (Sub):* $\frac{\varphi}{\varphi_q(\psi)}$

(where $\varphi_q(\psi)$ *results from* φ *after substitution of* ψ *in* φ *instead of* q*). For every intuitionistic formula* θ, *let* $Int{+}\theta$ *be a superintuitionistic logic that results from extending* Int *by a 'new' axiom that is* θ.

Definition 6. *Let* $\mathcal{K}$ *be a class of Kripke frames, and* L *be a superintuitionistic logic. The class* $\mathcal{K}$ characterizes *the logic* L, *if every formula that belongs to* L *is valid in all frames in* $\mathcal{K}$ *and every formula that does not belong to* L *is refutable by some frame in* $\mathcal{K}$.

We are particularly interested in the following three classes of frames (Figure 1).

Definition 7. *Let* $m > 0$ *be a positive integer.*

- *m-line* lin_m *consists of* m *linearly ordered elements.*
- *m-fan* fan_m *consists of* m *incompatible elements and the smallest one.*
- *m-top* top_m *consists of* m *incompatible elements, the smallest element, and the greatest one.*

Definition 8. *Let* L *be a superintuitionistic logic. Let us say that* L *has the* tabular property, *if it can be characterized by a finite set of finite frames. Let us say* L *has the* pretabular property *if it is maximal among non-tabular logics.*

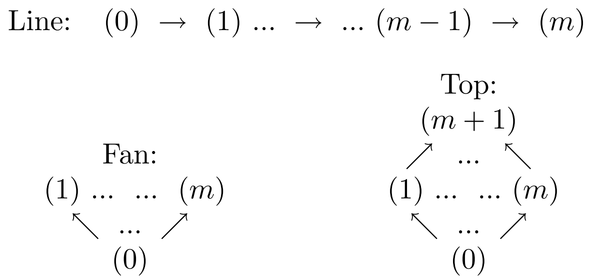

Figure 1. Frames of interest

It has been proved by L.L. Maksimova [4] that there exist exactly three pretabular superintuitionistic logics; these logics are denoted by LC, LP_2, LQ_3. LC is characterized by frames lin_n, $n \geq 1$, LP_2 — by frames fan_n, $n \geq 1$, and LQ_3 — by frames top_n, $n \geq 1$. The following statement has been proved in [5] and leads to a method of checking pretabularity.

Statement 1.
Let ψ be an intuitionistic formula and let L be superintuitionistic logic Int+ψ. Let N be the number of different intuitionistic variables in ψ, r be the total number of instances of '$\rightarrow$' and '$\neg$' in ψ (or 1, if ψ is '$\rightarrow$'- and '$\neg$'-free), and $m = \min(2^N, r)$. Then

1. *L=LC, iff ψ is valid in $lin_{(N+1)}$, but fan_2 and top_2 both refute ψ.*
2. *L=LP_2, iff ψ is valid in fan_m, but lin_3 refutes ψ.*
3. *L=LQ_3, iff ψ is valid in top_m, but fan_2 and lin_4 both refute ψ.*

It has been proved by A.V. Kuznetsov [3] that every superintuitionistic logic L that has not the tabular property is contained in some superintuitionistic logic that enjoys the pretabular property. It implies (see [5]) that a superintuitionistic logic has the tabular property iff it is not contained in any of the three logics LC, LP_2, or LQ_3 that have the pretabular property. In combination with statement 1, this argument implies correctness of the next statement [5]) that is a decision criterion for tabularity.

Statement 2.
Let ψ be an intuitionistic formula and let L be superintuitionistic logic Int+ψ. Let N be the number of different intuitionistic variables in ψ, r be the total number of instances of '$\rightarrow$' and '$\neg$' in ψ (or 1, if ψ is '$\rightarrow$'- and '$\neg$'-free), and $m = \min(2^N, r)$. Then L has the tabular property iff three frames $lin_{(N+1)}$, fan_m, and top_m altogether refute ψ.

It is also known that tabularity is $\mathcal{NP}$-complete, while pretabularity is $co\mathcal{NP}$-hard [6, 7].

3. Implementing decision criterion

The above statements 1 and 2 have been implemented first as an experimental PROLOG-program for checking pretabularity and tabularity. Unfortunately, efficiency is not a strong point of the experimental PROLOG-program. In particular, processing of the following formula

$$(\neg p \vee \neg(\neg p)) \wedge ((\neg(\neg p) \wedge ((q \to p) \to (r \to q)) \wedge ((r \to q) \to r)) \to r)$$

(that characterizes LQ_3) requires several hours. This experience leads us to the idea to try SMV-based and SAT-based approaches for checking the tabular property.

3.1. SMV-based approach

Definition 9. *Let $\mathcal{F} = (W, \leq)$ be a finite frame. For every world $u \in W$ let $next(u) = \{w \in W : u < w$ and there is no $v \in W$ such that $u < v < t\}$ be the set of all immediate successors of u in $(W, \leq)$. Then let the binary relation $\to$ be $\{(u, u), (u, w) : u, w \in W,$ and $w \in next(u)\}$.*

Observe that for every frame $\leq$ is the transitive closure of $\to$.

In accordance with the above definition, every finite intuitionistic Kripke frame can be considered as a transition system. Hence every finite intuitionistic Kripke model $(W, \leq, \models)$ can be considered as a labeled transition system $(W, \to, \models)$. It implies that semantics of Computation Tree Logic is defined in all finite intuitionistic Kripke models too.

Definition 10. *A formula ϕ of Intuitionistic Propositional Logic and a formula ψ of Computation Tree Logic are (semantically) equivalent iff for every intuitionistic Kripke model $(W, \leq, \models)$ and for every world w the following holds: $w \models \phi \Leftrightarrow w \models \psi$.*

Definition 11.

IPL frame checking problem *Input: a formula of Intuitionistic Propositional Logic ξ, a finite Intuitionistic Kripke frame $(W, \leq)$. Output: "Valid" if ξ is valid on $(W, \leq)$, and "Refutable" otherwise.*

CTL model checking problem *Input: a formula of Computation Tree Logic ξ, a finite labeled transition system $(W, \to, \models)$, a state $u \in W$. Output: "Valid" if $u \models \xi$, and "Invalid" otherwise.*

In particular (according to statement 2), for validation of the tabularity property of a superintuitionistic logic L = $Int+\psi$ (where ψ is an intuitionistic formula) one can frame check ψ on three frames $lin_{(N+1)}$, fan_m, and top_m (where N is the number of different intuitionistic variables in ψ, r is the total number of '$\to$' and '$\neg$' in ψ, and $m = \min(2^N, r)$): L enjoys the tabular property iff all these three frames $lin_{(N+1)}$, fan_m, and top_m altogether refute ψ.

The notion of semantic equivalence leads to the following idea how to frame check an intuitionistic formula ϕ in a finite frame $(W, \leq)$:

- first find a formula $\psi \in$CTL that is semantically equivalent to ϕ;
- for every possible monotone $\models$ on $(W, \leq)$ and every minimal world w of $(W, \leq)$, check $w \models (\neg\psi)$ by an available model checker;
- if $w \models (\neg\psi)$ for any monotone $\models$ and world w, then $(W, \leq)$ refutes ϕ, otherwise ϕ is valid on $(W, \leq)$.

Algorithm 1. Let *ctl* be the following recursive algorithm that translates formulas of Intuitionistic Propositional Logic to formulas of Computation Tree Logic:

- for every intuitionistic variable p, let $ctl(p) = p$;
- $ctl(\phi \wedge \psi) = ctl(\phi) \wedge ctl(\psi)$ and $ctl(\phi \vee \psi) = ctl(\phi) \vee ctl(\psi)$;
- $ctl(\neg\phi) = \mathbf{AG}(\neg(ctl(\phi)))$ and $ctl(\phi \to \psi) = \mathbf{AG}((ctl(\phi) \to ctl(\psi))$.

Statement 3. *Algorithm 1 translates every intuitionistic formula to a formula of CTL that is semantically equivalent in every finite intuitionistic Kripke model.*

The above statement 3 provides an opportunity for exploiting a model checker for CTL for frame checking IPL. Note that a popular model checker SMV [1] has some virtues for this application. It has proved its efficiency in practice and its input language is also relatively simple so that the overhead of repeated execution and model generation is low.

Complexity is the only obstacle for using a model checker for frame checking. In general, for a given frame $(W, \leq)$ the number of possible satisfiability relations is in $O(2^{N\times|W|})$, where N is the number of variables in a formula to be checked. In the case of tabularity checking $|W|$ is $O(2^N)$ (according to method 2, we have to check frames fan_m and top_m, where $m = O(2^N)$). Hence the number of labeled transition systems over these fans and tops is $O(2^{N\times 2^N})$. At the same time, CTL model checking complexity of a formula ϕ in a finite labeled transition system $(W, \leq, \models)$ is $O((|W| + |\leq|) \times |\phi|)$ [1]. Therefore the complexity of tabular property via model checking is

Figure 2. Examples of a sorted and an isomorphic non-sorted model

$O(2^N \times 2^{N \times 2^N})$, i.e. it is double exponential due to plenty of models over fans and tops.

However, in our case we deal with lines, fans, and tops only. Observe that many models over fans are "symmetric", i.e. isomorphic to each other. Similarly, many models over tops are symmetric.

Definition 12. *Labeled transition systems* $(W_1, \leq_1, \models_1)$ *and* $(W_2, \leq_2, \models_2)$ *are said to be symmetric (or isomorphic) iff there exists a mapping* $f: W_1 \overset{1-1}{\longrightarrow} W_2$ *such that*

$$\forall u, v \in W_1 : \ u \leq_1 v \Leftrightarrow f(u) \leq_2 f(v)$$

$$\forall u \in W_1, p \in Int : \ u \models_1 p \Leftrightarrow f(u) \models_2 p.$$

Observe that for frame checking via model checking we can restrict ourselves by non-symmetric labeled transition systems over fans and tops.

Definition 13. *Let* ϕ *be some IPL or CTL formula; let us adopt and fix a (linear) order on intuitionistic variables* q_0, ... q_n *in* ϕ. *Let* $(W, \leq, \models)$ *be a labeled transition system over* fan_m *or* top_m*; then for every state* w *in* (1), ... (m) *let* A_w *be a Boolean vector in* $\{T, F\}^n$ *such that for every* $i \in [0..n]$, $(A_w)_i = T$ *if* $w \models q_i$, *and* $(A_w)_i = F$ *otherwise.* $(W, \leq, \models)$ *is said to be sorted if a sequence of Boolean vectors* $A_{(1)}$, ... $A_{(m)}$ *is sorted in the alphabet order.*

It is straightforward that every labeled transition system over a fan or a top is isomorphic to some sorted one. (It is sufficient just to sort $A_{(1)}$, ... $A_{(m)}$ in the alphabet manner.) An example of a sorted labeled transition system (with a single anonymous variable) and an isomorphic non-sorted label transition system is presented in Fig. 2.

Let us summarize the above discussion in the SMV-based approach to tabularity checking.

Algorithm 2.

1. Input the formula ξ of IPL; count the number N of propositional variables and number r of instances of '$\neg$' and '$\rightarrow$' in ξ; define $m = min(r, 2^N)$;

2. translate ξ to a semantically equivalent formula $ctl(\xi)$ of CTL;
3. for every possible monotone $\models$ on lin_{N+1}, check $w \models ctl(\xi)$ in (0) by SMV; if SMV always answers "Valid", then the superintuitionistic logic $Int + \xi$ is not tabular and we are done, otherwise — continue from next step;
4. for every possible monotone and sorted $\models$ on fan_m, check $w \models ctl(\xi)$ in (0) by SMV; if SMV always answers "Valid", then the superintuitionistic logic $Int + \xi$ is not tabular and we are done, otherwise — continue from the next step;
5. for every possible monotone and sorted $\models$ on top_m, check $w \models ctl(\xi)$ in (0) by SMV; if SMV always answers "Valid", then the superintuitionistic logic $Int + \xi$ is not tabular and we are done, otherwise — continue from the next step;
6. in this case the superintuitionistic logic $Int + \xi$ is tabular and we are done.

Let us estimate the number of iterations of SMV launches. It is straightforward that the number of monotone $\models$ on lin_{N+1} is $(N+2)^N$. Hence the number of SMV launches at step 3 is $O((N+2)^N)$. We have to make some computations for a correct estimation of the number of SMV launches at steps 4 and 5.

For every $m, v \geq 1$ let $M(m, v)$ be

$$|\{(a_1, ...a_m) : a_1, ...a_m \in [1..v] \text{ and } a_1 \leq ... \leq a_m\}|.$$

Lemma 1. $M(m, v) = \begin{pmatrix} m+v-1 \\ m \end{pmatrix}$.

Statement 4. *Let N be the number of propositional variables. The number of sorted intuitionistic Kripke models on fan_{2^N} and top_{2^N} is in* $O\left(\frac{4^{2^N}}{2^{N/2}}\right)$

Hence the number of SMV launches at steps 4 and 5 of SMV-based approach is $O\left(\frac{4^{2^N}}{2^{N/2}}\right)$. SMV implements the model checking algorithm with complexity $O((|W| + |\leq|) \times |\phi|)$ [1]. Therefore the overall complexity of this approach is $O(2^{N/2}4^{2^N})$.

3.2. SAT-based approach

Algorithm 3.
Let $\mathcal{F} = (W, \leq)$ be a finite frame. For every world $u \in W$ let $next(u) = \{w \in W : u < w \text{ and there is no } v \in W \text{ such that } u < v < w\}$ be the

set of all immediate successors of u in $(W, \leq)$. Let θ be an intuitionistic formula. For every subformula ξ of θ and every $u \in W$, let p^u_ξ be a fresh classical variable. Let $\mathcal{F}(\theta)$ be the following classical propositional formula $(\bigwedge_{u \in W,\ \xi \in \theta} \varepsilon^u_\xi)$, where ε^u_ξ is defined according to the syntax of ξ:

- if ξ is some intuitionistic variable then ε^u_ξ is $p^u_\xi \to \bigwedge_{w \in next(u)} p^w_\xi$;
- if ξ is a 'combined' formula then

$$\begin{cases} p^u_{\varphi\wedge\psi} \leftrightarrow p^u_\varphi \wedge p^u_\psi \\ p^u_{\varphi\vee\psi} \leftrightarrow p^u_\varphi \vee p^u_\psi \\ p^u_{\neg\varphi} \leftrightarrow \neg p^u_\varphi \wedge (\bigwedge\limits_{w \in next(u)} p^w_{\neg\psi}) \\ p^u_{\varphi\to\psi} \leftrightarrow (p^u_\varphi \to p^u_\psi) \wedge (\bigwedge\limits_{w \in next(u)} p^w_{\varphi\to\psi}) \\ p^u_{\varphi\leftrightarrow\psi} \leftrightarrow (p^u_\varphi \leftrightarrow p^u_\psi) \wedge (\bigwedge\limits_{w \in next(u)} p^w_{\varphi\leftrightarrow\psi}) \end{cases}.$$

Statement 5.
For every finite frame $\mathcal{F} = (W, \leq)$ and every intuitionistic propositional formula θ, there exists a propositional formula $\mathcal{F}(\theta)$ such that the following is equivalent (where $min(W)$ is the set of all minimal elements of $(W, \leq)$):

- *$\mathcal{F}$ refutes the intuitionistic formula θ;*
- *the classical propositional formula $\mathcal{F}(\theta) \wedge (\bigvee_{v \in min(W)} (\neg p^v_\theta))$ is satisfiable.*

The formula $\mathcal{F}(\theta)$ can be constructed in time quadratic in the number of worlds in the frame and linear in the number of connectives in the intuitionistic formula.

Let $\mathcal{F} = (W, \leq)$ be a finite frame and θ be an intuitionistic formula. Note that $\mathcal{F}(\theta) \wedge (\bigvee_{v \in min(W)} (\neg p^v_\theta))$ is 'almost' a CNF-formula: it remains to transform all 'building-blocks' ε^u_ξ into CNF format. For every world $u \in W$ and every subformula $\xi \in \theta$, let $CNF(\varepsilon^u_\xi)$ be the CNF-formula that corresponds to ε^u_ξ. Let $CNF(\mathcal{F}, \theta)$ be $\bigwedge_{u \in W, \xi \in \theta} CNF(\varepsilon^u_\xi)$.

Observe also that lines, fans and tops have a single minimal world 0. This observation together with statements 2 and 5 immediately implies the following SAT-based algorithm for validation of the tabular property.

Algorithm 4.
Let ψ be an intuitionistic formula and L be superintuitionistic logic $Int{+}\psi$. Let N be the number of different intuitionistic variables in ψ, r be the total number of instances of '$\to$' and '$\neg$' in ψ (or 1, if ψ is '$\to$'- and '$\neg$'-free),

and $m = \min(2^N, r)$. Then L has the tabular property iff the following propositional CNF-formulae are satisfiable:

- $(\neg p^0_\theta) \wedge CNF(lin_{(N+1)}, \psi)$,
- $(\neg p^0_\theta) \wedge CNF(fan_m, \psi)$.
- $(\neg p^0_\theta) \wedge CNF(top_m, \psi)$.

Algorithm 4 has been implemented as a LISP-program coupled with SAT-solver Z-Chaff [8] for checking tabularity. The LISP-program translates input intuitionistic formula ψ into three classical propositional formulae as prescribed by the algorithm. (We refer to these formulae as the line, fan and top components, respectively.) The syntax of output classical formulae is a so-called DIMACS cnf-format [9], the input format of Z-Chaff. Each of three generated components is passed to Z-Chaff for further processing.

4. Experimental results and conclusion

Table 2 represents the "basic" test suite that consists of 18 formulae. These formulae are given in the input syntax, where '&' stays for conjunction, '%' — for disjunction, '∼' — for negation, '$->$' – for implication, and '$<->$' — for equivalence. PROLOG, SMV-based, and SAT-based approaches have been tested against each other on these tests, all approaches have given correct answers in all basic cases.

Table 2 represents the corresponding experimental data for SAT-based and SMV-based approaches for basic tests. In this table every row has two lines:

- the first is the overall run time of the boolean-translator and Z-Chaff,
- the second is the overall run time of CTL-translator and SMV.

Tests have been executed on computer Celeron 733 RAM 512, time is in seconds. These data have proved efficiency and perspectives of SAT-based approach in comparison with SMV-based approach to tabularity checking. (Recall that PROLOG-program executes test #18 for several hours.)

The next Table 1 represents SAT-approach experimental data for three randomly generated "large" formulae. These formulae contains 10 intuitionistic variables and 200 connectives '¬' and '→' (i.e. $N = 10$, $r = 200$, and hence $m = 200$). After translation to cnf, the most complicated classical formula has about 2,000 classical variables and approximately 74,000 clauses. The table gives two figures for each of the three intuitionistic formula and each frame in lin_{N+1}, fan_m, and top_m:

- translation time to a corresponding component,
- the time for solving the component by Z-Chaff.

Table 1. Experimental data for "large" formulae

#	*lin*	*fan*	*top*
1	32.796604	280.14243	348.19058
	0.018997	1.59376	0.147978
2	35.170944	279.77386	310.212
	0.018997	1.59776	0.292955
3	775.3111	260.6953	309.8433
	0.007999	1.02184	0.112984

Experiments have been performed on the same computer Celeron 733 RAM 512.

Experimental data demonstrates that translation is the most time-expensive part of SAT-base approach: in our experiments it requires up to minutes while Z-Chaff works one and a half second at most. Hence more efficient implementation of translation is the main opportunity for faster SAT-based automatic recognition of tabularity of superintuitionistic logics.

References

[1] Clarke E.M., Grumberg O., Peled D. Model Checking. — MIT Press, 1999.

[2] Hoos H. H., Stützle T. SATLIB: An Online Resource for Research on SAT // SAT 2000. — IOS Press, 2000. — P. 283–292. — Available online at `www.satlib.org`.

[3] Kuznetsov A.V. Some properties of the lattice of varieties of pseudo-boolean algebras // 11th Sovjet Algebraic Colloquium, Abstracts, Kishinev, 1971. — P. 255–256.

[4] Maksimova L.L. Pretabular superintuitionistic logics // Algebra and Logic. — 1972. — N 11. — P. 558–570.

[5] Maksimova L.L., Schreiner P.A. The algorithms for recognition of tabularity and pretabularity in extensions of the intuitionistic calculus (in Russian) // Vestnik of Novosibirsk State University, 2006 (to appear).

[6] Maksimova L.L., Voronkov A. Complexity of some problems in modal and superintuitionistic logics // Bull. Symbol. Logic. — 2000. — N 6. — P. 118–119.

[7] Gabbay D.M., Maksimova L.L. Interpolation and Definability. Model and Intuitionistic Logic. — Oxford Press, 2005.

[8] Moskewicz M.W., Madigan C.F., Zhao Y. et al. Chaff: Engineering an Efficient SAT Solver // 38th Design Automation Conf. (DAC '01), 2001.

[9] DIMACS Challenge — Satisfiability: Suggested Format
http://www.intellektik.informatik.tu-darmstadt.de/SATLIB/
Benchmarks/SAT/satformat.ps

Table 2. Basic test suite

#	*Formula*	*lin*	*fan*	*top*
1	((∼ ((∼ p)&((q− > p)− > q)))− > q)	0.064607	0.175381	0.152623
		0.142459	0.026834	0.027246
2	(p%(p− > (q%(∼ q))))	0.084462	0.048921	0.072007
		0.261766	0.438981	0.156537
3	((p− > q)%(q− > p))	0.094389	0.039593	0.05517
		0.414965	0.129391	0.288677
4	(r%(r− > ((p− > q)%(q− > p))))	0.02207	0.037222	0.106619
		3.299572	5.036643	7.817828
5	((p− > (q%r))− > ((p− > q)%(p− > r)))	0.028764	0.028486	0.039976
		3.303151	2.137837	4.469693
6	((∼ p)%((∼ p)− > q))	0.083483	0.028559	0.039756
		0.419430	0.053062	1.045101
7	((p%(p− > (q%(∼ q))))&((∼ p)%(∼ (∼ p))))	0.011057	0.011583	0.015712
		0.264428	0.053096	0.160968
8	((∼ p)%(∼ (∼ p)))	0.006074	0.008622	0.012235
		0.077418	0.052490	0.129389
9	((p%(p− > (q%(∼ q))))&	0.102516	0.082201	0.168199
	&(((p− > q)%(q− > p))%(p < − > (∼ q))))	0.266742	0.183699	0.162006
10	(((p− > q)%(q− > p))%(p < − > (∼ q)))	0.070131	0.176502	0.16075
		0.415192	0.183514	0.396760
11	(p%(∼ p))	0.008256	0.011906	0.072834
		0.051440	0.050803	0.051778
12	(p%(p− > q))	0.03509	0.059681	0.080155
		0.262085	0.051673	0.052829
13	(r%(r− > (p%(p− > q))))	0.069186	0.018838	0.027349
		2.295202	1.994148	0.259426
14	((p%(p− > q))%(q− > r))	0.043491	0.123054	0.091374
		2.293015	0.131198	0.270157
15	(q%(q− > ((∼ p)%(∼ (∼ p)))))	0.03584	0.100505	0.062759
		0.414598	1.224612	1.552051
16	(r%(r− > (p%(p− > (q%(∼ q))))))	0.016239	0.021479	0.030727
		2.291307	5.043855	7.820232
17	(((∼ (∼ p))&((q− > p)− > q))− > q)	0.018376	0.074069	0.02684
		0.182351	1.203456	0.684214
18	(((∼ p)%(∼ (∼ p)))&	0.029243	0.04943	0.125219
	&(r%(r− > (p%(p− > (q%(∼ q)))))))	2.190208	0.054770	62.888234