Atomless Boolean Algebras
Algorithms and Applications

Ohad Asor (ohad@idni.org)

21 Apr 2024

Video 1

1. Preliminary material
2. Quantifier eliminiation and decision procedures in atomless BA.
3. NSO (Nullary second order logic)
4. Guarded Successor
5. Applications

Prereqs:

1. You need to know what is a ring, and very, very basic ring theory.
2. You need to know what is a BA.
3. people mix the two-element BA with general BA. I assume that you know the two-element BA quite well. CNF, BDD, SAT.
4. First Order Logic.
5. Temporal logics.

Video 2 - basic definitions and terminology.

1. A BR is any ring satisfying $xx = x$. In any BR, $x + x = 0$

   $$(x + x)(x + x) = x + x$$

   $$x + x = 0$$

   $$x = -x$$

   any BR is commutative:

   $$(x + y)(x + y) = x + y$$

   $$xy = -yx$$

   $$xy = yx$$

2. Ring has the signature $(\cdot, +, 0, 1)$. BA, as you know, has the signature $(\cap, \cup, ')$. Any BR is a BA and vice versa, by:

$$\cdot = \cap$$

$$a \cup b = a + b + ab$$
$$a' = 1 + a$$
$$a + b = ab' \cup a'b$$

3. a BF is [multivariate] polynomial function over a BR. in ANF (algebraic normal form):
$$f(x) = ax + b$$

4. SBF is a BF where all constants appearing in it are either 0 or 1.

5. Boole's normal form, or decomposition, Shannon's decomposition (mistakingly),
$$\begin{aligned} f(x) &= ax + bx' \\ &= ax \cup bx' \\ &= xf(1) \cup x'f(0) \end{aligned}$$
disjoint union is same as disjoint symdiff.

6. Minterm normal form: a minterm is of the form $X^A = \prod_i x_i^{a_i}$ where $x^1 = x$ and $x^0 = x'$.
$$xyz$$
$$xy'z$$
$$xy'z'$$
any two different minterms are disjoint.
$$f(X) = \bigcup_{A \in 2^n} f(A) X^A$$
as a conclusion, any BF is fully determined by its values on $0, 1$.
$$f(x) = \{1, 2\} \cap x$$
$$1 \equiv \mathbb{N}$$
$$f(1) = \{1, 2\}$$

Video 3: Zeros

$$f(x) = ax \cup bx' = 0$$

$$f(x) = 0 \leftrightarrow b \le x \le a'$$

$$a = f(1)$$

$$b = f(0)$$

$$x \le y \leftrightarrow xy' = 0 \leftrightarrow xy = x$$

in partular, a zero exists, iff $b \le a'$ alternatively iff $ab = 0$ iff $f(0) f(1) = 0$. Boole's consistency condition.

$$\left[0 = \prod_{A \in 2^n} f(A)\right] \leftrightarrow \exists X. f(X) = 0$$

if an SBF has a zero in one BA, then it has a zero in all BAs.

General Reproductive Solution. given $f(x)$ and assume it has a zero. define $g(x) = x + f(x)$.

1. the range of g is precisely all zeros of f.
   (a) if $f(x) = 0$ then x is in the range of g. $g(x) = x + f(x) = x$
   (b) $f(g(x)) = 0$

$$f(x + f(x)) = f(g(x)) = abx' + abx = 0$$

2. all zeros of f are fixed-points of g.

Video 4: Systems of Equations and Inequations

1. A [finite] set of BFs over atomless BA has a common nonzero iff none of the BFs is identically zero.

$$\left[\exists X. \bigwedge_i g_i(X) \ne 0\right] \leftrightarrow \left[\bigwedge_i \exists X. g_i(X) \ne 0\right]$$

2. Rudeanu's terminology. "Boolean functions and equations". Elementary GSBE

$$\bigwedge_i f_i(X) = 0$$

$$\bigwedge_i g_i(X) \ne 0$$

3. "squeeze the positives".

$$f(X) = 0$$

$$\bigwedge_i g_i(X) \ne 0$$

$$f(X) = 0 \wedge h(X) = 0 \leftrightarrow f(X) \cup g(X) = 0$$

4. wlog, consider the univariate case

$$\exists x.$$

$$f(x) = 0$$

$$\bigwedge_i g_i(x) \neq 0$$

$$f(0) f(1) = 0$$

$$\exists x. \bigwedge_i g_i(x + f(x)) \neq 0$$

$$f(0) f(1) = 0$$

$$\bigwedge_i g_i(f(0)) \cup g_i(f'(1)) \neq 0$$

$$f(0) f(1) = 0$$

$$\bigwedge_i \exists x. f'(x) g_i(x) \neq 0$$

$$\bigwedge_i f'(0) g_i(0) \cup f'(1) g_i(1) \neq 0$$

5. qelim in the theory of BA. the "standard" theory of BA has the signature $(\cap, \cup, ', 0, 1)$. here, we assume that the theory of BA is interpreted in a fixed structure, and, that there is a constant symbol in the signature interpreted in the BA, for each BA element.

Video 5: Additional properties of BFs.

1. Hall's marriage theorem. set theoretic. "system of distinct representatives".

2. minterm normal form of formulas. so far, atomic formula in the lang of BA, was of the form $f(X) = 0$. in MNT of formulas, each atomic formula is of the form $aX^A = 0$.

$$f(X) = \bigcup_{A \in 2^n} f(A) X^A = 0$$

$$\bigwedge_{A \in 2^n} f(A) X^A = 0$$

3. justify set theoretic language. Stone's representation theorem for Boolean algebras: in any BA, each element can be identified with a set, while the boolean operations are the standard set operations.

4. Lemma: in any BA, the system

$$X^{A_i} \geq b_i$$

has a solution, iff $b_i b_j = 0$ whenever $A_i \neq A_j$. Necessity is immediate. By induction on the number of variables. First assume that all A_i are different, because if not, then

$$X^{A_1} \geq b_1$$

$$X^{A_1} \geq b_2$$

$$X^{A_1} \geq b_1 \cup b_2$$

for single variable,

$$x \geq b_1$$

$$x' \geq b_2$$

$$b_1 \geq b_1$$

$$b_1 b_2 = 0$$

for the induction step, separate a distinguished variable x, and write the system as

$$\bigwedge_{i \in I} x X^{A_i} \geq b_i$$

$$\bigwedge_{i \in J} x' X^{B_j} \geq c_j$$

suppose we have an X satsifying

$$\bigwedge_{i \in I} X^{A_i} \geq b_i$$

$$\bigwedge_{i \in J} X^{B_j} \geq c_j$$

and suppose $|I| \neq 0$, set $x = \bigcup_i b_i$. then

$$\bigwedge_{i \in I} X^{A_i} \bigcup_m b_m = \bigwedge_{i \in I} \bigcup_m b_m X^{A_i} \geq \bigwedge_{i \in I} b_i X^{A_i} \geq b_i$$

$$\bigwedge_{i \in J} X^{B_j} \bigcap_m b'_m \geq \bigwedge_{i \in J} X^{B_j} c_j \geq c_j$$

$$c_j \leq \bigcap_m b'_m$$

$$\bigcup_m b_m c_j = 0$$

5. when we have
$$a_i X^{A_i} \neq 0$$
$$a_i X^{A_i} \geq b_i$$
but in atomless BA, all cardianlities are infinite.

6. for any BF f we have
$$\bigcap_x f(x) = \bigcap_x f(x) f(x') = f(0) f(1)$$
$$f(x) f(x') = f(0) f(1)$$
$$ax + bx'$$
$$ax' + bx$$
$$abx + abx' = ab$$
$$\bigcup_x f(x) = f(0) \cup f(1)$$
$$\sum_x f(x) = f(0) + f(1)$$

7. tri-linearity.
$$f(x + y + z) = f(x) + f(y) + f(z)$$
$$f(x) = ax + b$$
$$f(x + y + z) = ax + ay + az + b$$

8. $f(f(f(x))) = f(x)$.

9. weakly ω-categorical theory: an equivalent definition of ω-categorical theory if the following: there are only finitely many formulas, with finitely many fixed free variable symbols, up to logical equivalence. in weakly ω-categorical theory, there are only finitely many formulas, with finitely many fixed free variable symbols and constant symbols, up to logical equivalence.

10. the theory of atomless BA is weakly ω-categorical: first it is enough to consider quantifer-free formula, since we have quantifier elimination. consider formulas in minterm normal form. they are boolean combination, those are SBFs of atomic formulas. in N vars, there are 2^{2^N} SBFs. how many atomic formulas we can form with n variables and k constants? so in our case, $N = k2^n$ so the final number is $2^{2^{k2^n}}$.
$$cX^A$$
$$k2^n$$

11. in weakly ω-categorical theories, we can have recurrence relations.

$$\phi_n(x) = \exists y.\phi_{n-1}(y) \wedge \psi(x,y)$$

,

$$\phi_1(x) = \dots$$
$$f_n(x,y) = f_{n-1}(y,x) \cup g(x)$$
$$f_1(x) = \dots$$

while unfolding, due to the finiteness property, there will be a loop, either a fixed point, or a partial fixed point.

Video 6: Complexity, quantification over [higher-order] BFs (SBFs), increase complexity.

1. Kozen. The satisfiability of a formula in the lang of ba, for infinite ba, is *complete*

$$NEXPTIME \subset \bigcup_{c>0} STA(*, c^n, n) \subset EXPSPACE$$

for finite bas, this is very simply PSPACE complete by direct reduction QBF.

2. suppose

$$\exists f(x,y).f(f(x,x),y) = f(y,f(y,x))$$
$$f(x) = \sum_{A\in 2^n} f(A)X^A$$

you can convert a single arity n function quantifier, into two $n-1$ arity function quantifiers: simply Boole's decomposition

$$f(x,X) = xg(X) + x'h(X)$$

$$f(x,X) = xg(X) + h(X)$$

3. the case for higher-order BFs is completely analogous. same for SBFs. which gives the theory of BAs of cardinalities 2^{2^n}. all countable atomless BAs, are isomorphic. some manifestations of those:

   (a) each BA element is a finite union of left-closed right-open intervals over the rational numbers.

   (b) the BA of SBFs with unboundedly many variables.

   (c) clopen sets in the Cantor set (middle third).

   (d) LTA (Lindenbaum-Tarski algebras) of logics with infinite signature. up logical equivlance.

$$\forall x.x \neq 0 \rightarrow \exists y.0 < y < x$$

4. all atomless BAs are elementarily equivalent.

5. finite model theory: model-checking finite models with a second-order logic formula, captures the PH. afaik, no remotely good algorithms exist. my definition of good algorithm: is an algorithm that runs fast on many "easy" instances. so if the domain is of size n, then you need $\log n$ bits for each element. then you can convert the model to one with domain of size 2, by increasing all arities times $\log n$. similarly you modify the formula. now any k-ary relation is nothing but a SBF with $k \log n$. all is left to be done is to encode the structure as a formula, conjunct with the input formula, and check for satisfiability.

6. extend the theory of BA to make it able to "solve" any problem in ELEMENTARY, so this should be complete for a certain NONELEMENTARY complexity class.

7. from finite model theory, we know that HOL captures ELEMENTARY.

Video 7: NSO, Nullary Second Order Logic

1. the main point, is to have a language that can speak of its own sentences.

2. fix a logic $\mathcal{L}$ that makes a countable atomless LTA. NSO[$\mathcal{L}$] will be decidable iff $\mathcal{L}$ is decidable.

3. NSO[$\mathcal{L}$] is going to be theory of BA interpreted in NSO[$\mathcal{L}$].

4. in NSO[$\mathcal{L}$] each constant is in curly brackets.

5. jump right away to an example:

$$\forall x.x \cup \{\exists x.x = x'\} = 1$$

$$\neg\exists x.\,(x \cup \{\exists x.x = x'\}) + 1 \neq 0$$

$$\{\exists x.x = x'\} = 1$$

$$0 = 1$$

at the bottom of the recursion, there is no curly brackets, so the only constants appearing in this bottom level, are $0, 1$.

6. product algebra, or in other words, the many-sorted theory of BA.

7. add infinitely many symbols to the signature to make the LTA of NSO atomless.

Video 8: GS (Guarded Successor)

1. Time-Compatible Structure (TC structure). given some domain $\mathcal{D}$,a finite-time TC stucture over $\mathcal{D}$ will have as its domain $\mathcal{D}^*$, and an infinite time TC structure $\mathcal{D}^\omega$. finite/infinite are going to behave the same mostly due to Lowenheim-Skolem. a TC structure is also equipped with a prefix-preserving (aka TC) function $f : \mathcal{D}^* \to \mathcal{D}^*$. so if s_1 is a *strict* prefix s_2, then $f(s_1)$ is a strict prefix of s_2.

2. note that this models programs.

3. this extends to trees. by considering multiple successor relations.

4. $f : \mathcal{D}^* \to \mathcal{D}^*$ can also be written as $f : (\mathbb{N} \to \mathcal{D}) \to (\mathbb{N} \to \mathcal{D})$, but if f is TC then can be typed as $f : \mathbb{N} \to (\mathcal{D} \to \mathcal{D})$

5. Bounded Lookback (BL). a TC structure is of BL$[k]$, if $f(n)$ depends only on $f(n-1), \dots, f(n-k)$.

$$f(n) = f(n-1) + 1$$

$$f(1) = 5$$

$$x_n = x_{n-1} + 1$$

6. a function of BL$[k]$ can be typed as a pair of functions, one of type $f : (\mathcal{D} \to \mathcal{D})^k \to (\mathcal{D} \to \mathcal{D})$ and another of type $[k] \to (\mathcal{D} \to \mathcal{D})$.

7. any formula, in virtually any logic, with $2k+2$ free variables, can be seen as defining a set of BL$[k]$ structures.

$$\forall x_1 \exists y_1 \forall x_2 \exists y_2 \dots \bigwedge_{n=k}^{\infty} \phi(x_n, y_n, x_{n-1}, y_{n-1}, \dots x_{n-k}, y_{n-k}, y_1, y_2, \dots)$$

8. for satisfiability. given $\phi(x_n, x_{n-1}, y_n)$ define $\phi_n(x_1)$ to be a formula that says "exists a TC structure of length n starting with x_1"

$$\phi_2(x_1) = \forall y_2 \exists x_2 . \phi(x_2, x_1, y_2)$$

$$\phi_n(x_1) = \forall y_2 \exists x_2 \forall y_3 \exists x_3 . \phi(x_2, x_1, y_2) \wedge \phi_{n-1}(x_3, x_2, y_3)$$

$$\exists x_1 . \phi_n(x_1)$$

9. temporal logic, over infinite alphabet, that comes not only with equality, but with a rich theory: atomless ba.

10. moreover, it is decidable whether "forall input exists a time-compatible output".

11. two major deicdable fragments of fol are the two-vars and the guarded.

12. for example $\phi(x_n, x_{n-1}, y_n)$ can be written as

$$\forall nk . s(n,k) \to \phi(x(n), x(k), y(k))$$

$$\forall nk . s(n,k) \wedge s(k,m) \to \phi(x(n), x(k), x(m), y(k))$$

$$\forall nk . s(n,k) \to (s(k,m) \to \phi(x(n), x(k), x(m), y(k)))$$

13. so succesor relations appear only as guards, and s.t. the relative position of all position variables is fully determined.

$$\exists nk.s\left(n,k\right)\wedge s\left(k,m\right)\wedge\phi\left(x\left(n\right),x\left(k\right),x\left(m\right),y\left(k\right)\right)$$

14. quantifier collapse:

$$\forall n\exists k.s\left(n,k\right)\rightarrow\ldots$$

$$\forall n\forall k.s\left(n,k\right)\rightarrow\ldots$$

any quantifier prefix over succesors appearing only in guards, can be collapsed to no alternation.

$$x_n=0\vee x_n=1$$

$$\left(\forall n.x_n=0\right)\vee\left(\forall n.x_n=1\right)$$

$$x_n>0\wedge x_n<1$$

$$\left(\forall n.x_n>0\right)\wedge\left(\forall n.x_n<1\right)$$

15. to decide disjunction $\phi\vee\psi$ wrt sets of models, simply check if one of them is sat.

16. so assume a boolean combination of sets of models, write it in DNF. now decide emptiness for each DNF clause.

17. conjunctions are well-behaved, so each DNF clause has one positive and multiple negatives:

$$\phi\wedge\bigwedge_i\neg\psi_i$$

18. automata no output, transducers are automata with output, all with finite alphabet.

19. let's consider a "fourth law of robotics": any future sw update to the robot, should admit the laws of robotics.

20. the robot and the update, are written in the same language. so it has to be a language, that can take as input statement in the same lang, and check sat against other such sentences.

21. not only the tau language (tau=nso+gs) can speak of its own formulas, but 1. it is also rich enough to express sw 2. inputs/outputs of this sw are nothing but sentences in various logics incl the tau lang.

22. which bas? 1. of tau formulas 2. fixed-finite bas. 3. two-var+counting 4. tables 5. bfs and sbf

23. and ofc all equipped with additional conservative extensions to the theory of ba as described in taba.pdf