A Tableaux-Based Decision Procedure for Multi-Parameter Propositional Schemata

David M. Cerna
Theory and logic group
Technical University of Vienna

CICM July 9, 2014

Background

- In (Aravantinos et al. 2011) the **ST** procedure for STAB (**S**chemata **TAB**leaux) is provided deciding the satisfiability problem for an expressive class of propositional schemata, the class of regular propositional schemata.
- In (Aravantinos et al. 2013) a resolution calculus is provided deciding the satisfiability problem for a class of schematic clause sets which can encode regular schemata and to some extent propositional schemata with multiple parameters.
- In our work, we investigate which subclasses of the class of propositional schemata with multiple parameters can be decided using a slight extension of STAB.

Results

- Our goal is to find subclasses of the class of propositional schemata with multiple parameters which have a decision procedure for satisfiability while avoiding the extra machinery of normalized clause sets, introduced in (Aravantinos et al. 2013), as well as the transformation of propositional schemata into CNF formulae.
- We provide two classes of propositional schemata extending regular schemata which both have a decision procedure based on the tableaux procedure of (Aravantinos et al. 2011) and allow for restricted use of multiple parameters.

Overview

- First, we will provide a short description of the class of propositional schemata, and in particular, the class of regular schemata.
- We introduce the class of linked schemata and pure overlap schemata.
- Finally, we show how the **ST** procedure (Aravantinos et al. 2011) can be augmented to decide the satisfiability problem for these two classes of schema.

Propositional Schemata Basics

- All propositions have an index in the language of *linear expressions*, i.e. $P_{S(S(0))}$. Can be non-monadic.

Example

Linear expressions are essentially polynomials with exponents of either 0 or 1, built using the alphabet $\Sigma = \{0, S\}$ and variables ranging over Σ^*.

n	$n + S(0)$	$n + m + k$	$4n + m + S(S(0))$

- Given a,b which are linear expressions and f a linear expression containing i, an *iteration* is of the form:

$$\bigwedge_{i=a}^{b} \varphi_{f(i)} \quad \text{or} \quad \bigvee_{i=a}^{b} \varphi_{f(i)}$$

- We call i the *variable bounded by the iteration* and any variable not bounded by an iteration is a *free parameter*.

Propositional Schemata Basics

- Linear orderings can be expressed as follows:

$$a < b \equiv \bigvee_{i=a+1}^{b} \top$$

Example

$$(n \geq 0) \wedge P_0 \wedge \bigwedge_{i=1}^{n} (\neg P_{i-1} \vee P_i) \wedge \neg P_n \qquad (1)$$

- A schema is satisfiable if given a substitution σ for the free parameters the resulting sentence is satisfiable.

Results Propositional Schemata

Fact (Situation)

! *Most subclasses of the class of propositional schemata are undecidable for satisfiability, even in the monadic case.*

Definition (Bounded-Linear Schemata)

Only allowed one free parameter and indices can only have one variable bounded by an iteration. Propositions are monadic.

Example (Bounded-Linear Schema)

$$\bigwedge_{i=0}^{n} \bigvee_{j=i+2}^{2n+4} P_{3n+j+2} \rightarrow P_{i+4-8n} \quad (2)$$

The free parameter is n and i,j are the bound parameters. P_{i+j} and P_{i+2j} are not allowed.

Results Propositional Schemata

Definition (Regular Schemata)

- Only one parameter is allowed.
- no nested iterations.
- only one index which has either a free parameter, a bounded parameter or neither. Coefficients on parameters are 0 or 1.
- All iterations are the same size.

Theorem (Aravantinos et al. 2011)

The satisfiability problem for the class of bounded linear schemata is reducible to the problem for the class of regular schemata.

Theorem (Aravantinos et al. 2011)

The class of regular schemata is decidable using the **ST**.

Concept Behind Linked Schemata

- Allowing unrestricted use of multiple parameters is undecidable for satisfiability.
- However certain restrictions are easily reduced to schemata which are regular schemata like.

$$\left(\bigwedge_{i=1}^{n} p_i\right) \wedge \left(\bigvee_{j=n+1}^{m} \neg p_i\right) \equiv_S \left(\bigwedge_{i=1}^{n} p_i\right) \wedge \left(\bigvee_{j=1}^{m} \neg q_i\right). \quad (3)$$

- If there is no overlap of the intervals than it is as if we are working with two regular schema which are propositionally connected.

Definitions needed defining Linked Schemata

Definition

Let $p \in \mathcal{P}$ be a propositional symbol and φ a propositional schema, then $occ(p, \varphi) = 1$ iff p occurs in φ, otherwise it is $occ(p, \varphi) = 0$.

Definition

Given a schema φ we can construct the set of *principal objects* $\mathcal{P}(\varphi)$ using the following inductive definition:

- $\mathcal{P}(P_a) \Rightarrow \{P_a\}$, $\mathcal{P}(\bigvee_{i=a}^{b} \psi) \Rightarrow \left\{\bigvee_{i=a}^{b} \psi\right\}$, $\mathcal{P}(\bigwedge_{i=a}^{b} \psi) \Rightarrow \left\{\bigwedge_{i=a}^{b} \psi\right\}$
- $\mathcal{P}(\phi \vee \psi) \Rightarrow \mathcal{P}(\phi) \cup \mathcal{P}(\psi)$, $\mathcal{P}(\phi \wedge \psi) \Rightarrow \mathcal{P}(\phi) \cup \mathcal{P}(\psi)$ $\mathcal{P}(\neg\psi) \Rightarrow \mathcal{P}(\psi)$

- We will abbreviate the set of propositional connectives used as $\mathcal{O} = \{\wedge, \vee, \neg\}$. By $\psi \in cl_{\mathcal{O}}(\Phi)$, we mean that ψ can be constructed using the set of propositional schema Φ and the logical connective set $\mathcal{O}$.

Linked Schemata

Definition

Let us consider the class Λ of all finite sets Φ of regular schemata such that for all propositional symbols p, we have that $\left(\sum_{\phi\in\Phi} occ(p,\phi)\right)$ is either 1 or 0, we define the class **LS** of *linked schemata* as

$$\mathbf{LS} = \bigcup_{\Phi\in\Lambda} cl_{\mathcal{O}}\left(\bigcup_{\phi\in\Phi}\mathcal{P}(\phi)\right)$$

Lemma

If φ is a regular schema, then it is a linked schema.

Theorem

The class of regular schemata is contained but not equal to the class of linked schemata.

Example of a Linked Schemata

$$\left(\left(\bigvee_{i=1}^{k} \neg P_i \rightarrow \bigvee_{i=1}^{m} Q_i\right) \wedge \left(\bigvee_{i=1}^{m} R_i \rightarrow \bigwedge_{i=1}^{n} M_i\right) \wedge \bigwedge_{i=1}^{m} (Q_i \leftrightarrow R_i)\right) \rightarrow \left(\bigvee_{i=1}^{k} \neg P_i \rightarrow \bigwedge_{i=1}^{n} M_i\right).$$

Concept Behind Pure Overlap Schemata

- Linked schemata only allow propositional symbols to occur in the scope of at most one free parameter.
- Can we weaken this requirement?

$$0 \leq n \wedge \left(\bigwedge_{i=0}^{n} p_i \right) \vee \left(\bigwedge_{i=0}^{m} \neg p_i \right) \wedge 0 \leq m$$

- If we consider the propositional tableaux extension rules, the two parameters will be put into two different branches and are essentially in different scopes.
- Note that changing either occurrence of p to another propositional symbol is not logically equivalent to the above formula.

Iteration Invariant DNF and Relatively Pure Literals

$$\left(6 \leq n \wedge p_6 \wedge \left(\bigvee_{i=5}^{n} \neg p_i \vee p_{i+1}\right) \wedge \neg p_7 \wedge \neg(6 \leq m) \wedge p_m\right) \vee$$

$$\left(\bigwedge_{i=0}^{n} \neg p_i\right)$$

- If we do not consider iterations as unrollable, the above formula is in DNF.
- The propositional symbol p with index m is relatively pure with respect to the negative occurrences of p in the left most clause.

Lemma

Given a set of regular schemata Φ, *for all* $\psi \in cl_{\mathcal{O}}\left(\bigcup_{\phi\in\Phi} \mathcal{P}(\phi)\right)$ *there exists an IIDNF of* ψ.

More on Relatively Pure Literals

- The relatively pure literals of a schema remain relatively pure regardless of the schema being in IIDNF or not.
- Given a set of regular schemata Φ, let $cl^{rp}_{\mathcal{O}}(\Phi)$ be the set of all schema which can be constructed using the logical connectives $\mathcal{O}$, such that they are relatively pure.

Pure Overlap Schemata

Definition (The class of Pure Overlap Schemata)

Let us consider the class Λ of all finite sets Φ of regular schemata. We define the class of *pure overlap schemata* as

$$\mathbf{POS} = \bigcup_{\Phi \in \Lambda} cl^{rp}_{\mathcal{O}} \left(\bigcup_{\phi \in \Phi} \mathcal{P}(\phi) \right)$$

Lemma

If φ is a linked schema, then it is a pure overlap schema.

Theorem

The class of linked schemata is contained but not equal to the class of pure overlap schemata.

Decision Procedure for Pure Overlap schemata

- Being that linked schemata are a subset of pure overlap schemata we only need to provide a decision procedure for pure overlap schemata.
- We use the **ST** procedure as a sub-routine for the decision procedure of pure overlap schemata.
- Interpretations are constructed the same way as they are constructed for regular schemata (Aravantinos et al. 2011), except the number of interpretations increases.
- We add a branching rule to the **ST** decision procedure which branches on parameters.

Decision Procedure for Pure Overlap schemata

Algorithm ($\mathsf{ST}^{\mathbf{POS}}$ Procedure)

*Given a schema $\varphi \in$ **POS** in negation normal form. The following algorithm, called the $ST^{\mathbf{POS}}$ procedure, decides the satisfiability of φ:*

1) *Apply STAB propositional extension rules with highest priority until no more can be applied. This results in m sets of atoms and iterations referred to as $B_1, \ldots, B_m$.*
2) *For each B_i, we separate B_i into n (the number of parameters in B_i) sub-branches $B_{(i,1)}, \cdots B_{(i,n)}$, where each $B_{(i,j)}$ contains iterations and atoms indexed by a single parameter. Atoms without a free parameter in the indices can be added to every $B_{(i,j)}$. We will mark such a sub-branching with $\otimes_n$ where n is the number of parameters on the branch.*
3) *Run the ST procedure (Aravantinos et al. 2010) on the sub-branch $B_{(i,j)}$.*
4) *For any branch B_i, if one of its sub-branches $B_{(i,j)}$ has a closed tableau after following the ST procedure, then the branch B_i is closed.*

Conclusion

- The concept behind the procedure is if any of the sub-branches $B_{(i,j)}$ is on unsatisfiable, then the entire branch is unsatisfiable, because these branches are essentially conjunctions modulo the iterations.
- Being that the concept of relatively pure guarantees that the intervals are independent the $\otimes_n$ rule is sound.
- As for future work, we would like to consider using the same methods outlined here on other classes of propositional schemata not yet considered, i.e nested regular schemata (Aravantinos et al. 2010).
- Also, we are interested in investigating the relationship between the introduced classes of schemata and temporal logics.

Thank you for your time.