My Attempts To Save Politeness

Yoni Zohar
Bar-Ilan

BIU CS Colloquium
November 20, 2025

Disclamer

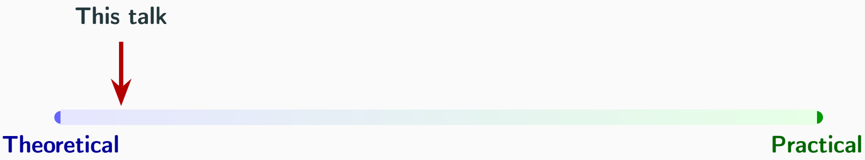
- This talk is sponsored by...

FM@BIU

FM@BIU

Theoretical — **Practical**

FM@BIU



Acknowledgements

- Clark Barrett, Stanford
- Cesare Tinelli, UIowa
- Jane Lange, Stanford, MIT
- Ying Sheng, Stanford, UCLA
- Pascal Fontaine, ULiege
- Christophe Ringeissen, Inria
- Andrew Reynolds, UIowa

- Guilherme Toledo, Bar-Ilan
- Ben Przybocki, Stanford, Cambridge, CMU

Satisfiability Modulo Theories (SMT)

Satisfiability Modulo Theories (SMT)

Theory Combination

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Politeness

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Politeness

My Attempts To Save Politeness

Outline

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Politeness

My Attempts To Save Politeness

Satisfiability

What does it mean that a formula is satisfiable?

- ▷ It is consistent
- ▷ It does not entail a contradiction
- ▷ It has a model

What is a model?

- ▷ Domain + Interpretation of symbols

▷ Example 1: $\exists x.P(x) \wedge P(a)$

- Domain: $\mathbb{N}$
- Interpretation of P: Even numbers
- Interpretation of a: 2

▷ Example 2: $\forall x \exists y.Q(x, y) \wedge Q(a, b)$

- Domain: $\{1\}$
- Interpretation of Q: $\{1, 1\}$
- Interpretations of a and b: 1

What Modulo What?

- The story of SMT

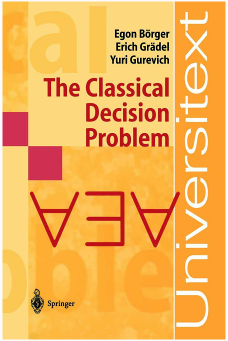
  (the way I tell it)

```
...
(assert (forall x forall y ... (= (f x y) x)))
(assert (forall x ... (P x) ...))
(assert (forall ...))

...

(assert (= (f c d) d))
(assert (not (=> (P (g c c)) (= c d))))
...

(check-sat)
```



Satisfiability Modulo Linear Arithmetic

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
(assert (> (+ x x x y y) 1))
(assert (> (+ z z y y y y) 0))
```

```
(check-sat)
```

Satisfiability Modulo Linear Arithmetic

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
(assert (> (+ x x x y y w w) w))
(assert (> (+ w w d 1 1 1) 0))
```

```
(check-sat)
```

Satisfiability Modulo Linear Arithmetic

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
(assert (> (+ x1 x1 x2 x2 x2 x3) w))
(assert (> (+ x4 x5 x7) 0))
(assert (> (+ x4 x4 x4) x4))
```

```
(check-sat)
```

Satisfiability Modulo Linear Arithmetic

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
(assert (> (+ x1 x1 x1 y y w w) w))
(assert (> (+ w1 w2 d1 1 1 1) 0))
(assert (> (+ x14 x14 x13 y1 y1 w0 w0) w))
(assert (> (+ w1 w2 d1 1 1 1) 0))
```

```
(check-sat)
```

Satisfiability Modulo Linear Arithmetic

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
(assert (> (+ x1 x2 x3 x4 x5 x6 x7) w))
(assert (> (+ x1 x1 x1 x1) w))
(assert (> (+ x3) 1))
(assert (> (+ y1 y2 y3 y4 y5 y6 y7) w))
(assert (> (+ z1 z2 z3 z4 z5 z6 z7) w))
(assert (> (+ x1 x2 x3 x4 x5 x6 x7) x8))
```

```
(check-sat)
```

Satisfiability Modulo Linear Arithmetic

```
(set-logic QF_LIA)
```

```
(assert (> (+ x1 x2 x3 x4 x5 x6 x7) w))
(assert (> (+ x1 x1 x1 x1) w))
(assert (> (+ x3) 1))
(assert (> (+ y1 y2 y3 y4 y5 y6 y7) w))
(assert (> (+ z1 z2 z3 z4 z5 z6 z7) w))
(assert (> (+ x1 x2 x3 x4 x5 x6 x7) x8))
```

```
(check-sat)
```

* `check-sat` now means `check-LIA-sat`.

Satisfiability Modulo Lists

```
...
(assert (forall l,a. (distinct nil (cons a l))))
(assert (forall l (=> (= l (cons a ll)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a ll)) (= ( tail l) ll))))
...
```

```
 (assert (distinct i j))
 (assert (= (const a i) (cons a j)))
```

```
(check-sat)
```

Satisfiability Modulo Lists

```
...
(assert (forall l,a. (distinct nil (cons a l))))
(assert (forall l (=> (= l (cons a ll)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a ll)) (= ( tail l) ll))))
...
```

```
(assert (distinct a b))
(assert (distinct (cons i (cons a l)) nil))
(assert (distinct a b c d e))
```

```
(check-sat)
```

Satisfiability Modulo Lists

```
...
(assert (forall l,a. (distinct nil (cons a l))))
(assert (forall l (=> (= l (cons a ll)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a ll)) (= ( tail l) ll))))
...
```

```
 (assert (= a b))
 (assert (distinct (cons i (cons a l)) nil))
 (assert (= a b c d e))
 (assert (distinct nil (cons i (cons a (cons b l))) ))
 (assert (= a b))
```

```
(check-sat)
```

Satisfiability Modulo Lists

```
...
(assert (forall l,a. (distinct nil (cons a l))))
(assert (forall l (=> (= l (cons a ll)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a ll)) (= ( tail l) ll))))
...
```

```
 (assert (= a b))
```

```
(check-sat)
```

Satisfiability Modulo Lists

```
...
(assert (forall l,a. (distinct nil (cons a l))))
(assert (forall l (=> (= l (cons a l1)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a l1)) (= ( tail l) l1))))
...
```

```
 (assert (= a1 b1))
 (assert (= a2 b2))
 (assert (= a3 b3))
 (assert (= a b c d e))
 (assert (= (head (cons a i)) (head l)))
```

```
(check-sat)
```

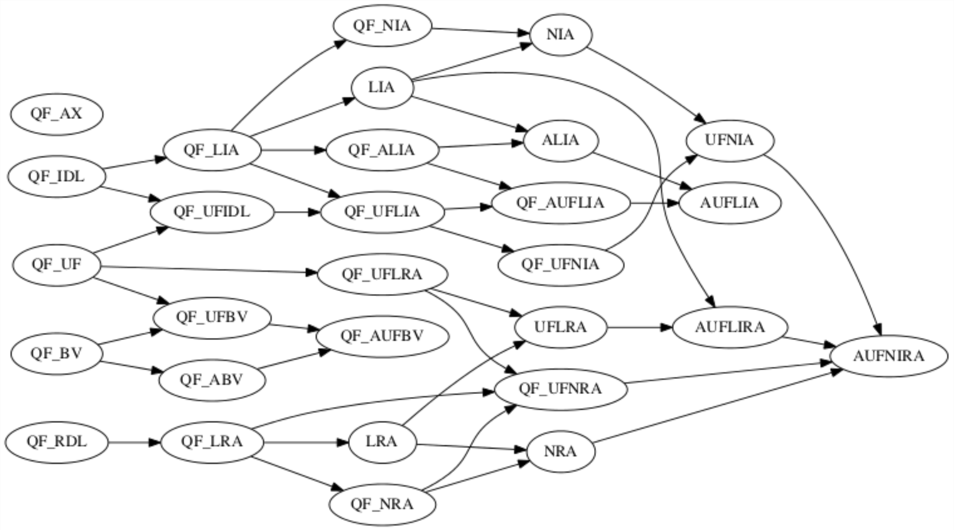
Satisfiability Modulo Lists

```
(set-logic QF_DT)
```

```
(assert (= a1 b1))
(assert (= a2 b2))
(assert (= a3 b3))
(assert (= a b c d e))
(assert (= (head (cons a i)) (head l)))
```

```
(check-sat)
```

Logic Zoo


QF_NIA
NIA
LIA
QF_AX
QF_LIA
QF_ALIA
ALIA
UFNIA
QF_IDL
QF_UFIDL
QF_UFLIA
QF_AUFLIA
AUFLIA
QF_UF
QF_UFNIA
QF_UFLRA
QF_UFBV
QF_AUFBV
UFLRA
AUFLIRA
QF_BV
AUFNIRA
QF_ABV
QF_UFNRA
QF_RDL
QF_LRA
LRA
NRA
QF_NRA

Satisfiability Modulo Theories (SMT)

Theory Combination

Politeness

My Attempts To Save Politeness

Theory Combination

```
...
(assert (forall x forall y (= (+ x (+ y 1)) (+ (+ x y) 1))))
(assert (forall x (= (+ x 0) x)))
(assert (forall x forall y (=> (> x y) (exists ...))))
...
```

```
...
(assert (forall l,a. (distinct nil (cons l a))))
(assert (forall l (=> (= l (cons a ll)) (= ( head l) a))))
(assert (forall l (=> (= l (cons a ll)) (= ( tail l) ll))))
...
```

```
...
(assert (< (head a) (head (tail b))))
...
```

Theory Combination

```
(set-logic QF_DTLIA)
```

```
...
(assert (< (head a) (head (tail b))))
...
```

Theory Combination

The Question

Given algorithms for T_1 and T_2, can we construct an algorithm for $T_1 \cup T_2$?

Formally

Suppose T_1 and T_2 are decidable. Is $T_1 \cup T_2$ decidable? Hopefully with a constructive proof?

Theory Combination

The Question

Given algorithms for T_1 and T_2, can we construct an algorithm for $T_1 \cup T_2$?

Formally

Suppose T_1 and T_2 are decidable. Is $T_1 \cup T_2$ decidable? Hopefully with a constructive proof?

The Decision Problem for T:
Given a QF formula, is it T-SAT?

First Step: Purification

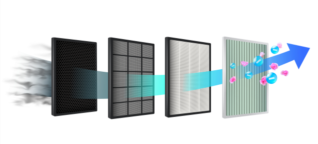
DT Solver	Arith Solver

$$a_0, a_1, a_2 \neq nil$$

$$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$$

$$tail(a_0) = tail(a_1) = tail(a_2) = nil$$

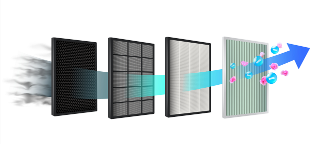
$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	

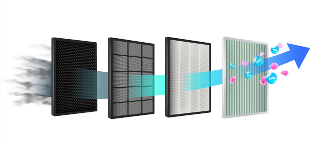
$$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$$
$$tail(a_0) = tail(a_1) = tail(a_2) = nil$$
$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	

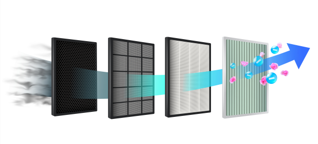
$$tail(a_0) = tail(a_1) = tail(a_2) = nil$$
$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$	

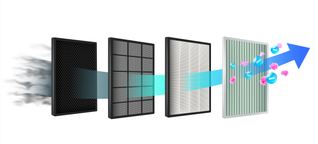
$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$	

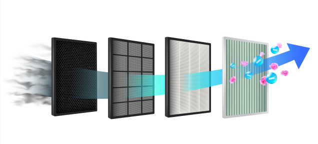
$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$



First Step: Purification

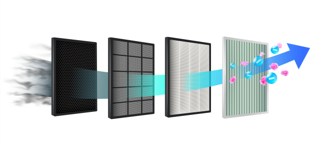
DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	

$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$



First Step: Purification

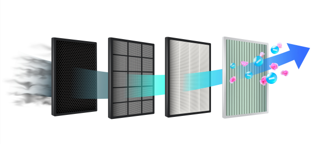
DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$	
	$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$	

$$0 \leq head(a_0), head(a_1), head(a_2) \leq 1$$

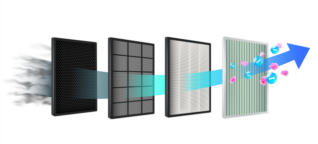


First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	

$$0 \leq x, y, z \leq 1$$

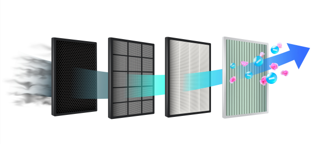
$$x = head(a_0), y = head(a_1), z = head(a_2)$$



First Step: Purification

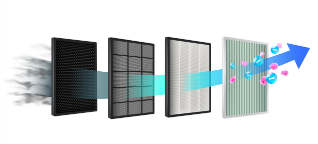
DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
	$0 \leq x, y, z \leq 1$

$$x = head(a_0), y = head(a_1), z = head(a_2)$$



First Step: Purification

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
$x = head(a_0), y = head(a_1), z = head(a_2)$	$0 \leq x, y, z \leq 1$



Naive Method

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$A \text{ is } T_1\text{-SAT and } B \text{ is } T_2\text{-SAT.}$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$ $x = head(a_0), y = head(a_1), z = head(a_2)$	$0 \leq x, y, z \leq 1$
SAT	SAT

The formlua is UNSAT  Method says SAT

Greg
@Nelson

We are happy to announce a theory combination method! For it to work, the two combined theories must be stably infinite! ♾️

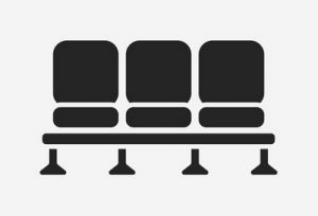
12:00 PM · Jun 1, 1979 · Twitter for rotary phones

1 Retweet **2** Quote Tweets **70** Likes

Exchanging Information

Arrangements

- Equivalent classes of variables
- Which variables should have the same value
- Conjunction of (dis)equalities between variables



Second Step: Combination

Naive Method

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$A \text{ is } T_1\text{-SAT and } B \text{ is } T_2\text{-SAT.}$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$ $x = head(a_0), y = head(a_1), z = head(a_2)$	$0 \leq x, y, z \leq 1$
SAT	SAT

The formlua is UNSAT  Method says SAT

Second Step: Combination

Nelson-Oppen Method **[Nelson & Oppen, 1979]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	$0 \leq x, y, z \leq 1$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
$x = head(a_0), y = head(a_1), z = head(a_2)$	
$\delta : x = y = z$	$\delta : x = y = z$
UNSAT	SAT

Second Step: Combination

Nelson-Oppen Method **[Nelson & Oppen, 1979]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	$0 \leq x, y, z \leq 1$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
$x = head(a_0), y = head(a_1), z = head(a_2)$	
$\delta : x = y \neq z$	$\delta : x = y \neq z$
UNSAT	SAT

Second Step: Combination

Nelson-Oppen Method **[Nelson & Oppen, 1979]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	$0 \leq x, y, z \leq 1$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
$x = head(a_0), y = head(a_1), z = head(a_2)$	
$\delta : x \neq y = z$	$\delta : x \neq y = z$
UNSAT	SAT

Second Step: Combination

Nelson-Oppen Method **[Nelson & Oppen, 1979]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT}.$$

DT Solver	Arith Solver
$a_0, a_1, a_2 \neq nil$	$0 \leq x, y, z \leq 1$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$tail(a_0) = tail(a_1) = tail(a_2) = nil$	
$x = head(a_0), y = head(a_1), z = head(a_2)$	
$\delta : x \neq y \neq z$	$\delta : x \neq y \neq z$
SAT	UNSAT

The formlua is UNSAT  Method says UNSAT

Second Step: Combination

Nelson-Oppen Method [Nelson & Oppen, 1979]

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT}.$$

- ▷ Nelson-Oppen works for lists of integers
- ▷ What about lists of bit-vectors?

Second Step: Combination

Nelson-Oppen Method **[Nelson & Oppen, 1979]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } vars(A) \cap vars(B) \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } B \wedge \delta \text{ is } T_2\text{-SAT}.$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $tail(a_0) = tail(a_1) = tail(a_2) = nil$ No shared variables Trivial δ	$\top$ No shared variables Trivial δ
SAT	SAT

The formlua is UNSAT  Method says SAT

Satisfiability Modulo Theories (SMT)

Theory Combination

Politeness

My Attempts To Save Politeness

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Politeness

My Attempts To Save Politeness

Christophe @ThePoliteGuy · Jun 1, 2005 ···

If anyone would like to combine bit-vectors with other theories, you can now do it! 💪

@Ranise @Zarba and me introduced the polite combination method. 🙏

One theory has to be *polite*. The other has no requirements. 🏃

415 148 935

Bitter @BitVecGuy · Jun 1, 2005 ···

That is wonderful! I can finally work with bit-vectors! 🙏 What's the catch?

44 251 4.5K

Christophe @Christophe · Jun 1, 2005 ···

You need to consider more arrangements in the algorithm. Also, politeness is harder to prove than stable infiniteness. 😀

3 6 42

Second Step: Combination

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	BV[1] Solver
$wit(a_0, a_1, a_2 \neq nil)$	$\top$
$wit(a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2)$	
$wit(tail(a_0) = tail(a_1) = tail(a_2) = nil)$	
$\delta = ?$	$\delta = ?$

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT}.$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$	$\top$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$\bigwedge_{i=0}^{2} a_i = cons(v_i, nil)$	
$\delta = ?$	$\delta = ?$

Second Step: Combination

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$	$\top$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$\bigwedge_{i=0}^{2} a_i = cons(v_i, nil)$	
$\delta : v_0 = v_1 = v_2$	$\delta : v_0 = v_1 = v_2$
UNSAT	SAT

Second Step: Combination

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$	$\top$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$\bigwedge_{i=0}^{2} a_i = cons(v_i, nil)$	
$\delta : v_0 = v_1 \neq v_2$	$\delta : v_0 = v_1 \neq v_2$
UNSAT	SAT

Second Step: Combination

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$

$$\Leftrightarrow$$

$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$

$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT}.$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$	$\top$
$a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$	
$\bigwedge_{i=0}^{2} a_i = cons(v_i, nil)$	
$\delta : v_0 \neq v_1 = v_2$	$\delta : v_0 \neq v_1 = v_2$
UNSAT	SAT

Second Step: Combination

Polite Combination **[Ranise, Ringeissen, Zarba, 2005]**

$$A \wedge B \text{ is } (T_1 \cup T_2)\text{-SAT}$$
$$\Leftrightarrow$$
$$\exists \delta \text{ over } \boxed{vars(B)} \text{ s.t.}$$
$$A \wedge \delta \text{ is } T_1\text{-SAT and } \boxed{\text{wit(B)}} \wedge \delta \text{ is } T_2\text{-SAT.}$$

DT Solver	BV[1] Solver
$a_0, a_1, a_2 \neq nil$ $a_0 \neq a_1, a_1 \neq a_2, a_0 \neq a_2$ $\bigwedge_{i=0}^{2} a_i = cons(v_i, nil)$ $\delta : v_0 \neq v_1 \neq v_2$	$\top$ $\delta : v_0 \neq v_1 \neq v_2$
SAT	UNSAT

The formlua is UNSAT  Method says UNSAT

Outline

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

Politeness

My Attempts To Save Politeness

Politeness = Smoothness + Finite Witnessibility

Smoothness

T is smooth if

- for every T-SAT formula A and T-model of it $\mathcal{I}$
- A is satisfiable by a T-model with cardinality $\geq$ that of $\mathcal{I}$

Example

- Theory of lists is smooth
- Theory of BVs is not

This property generalizes stable infiniteness!

T is Finitely Witnessibile:

If A is T-SAT then it has a T-model $\mathcal{I}$ with domain $vars(A)^{\mathcal{I}}$

Example

- $nil = nil$: empty domain?


T is Finitely Witnessibile:

If $wit(A)$ is T-SAT then it has a T-model $\mathcal{I}$ with domain $vars(wit(A))^{\mathcal{I}}$

Example

- $wit(nil = nil)$: empty domain?


Politeness = Smoothness + Finite Witnessibility

T is Finitely Witnessibile:

If $wit(A)$ is T-SAT then it has a T-model $\mathcal{I}$ with domain $vars(wit(A))^{\mathcal{I}}$

- wit is computable
- A is T-equivalent to $\exists \overline{w}.wit(A)$, $\overline{w}$ = new variablesd from *wit*

Example

- $nil = nil \land x = x$: non-empty domain


Politeness Implies Combinability

[Ranise, Ringeissen, Zarba, 2005]

Combining Data Structures with Nonstably Infinite Theories Using Many-Sorted Logic*

Silvio Ranise[1], Christophe Ringeissen[1], and Calogero G. Zarba[2]

[1] LORIA and INRIA-Lorraine
[2] University of New Mexico

Theorem 15 (Correctness and complexity). *Let T_i be a Σ_i-theory, for $i = 1, 2$. Assume that:*

- *the quantifier-free satisfiability problem of T_i is decidable, for $i = 1, 2$;*
- $\Sigma_1^{\mathrm{F}} \cap \Sigma_2^{\mathrm{F}} = \emptyset$ *and* $\Sigma_1^{\mathrm{P}} \cap \Sigma_2^{\mathrm{P}} = \emptyset$;
- T_2 *is polite with respect to* $\Sigma_1^{\mathrm{S}} \cap \Sigma_2^{\mathrm{S}}$.

Then the quantifier-free satisfiability problem of is decidable.

In other words:

Polite combination works for polite theories.

Politeness Implies Combinability

[Ranise, Ringeissen, Zarba, 2005]

Combining Data Structures with Nonstably Infinite Theories Using Many-Sorted Logic*

Silvio Ranise[1], Christophe Ringeissen[1], and Calogero G. Zarba[2]

[1] LORIA and INRIA-Lorraine
[2] University of New Mexico

PROOF.

(2 $\Rightarrow$ 1). Let $\mathcal{A}$ be a T_1-interpretation satisfying $\Gamma_1 \cup arr(V, E)$, and let $\mathcal{B}$ be a T_2-interpretation satisfying $\{\psi_2\} \cup arr(V, E)$. Since T_2 is finitely witnessable, we can assume without loss of generality that $B_\sigma = V_\sigma^{\mathcal{B}}$, for each $\sigma \in S$.

In other words:

... If $wit(\phi_2) \wedge \delta$ is T_2-SAT, it has a T_2-"small" model. ...

Dejan @CVC4Guy · Jun 1, 2010

Check out our new SMT Solver! With @Barrett. It is based on polite combination. See first comment for details. 😻

39 27 952

Dejan @CVC4Guy · Jun 1, 2010

What is important to know, is that the polite combination method requires *strong politeness*, not only *politeness*. 🤷‍♂️

342 771 12.5K

John @SMTFan · Jun 1, 2010

Working like magic! 🏆

353 5.3K 23K

Counterexample

- T: $\exists x, y. x \neq y$
- T is decidable
- T is polite with $wit(\phi) = \phi \wedge w_1 = w_1 \wedge w_2 = w_2$
- ϕ: $x = x$
- δ: $x = w_1 = w_2$
- $wit(\phi) \wedge \delta$ is T-SAT, but not by a "small" model.

Polite Theories Revisited*

Dejan Jovanović and Clark Barrett

New York University
{dejan,barrett}@cs.nyu.edu

by $\exists x\, y.\, x \neq y$). Note that the combination of these two theories contains no structures, and hence no formula is satisfiable in $T_1 \oplus T_2$.

Theory T_2 is clearly smooth with respect to σ. To be polite, T_2 must also be finitely witnessable with respect to σ. Consider the following candidate witness function:

$$witness(\phi) \triangleq \phi \wedge w_1 = w_1 \wedge w_2 = w_2 \ ,$$

where w_1 and w_2 are fresh variables of sort σ not appearing in ϕ.

Let ϕ be a conjunction of flat Σ-literals, let $\psi = witness(\phi)$, and let $V = vars(\psi)$. It is easy to see that the first condition for finite witnessability holds: ϕ is satisfied in a T_2 model iff $\exists w_1 w_2.\, \psi$ is. Now, consider the second condition according to [8] (i.e. without the arrangement). We must show that if ψ is T_2-satisfiable (in interpretation $\mathcal{B}$, say), then there exists a T_2-interpretation $\mathcal{A}$ satisfying ψ such that $A_\sigma = [V]^{\mathcal{A}}$. The obvious candidate for $\mathcal{A}$ is obtained by setting $A_\sigma = [V]^{\mathcal{B}}$ and by letting $\mathcal{A}$ interpret only those variables in V (interpreting them as in $\mathcal{B}$). Clearly $\mathcal{A}$ satisfies ψ. However, if $[V]^{\mathcal{B}}$ contains only one element, then $\mathcal{A}$ is not a T_2-interpretation. But in this case, we can always first modify the way variables are interpreted in $\mathcal{B}$ to ensure that $w_2^{\mathcal{B}}$ is different from $w_1^{\mathcal{B}}$ ($\mathcal{B}$ is a T_2-interpretation, so B_σ must contain at least two different elements). Since w_2 does not appear in ϕ, this change cannot affect the satisfiability of ψ in $\mathcal{B}$. After making this change, $[V]^{\mathcal{B}}$ is guaranteed to contain at least two elements, so we can always construct $\mathcal{A}$ as described above. Thus, the second condition for finite witnessability is satisfied and the candidate witness function is indeed a witness function according to [8].

As we will see below, however, this witness function leads to problems. Notice that according to the definition of finite witnessability in this paper, the candidate witness function is not acceptable. To see why, consider again the second condition. Let δ_V be an arrangement of V. According to our definition, we must show that if $\psi \wedge \delta_V$ is satisfied by T_2-interpretation $\mathcal{B}$, then there exists a T_2-interpretation $\mathcal{A}$ satisfying $\psi \wedge \delta_V$ such that $A_\sigma = [V]^{\mathcal{A}}$. We can consider the same construction as above, but this time, the case when $[V]^{\mathcal{B}}$ contains only one element cannot be handled as before. This is because δ_V requires $\mathcal{A}$ to preserve equalities and disequalities in V. In particular, δ_V may include $w_1 = w_2$. In this case, there is no way to construct an appropriate interpretation $\mathcal{A}$.

Now, we show what happens if the candidate witness function given above is allowed. Consider using Proposition 2 to check the satisfiability of $x = x$ (where x is a variable of sort σ). Although this is trivially satisfiable in any theory that has at least one structure, it is not satisfiable in $T_1 \oplus T_2$ since there are no structures to satisfy it. To apply the proposition we let $\Gamma_1 = \emptyset$, $\Gamma_2 = \{x = x\}$,

Problem

Counterexample

- T: $\exists x, y. x \neq y$
- T is decidable
- T is polite with $wit(\phi) = \phi \wedge w_1 = w_1 \wedge w_2 = w_2$
- ϕ: $x = x$
- δ: $x = w_1 = w_2$
- $wit(\phi) \wedge \delta$ is T-SAT, but not by a "small" model.

Important!

- This is a counterexample for (a step in) the proof
- Not for the theorem
- T can be combined with any decidable theory

Polite Theories Revisited*

Dejan Jovanović and Clark Barrett

New York University
{dejan,barrett}@cs.nyu.edu

by $\exists x\, y.\, x \neq y$). Note that the combination of these two theories contains no structures, and hence no formula is satisfiable in $T_1 \oplus T_2$.

Theory T_2 is clearly smooth with respect to σ. To be polite, T_2 must also be finitely witnessable with respect to σ. Consider the following candidate witness function:

$$witness(\phi) \triangleq \phi \wedge w_1 = w_1 \wedge w_2 = w_2 \ ,$$

where w_1 and w_2 are fresh variables of sort σ not appearing in ϕ.

Let ϕ be a conjunction of flat Σ-literals, let $\psi = witness(\phi)$, and let $V = vars(\psi)$. It is easy to see that the first condition for finite witnessability holds: ϕ is satisfied in a T_2 model iff $\exists w_1 w_2.\, \psi$ is. Now, consider the second condition according to [8] (i.e. without the arrangement). We must show that if ψ is T_2-satisfiable (in interpretation $\mathcal{B}$, say), then there exists a T_2-interpretation $\mathcal{A}$ satisfying ψ such that $A_\sigma = [V]^{\mathcal{A}}$. The obvious candidate for $\mathcal{A}$ is obtained by setting $A_\sigma = [V]^{\mathcal{B}}$ and by letting $\mathcal{A}$ interpret only those variables in V (interpreting them as in $\mathcal{B}$). Clearly $\mathcal{A}$ satisfies ψ. However, if $[V]^{\mathcal{B}}$ contains only one element, then $\mathcal{A}$ is not a T_2-interpretation. But in this case, we can always first modify the way variables are interpreted in $\mathcal{B}$ to ensure that $w_2^{\mathcal{B}}$ is different from $w_1^{\mathcal{B}}$ ($\mathcal{B}$ is a T_2-interpretation, so B_σ must contain at least two different elements). Since w_2 does not appear in ϕ, this change cannot affect the satisfiability of ψ in $\mathcal{B}$. After making this change, $[V]^{\mathcal{B}}$ is guaranteed to contain at least two elements, so we can always construct $\mathcal{A}$ as described above. Thus, the second condition for finite witnessability is satisfied and the candidate witness function is indeed a witness function according to [8].

As we will see below, however, this witness function leads to problems. Notice that according to the definition of finite witnessability in this paper, the candidate witness function is not acceptable. To see why, consider again the second condition. Let δ_V be an arrangement of V. According to our definition, we must show that if $\psi \wedge \delta_V$ is satisfied by T_2-interpretation $\mathcal{B}$, then there exists a T_2-interpretation $\mathcal{A}$ satisfying $\psi \wedge \delta_V$ such that $A_\sigma = [V]^{\mathcal{A}}$. We can consider the same construction as above, but this time, the case when $[V]^{\mathcal{B}}$ contains only one element cannot be handled as before. This is because δ_V requires $\mathcal{A}$ to preserve equalities and disequalities in V. In particular, δ_V may include $w_1 = w_2$. In this case, there is no way to construct an appropriate interpretation $\mathcal{A}$.

Now, we show what happens if the candidate witness function given above is allowed. Consider using Proposition 2 to check the satisfiability of $x = x$ (where x is a variable of sort σ). Although this is trivially satisfiable in any theory that has at least one structure, it is not satisfiable in $T_1 \oplus T_2$ since there are no structures to satisfy it. To apply the proposition we let $\Gamma_1 = \emptyset$, $\Gamma_2 = \{x = x\}$,

The Fix of [Jovanovic & Barrett, 2010]

Finite Witnessibility

T is finitely witnessible if there exists a function *wit* such that:

- A is T-equivalent to $\exists \overline{w}\, wit(A)$, $w = vars(wit(A)) \setminus vars(A)$
- if $wit(A)$ is T-SAT then it is T-SAT by a model with *domain* $vars(wit(A))^{\mathcal{I}}$.

Definition

T is polite if it is smooth and finitely witnessable.

Theorem? [Ranise, Ringeissen, Zarba, 2005]

Polite combination works for polite theories.

$(2 \Rightarrow 1)$. Let $\mathcal{A}$ be a T_1-interpretation satisfying $\Gamma_1 \cup arr(V, E)$, and let $\mathcal{B}$ be a T_2-interpretation satisfying $\{\psi_2\} \cup arr(V, E)$. Since T_2 is finitely witnessable, we can assume without loss of generality that $B_\sigma = V_\sigma^{\mathcal{B}}$, for each $\sigma \in S$.

The Fix of [Jovanovic & Barrett, 2010]

Strong Finite Witnessibility

T is strongly finitely witnessible if there exists a function *wit* such that:

- A is T-equivalent to $\exists \overline{w}\, wit(A)$, $w = vars(wit(A)) \setminus vars(A)$
- $\forall\ \delta$: if $wit(A) \wedge \delta$ is T-SAT then it is T-SAT by a model with *domain* $vars(wit(A) \wedge \delta)^{\mathcal{I}}$.

Definition

T is strongly polite if it is smooth and strongly finitely witnessable.

Theorem! **[Jovanovic & Barrett, 2010]**

Polite combination works for strongly polite theories.

strongly

(2 $\Rightarrow$ 1). Let $\mathcal{A}$ be a T_1-interpretation satisfying $\Gamma_1 \cup arr(V, E)$, and let $\mathcal{B}$ be a T_2-interpretation satisfying $\{\psi_2\} \cup arr(V, E)$. Since T_2 is finitely witnessable, we can assume without loss of generality that $B_\sigma = V_\sigma^{\mathcal{B}}$, for each $\sigma \in S$.

Yoni @SequentCalculus · Jun 1, 2017

All of you SMT fans -- does anyone know if politeness alone can be used for theory combination?

Yoni @SequentCalculus · Jun 1, 2017

All of you SMT fans -- does anyone know if politeness alone can be used for theory combination?

Yoni

@SequentCalculus

Account suspended

Twitter suspends accounts that violate the Twitter Rules.

Outline

Satisfiability Modulo Theories (SMT)

Polite Theory Combination

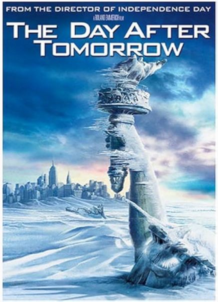
Politeness

My Attempts To Save Politeness

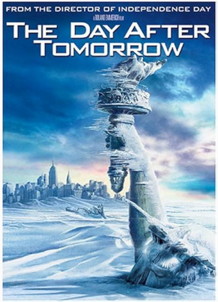
Save Politeness?

Save Politeness from What?

- Extinction



Save Politeness?



Save Politeness from What?

- ▷ Extinction
- ▷ The Polite Combination Method is great
  - Revolutionary
  - Non-symmetric
  - Only requires a solver
- ▷ The Politeness Property is:
  - Not known to be enough for polite combination method
  - Replaced by strong politeness
    - A more complicated property
  - Can we do something else with politeness?
    - Practiaclly: it would be easier to prove combinability

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

My Attempts To Save Politeness

Attempt 1 | Attempt 2 | Attempt 3 | Attempt 4

Additivity

Additive Politeness $\Rightarrow$ Strong politeness

Attempt 1

Additive Politeness **[Sheng et al. 2020]**

wit is additive if $wit(wit(\phi) \wedge \delta) \equiv wit(\phi) \wedge \delta$ where δ is an arrangement.

- ▷ The witness should be invariant to arrangements

Proposition

If a witness is additive, then it is a strong.

Was Attempt 1 a Success?

- Strong politeness of DT, sets, arrays [Sheng et al. 2020, Raya & Ringeissen, 2025]
- strong politeness = additive politeness[1] [Przybocki, Toledo, Zohar, 2025]
- We want to use vanilla politeness!

[1] a different but just as good definition of additivity is needed.

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Additive Politeness $\Rightarrow$ Strong politeness

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Additive Politeness $\Rightarrow$ Strong politeness

My Attempts To Save Politeness

Attempt 1 | Attempt 2 | Attempt 3 | Attempt 4

Additivity

Equality

Maybe: politeness = strong politeness

Attempt 2

Politeness = Strong Politeness ?

- If so, the original proof is actually correct

Proof.

strongly

(2 ⇒ 1). Let $\mathcal{A}$ be a T_1-interpretation satisfying $\Gamma_1 \cup arr(V, E)$, and let $\mathcal{B}$ be a T_2-interpretation satisfying $\{\psi_2\} \cup arr(V, E)$. Since T_2 is finitely witnessable, we can assume without loss of generality that $B_\sigma = V_\sigma^{\mathcal{B}}$, for each $\sigma \in S$.

Attempt 2

Politeness = Strong Politeness ?

- ▷ If so, the original proof is actually correct

PROOF.

strongly

(2 ⇒ 1). Let $\mathcal{A}$ be a T_1-interpretation satisfying $\Gamma_1 \cup arr(V, E)$, and let $\mathcal{B}$ be a T_2-interpretation satisfying $\{\psi_2\} \cup arr(V, E)$. Since T_2 is finitely witnessable, we can assume without loss of generality that $B_\sigma = V_\sigma^{\mathcal{B}}$, for each $\sigma \in S$.

Politeness ≠ Strong Politeness !

- ▷ We found a theory that is polite but not strongly polite [Sheng et al. 2021]
- ▷ Two sorts, cardinalities $(2, \infty)$ or $(\geq 3, \geq 3)$.
  - Originally introduced in the context of shiny combination [Casal & Rasga 2018]

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Maybe: politeness = strong politeness

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Maybe: politeness = strong politeness

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Maybe: politeness implies combinability but with a different proof?

Attempt 3

Theorem **[Toledo, Przybocki, Zohar, 2025]**

- There are two decidable theories such that one is polite and their union is undecidable.

Axioms[2]

Name	Axiomatization
$T_=$	$\{P_n \rightarrow \psi_{=n} : n \in \mathbb{N}^*\}$
T_f	$\{[\psi^{=}_{\geq f_1(k)} \wedge \psi^{\neq}_{\geq f_0(k)}] \vee \bigvee_{i=1}^{k}[\psi^{=}_{=f_1(i)} \wedge \psi^{\neq}_{=f_0(i)}] : k \in \mathbb{N}^*\}$

Proof.

- $T_=$ and T_f are decidable
- T_f is polite.
- $T_= \cup T_f$ is undecidable (If it were, f would be computable.) □

[2] f is a non-computable function.

My Attempts To Save Politeness

Attempt 1 | Attempt 2 | Attempt 3 | Attempt 4

Additivity

Equality

Unnecessity

Maybe: politeness implies combinability but with a different proof?

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Maybe: politeness implies combinability but with a different proof?

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Partiality

Maybe: politeness is useful for some class of theories

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable.
- $G(X)$ is the set of theories combinable with every theory of X.

Connections

- $SI \subseteq G(SI)$ from Nelson-Oppen
- $SP \subseteq G(D)$ from strong politeness theorem

Notation

- SI: Stable Infinite Theories
- P: Polite Theories
- SP: Strongly Polite Theories
- D: All Decidable Theories

What about P ?

$G(P) = ???$

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable.
- $G(X)$ is the set of theories combinable with every theory of X.

Connections

- $SI \subseteq G(SI)$ from Nelson-Oppen
  - But actually, $SI = G(SI)$
- $SP \subseteq G(D)$ from strong politeness theorem
  - But actually, $SP = G(D)$

Notation

- SI: Stable Infinite Theories
- P: Polite Theories
- SP: Strongly Polite Theories
- D: All Decidable Theories

What about P ?

$G(P) = ???$

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

galois

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable.
- $G(X)$ is the set of theories combinable with every theory of X.

Connections

- $SI \subseteq G(SI)$ from Nelson-Oppen
  - But actually, $SI = G(SI)$
- $SP \subseteq G(D)$ from strong politeness theorem
  - But actually, $SP = G(D)$

Notation

- SI: Stable Infinite Theories
- P: Polite Theories
- SP: Strongly Polite Theories
- D: All Decidable Theories

What about P ?

$G(P) = \$\$\$$

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable.
- $G(X)$ is the set of theories combinable with every theory of X.

Connections

- $SI \subseteq G(SI)$ from Nelson-Oppen
  - But actually, $SI = G(SI)$
- $SP \subseteq G(D)$ from strong politeness theorem
  - But actually, $SP = G(D)$

Notation

- SI: Stable Infinite Theories
- P: Polite Theories
- SP: Strongly Polite Theories
- D: All Decidable Theories

What about P ?

$G(P) =$ ₪₪₪

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable
- $G(X)$ is t

Connections

- $SI \subseteq G(S$
  - But a
- $SP \subseteq G(L$
  - But a

ite Theories

ories

olite Theories

ble Theories

What about P ?

$G(P) =$ ₪₪₪

Attempt 4

A Galois Connection **[Przybocki, Toledo, Zohar, POPL'26]**

|galois|

- T_1 and T_2 are combinable if $T_1 \cup T_2$ is decidable.
- $G(X)$ is the set of theories combinable with every theory of X.

Connections

- $SI \subseteq G(SI)$ from Nelson-Oppen
  - But actually, $SI = G(SI)$
- $SP \subseteq G(D)$ from strong politeness theorem
  - But actually, $SP = G(D)$

Notation

- SI: Stable Infinite Theories
- P: Polite Theories
- SP: Strongly Polite Theories
- D: All Decidable Theories

What about P ?

$$G(P) = SI$$

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Partiality

Maybe: politeness is useful for some class of theories

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Partiality

Maybe: politeness is useful for some class of theories

My Attempts To Save Politeness

Attempt 1 → Attempt 2 → Attempt 3 → Attempt 4

Additivity

Equality

Unnecessity

Partiality

Attempt 5?

- $G(P) = SI$ was only proven for single-sorted theories.
- Maybe it does not hold for many-sorted ones?
- To be determined

Conclusion

Polite Combination:

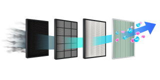
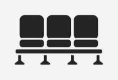
- ▷ Revolutionary
- ▷ Optimal ($G(SP) = D$)
- ▷ Only needs a solver
- ▷ Used in cvc5

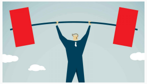
Politeness:

- ▷ Not enough
- ▷ You really need strong/additive politeness
- ▷ Does not improve Nelson Oppen

Future:

- ▷ $G(P) = SI$ for many-sorted signatures?
- ▷ Connections between more properties






|galois|

Polite Combination:

- Revolutionary
- Optimal ($G(SP) = D$)
- Only needs a solver
- Used in cvc5

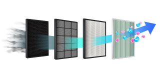
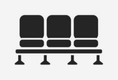
Politeness:

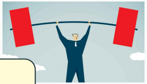
- Not enough
- You really ne
- Does not improve Nelson Oppen

Future:

- $G(P) = SI$ for many-sorted signatures?
- Connections between more properties

Stably-infinite	Smooth	S.F. witnessable	Example
T	T	T	$T_{\geq n}$
		F	T_{∞}
	F	T	
		F	$T_{n,\infty}$
F	F	T	$T_{\leq n}$
		F	$T_{m,n}$



Conclusion

Polite Combinat

- ▷ Revolutionary
- ▷ Optimal ($G($
- ▷ Only needs a
- ▷ Used in cvc5

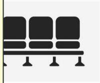
Politeness:

- ▷ Not enough
- ▷ You really ne
- ▷ Does not imp

Future:

- ▷ $G(P) = SI$ f
- ▷ Connections

								Empty		Non-empty		
SI	**SM**	**FW**	**SW**	**CV**	**FM**	**SF**	**CF**	OS	MS	OS	MS	$N^{\underline{o}}$
T	T	T	T	T	T	T	T	$\mathcal{T}_{\geq n}$	$(\mathcal{T}_{\geq n})^2$	$(\mathcal{T}_{\geq n})_s$	$((\mathcal{T}_{\geq n})^2)_s$	1
							F	Theorem 5		Unicorns 2.0		2
				F	T	T	T	[24]		$(\mathcal{T}_{\geq n})_\vee$	$((\mathcal{T}_{\geq n})^2)_\vee$	3
							F			Unicorns 2.0		4
			F	T	T	T	T	[25]		Unicorns 3.0		5
							F			$\mathcal{T}_f$	$(\mathcal{T}_f)^2$	6
						F	T	[25]	$\mathcal{T}_{2,3}$	[25]	$(\mathcal{T}_{2,3})_s$	7
							F		Theorem 5		$\mathcal{T}_{XVII}$	8
				F	T	T	T	[24]		Unicorns 3.0		9
							F			$\mathcal{T}_f^s$	$(\mathcal{T}_f^s)^2$	10
						F	T			[25]	$(\mathcal{T}_{2,3})_\vee$	11
							F				$\mathcal{T}_{XVIII}$	12
		F	F	T	T	T	F	[24]		$\mathcal{T}_\varsigma^s$	$\mathcal{T}_\varsigma^=$	13
						F	F			[25]	$\mathcal{T}_\varsigma^2$	14
					F	F	T	$\mathcal{T}_\infty$	$(\mathcal{T}_\infty)^2$	$(\mathcal{T}_\infty)_s$	$((\mathcal{T}_\infty)^2)_s$	15
							F	Theorem 5		$\mathcal{T}_{XV}$	$(\mathcal{T}_{XV})^2$	16
				F	T	T	F	[24]		$\mathcal{T}_\varsigma^\vee$	$(\mathcal{T}_\varsigma^\vee)^2$	17
						F	F			[25]	$\mathcal{T}_{\varsigma\vee}^=$	18
					F	F	T			$(\mathcal{T}_\infty)_\vee$	$((\mathcal{T}_\infty)^2)_\vee$	19
							F			$\mathcal{T}_{XVI}$	$(\mathcal{T}_{XVI})^2$	20
	F	T	F	T	T	T	T	$\mathcal{T}_{even}^\infty$	$(\mathcal{T}_{even}^\infty)^2$	$(\mathcal{T}_{even}^\infty)_s$	$((\mathcal{T}_{even}^\infty)^2)_s$	21
							F	$\mathcal{T}_I$	$(\mathcal{T}_I)^2$	$(\mathcal{T}_I)_s$	$((\mathcal{T}_I)^2)_s$	22
						F	T	[25]	$\mathcal{T}^\infty$	[25]	$(\mathcal{T}^\infty)^2$	23
							F		$\mathcal{T}_{II}$		$(\mathcal{T}_{II})^2$	24
				F	T	T	T	[24]		$(\mathcal{T}_{even}^\infty)_\vee$	$((\mathcal{T}_{even}^\infty)^2)_\vee$	25
							F			$(\mathcal{T}_I)_\vee$	$((\mathcal{T}_I)^2)_\vee$	26
						F	T			[25]	$(\mathcal{T}^\infty)_\vee$	27
							F				$(\mathcal{T}_{II})_\vee$	28
		F	F	T	T	T	F	$\mathcal{T}_\varsigma$	$(\mathcal{T}_\varsigma)^2$	$(\mathcal{T}_\varsigma)_s$	$((\mathcal{T}_\varsigma)^2)_s$	29
						F	F	[25]	$\mathcal{T}_\varsigma^\infty$	[25]	$(\mathcal{T}_\varsigma^\infty)_s$	30
					F	F	T	$\mathcal{T}_{n,\infty}$	$(\mathcal{T}_\infty)^2$	$(\mathcal{T}_{n,\infty})_s$	$((\mathcal{T}_{n,\infty})^2)_s$	31
							F	Theorem 7	$\mathcal{T}_{III}$	$\mathcal{T}_{VII}$	$(\mathcal{T}_{III})_s$	32
				F	T	T	F	[24]		$(\mathcal{T}_\varsigma)_\vee$	$((\mathcal{T}_\varsigma)^2)_\vee$	33
						F	F			[25]	$(\mathcal{T}_\varsigma^\infty)_\vee$	34
					F	F	T			$(\mathcal{T}_{n,\infty})_\vee$	$((\mathcal{T}_{n,\infty})^2)_\vee$	35
							F			$\mathcal{T}_{VIII}$	$(\mathcal{T}_{III})_\vee$	36
F	F	T	T	T	T	T	T	$\mathcal{T}_{\leq 1}$	$(\mathcal{T}_{\leq 1})^2$	$(\mathcal{T}_{\leq 1})_s$	$((\mathcal{T}_{\leq 1})^2)_s$	37
							F	Theorem 6	Unicorns 2.0			38
				F	T	T	T	$\mathcal{T}_{\leq n}$	$(\mathcal{T}_{\leq n})^2$	$(\mathcal{T}_{\leq n})_s$	$((\mathcal{T}_{\leq n})^2)_s$	39
							F	Theorem 6	Unicorns 2.0			40
			F	T	T	T	T	[24]	$\mathcal{T}_1^{odd}$	$\mathcal{T}_{odd}^{\neq}$	$(\mathcal{T}_1^{odd})_s$	41
							F		$\mathcal{T}_{XI}$	$\mathcal{T}_{XIV}$	$(\mathcal{T}_{XI})_s$	42
						F	T	[25]	$\mathcal{T}_{2,3}^3$	[25]	$\mathcal{T}_{\neq}^\infty$	43
							F		$\mathcal{T}_{II}^3$		$\mathcal{T}_{XIII}$	44
				F	T	T	T	$\mathcal{T}_{\langle m,n\rangle}$	$(\mathcal{T}_{\langle m,n\rangle})^2$	$(\mathcal{T}_{\langle m,n\rangle})_s$	$((\mathcal{T}_{\langle m,n\rangle})^2)_s$	45
							F	Theorem 6	$\mathcal{T}_X$	$\mathcal{T}_{XII}$	$(\mathcal{T}_X)_s$	46
						F	T	[25]	$\mathcal{T}_{m,n}^\infty$	[25]	$(\mathcal{T}_{m,n}^\infty)_s$	47
							F		$\mathcal{T}_{VI}$		$(\mathcal{T}_{VI})_s$	48
		F	F	T	T	T	F	[25]	$\mathcal{T}_1^\varsigma$	$\mathcal{T}_{\varsigma,1}^{\neq}$	$(\mathcal{T}_1^\varsigma)_s$	49
						F	F		$\mathcal{T}_\varsigma^{\infty,3}$	[25]	$\mathcal{T}_{\varsigma\neq}^\infty$	50
					F	F	T		$\mathcal{T}_1^\infty$	$\mathcal{T}_{1,\infty}^{\neq}$	$(\mathcal{T}_1^\infty)_s$	51
							F		$\mathcal{T}_I^3$	$\mathcal{T}_{IX}$	$(\mathcal{T}_{IX})^2$	52
				F	T	T	F		$\mathcal{T}_n^\varsigma$	$\mathcal{T}_n^\varsigma$	$(\mathcal{T}_n^\varsigma)_s$	53
						F	F		$\mathcal{T}_{m,n}^\varsigma$	[25]	$(\mathcal{T}_{m,n}^\varsigma)_s$	54
					F	F	T		$\mathcal{T}_2^\infty$	$\mathcal{T}_{2,\infty}^{\neq}$	$(\mathcal{T}_2^\infty)_s$	55
							F		$\mathcal{T}_V$	$\mathcal{T}_{IV}$	$(\mathcal{T}_V)_s$	56



galois

Conclusion

Polite Combination:

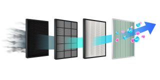
- Revolutionary
- Optimal ($G(SP) = D$)
- Only needs a solver
- Used in cvc5

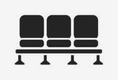
Politeness:

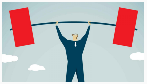
- Not enough
- You really need strong/additive politeness
- Does not improve Nelson Oppen

Future:

- $G(P) = SI$ for many-sorted signatures?
- Connections between more properties







Thank You!

Back up

Back up slides

Axioms

Name	Axiomatization
$T_=$	$\{P_n \to \psi_{=n} : n \in \mathbb{N}^*\}$
T_f	$\{[\psi^{=}_{\geq f_1(k)} \land \psi^{\neq}_{\geq f_0(k)}] \lor \bigvee_{i=1}^{k}[\psi^{=}_{=f_1(i)} \land \psi^{\neq}_{=f_0(i)}] : k \in \mathbb{N}^*\}$

- $f : \mathbb{N}^* \to \{0, 1\}$ is assumed to be a non-computable function, such that $f(1) = 1$ and, for every $k \geq 0$, f maps half of the numbers between 1 and 2^k to 1, and the other half to 0.
- $f_i(k)$ is the number of numbers between 1 and k that are mapped by f to i.

- $\psi^{=}_{\geq n} = \exists \overrightarrow{x}. [n \not\approx \land \bigwedge_{i=1}^{n} p(x_i)]$, $\quad \psi^{\neq}_{\geq n} = \exists \overrightarrow{x}. [n \not\approx \land \bigwedge_{i=1}^{n} \neg p(x_i)]$
- $\psi^{=}_{=n} = \exists \overrightarrow{x}. [n \not\approx \land \bigwedge_{i=1}^{n} p(x_i) \land \forall x. [p(x) \to \bigvee_{i=1}^{n} x = x_i]]$
- $\psi^{\neq}_{=n} = \exists \overrightarrow{x}. [n \not\approx \land \bigwedge_{i=1}^{n} \neg p(x_i) \land \forall x. [\neg p(x) \to \bigvee_{i=1}^{n} x = x_i]]$
- $p(x)$ stands for $s(x) = x$.

Proof

Theorem

$G(P) = SI$.

Proof.

- Let U be an undecidable set of numbers
- For each n, let T_n be:
  $\{P_m \rightarrow \psi_{\geq n+1} \mid m \in U\} \cup \{P_i \rightarrow \not P_j \mid i \neq j\}$
- For each n, T_n is polite.
- Let T be a theory that can be combined with every polite theory.
- Then T can be combined with every T_n.
- So $T \cup T_n$ is decidable.

□

Proof

Theorem

$G(P) = SI$.

Proof.

- ▷ Suppose T is not SI.
- ▷ By compactness, the models of some formula φ have a maximal finite cardinality n.
- ▷ $\varphi \wedge P_k$ is $T \cup T_n$-SAT iff $k \notin U$.
- ▷ But $T \cup T_n$ is decidable as T_n is polite and T can be combined with every polite theory.
- ▷ We solved an undecidable problem.

□