— TTVSI, London 2008 —

Decision Procedures for Parametric Theories

Sava Krstić*, Amit Goel*, Jim Grundy*, Cesare Tinelli**

*Intel Strategic CAD Labs
**The University of Iowa

This Talk

Based on work in

- S. Krstić, A. Goel, J. Grundy, and C. Tinelli
  Combined Satisfiability Modulo Parametric Theories
  TACAS, 2007.

- S. Krstić and A. Goel
  Architecting Solvers for SAT Modulo Theories: Nelson-Oppen with DPLL
  FroCoS, 2007.

Contribution

Nelson-Oppen framework for theories in parametrically polymorphic logics—a fresh foundation for design of SMT solvers.

- Endowing SMT with a rich typed input language that can model arbitrarily nested data structures
- Completeness of the Nelson-Oppen-style combination method proved for theories of all common datatypes
- Troublesome stable infinity condition replaced by a natural notion of type parametricity
- Issue of finite-cardinality constraints exposed as crucial for completeness

SAT Modulo Theories (SMT)

There are decision procedures for (fragments of) logical theories of common datatypes.

Use them to decide validity/satisfiability of formulas that involve symbols from several theories.

- $f(x) = x \;\Rightarrow\; f(2x - f(x)) = x$ $\qquad [\mathcal{T}_{\mathsf{UF}} + \mathcal{T}_{\mathsf{Int}}]$
- $\mathsf{head}(a) = f(x) + 1 \;\ldots$ $\qquad [\mathcal{T}_{\mathsf{UF}} + \mathcal{T}_{\mathsf{Int}} + \mathcal{T}_{\mathsf{List}}]$

The underlying logic is the classical (unsorted or multi-sorted) first-order logic.

SMT Solvers

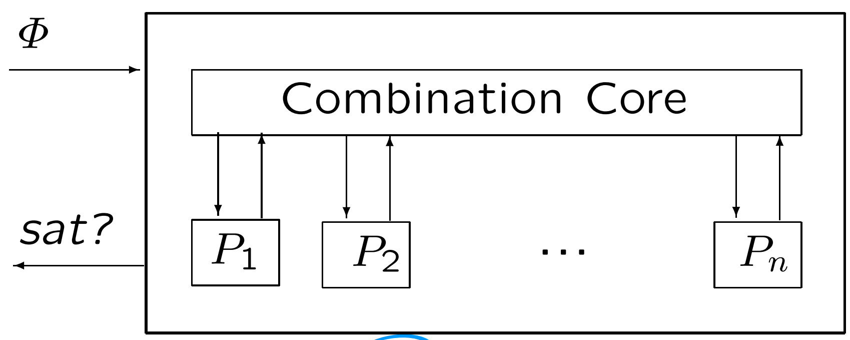
G. Nelson, D.C.Oppen *Simplification by cooperating decision procedures*, 1979

Input:

- theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ with disjoint signatures $\Sigma_1, \ldots, \Sigma_n$
- decision procedures P_i for satisfiability of sets of $\mathcal{T}_i$-literals

Output:

- a decision procedure for $(\mathcal{T}_1 + \cdots + \mathcal{T}_n)$-satisfiability of sets of $(\Sigma_1 + \cdots + \Sigma_n)$-literals.



SMT Solvers

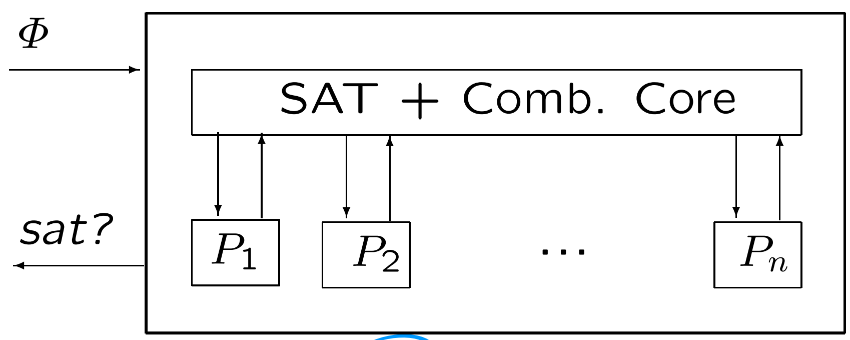
G. Nelson, D.C.Oppen *Simplification by cooperating decision procedures*, 1979

Input:

- theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ with disjoint signatures $\Sigma_1, \ldots, \Sigma_n$
- decision procedures P_i for satisfiability of sets of $\mathcal{T}_i$-literals

Output:

- a decision procedure for satisfiability of quantifier-free $(\mathcal{T}_1 + \cdots + \mathcal{T}_n)$-formulas.



Nelson-Oppen Combination Algorithm

1 Transform ("purify") the mixed input formula Φ into an equisatisfiable set of pure formulas $\Phi_1, \ldots, \Phi_n$

2 If $\Phi_i \models x = y$, update all Φ_j with $\Phi_j \cup \{x = y\}$

3 Repeat 2 until $\Phi_i \vdash \perp$ for some i (return **UNSAT**), or 2 no longer applies (return **SAT**)

Nelson-Oppen: Example

$\mathcal{T}_1 =$ theory of lists $\qquad \mathcal{T}_2 =$ linear arithmetic

Input set:

$$\Phi = \begin{cases} l_1 \neq l_2 \\ \mathsf{head}(l_2) \leq x \\ l = \mathsf{tail}(l_2) \\ l_1 = x :: l \\ \mathsf{head}(l) - \mathsf{head}(\mathsf{tail}\ l_1) + x \leq \mathsf{head}(l_2) \end{cases}$$

Purified sets:

$$\Phi_1 = \begin{cases} l_1 \neq l_2 \\ y_1 = \mathsf{head}(l_2) \\ l = \mathsf{tail}(l_2) \\ l_1 = x :: l \\ y_2 = \mathsf{head}(l) \\ y_3 = \mathsf{head}(\mathsf{tail}\ l_1) \end{cases} \qquad \Phi_2 = \begin{cases} y_1 \leq x \\ y_2 - y_3 + x \leq y_1 \end{cases}$$

Nelson-Oppen: Example [ctd]

Φ_1	Φ_2
$l_1 \neq l_2$	$y_1 \leq x$
$y_1 = \mathsf{head}(l_2)$	$y_2 - y_3 + x \leq y_1$
$l = \mathsf{tail}(l_2)$	
$l_1 = x :: l$	
$y_2 = \mathsf{head}(l)$	
$y_3 = \mathsf{head}(\mathsf{tail}\ l_1)$	

Nelson-Oppen: Example [ctd]

Φ_1	Φ_2
$l_1 \neq l_2$	$y_1 \leq x$
$y_1 = \mathsf{head}(l_2)$	$y_2 - y_3 + x \leq y_1$
$l = \mathsf{tail}(l_2)$	
$l_1 = x :: l$	
$y_2 = \mathsf{head}(l)$	
$y_3 = \mathsf{head}(\mathsf{tail}\ l_1)$	
$\longrightarrow\!\!\!\!\twoheadrightarrow$	$y_2 = y_3$

Nelson-Oppen: Example [ctd]

Φ_1	Φ_2
$l_1 \neq l_2$	$y_1 \leq x$
$y_1 = \text{head}(l_2)$	$y_2 - y_3 + x \leq y_1$
$l = \text{tail}(l_2)$	
$l_1 = x :: l$	
$y_2 = \text{head}(l)$	
$y_3 = \text{head}(\text{tail } l_1)$	
$\twoheadrightarrow$	$y_2 = y_3$
$x = y_1$	$\twoheadleftarrow$

Nelson-Oppen: Example [ctd]

Φ_1	Φ_2
$l_1 \neq l_2$	$y_1 \leq x$
$y_1 = \mathsf{head}(l_2)$	$y_2 - y_3 + x \leq y_1$
$l = \mathsf{tail}(l_2)$	
$l_1 = x :: l$	
$y_2 = \mathsf{head}(l)$	
$y_3 = \mathsf{head}(\mathsf{tail}\ l_1)$	
$\twoheadrightarrow$	$y_2 = y_3$
$x = y_1$	$\twoheadleftarrow$
UNSAT	

Completeness of Nelson-Oppen

- Suppose N-O halts in a state $\{\Phi_1, \Phi_2\}$ and E is the set of all equalities between shared variables that have been derived.
- Let $\Delta = E \cup \{x \neq y \mid x = y \notin E\}$ (arrangement)
  - $\Phi_1 \cup \Delta$ has a $\mathcal{T}_1$-model M_1
  - $\Phi_2 \cup \Delta$ has a $\mathcal{T}_2$-model M_2
- If M_1 and M_2 have the same cardinality*, we can "glue" them and obtain a model of $\Phi_1 \cup \Phi_2 \cup \Delta$.

*Single-sorted *FOL*!

Completeness of Nelson-Oppen

- Suppose N-O halts in a state $\{\Phi_1, \Phi_2\}$ and E is the set of all equalities between shared variables that have been derived.
- Let $\Delta = E \cup \{x \neq y \mid x = y \notin E\}$ (arrangement)
  - $\Phi_1 \cup \Delta$ has a $\mathcal{T}_1$-model M_1
  - $\Phi_2 \cup \Delta$ has a $\mathcal{T}_2$-model M_2
- If M_1 and M_2 have the same cardinality, then we can "glue" them and obtain a model of $\Phi_1 \cup \Phi_2 \cup \Delta$.

Q But what if the cardinalities of M_1 and M_2 are not the same?

Problem with Cardinality Mismatch

$$\begin{aligned} \mathcal{T}_1 &= \text{“theory of uninterpreted functions”} \\ \mathcal{T}_2 &= \text{theory of bits (not stably-infinite)} \end{aligned}$$

Purified Input:

Φ_1	Φ_2
$f(f(x)) \neq x$	$x = 0$
$f(f(y)) \neq y$	$y = 1$

- There are no equations to propagate; N-O returns **SAT**
- Φ_1 requires at least 3 elements for a model

The Notorious Stable Infiniteness Restriction

Definition: A first-order theory $\mathcal{T}$ is stably infinite if every satisfiable formula is satisfiable in an infinite model.

- This condition guarantees completeness of N-O, but
  - it's not easy to prove
  - it's not true in some important cases (eg, bitvectors)
- General understanding: the condition doesn't matter much
- Lot of research shows completeness of N-O without it: [Tinelli-Zarba'04], [Fontaine-Gribomont'04], [Zarba'04], [Ghilardi-Nicolini-Zuchelli'07], [Ranise-Ringeissen-Zarba'05]

Parametricity, Not Stable Infiniteness: Example

- $\Phi = \left\{ \begin{array}{c} \mathsf{head}(\mathsf{tail}\, a) = \mathsf{head}\, a + x \\ \mathsf{head}\, b = \mathsf{head}\, a + x \\ \mathsf{tail}\, a = \mathsf{tail}\, b \end{array} \right\}$

- $\Phi_{\mathsf{List}} = \left\{ \begin{array}{c} \mathsf{tail}\, a = \mathsf{tail}\, b \\ u = \mathsf{head}\, a \\ v = \mathsf{head}\, b \\ w = \mathsf{head}(\mathsf{tail}\, a) \end{array} \right\}$ $\quad \Phi_{\mathsf{Int}} = \left\{ \begin{array}{c} w = u + x \\ v = u + x \end{array} \right\}$ $\quad \Delta = \{v = w \neq u\}$

- $\begin{pmatrix} u & v & w & a & b \\ 5 & 9 & 9 & [5, 9] & [9, 9] \end{pmatrix} \models \Phi_{\mathsf{List}} \cup \Delta$ $\qquad \begin{pmatrix} u & v & w & x \\ \mathbf{1} & \mathbf{2} & \mathbf{2} & 1 \end{pmatrix} \models \Phi_{\mathsf{Int}} \cup \Delta$

- $\mathcal{T}_{\mathsf{List}}$ knows nothing about $\mathbb{Z}$ and cannot distinguish the pair $(5, 9)$ from any pair (⚽, ☕) of distinct symbols:

$$\begin{pmatrix} u & v & w & a & b \\ ⚽ & ☕ & ☕ & [⚽, ☕] & [☕, ☕] \end{pmatrix} \models \Phi_{\mathsf{List}} \cup \Delta$$

$\therefore$ To construct a model for $\Phi_{\mathsf{List}} \cup \Phi_{\mathsf{Int}} \cup \Delta$, use the blue assignment to u, v, w

All Theories of Practical Interest Are Parametric

$$\Sigma_{\mathsf{Int}} = \langle \mathsf{Int} \mid 0^{\mathsf{Int}}, 1^{\mathsf{Int}}, (-1)^{\mathsf{Int}}, \ldots, +^{\mathsf{Int}^2 \to \mathsf{Int}}, -^{\mathsf{Int}^2 \to \mathsf{Int}}, \times^{\mathsf{Int}^2 \to \mathsf{Int}}, \leq^{\mathsf{Int}^2 \to \mathsf{Bool}}, \ldots \rangle$$

$$\Sigma_{\mathsf{Array}} = \langle \mathsf{Array} \mid \mathsf{mk\_arr}^{\beta \to \mathsf{Array}(\alpha,\beta)}, \mathsf{read}^{[\mathsf{Array}(\alpha,\beta),\alpha] \to \beta}, \mathsf{write}^{[\mathsf{Array}(\alpha,\beta),\alpha,\beta] \to \mathsf{Array}(\alpha,\beta)} \rangle$$

$$\Sigma_{\mathsf{List}} = \langle \mathsf{List} \mid \mathsf{cons}^{[\alpha,\mathsf{List}(\alpha)] \to \mathsf{List}(\alpha)}, \mathsf{nil}^{\mathsf{List}(\alpha)}, \mathsf{head}^{\mathsf{List}(\alpha) \to \alpha}, \mathsf{tail}^{\mathsf{List}(\alpha) \to \mathsf{List}(\alpha)} \rangle$$

$$\Sigma_{\mathsf{UF}} = \langle \Rightarrow \mid @^{[\alpha \Rightarrow \beta, \alpha] \to \beta} \rangle$$

$$\Sigma_{\times} = \langle \times \mid \langle \_, \_ \rangle^{[\alpha,\beta] \to \alpha \times \beta}, \mathsf{fst}^{\alpha \times \beta \to \alpha}, \mathsf{snd}^{\alpha \times \beta \to \beta} \rangle$$

$$\Sigma_{\mathsf{BitVec32}} = \ldots$$

$$\Sigma_{\mathsf{Sets}} = \ldots$$

$$\Sigma_{\mathsf{Multisets}} = \ldots$$

Syntax for Parametric Theories

- Parametrically typed FOL (**PTFOL**), an applicative fragment of HOL
  - A **signature** is a pair $\Sigma = \langle O \mid K \rangle$, where O is the set of type operators and K is a set of constants
  - A **type operator** is symbol, with an arity ≥ 0
  - Build **types** using type operators and **type variables**, as usual
  - A **constant** is a symbol, with an arity $[\sigma_1, \ldots, \sigma_k] \to \sigma$
  - Build (well-typed) **terms** by applying costants to **term variables** and simpler terms

- The **logical signature**, to be included in all others:

$$\boxed{\Sigma_{\mathsf{Eq}} = \langle \mathsf{Bool} \mid =^{\alpha^2 \to \mathsf{Bool}}, \mathsf{ite}^{[\mathsf{Bool}, \alpha, \alpha] \to \alpha}, \mathsf{true}^{\mathsf{Bool}}, \mathsf{false}^{\mathsf{Bool}}, \neg^{\mathsf{Bool} \to \mathsf{Bool}}, \wedge^{\mathsf{Bool}^2 \to \mathsf{Bool}}, \ldots \rangle}$$

Theory Semantics: Parametric Structures

- $\mathcal{T}_{\mathsf{Array}}$—like any other datatype theory—is a theory of a single model: there is a unique semantic structure associated with the signature

$$\Sigma_{\mathsf{Array}} = \langle \mathsf{Array} \mid \mathsf{mk\_arr}^{\beta \to \mathsf{Array}(\alpha,\beta)}, \mathsf{read}^{[\mathsf{Array}(\alpha,\beta),\alpha] \to \beta}, \mathsf{write}^{[\mathsf{Array}(\alpha,\beta),\alpha,\beta] \to \mathsf{Array}(\alpha,\beta)} \rangle$$

- Parametric types: Array is a binary set operation: $\mathsf{Array}(I, E)$ is the set of arrays indexed by I, with elements in A

- Parametric elements: read is an indexed family of binary operations $\mathsf{read}_{I,E} : [\mathsf{Array}(I, E), I] \to E$

Q But what does it mean that Array and read are <u>parametric</u>?

Semantics: Parametric Set Operators

- The meaning of the type operator List is the function $[\text{List}] : \mathcal{U} \longrightarrow \mathcal{U}$, where $\mathcal{U}$ is a universe of sets
- List also acts on relations: for every $R\colon A \leftrightarrow B$, there is an induced relation $[\text{List}]^{\#}(R)\colon [\text{List}](A) \longleftrightarrow [\text{List}](B)$
- Functoriality:
  - $[\text{List}]^{\#}(\text{id}_A) = \text{id}_{[\text{List}](A)}$
  - $[\text{List}]^{\#}(R \circ R') = [\text{List}]^{\#}(R) \circ [\text{List}]^{\#}(R')$
- We require that interpretations $[F]$ of all type operators F be **functorial on partial bijections**
  - This captures the uniformity of [List] and other parametric set operations

NB Reynolds Parametricity is related, but not the same

Semantics: Parametric Elements

- The meaning of the constant $\mathsf{fst}^{\alpha\times\beta\to\alpha}$ is the family of functions $\mathsf{fst}_{A,B} : A \times B \to A$
- This family is **parametric**:

$$\mathsf{fst}_{A,B}(a,b)\ (R\times S)\ \mathsf{fst}_{A',B'}(a',b')$$

holds for all partial bijections $R\colon A \leftrightarrow A'$, $S\colon B \leftrightarrow B'$, and related elements $a\,R\,a'$, $b\,S\,b'$

Semantics: Satisfiability

Fix a theory $\mathcal{T}$

- Interpretation of types $[\![\tau]\!]\,\iota$ requires a type environment ι — a map from type variables to sets
- Interpretation of terms $[\![t]\!]\,\iota\,\rho$ requires a type environment and a term environment ρ — a map from term variables to elements of sets that ι associates to the types of these variables
- A formula (term of boolean type) is **$\mathcal{T}$-satisfiable** if $[\![\phi]\!]\,\iota\,\rho = \mathsf{true}$ for some ι and ρ

(Nothing fancy here)

Combination of Theories

- The signatures $\Sigma = \langle O \mid K \rangle$ and $\Sigma' = \langle O' \mid K' \rangle$ are **disjoint** if the $O \cap O' = \varnothing$ and $K \cap K' = \varnothing$
- . . . and then their **union signature** is $\Sigma + \Sigma' = \langle O \cup O' \mid K \cup K' \rangle$
- If $\mathcal{T}$ is a Σ-theory and $\mathcal{T}'$ is a Σ'-theory, there is a well-defined **union theory** $\mathcal{T} + \mathcal{T}'$ over $(\Sigma + \Sigma')$

Mixed Formulas and Purification

- In *FOL*, the formula $1 + f(x) = f(1 + f(x))$ purifies into $\Phi_{\mathsf{UF}} = \{y = f(x), u = f(z)\}$, $\Phi_{\mathsf{Int}} = \{z = 1 + y, z = u\}$
- For us, $\Phi_{\mathsf{UF}} = \{y^{\mathsf{Int}} = f^{\mathsf{Int} \Rightarrow \mathsf{Int}}\, x^{\mathsf{Int}}, u^{\mathsf{Int}} = f^{\mathsf{Int} \Rightarrow \mathsf{Int}}\, z^{\mathsf{Int}}\}$
- This Φ_{UF} is not Σ_{UF}-pure because of presence of Int
  - Φ_{UF} is **semipure**—its impurity is at the type level only
- We can use a pure appoximation

$$\Phi_{\mathsf{UF}}^{\mathsf{pure}} = \{y^{\alpha} = f^{\alpha \Rightarrow \alpha}\, x^{\alpha}, u^{\alpha} = f^{\alpha \Rightarrow \alpha}\, z^{\alpha}\}$$

- . . . but this transformation is not safe!

Cardinality Constraints

$$\Phi : \mathsf{distinct}(x_1^{\mathsf{List}(\alpha)}, \ldots, x_5^{\mathsf{List}(\alpha)}) \qquad \mathsf{tail}(\mathsf{tail}\ x_i^{\mathsf{List}(\alpha)}) = \mathsf{nil}$$

$$\Phi_1 : \mathsf{distinct}(x_1^{\mathsf{List}(\mathsf{Bool})}, \ldots, x_5^{\mathsf{List}(\mathsf{Bool})}) \quad \mathsf{tail}(\mathsf{tail}\ x_i^{\mathsf{List}(\mathsf{Bool})}) = \mathsf{nil} \qquad (i = 1, \ldots, 5)$$

$$\Phi_2 : \mathsf{distinct}(x_1^{\mathsf{List}(\mathsf{Int})}, \ldots, x_5^{\mathsf{List}(\mathsf{Int})}) \qquad \mathsf{tail}(\mathsf{tail}\ x_i^{\mathsf{List}(\mathsf{Int})}) = \mathsf{nil}$$

- $\Phi = \Phi_1^{\text{pure}}$, $\Phi = \Phi_2^{\text{pure}}$; Φ_2 is satisfiable, Φ_1 is not
- Instead of Φ_1, the $\mathcal{T}_{\text{List}}$-solver gets Φ together with the cardinality constraint $\alpha \doteq 2$
- Instead of Φ_2, the $\mathcal{T}_{\text{List}}$-solver just gets Φ

Lemma A semipure query Φ is satisfiable iff Φ^{pure} is satisfiable together with the cardinality constraints

Purification

Turn a mixed $(\mathcal{T}_1 + \cdots + \mathcal{T}_n)$-query Φ into the **purified form**

$$\Phi_{\mathbb{B}} \cup \Phi_{\mathbb{E}} \cup \Phi_1 \cup \cdots \cup \Phi_n$$

where

- $\Phi_{\mathbb{B}}$ is a set of propositional formulas
- $\Phi_{\mathbb{E}} = \{\ldots\ldots \; p^{\mathsf{Bool}} \leftrightarrow x^{\tau} = y^{\tau} \; \ldots\ldots\}$
- $\Phi_i = \{\ldots\ldots \; p^{\mathsf{Bool}} \leftrightarrow \psi \; \ldots\ldots\} \cup \{\ldots\ldots \; x^{\tau} = t \; \ldots\ldots\}$ (i-semipure)

Ex: $f(x) = x \vee f(2 * x - f(x)) > x$ becomes

$\Phi_{\mathbb{B}} = \{p \vee q\}$	$\Phi_{\mathbb{E}} = \{p \leftrightarrow y = x\}$,
$\Phi_{\mathsf{UF}} = \{y = f(x),\ u = f(z)\}$	$\Phi_{\mathsf{Int}} = \{q \leftrightarrow u > x,\ z = 2 * x - y\}$

A PTFOL Nelson-Oppen Combination Theorem

Theorem The query

$$\Phi \;=\; \Phi_{\mathbb{B}} \cup \Phi_{\mathbb{E}} \cup \Phi_1 \cup \cdots \cup \Phi_n$$

is $(\mathcal{T}_1 + \cdots + \mathcal{T}_n)$-satisfiable iff there exist

- an assignment M of the atoms in $\Phi_{\mathbb{B}}$
- an arrangement Δ of the non-Bool variables in Φ

such that

- $M \models \Phi_{\mathbb{B}}$
- $M, \Delta \models \Phi_{\mathbb{E}}$
- $(\Phi_i \cup M \cup \Delta)^{\text{pure}} \cup \Phi_i^{\text{card}}$ is $\mathcal{T}_i$-satisfiable (for $i = 1, \ldots, n$)

Theoretical Requirement: Flexible Structures

A theory $\mathcal{T}$ is **flexible** if for

- every query Φ
- every injective $\langle \iota, \rho \rangle$ such that $\langle \iota, \rho \rangle \models \Phi$
- every α in the domain of ι
- every $\kappa > |\iota(\alpha)|$

there exist injective $\langle \iota^{\mathrm{up}(\kappa)}, \rho^{\mathrm{up}(\kappa)} \rangle$ and $\langle \iota^{\mathrm{down}}, \rho^{\mathrm{down}} \rangle$ satisfying Φ such that

- $\iota^{\mathrm{up}(\kappa)}(\beta) = \iota(\beta) = \iota^{\mathrm{down}}(\beta)$ for every $\beta \neq \alpha$
- $\iota^{\mathrm{up}(\kappa)}(\alpha)$ has cardinality κ [up-flexibility]
- $\iota^{\mathrm{down}}(\alpha)$ is countable [down-flexibility]

¿Lemma? *All parametric theories are flexible.*

- Proved for a large class, including all datatype theories

Combined Solver: Architecture & Implementation

- **NODPLL**: Top-level architecture of an SMT solver
  - Nelson-Oppen with DPLL
  - presented as a transition system
  - precisely formulates the main algorithmic features
  - guides the implementation
- **DPT**: Decision Procedure Toolkit
  - SMT solver developed at Intel Strategic CAD Labs
  - written in OCaml
  - open source (SourceForge)
  - clarity with competitive performance