

# Hierarchic reasoning in local theory extensions

Viorica Sofronie-Stokkermans

Max-Planck-Institut für Informatik

Saarbrücken

CADE 20, Tallinn, July 24–28, 2005

# Motivation

---

## Hierarchic reasoning in extensions of theories

### Example (Mathematics, Verification)

$\mathcal{T}_0 =$  real numbers with ordering

$\mathcal{T}_1 =$  free or monotone functions on real numbers

**Task:** prove some property of free (monotone) functions  
use a prover for the real numbers as a “black-box”

# Motivation

---

## Hierarchic reasoning in extensions of theories

**Example** (Knowledge representation)

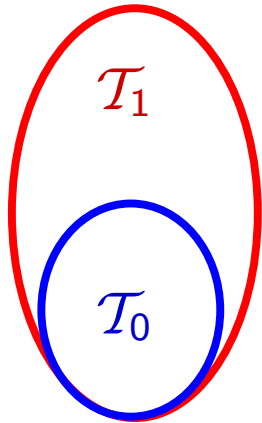
$\mathcal{T}_0 =$  (semi)lattices (distributive lattices, Boolean algebras, ...)

$\mathcal{T}_1 =$  free or monotone functions on  $\mathcal{T}_0$

**Task:** prove some property of free (monotone) functions  
use a prover for  $\mathcal{T}_0$  as a “black-box”

# Motivation

---



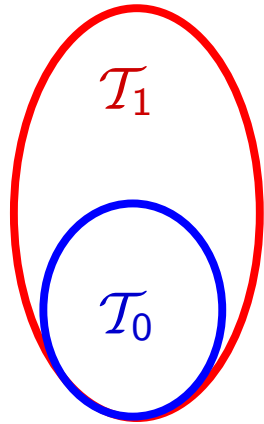
$\mathcal{T}_1$ :  $\Sigma_1$ -theory;  $\mathcal{T}_0 \subseteq \mathcal{T}_1$   $\Sigma_1$  extension of  $\Sigma_0$ .

$\mathcal{T}_0$ :  $\Sigma_0$ -theory.

Can use a prover for  $\mathcal{T}_0$  as a black-box to prove theorems in  $\mathcal{T}_1$ ?

# Motivation

---



$\mathcal{T}_1$ :  $\Sigma_1$ -theory;  $\mathcal{T}_0 \subseteq \mathcal{T}_1$   $\Sigma_1$  extension of  $\Sigma_0$ .

$\mathcal{T}_0$ :  $\Sigma_0$ -theory.

Can use a prover for  $\mathcal{T}_0$  as a black-box to prove theorems in  $\mathcal{T}_1$ ?

- extensions with **relations** [Bachmair, Ganzinger, Waldmann'94]
- extensions with **partial functions** [Ganzinger, Waldmann, VS'04]

- constraint superposition calculus  $\mapsto$  **hierarchic reasoning**

strong conditions on base theory: compact, universal.

# This talk

---

Extensions with function symbols:

Identify situations when

- hierarchic reasoning is possible

... and simple:

- no sophisticated implementations are necessary

# This talk

---

Extensions with function symbols:

Identify situations when

- hierarchic reasoning is possible

... and simple:

- no sophisticated implementations are necessary
- complexity of the  $\forall$  fragment of extension expressible in terms of complexity of the  $\forall$  (or  $\forall\exists$ ) fragment of base theory

# Example: The sum of two Lipschitz functions

---

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$(\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c|$$

$$(\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c|$$

## Problems:

- A prover for  $\mathbb{R}$  does not know about  $f, g$
- A prover for first-order logic may have problems with the reals
- Nelson-Oppen reasoning in theory combinations not possible

# Example: The sum of two Lipschitz functions

---

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$(\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c|$$

$$(\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c|$$

- Hierarchic reasoning

reduce to the problem of checking a family of constraints over  $\mathbb{R}$

- Modular reasoning

if we separate the information about  $f$  and  $g$  at the beginning:

no need to combine the information again at a later point

# Idea

---

$\Sigma_1$  extension of  $\Sigma_0$  with **function symbols**

$\mathcal{K}$  set of  $\Sigma_1$ -clauses;  $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

**Task:** Check whether  $\mathcal{T}_1 \cup G \models \perp$

**Approach:** Consider a **relational approximation** of the problem:  
(approximate extension functions with functional relations)

$$\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \models \perp$$

# Idea

---

$\Sigma_1$  extension of  $\Sigma_0$  with **function symbols**

$\mathcal{K}$  set of  $\Sigma_1$ -clauses;  $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

**Task:** Check whether  $\mathcal{T}_1 \cup G \models \perp$

**Approach:** Consider a **relational approximation** of the problem:  
(approximate extension functions with functional relations)

$$\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \models \perp$$

**Soundness:**  $\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \models \perp \implies \mathcal{T}_1 \cup G \models \perp$

# Idea

---

$\Sigma_1$  extension of  $\Sigma_0$  with **function symbols**

$\mathcal{K}$  set of  $\Sigma_1$ -clauses;  $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

**Task:** Check whether  $\mathcal{T}_1 \cup G \models \perp$

**Approach:** Consider a **relational approximation** of the problem:  
(approximate extension functions with functional relations)

$$\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \models \perp$$

**Soundness:**  $\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \models \perp \implies \mathcal{T}_1 \cup G \models \perp$

**Completeness:**  $\mathcal{T}_0 \cup \mathcal{K}^* \cup G^* \not\models \perp \implies \exists$  “partial” model

**If** every “partial” model can be embedded into  
a total model of  $\mathcal{T}_0 \cup \mathcal{K}$

**then**  $\mathcal{T}_1 \cup G \not\models \perp$

# Example: The sum of two Lipschitz functions

---

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$(\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c|$$

$$(\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c|$$

## Partial models can be made total

- approximate extension functions with functional relations

$$\mathcal{T}_0 \cup \mathcal{K}^* \cup \mathcal{G}^* \models \perp \quad \mapsto \quad \text{sound and complete}$$

## Local theory extension

- can even do better:  $\mathcal{T}_0 \cup \mathcal{K}[G]^* \cup \mathcal{G}^* \models \perp$

# Overview

---

## Local theory extensions

- various notions of locality of an extension
- link with embeddability of partial algebras into total algebras
- hierarchic reasoning (direct argument)
- parameterized complexity

# Locality, tractability, embeddability

---

$\mathcal{T}_1 := \mathcal{K}$  set of equational Horn clauses;

$\mathcal{K}$  is **local**, if for ground Horn clauses  $C$ ,  $\mathcal{K} \models C$  iff  $\mathcal{K}[C] \models C$

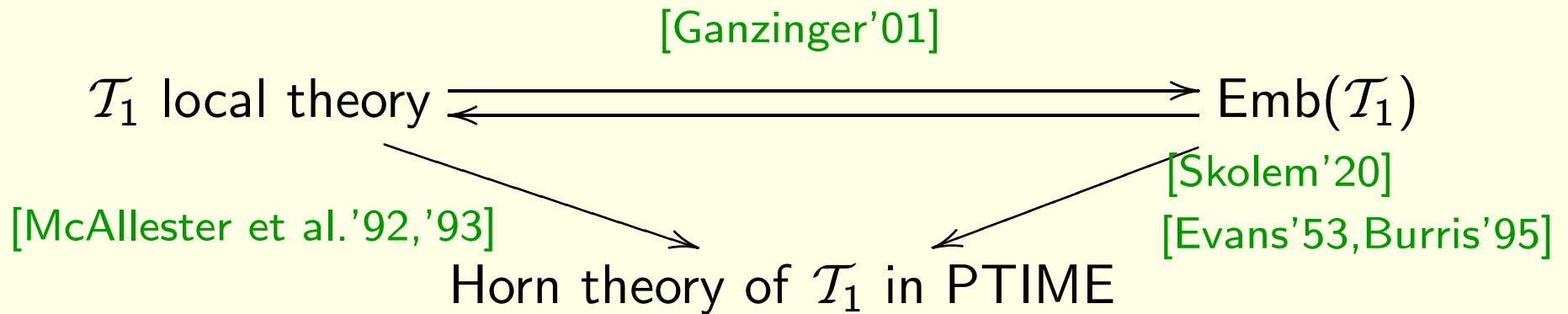
Local theories [Givan, McAllester'92] capture PTIME

# Locality, tractability, embeddability

---

$\mathcal{T}_1 := \mathcal{K}$  set of equational Horn clauses;

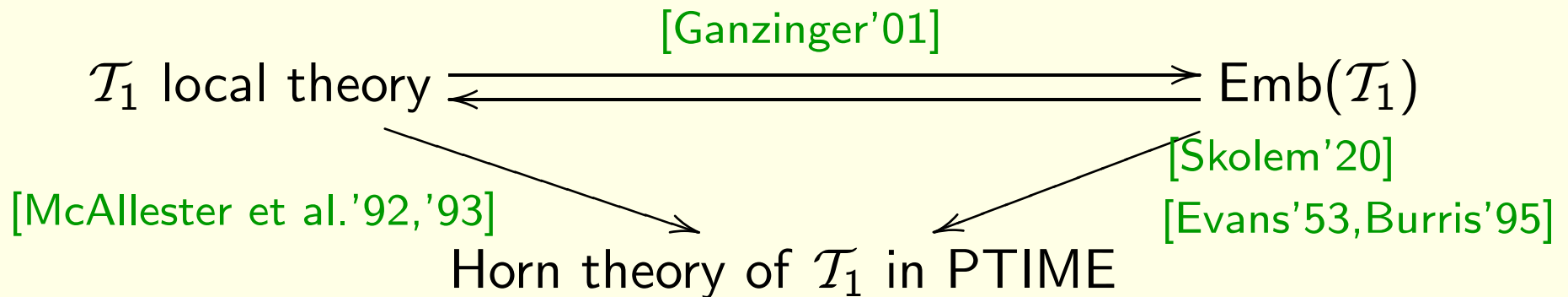
$\mathcal{K}$  is **local**, if for ground Horn clauses  $C$ ,  $\mathcal{K} \models C$  iff  $\mathcal{K}[C] \models C$



# Locality, tractability, embeddability

$\mathcal{T}_1 := \mathcal{K}$  set of equational Horn clauses;

$\mathcal{K}$  is **local**, if for ground Horn clauses  $C$ ,  $\mathcal{K} \models C$  iff  $\mathcal{K}[C] \models C$



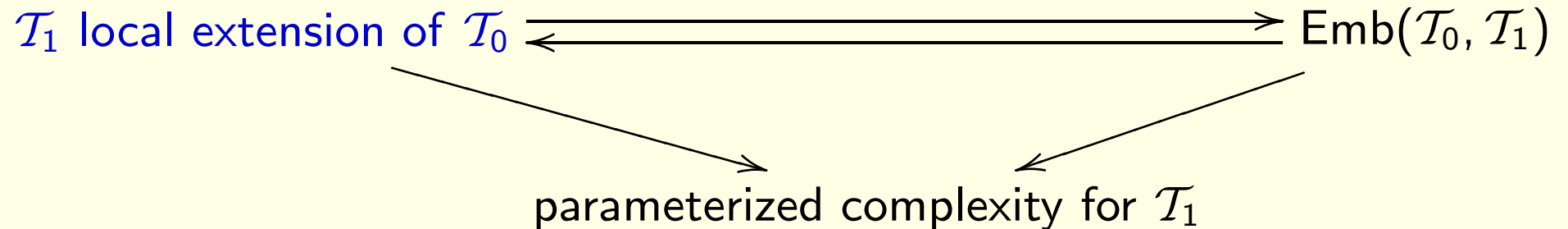
$\mathcal{K}$  is **stably local**, if for ground Horn clauses  $C$ ,  $\mathcal{K} \models C$  iff  $\mathcal{K}^{[C]} \models C$

# Local extensions

---

$\mathcal{K}$  set of equational clauses;  $\mathcal{T}_0$  theory;  $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

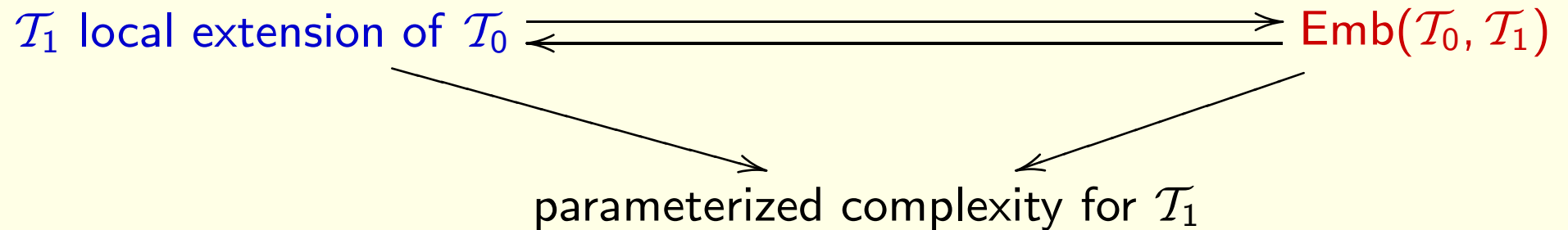
(Loc)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  is local, if for ground clauses  $G$ ,  
 $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \perp$  iff  $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$  has no (partial) model



# Local extensions

$\mathcal{K}$  set of equational clauses;  $\mathcal{T}_0$  theory;  $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

(Loc)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  is **local**, if for ground clauses  $G$ ,  
 $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \perp$  iff  $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$  has no (partial) model

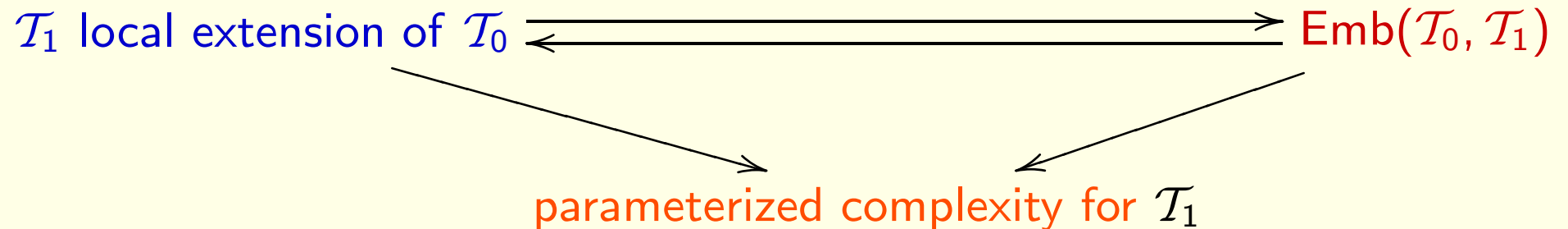


(SLoc)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  is **stably local**, if for ground clauses  $G$ ,  
 $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \perp$  iff  $\mathcal{T}_0 \cup \mathcal{K}^{[G]} \cup G$  has no (partial) model

# Local extensions

$\mathcal{K}$  set of equational clauses;  $\mathcal{T}_0$  theory;  $\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$

(Loc)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  is **local**, if for ground clauses  $G$ ,  
 $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \perp$  iff  $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$  has no (partial) model



(SLoc)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  is **stably local**, if for ground clauses  $G$ ,  
 $\mathcal{T}_0 \cup \mathcal{K} \cup G \models \perp$  iff  $\mathcal{T}_0 \cup \mathcal{K}^{[G]} \cup G$  has no (partial) model

# Validity of clauses in partial algebras

---

- Evans validity (equational case: [Evans'53])
- weak equality

Horn clauses: [Burris'95], [Ganzinger'01]

# Evans validity

$C := \bigvee_i u_i \not\approx v_i \vee \bigvee_j s_j \approx t_j \vee \bigvee_k L_k$      $C$  true iff one of its literals is true

$s \approx t$  true if -  $s, t$  defined and equal, or  
-  $s$  and  $t$  undefined, or  
-  $s$  or  $t$  **irrelevant**  
(proper subterm undefined)

$u \not\approx v$  true if -  $u, v$  defined  
and different, or  
-  $u$  or  $v$  undefined  

---

 $[\neg]P(t_1, \dots, t_n)$  similarly

**Example:** Hold in standard partial model?

**Yes:**  $\text{car}(\text{nil}) \approx \text{cdr}(\text{nil})$      $\text{car}(\text{nil}) \not\approx \text{cdr}(\text{nil})$     **No:**  $\text{car}(\text{nil}) \approx \text{nil}$   
 $\text{car}(\text{cdr}(\text{nil})) \approx \text{nil}$      $\text{car}(\text{cdr}(\text{nil})) \not\approx \text{nil}$

# Weak validity

$C := \bigvee_i u_i \not\approx v_i \vee \bigvee_j s_j \approx t_j \vee \bigvee_k L_k$      $C$  true iff one of its literals is true

$s \approx t$  true if -  $s, t$  defined  
                  and equal, or  
                  -  $s$  or  $t$  undefined

---

$P(t_1, \dots, t_n)$     similarly

$u \not\approx v$  true if -  $u, v$  defined  
                  and different, or  
                  -  $u$  or  $v$  undefined

---

$\neg P(t_1, \dots, t_n)$     similarly

**Example:** Hold in standard partial model?

**Yes:**     $\text{car}(\text{nil}) \approx \text{cdr}(\text{nil})$      $\text{car}(\text{nil}) \not\approx \text{cdr}(\text{nil})$     **Yes:**     $\text{car}(\text{nil}) \approx \text{nil}$   
           $\text{car}(\text{cdr}(\text{nil})) \approx \text{nil}$      $\text{car}(\text{cdr}(\text{nil})) \not\approx \text{nil}$

# Embeddability of partial into total models

---

$\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ ,  $\mathcal{K}$  set of clauses;  $\Pi_0 = (\Sigma_0, \text{Pred})$ ,  $\Pi = (\Sigma_0 \cup \Sigma_1, \text{Pred})$

## Notations:

$\text{PMod}(\Sigma_1, \mathcal{T}_1)$  Evans partial models of  $\mathcal{K}$  which are total models of  $\mathcal{T}_0$

$\text{PMod}_w(\Sigma_1, \mathcal{T}_1)$  Weak partial models of  $\mathcal{K}$  which are total models of  $\mathcal{T}_0$

## Embeddability conditions

(Emb) Every  $A \in \text{PMod}(\Sigma_1, \mathcal{T}_1)$  weakly embeds into a total model of  $\mathcal{T}_1$ .

(Emb<sub>w</sub>) Every  $A \in \text{PMod}_w(\Sigma_1, \mathcal{T}_1)$  weakly embeds into a total model of  $\mathcal{T}_1$ .

(Emb)<sup>f</sup>, (Emb<sub>w</sub>)<sup>f</sup> refer only to **finite** partial algebras

**Lemma:**  $\text{PMod}(\Sigma_1, \mathcal{T}_1) \subseteq \text{PMod}_w(\Sigma_1, \mathcal{T}_1)$ . Therefore (Emb<sub>w</sub>)  $\Rightarrow$  (Emb)

# Embeddability implies locality

---

## Theorem

$\mathcal{K}$  set of  $\Sigma_1$ -flat,  $\Sigma_1$ -linear clauses. Then for  $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ :

- (1)  $(\text{Emb}_w) \Rightarrow (\text{Loc})$ .
- (2)  $\mathcal{T}_0$  locally finite, universal &  $\mathcal{K}$ : finitely many ground terms:  
 $(\text{Emb}_w^f) \Rightarrow (\text{Loc}^f)$ .

Similar relationships between embeddability of Evans partial models and stable locality ( $\mathcal{T}_0$  required to be universal;  $\mathcal{K}$  can be arbitrary).

# Embeddability implies locality

---

## Theorem

$\mathcal{K}$  set of  $\Sigma_1$ -flat,  $\Sigma_1$ -linear clauses. Then for  $\mathcal{T}_0 \subseteq \mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ :

- (1)  $(\text{Emb}_w) \Rightarrow (\text{Loc})$ .
- (2)  $\mathcal{T}_0$  locally finite, universal &  $\mathcal{K}$  finitely many ground terms:  
 $(\text{Emb}_w^f) \Rightarrow (\text{Loc}^f)$ .

Similar relationships between embeddability of Evans partial models and stable locality ( $\mathcal{T}_0$  required to be universal;  $\mathcal{K}$  can be arbitrary).

**Consequence:** Can identify local and stably local extensions by proving that embedding properties hold

# Examples of local extensions

---

Extensions of a theory  $\mathcal{T}_0$ :

- with **free function symbols**
- with **selectors**  $\{s_1, \dots, s_n\}$  for an  $n$ -ary function  $c$ , injective in  $\mathcal{T}_0$ .

$$s_i(c(x_1, \dots, x_n)) = x_i$$

$$x = c(x_1, \dots, x_n) \rightarrow c(s_1(x), \dots, s_n(x)) = x$$

- with **monotone functions** for:

$$\mathcal{T}_0 \in \{\text{Posets, TotOrd, DenseTotOrd, Lat, SLat, DLat, BoolAlg}\}$$

$$\mathcal{T}_0 = \mathbb{R} \text{ theory of real numbers}$$

- with  **$\lambda$ -Lipschitz functions** at  $c$  for:

$$\mathcal{T}_0 = \mathbb{R} \text{ theory of real numbers}$$

# Example

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$\left. \begin{array}{l} (\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c| \\ (\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c| \end{array} \right\} \text{ set of clauses } \mathcal{K}$$

**Solution:**  $\mathbb{R} \cup \mathcal{K} \cup \underbrace{|f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |x - c|}_{G} \models \perp$

# Example

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$\left. \begin{array}{l} (\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c| \\ (\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c| \end{array} \right\} \text{ set of clauses } \mathcal{K}$$

**Solution:**  $\mathbb{R} \cup \mathcal{K} \cup \underbrace{|f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |x - c|}_G \models \perp$

Locality condition:

$\mathbb{R} \cup \mathcal{K}[G] \cup G$  has no partial model where  $f(c), f(d), g(c), g(d)$  defined

# Example

$$\mathbb{R} \cup (\mathbb{L}_{c,\lambda_1}^f) \cup (\mathbb{L}_{c,\lambda_2}^g) \models \forall x |f(x)+g(x)-(f(c)+g(c))| \leq (\lambda_1+\lambda_2) \cdot |x-c|$$

$$\left. \begin{array}{l} (\mathbb{L}_{c,\lambda_1}^f) \quad \forall x |f(x) - f(c)| \leq \lambda_1 \cdot |x - c| \\ (\mathbb{L}_{c,\lambda_2}^g) \quad \forall x |g(x) - g(c)| \leq \lambda_2 \cdot |x - c| \end{array} \right\} \text{ set of clauses } \mathcal{K}$$

**Solution:**  $\mathbb{R} \cup \mathcal{K} \cup \underbrace{|f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |x - c|}_{G} \models \perp$

**Locality condition:**

$\mathbb{R} \cup \mathcal{K}[G] \cup G$  has no partial model where  $f(c), f(d), g(c), g(d)$  defined

$$\mathcal{K}[G] : \quad |f(d) - f(c)| \leq \lambda_1 \cdot |d - c| \wedge |f(c) - f(c)| \leq \lambda_1 \cdot |c - c| \\ |g(d) - g(c)| \leq \lambda_2 \cdot |d - c| \wedge |g(c) - g(c)| \leq \lambda_2 \cdot |c - c|$$

**size:** polynomial in  $|\text{st}(G)|$  for a fixed  $\mathcal{K}$

# Relational translations

---

$\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ ,  $\mathcal{K}$  set of clauses;  $\Pi_0 = (\Sigma_0, \text{Pred})$ ,  $\Pi = (\Sigma_0 \cup \Sigma_1, \text{Pred})$

**Step 1:** **purify** and **flatten** (abstract out  $\Sigma_1$ -terms)

**Example:**  $|f(c + d) - f(d)| \leq l \cdot |c| \mapsto \begin{cases} c + d = c_1 \\ f(c_1) = c_2 \\ f(d) = c_3 \\ |c_2 + c_3| \leq l \cdot |c| \end{cases}$

# Relational translations

---

$\mathcal{T}_1 = \mathcal{T}_0 \cup \mathcal{K}$ ,  $\mathcal{K}$  set of clauses;  $\Pi_0 = (\Sigma_0, \text{Pred})$ ,  $\Pi = (\Sigma_0 \cup \Sigma_1, \text{Pred})$

**Step 1:** purify and flatten (abstract out  $\Sigma_1$ -terms)

$$\text{Example: } |f(c + d) - f(d)| \leq l \cdot |c| \mapsto \begin{cases} c + d = c_1 \\ f(c_1) = c_2 \\ f(d) = c_3 \\ |c_2 + c_3| \leq l \cdot |c| \end{cases}$$

**Step 2:** encode functions in  $\Sigma_1$  as relations

$$\text{Example: } f(d) = c_3 \mapsto r_f(d, c_3)$$

**Step 3:** express functionality of relations  $r_f$  (ground instances [G])

# Example

---

**Show:**  $\mathbb{R} \cup \mathcal{K}[G] \cup G$  no partial model with  $f(c), f(d), g(c), g(d)$  defined

$$\begin{aligned} \mathcal{K}[G] \quad & |f(d) - f(c)| \leq \lambda_1 \cdot |d - c| \wedge |f(c) - f(c)| \leq \lambda_1 \cdot |c - c| \\ & |g(d) - g(c)| \leq \lambda_2 \cdot |d - c| \wedge |g(c) - g(c)| \leq \lambda_2 \cdot |c - c| \end{aligned}$$

$$G \quad |f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

## 1. Flatten $\mathcal{K}[G] \cup G$

$$f(d) = d_1 \quad |d_1 - c_1| \leq \lambda_1 \cdot |d - c|$$

$$f(c) = c_1 \quad |c_1 - c_1| \leq \lambda_1 \cdot |c - c|$$

$$g(d) = d_2 \quad |d_2 - c_2| \leq \lambda_2 \cdot |d - c|$$

$$g(c) = c_2 \quad |c_2 - c_2| \leq \lambda_2 \cdot |c - c|$$

$$|d_1 + d_2 - c_1 - c_2| \not\leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

# Example

---

**Show:**  $\mathbb{R} \cup \mathcal{K}[G] \cup G$  no partial model with  $f(c), f(d), g(c), g(d)$  defined

$$\begin{aligned} \mathcal{K}[G] \quad & |f(d) - f(c)| \leq \lambda_1 \cdot |d - c| \quad \wedge \quad |f(c) - f(c)| \leq \lambda_1 \cdot |c - c| \\ & |g(d) - g(c)| \leq \lambda_2 \cdot |d - c| \quad \wedge \quad |g(c) - g(c)| \leq \lambda_2 \cdot |c - c| \end{aligned}$$

$$G \quad |f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

2. Replace functions with functional relations in  $\mathcal{K}[G] \cup G$

$$r_f(d, d_1) \quad |d_1 - c_1| \leq \lambda_1 \cdot |d - c| \quad c = d \wedge r_f(c, c_1) \wedge r_f(d, d_1) \rightarrow c_1 = d_1$$

$$r_f(c, c_1) \quad |c_1 - c_1| \leq \lambda_1 \cdot |c - c|$$

$$r_g(d, d_2) \quad |d_2 - c_2| \leq \lambda_2 \cdot |d - c| \quad c = d \wedge r_g(c, c_2) \wedge r_g(d, d_2) \rightarrow c_2 = d_2$$

$$r_g(c, c_2) \quad |c_2 - c_2| \leq \lambda_2 \cdot |c - c|$$

$$|d_1 + d_2 - c_1 - c_2| \not\leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

# Example

---

**Show:**  $\mathbb{R} \cup \mathcal{K}[G] \cup G$  no partial model with  $f(c), f(d), g(c), g(d)$  defined

$$\mathcal{K}[G] \quad |f(d) - f(c)| \leq \lambda_1 \cdot |d - c| \wedge |f(c) - f(c)| \leq \lambda_1 \cdot |c - c| \\ |g(d) - g(c)| \leq \lambda_2 \cdot |d - c| \wedge |g(c) - g(c)| \leq \lambda_2 \cdot |c - c|$$

$$G \quad |f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

(3) resolve away the functional relations

$$r_f(d, d_1) \quad |d_1 - c_1| \leq \lambda_1 \cdot |d - c| \quad c = d \rightarrow c_1 = d_1$$

$$r_f(c, c_1) \quad |c_1 - c_1| \leq \lambda_1 \cdot |c - c|$$

$$r_g(d, d_2) \quad |d_2 - c_2| \leq \lambda_2 \cdot |d - c| \quad c = d \rightarrow c_2 = d_2$$

$$r_g(c, c_2) \quad |c_2 - c_2| \leq \lambda_2 \cdot |c - c|$$

$$|d_1 + d_2 - c_1 - c_2| \not\leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

# Example

---

**Show:**  $\mathbb{R} \cup \mathcal{K}[G] \cup G$  no partial model with  $f(c), f(d), g(c), g(d)$  defined

$$\mathcal{K}[G] \quad |f(d) - f(c)| \leq \lambda_1 \cdot |d - c| \wedge |f(c) - f(c)| \leq \lambda_1 \cdot |c - c| \\ |g(d) - g(c)| \leq \lambda_2 \cdot |d - c| \wedge |g(c) - g(c)| \leq \lambda_2 \cdot |c - c|$$

$$G \quad |f(d) + g(d) - (f(c) + g(c))| \leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

(4) check the satisfiability of a set of constraints over  $\mathbb{R}$

$$r_f(d, d_1) \quad |d_1 - c_1| \leq \lambda_1 \cdot |d - c| \quad c = d \rightarrow c_1 = d_1$$

$$r_f(c, c_1) \quad |c_1 - c_1| \leq \lambda_1 \cdot |c - c|$$

$$r_g(d, d_2) \quad |d_2 - c_2| \leq \lambda_2 \cdot |d - c| \quad c = d \rightarrow c_2 = d_2$$

$$r_g(c, c_2) \quad |c_2 - c_2| \leq \lambda_2 \cdot |c - c|$$

$$|d_1 + d_2 - c_1 - c_2| \not\leq (\lambda_1 + \lambda_2) \cdot |d - c|$$

# Parameterized complexity

---

$\mathcal{T}_0 \cup \mathcal{K} \cup G$	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$	polynomial in $ \text{st}(G) $
	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup D^* \cup \text{Fun}(D^*)$	quadratic
	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup N_0$	constant

constraint solver for base theory to check satisfiability

$$N_0 = \{ \bigwedge_{i=1}^n c_i \approx d_i \rightarrow c = d \mid r^f(\bar{c}, c), r^f(\bar{d}, d) \in D^* \}.$$

**Theorem.** The following are equivalent

- (1)  $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$  has a partial model (all ground terms defined)
- (2)  $\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup N_0$  has a (total) model.

# Parameterized complexity

$$\begin{aligned}
 \mathcal{T}_0 \cup \mathcal{K} \cup G &\mapsto \mathcal{T}_0 \cup \mathcal{K}[G] \cup G && \text{polynomial in } |\text{st}(G)| \\
 &\mapsto \mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup D^* \cup \text{Fun}(D^*) && \text{quadratic} \\
 &\mapsto \mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup N_0 && \text{constant}
 \end{aligned}$$

constraint solver for base theory to check satisfiability

$$N_0 = \{ \bigwedge_{i=1}^n c_i \approx d_i \rightarrow c = d \mid r^f(\bar{c}, c), r^f(\bar{d}, d) \in D^* \}.$$

**Theorem.** The following are equivalent

- (1)  $\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$  has a partial model (all ground terms defined)
- (2)  $\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup N_0$  has a (total) model.

$\downarrow$        $\downarrow$   
ground      ground

ground if all variables in  $\mathcal{K}$  shielded by an extension function  
with free variables otherwise

# Parameterized complexity

---

$\mathcal{T}_0 \cup \mathcal{K} \cup G$	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}[G] \cup G$	polynomial in $ \text{st}(G) $
	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup D^* \cup \text{Fun}(D^*)$	quadratic
	$\mapsto$	$\mathcal{T}_0 \cup \mathcal{K}_0 \cup G_0 \cup N_0$	constant

constraint solver for base theory to check satisfiability

$$N_0 = \{ \bigwedge_{i=1}^n c_i \approx d_i \rightarrow c = d \mid r^f(\bar{c}, c), r^f(\bar{d}, d) \in D^* \}.$$

**Theorem.** Assumptions: (1)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  has (Loc<sup>f</sup>) or  
 (2)  $\mathcal{T}_0 \subseteq \mathcal{T}_1$  has (SLoc<sup>f</sup>),  $\mathcal{T}_0$  locally finite.

(a) If variables in  $\mathcal{K}$  shielded:  $Th_{\forall}(\mathcal{T}_0)$  decidable  $\rightarrow$   $Th_{\forall}(\mathcal{T}_1)$  decidable.

(b) Otherwise:  $Th_{\exists\forall}(\mathcal{T}_0)$  decidable  $\rightarrow$   $Th_{\forall}(\mathcal{T}_1)$  decidable.

# Conclusions

---

## Local theory extensions

- hierarchical reasoning
- parameterized complexity results

## Future work

- go beyond the universal fragment
- modular reasoning: combinations of local theory extensions
- use results to generate interpolants (applications in verification)