Lecture 6

Translation Schemes:

Main definitions and examples

- The framework of translation schemes
  - The induced maps
  - The fundamental lemma
  - Reductions
- The Museum of examples

Definition 1 (Translation Schemes Φ)

- Let τ and $\sigma = \{R_1, \ldots, R_m\}$ be two vocabularies with $\rho(R_i)$ be the arity of R_i.

- Let $\mathcal{L}$ be a fragment of SOL, such as FOL, $MSOL$, $\exists MSOL$, etc.

- Let $\Phi = \langle \phi, \psi_1, \ldots, \psi_m \rangle$ be formulae of $\mathcal{L}(\tau)$ such that ϕ has exactly k distinct free first order variables and each ψ_i has $k\rho(R_i)$ distinct free first order variables.
  We say that Φ is **k–feasible (for σ over τ)**.

- A k-feasible $\Phi = \langle \phi, \psi_1, \ldots, \psi_m \rangle$ is called a **k–τ–σ–$\mathcal{L}$–translation scheme** or, in short, a **translation scheme**, if the parameters are clear in the context.

Distinctions

If $k = 1$ we speak of **scalar** or **non–vectorized** translation schemes.

If $k \geq 2$ we speak of **vectorized** translation schemes.

If ϕ is such that $\forall \bar{x} \phi(\bar{x})$ is a tautology (always true) the translation scheme is **not relativized** otherwise it is **relativized**.

A translation scheme is **simple** if it is neither relativized nor vectorized.

Example 2 (τ_{words_3} and τ_{graphs})

τ_{words_3} consists of $\{R_<, P_0, P_1, P_2\}$ for three letters $\{0, 1, 2\}$.

τ_{graphs} consists of $\{E\}$

Put $k = 1$,
$\phi_1(x) = (P_0(x) \vee P_1(x))$ and
$\psi_E(x, y) = (P_0(x) \wedge P_1(y))$

$$\Phi_1 = \langle \phi_1(x), \psi_E(x, y) \rangle$$

is a **scalar** and **relativized** translation scheme in FOL.

If instead we look at $\phi_2(x) = (x \approx x)$ then

$$\Phi_2 = \langle \phi_2(x), \psi_E(x, y) \rangle$$

is a **simple** translation scheme.

Example 3 (τ_{words_2} and τ_{grids})

τ_{words_2} consists of $\{R_<, P_0, P_1\}$

τ_{grids} consists of $\{E_{NS}, E_{EW}\}$

Put $k = 2$,
$\phi(x) = ((x \approx x) \wedge (y \approx y))$
$\psi_{E_{NS}}(x_1, x_2, y_1, y_2) = (R_<(x_1, x_2) \wedge y_1 \approx y_2)$
$\psi_{E_{ES}}(x_1, x_2, y_1, y_2) = (R_<(y_1, y_2) \wedge x_1 \approx x_2)$

$$\Phi_3 =$$
$$\langle \phi(x, y), \psi_{E_{NS}}(x_1, x_2, y_1, y_2), \psi_{E_{EW}}(x_1, x_2, y_1, y_2) \rangle$$

is a **vectorized** but **not** relativized translation scheme in FOL.

Definition 4 (The induced transduction $\Phi^\star$)

Given a translation scheme Φ

$$\Phi^\star : Str(\tau) \to Str(\sigma)$$

is a (partial) function from τ–structures to σ–structures defined by $\Phi^\star(\mathcal{A}) = \mathcal{A}_\Phi$ and

(i) the universe of $\mathcal{A}_\Phi$ is the set $A_\Phi = \{\bar{a} \in A^k : \mathcal{A} \models \phi(\bar{a})\}$;

(ii) the interpretation of R_i in $\mathcal{A}_\Phi$ is the set

$$\mathcal{A}_\Phi(R_i) = \{\bar{a} \in {A_\Phi}^{\rho(R_i)\cdot k} : \mathcal{A} \models \psi_i(\bar{a})\}.$$

$\mathcal{A}_\Phi$ is a σ–structure of cardinality at most $\mid A \mid^k$.

As Φ is k–feasible for σ over τ, $\Phi^\star(\mathcal{A})$ is defined iff $\mathcal{A} \models \exists \bar{x} \phi$.

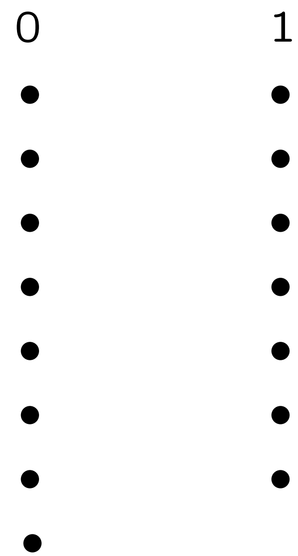
Example 5 (Words and graphs)

Let is compute $\Phi_1^\star$.

For the word

$$1001020102001022111$$

we get the graph



(1)

Example 6 (Words and grids)

Let is compute $\Phi_3^\star$.

For a word

$$0110101001$$

we get

$$\begin{array}{c}
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \\
\bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet \to \bullet
\end{array}$$

This is independent of the letters ¶$0, 1\}$.

Definition 7 (The induced translation $\Phi^\sharp$)

Given a translation scheme Φ we define a function $\Phi^\sharp : \mathcal{L}(\sigma) \rightarrow \mathcal{L}(\tau)$ from $\mathcal{L}(\sigma)$–formulae to $\mathcal{L}(\tau)$–formulae inductively as follows:

- For $R_i \in \sigma$ and $\theta = R_i(x_1, \ldots, x_m)$ let $x_{j,h}$ be new variables with $i \leq m$ and $h \leq k$ and denote by $\bar{x}_i = \langle x_{i,1}, \ldots, x_{i,k} \rangle$. We put

$$\Phi^\sharp(\theta) = \left(\psi_i(\bar{x}_1, \ldots, \bar{x}_m) \wedge \bigwedge_i \phi(\bar{x}_i) \right)$$

- This also works for equality and relation variables U instead of relation symbols R.

Definition 7 (Continued: booleans)

For the boolean connectives, the translation distributes, i.e.

- if $\theta = (\theta_1 \vee \theta_2)$ then
$$\Phi_\sharp(\theta) = (\Phi_\sharp(\theta_1) \vee \Phi_\sharp(\theta_2))$$
- if $\theta = \neg\theta_1$ then
$$\Phi_\sharp(\theta) = \Phi_\sharp(\neg\theta_1)$$
- similarly for $\wedge$ and $\rightarrow$.

Definition 7 (Continued: quantification)

- For the existential quantifier, we use relativization to ϕ:
  If $\theta = \exists y \theta_1$, let $\bar{y} = \langle y_1, \ldots, y_k \rangle$ be new variables. We put

  $$\theta_\Phi = \exists \bar{y}(\phi(\bar{y}) \wedge (\theta_1)_\Phi).$$

  This concludes the inductive definition for first order logic FOL.

- For second order quantification of variables U of arity ℓ and $\bar{a}$ a vector of length ℓ of first order variables or constants, we translate $U(\bar{a})$ by treating U as a relation symbol above and put

  $$\theta_\Phi = \exists V(\forall \bar{v}(V(\bar{v}) \rightarrow$$
  $$(\phi(\bar{v_1}) \wedge \ldots \phi(\bar{v}_\ell) \wedge (\theta_1)_\Phi)))$$

Example 8 (Computing $\Phi_1^\sharp$)

Recall

$$\Phi_1 = \langle \phi_1(x), \psi_E(x,y) \rangle$$

with $k = 1$,
$\phi_1(x) = (P_0(x) \vee P_1(x))$ and
$\psi_E(x,y) = (P_0(x) \wedge P_1(y))$

Let θ_{conn} be the formula which says the graph is connected:

$$\neg \left(\exists U \left(\exists x \neg U(x) \wedge \forall x \forall y (U(x) \wedge E(x,y) \rightarrow U(y)) \right) \right)$$

Example 8 (Continued)

- $U(x)$ is replaced by
$$(\phi_1(x) \wedge U(x)) = ((P_0(x) \vee P_1(x)) \wedge U(x))$$
- $E(x, y)$ is replaced by
$$(\phi_1(x) \wedge \phi_1(y) \wedge E(x, y)) =$$
$$((P_0(x) \vee P_1(x)) \wedge (P_0(y) \vee P_1(y)) \wedge E(x, y))$$
- $(x \approx y)$ is replaced by
$$(\phi_1(x) \wedge \phi_1(y) \wedge (x \approx y)) =$$
$$((P_0(x) \vee P_1(x)) \wedge (P_0(y) \vee P_1(y)) \wedge (x \approx y))$$
- Then we proceed inductively.

$(x \approx y)$ does not occur in θ_{conn}.

Proposition 9
(Preservation of tautologies I)

Let $\mathcal{L}$ be First Order Logic FOL.

$$\Phi = \langle \phi, \psi_1, \ldots, \psi_m \rangle$$

be a k–$(\tau - \sigma)$—$\mathcal{L}$–translation scheme, which is not relativizing, i.e. $\forall \bar{x} \phi(\bar{x})$ is a tautology. Let θ a σ-formula.

- If θ is a tautology (not satisfiable), so is $\Phi^{\sharp}(\theta)$.
- If ϕ is not a tautology, this is not true.
- There are formulas θ which are not tautologies (are satsifiable), such that $\Phi^{\sharp}(\theta)$ is a tautology (is not satisfiable).

Proof of proposition 9

Proof:
For FOL, the first two parts are by straight induction using the completeness theorem. What we observe is that proof sequences translate properly using $\Phi^\sharp$.

Generalizing to other logics needs regularity conditions.

If ϕ is not a tautology, $\exists x(x = x)$ is a tautology, but $\Phi^\sharp(\exists x(x = x)) = \exists x \phi(x) \wedge x = x$ is not a tautology.

Now let $\Phi = \langle \psi_R, \psi_S \rangle$ be defined by

$$\psi_R(x) = P(x) \text{ and } \psi_S(x) = \neg P(x).$$

$\exists x \theta_1$ be $R(x) \wedge S(x)$ and $\exists x \theta_2$ be $R(x) \vee S(x)$ are both satisfiable but not tautolgies. But
$\Phi^\sharp(\theta_1)$ is not satisfiable and
$\Phi^\sharp(\theta_2)$ is a tautology. $Q.E.D.$

Theorem 10 (Fundamental Property)

Let $\Phi = \langle \phi, \psi_1, \ldots, \psi_m \rangle$ be a k–$(\tau - \sigma)$–translation scheme in a logic $\mathcal{L}$. Then the transduction $\Phi^\star$ and the translation $\Phi^\sharp$ are in linked in $\mathcal{L}$.

In other words, given

- $\mathcal{A}$ be a τ-structure and
- θ be a $\mathcal{L}(\sigma)$–formula.

Then

$$\boxed{\mathcal{A} \models \Phi^\sharp(\theta) \text{ iff } \Phi^\star(\mathcal{A}) \models \theta}$$

Translation Scheme and its induced maps

in the Fundamental Property of theorem 10

Translation scheme Φ		
τ-structure	$\xrightarrow{\Phi^\star}$	σ-structure
$\mathcal{A}$		$\Phi^\star(\mathcal{A})$
τ-formulae	$\xleftarrow[\Phi^\sharp]{}$	σ-formulae
$\Phi^\sharp(\theta)$		θ
$\mathcal{A} \models \Phi^\sharp(\theta)$	iff	$\Phi^\star(\mathcal{A}) \models \theta$

Definition 11 ($\mathcal{L}$–Reductions)

Let $\mathcal{L}$ be a regular logic and Φ be a $(\tau_1 - \tau_2)$ translation scheme. We are given

- two classes K_1, K_2 of $\tau_1(\tau_2)$-structures closed under isomorphism

We say

(i) $\Phi^\star$ is a *weak reduction* of K_1 to K_2 if for every τ_1-structure $\mathfrak{A}$ with $\mathfrak{A} \in K_1$ we have $\Phi^\star(\mathfrak{A}) \in K_2$.

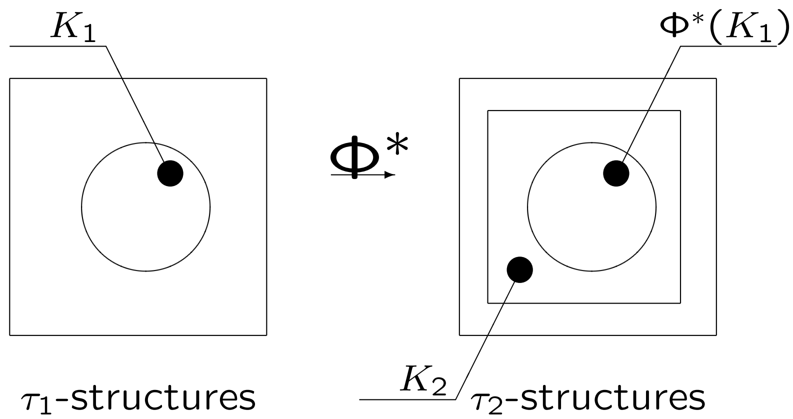
(ii) $\Phi^\star$ is a *reduction* of K_1 to K_2 if for every τ_1-structure $\mathfrak{A}$, $\mathfrak{A} \in K_1$ iff $\Phi^\star(\mathfrak{A}) \in K_2$.

Definition 11(Continued)

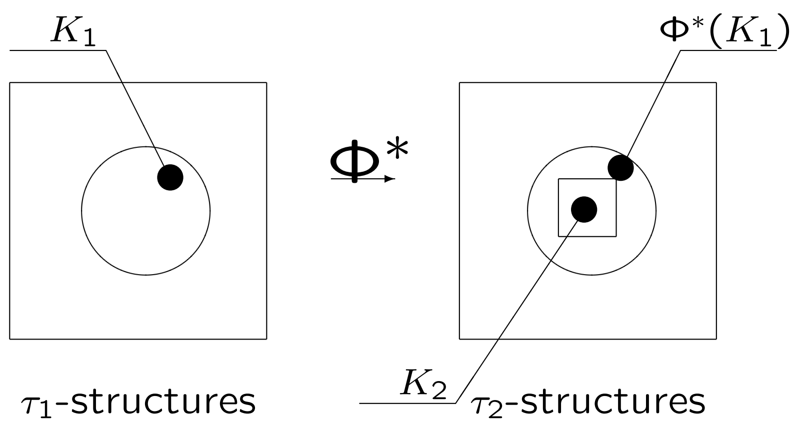
(iii) $\Phi^\star$ of K_1 to K_2 is *onto* if (additionally) for every $\mathfrak{B} \in K_2$ there is an $\mathfrak{A} \in K_1$ with $\Phi^\star(\mathfrak{A})$ isomorphic to $\mathfrak{B}$.

(iv) By abuse of language we say $\Phi^\star$ is a *translation of* K_1 *onto* K_2 also if $\Phi^\star$ is not a weak reduction but only $K_2 \subseteq \Phi^\star(K_1)$.

(v) We say that Φ induces a reduction (a weak reduction) of K_1 to K_2, if $\Phi^\star$ is a reduction (a weak reduction) of K_1 to K_2. For simplicity, we also say Φ is a reduction (a weak reduction) instead of saying that Φ induces a reduction (a weak reduction).



Weak reduction



ONTO

Definition 12 ($\mathcal{L}$–Reducibility)

(i) Let $k \in \mathbf{N}$.
We say that K_1 is *$\mathcal{L}$-k-reducible* to K_2
$(K_1 \lhd_{\mathcal{L}-k} K_2)$, if there is a $\mathcal{L}$-k-translation scheme Φ for τ_2 over τ_1, such that $\Phi^\star$ is a reduction of K_1 to K_2.

(ii) We say that K_1 is *$\mathcal{L}$-reducible* to K_2
$(K_1 \lhd_{\mathcal{L}} K_2)$, if $K_1 \lhd_{\mathcal{L}-k} K_2$ for some $k \in \mathbf{N}$.

(iii) We say that K_1 is *$\mathcal{L}$-bi-reducible* to K_2 and write $K_1 \bowtie_{\mathcal{L}} K_2$, if $K_1 \lhd_{\mathcal{L}-k} K_2$ and
$K_2 \lhd_{\mathcal{L}-k} K_1$ for some $k \in \mathbf{N}$.
Clearly, bi-reducibility is a symmetric relation.

Theorem 13 (Definability and Reducibility)

Let $\Phi^{\star}$ be an $\mathcal{L}$-reduction of K_1 to K_2.
If K_2 is $\mathcal{L}$-definable then K_1 is -definable.

Recall that a class of τ-structures K_2 is $\mathcal{L}$-definable if there is a $\mathcal{L}(\tau)$-sentence θ such that $K_2 = Mod(\theta)$.

Proof:

We use the Fundamental Property of Φ.

If K_2 is defined by θ, so K_1 is defined by $\Phi^{\sharp}(\theta)$.

Proposition 14

Hamiltonian graphs are not $MSOL$-definable (both in τ_{graphs_1} and τ_{graphs_2}).

Proof:

We use Φ_2 from example 2.

$\Phi_2^\star$ is a reduction from words 0^n1^m over $\{0,1\}$ to complete bipartite graphs $K_{n,m}$, which are $MSOL$-defined by θ_{co-bi}.

$K_{n,m}$ is Hamiltonian iff $n = m$.

So, if θ_{hamil} defined all Hamiltonian graphs,

$$\Phi_2^\sharp(\theta_{hamil} \wedge \theta_{co-bi})$$

defined the language $\{0^n1^n\}$.

But $\{0^n1^n\}$ is not regular, and hence, by Büchi's theorem, not $MSOL$-definable.

Q.E.D.

Proposition 15

Eulerian graphs are not $MSOL$-definable (both in τ_{graphs_1} and τ_{graphs_2}).

Proof: Let SET be the class of finite sets and $ODD \subseteq SET$ those of odd cardinality.
Let $CLIQUE$ be the class of complete graphs.
$CLIQUE$ is FOL-definable by some θ_{clique}.

Let the simple FOL translation scheme Φ be given by $\phi(x) = (x \approx x)$ and $\psi_E(x, y) = (\neg x \approx y)$.

$\Phi^\star$ is a reduction from SET to $CLIQUE$.

Now assume that there is $\theta_{euler} \in MSOL$,
with $EULER = Mod(\theta_{euler})$.
Put $\theta = (\theta_{clique} \wedge \theta_{euler})$.
$\Phi^\sharp(\theta)$ is equivalent to $\theta_{odd} \in MSOL$.

But this contradicts the fact that ODD $(EVEB)$ is not $MSOL$-definable.

Q.E.D.

Proof of theorem 10

We use induction over the construction of θ.

- If all the formulas ϕ, ψ_i of Φ and θ are atomic,
  both $\Phi^\star(\mathfrak{A}) = \mathfrak{A}$ and
  $\Phi^\sharp(\theta) = \theta$.

- Next we keep θ atomic and assume

  $$\Phi = \langle \phi(\bar{x}), \psi_{S_1}(\bar{x}), \ldots \psi_{S_m}(\bar{x}) \rangle$$

  $\Phi^\star(\mathfrak{A}) \models S_i(\bar{a})$ iff $\mathfrak{A} \models \psi_{S_i}(\bar{a})$
  by definition of $\Phi^\star$.

- Now the induction on θ uses that $\Phi^\sharp$ commutes with the logical constructs.

Q.E.D.

Proposition 16
(Preservation of tautologies II)

Let $\mathcal{L}$ be First Order Logic FOL.

$$\Phi = \langle \phi, \psi_1, \ldots, \psi_m \rangle$$

be a k–$(\tau - \sigma)$—$\mathcal{L}$–translation scheme. Let θ a σ-formula.

Assume that $\Phi^\star$ is onto all σ-structures, i.e.
for every σ-structure $\mathfrak{B}$ there is a τ-structure $\mathfrak{A}$ such that $\Phi^\star(\mathfrak{A}) = cong\mathfrak{B}$

- If θ is a tautology, so is $\Phi^\sharp(\theta)$.
- If additionally $\exists \bar{x} \phi(\bar{x})$ is a tautology and $\Phi\sharp(\theta)$ is a tautology then θ is a tautology.

Proof:
Use the fundamental property. $Q.E.D.$
Note that here the proof is semantical.

Example 17 (Renaming)

One of the simplest translations encountered in logic is the renaming of basic relations.

Let $\tau_1 = \{R_i : i \leq k\}$ and $\tau_2 = \{S_i : i \leq k\}$,
where R_i and S_i are of the same arity,
respectively.

Let Φ be the (τ_1, τ_2) translation scheme given by $\Phi = \langle x = x, R_1(\bar{u}), \ldots, R_k(\bar{v}) \rangle$.

Such a translation scheme and as well as its induced maps $\Phi^\star$ and $\Phi^\sharp$ are called **renaming**.

Example 18 (Cartesian Product)

Let us consider one example of vectorized translation scheme that defines Cartesian Product.

For simplicity, we assume that $k = 2$.

Let $\tau_1 = \{R_1(x_1, x_2)\}$ with R_1 binary
and $\tau_2 = \{R_2(x_1, x_2)\}$ with R_2 binary.

$$\Phi = \langle (x_1 = x_1 \vee x_2 = x_2), \\ (R_1(x_1, x_2) \wedge R_2(x_3, x_4))\rangle$$

It is easy to see that $\Phi^\star(\mathcal{A})$ is isomorphic to the Cartesian product $\mathcal{A}^2$.

The n-hold Cartesian product is defined in the same way.

Example 19 (Graphs)

$Graphs_1$ is the class of structures of the form $\langle V, E \rangle$ where E is a binary irreflexiv relation on the set of vertices V.

$Graphs_2$ is the class of structures of the form $\langle V \sqcup E; Src(v,e), Tgt(v,e) \rangle$ with the universe
consisting of **disjoint** sets of vertices and edges and $Src(v,e)$ ($Tgt(v,e)$) indicates that v is the source (target) of the directed edge e.

For a graph G we denote its representations by G_i for $G_i \in Graphs_i$ respectively.

We define a scalar translation scheme
$\Phi = \langle \phi, \psi_E \rangle$ from $Graphs_2$ to $Graphs_1$ by

$$\phi(v) = (\exists e(Src(v,e) \lor eTgt(v,e)) \lor$$
$$(v = v \land \neg \exists x(Src(x,v) \lor Tgt(x,v))$$
$$\phi_E(x,y) = \exists e\,((Src(x,e) \land Tgt(y,e))$$

Clearly, for every graph G we have

$$\Phi^{\star}(G_2) \cong G_1$$

Theorem 20 (Complexity of transductions)

If Φ is in FOL (or $\exists HornSOL$)
then $\Phi^\star$ is computable in polynomial time.

Proof:

We test all k-tuples $\bar{a}$ in $\mathfrak{A}$ of size n for

$$\mathfrak{A} \models \phi(\bar{a})$$

This takes $n^k \cdot TIME(\mathfrak{A}, \phi)$ time.

But we know that $TIME(\mathfrak{A}, \phi)$ is a polynomial in n.

For the ψ_{S_i} this is the same.

Q.E.D.

By a theorem of Grädel, this also holds for $HornSOL$, cf. the project page.

The Feferman-Vaught Theorem

and its algorithmic uses

Lecture originally prepared for the Tarski Centennial Conference

Warsaw, Poland, May, 2001

Vocabularies, structures and theories

Let τ vocabulary
(or a *similarity type* as Tarski used to call it)
given by a set of relation symbols, but no function symbols nor constants.

FOL(τ) denotes the set of τ-formulas in First Order Logic.

SOL(τ) and MSOL(τ) denote the set of τ-formulas in
Second Order and Monadic Second Order Logic.

For a class of τ-structures K
$Th_{\mathsf{FOL}}(K)$ is the set of
sentences of FOL(τ) true in all $\mathfrak{A} \in K$.
We write $Th_{\mathsf{FOL}}(\mathfrak{A})$ for $K = \{\mathfrak{A}\}$.

Similarly, $Th_{\mathsf{SOL}}(K)$ and $Th_{\mathsf{MSOL}}(K)$
for SOL and MSOL.

A. Tarski and E.W. Beth

Tarski published four short abstracts on model theory in 1949
(in the Bulletin of the AMS, vol. 55) and sent his manuscript for

Contribution to the theory of models, I

to E.W. Beth for publication.

Alfred Tarski (1901-1983)

Evert Willem Beth (1908-1964)

Inspired by these, E.W. Beth published two papers on model theory. In one of them he showed that

Beth's Theorem

Let $\mathfrak{A} = \langle A <_A \rangle$ and Let $\mathfrak{B} = \langle B <_B \rangle$ be two linear orders.

We denote by $\mathfrak{C} = \mathfrak{A} \sqcup_< \mathfrak{B}$ their ordered disjoint union,
defined by $C = A \sqcup B$, the disjoint union of A and B, and

$$<_C = <_A \sqcup B \cup A \times B$$

Theorem:(Beth 1952)

$Th_{\mathsf{FOL}}(\mathfrak{C})$ is uniquely determined by
$Th_{\mathsf{FOL}}(\mathfrak{A})$ and $Th_{\mathsf{FOL}}(\mathfrak{B})$.

In other words:
If $Th_{\mathsf{FOL}}(\mathfrak{A}) = Th_{\mathsf{FOL}}(\mathfrak{A}')$ and $Th_{\mathsf{FOL}}(\mathfrak{B}) = Th_{\mathsf{FOL}}(\mathfrak{B}')$ and $\mathfrak{C}' = \mathfrak{A}' \sqcup_< \mathfrak{B}'$ then
$Th_{\mathsf{FOL}}(\mathfrak{C}) = Th_{\mathsf{FOL}}(\mathfrak{C}')$

In Tarski's school it was asked in the early 1950ties

Let $\mathfrak{A}, \mathfrak{B}$ be τ-structures,
$\mathfrak{A} \times \mathfrak{B}$ their Cartesian product and
$\mathfrak{A} \sqcup \mathfrak{B}$ their disjoint union.

Assume we are given $Th_{\mathsf{FOL}}(\mathfrak{A})$ and $Th_{\mathsf{FOL}}(\mathfrak{B})$.

What can we say about $Th_{\mathsf{FOL}}(\mathfrak{A} \times \mathfrak{B})$ and $Th_{\mathsf{FOL}}(\mathfrak{A} \sqcup \mathfrak{B})$?

What happens in the case of infinite sums and products?

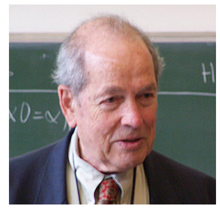
This triggered many landmark papers.

It also lead to the study of ultraproducts.

Tarski's pupils dealing with this question


A. Mostowski


A. Morel


S. Feferman


R. Vaught


J. Keisler

1938 Andrzei Mostowski
A. Mostowski, On direct product theories, JSL 17 (1952), pp. 1-31

1952 **Anne Morel**
T.E. Frayne, A.C. Morel and D.S. Scott, Reduced direct products, Fundamenta Mathematicae 51 (1962), pp.195-228

1954 Robert Vaught

1957 Solomon Feferman
S. Feferman and R.L. Vaught, The first order properties of algebraic systems, Fundamenta Mathematicae 47 (1959), pp. 57-103

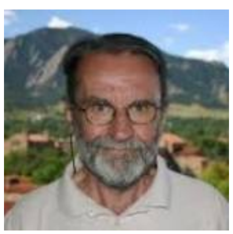
1961 Jerome Keisler
Many papers exploiting ultraproducts

Feferman and Vaught answered

Theorem A:(Feferman and Vaught, 1959)

$Th_{\mathsf{FOL}}(\mathfrak{A} \times \mathfrak{B})$ and $Th_{\mathsf{FOL}}(\mathfrak{A} \sqcup \mathfrak{B})$ are uniquely determined by $Th_{\mathsf{FOL}}(\mathfrak{A})$ and $Th_{\mathsf{FOL}}(\mathfrak{B})$.

- By combining it with transductions and interpretations this remains true for a wide variety of generalized products.
- Also true for infinite generalized sums and products,
provided the index structures are sufficiently MSOL indistinguishable.
- For MSOL still true for disjoint unions $\mathfrak{A} \sqcup \mathfrak{B}$ over infinite index sets,
(Ehrenfeucht, Läuchli, Shelah, Gurevich),
but **not true for** SOL.


A. Ehrenfeucht


H. Läuchli


S. Shelah


Y. Gurevich

Ehrenfeucht's proof theorem A

Use Ehrenfeucht-Fraïssé Games

This gives actually more:

Let $q \in \mathbb{N}$
and let $\mathsf{FOL}^q(\tau)$ denote the sentences of FOL of quantifier rank at most q.

Put $Th^q_{\mathsf{FOL}}(\mathfrak{A}) = Th_{\mathsf{FOL}}(\mathfrak{A}) \cap \mathsf{FOL}^q(\tau)$.

Theorem B:(Feferman and Vaught, 1959)

$Th^q_{\mathsf{FOL}}(\mathfrak{A} \times \mathfrak{B})$ and $Th^q_{\mathsf{FOL}}(\mathfrak{A} \sqcup \mathfrak{B})$
are uniquely determined by $Th^q_{\mathsf{FOL}}(\mathfrak{A})$ and $Th^q_{\mathsf{FOL}}(\mathfrak{B})$.

For MSOL still true for $\mathfrak{A} \sqcup \mathfrak{B}$.

Feferman and Vaught's proof

Use Reduction Sequences

Here one proves by induction, say for disjoint union

Theorem C:(Feferman and Vaught, 1959)

For every formula $\phi \in \mathsf{FOL}^q(\tau)$ one can compute a sequence of formulas

$$\langle \psi_1^A, \ldots \psi_m^A, \psi_1^B, \ldots \psi_m^B \rangle \in \mathsf{FOL}^q(\tau)^{2m}$$

and a Boolean function
$B_\phi : \{0,1\}^{2m} \rightarrow \{0,1\}$ such that

$$\mathfrak{A} \sqcup \mathfrak{B} \models \phi$$

iff

$$B_\phi(b_1^A, \ldots b_m^A, b_1^B, \ldots b_m^B) = 1$$

where $b_j^A = 1$ iff $\mathfrak{A} \models \psi_j^A$ and $b_j^B = 1$ iff $\mathfrak{B} \models \psi_j^B$

Similarly for MSOL

Algorithmic problems on finite structures

From 1970 on interest in computing focused on computing on finite structures:

- $\mathbf{NP}$-complete problems and complexity hierarchies.
- Graph algorithms and other finite data structures.
- Generalizing finite automata to trees and beyond.

In all these fields Use Ehrenfeucht-Fraïssé Games and the Feferman-Vaught Theorem play crucial rôles.

Operations on coloured, (un)-directed graphs.

Instead for the disjoint union we can prove
theorem B and C for the following operations:

- Concatenation of words, $v \circ w$.
- Joining two trees
  at a new common root, $T_1 \bullet T_2$.
- H-sums of graphs: For $i = 1, 2$
  let $G_i = \langle V(G_i), E(G_i) \rangle$ and
  $V(G_1) \cap V(G_2) = V(H)$ and
  $E(H) = E(G_1) \cap V(H)^2 = E(G_2) \cap V(H)^2$.

  Then $G = G_1 \oplus_H G_2$ is given by
  $V(G) = V(G_1) \cup V(G_2)$ and
  $E(G) = E(G_1) \cup E(G_2)$.

  and similarly for edge and vertex coloured graphs

MSOL-smooth operations

We generalize the previous operations to operations satisfying theorem B or C:

Definition:
A n-ary operation $\mathcal{O}$ on τ-structures
is MSOL-smooth if
for every $q \in \mathbb{N}$ and every $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_n$

$$Th^q_{\mathrm{MSOL}}(\mathcal{O}(\mathfrak{A}_1, \ldots, \mathfrak{A}_n))$$

depends only on $Th^q_{\mathrm{MSOL}}(\mathfrak{A}_i)$ for $1 \leq i \leq n$.

$\mathcal{O}$ is effectively MSOL-smooth if there is an algorithm which computes for every $\phi \in \mathrm{MSOL}(\tau)$ a reduction sequence, i.e. a sequence of formulas as described in theorem C.

Examples of MSOL-smooth operations

- The disjoint union is effectively MSOL-smooth.
- Quantifier free MSOL-transductions are effectively MSOL-smooth.
- The fusion operation $\mathbf{fus}_P$ i.e. identifying all elements which satisfy a given unary predicate P is MSOL-smooth, but we don't know whether it is effectively so.

Open problem:
Are there more examples?
Are there MSOL-smooth operations which are not effectively MSOL-smooth ?

MSOL-inductive classes

(graph grammars)

Definition:

A class K of τ-structures is MSOL-**inductive** if it is defined inductively using a finite set of MSOL-smooth operations.

K is **effectively** MSOL-**inductive** if it is defined inductively using a finite set of effectively MSOL-smooth operations.

Open problem:
Are there MSOL-inductive classes K which are not effectively MSOL-inductive ?

Effectively MSOL-inductive classes of structures

- Words $\Sigma^\star$ are defined inductively by
  (i) the empty word is a word
  (ii) one letter words are words
  (iii) words are closed under concatenation

- Coloured trees (forests) are defined similarly:
  (i) one leave trees are trees
  (ii) trees are closed under root joining
  (iii) forests are closed under disjoint unions

- Series-parallel (SP) graphs are defined by
  (i) one edge graphs are SP.
  (ii) SP graphs are closed under disjoint unions
  (iii) SP graphs are closed under H-sums for all H with at most two vertices.

More effectively MSOL-inductive classes

- Graphs of tree width at most k TW_k can be defined inductively by looking at vertex coloured graphs with at most $k+1$ colours:
  (i) All graphs with at most $k+1$ vertices are in TW_k.
  (ii) TW_k is closed under disjoint union.
  (iii) TW_k is closed under renaming of colours.
  (iv) TW_k is closed under fusion, i.e. contraction of
  all vertices of a specific colour into one vertex.

- Similarly, for graphs of clique width at most k CW_k
  (i) All graphs with at most 1 vertex are in CW_k.
  (ii) CW_k is closed under disjoint union.
  (iii) CW_k is closed under renaming of colours.
  (iv) CW_k is closed under adding all possible edges
  between to sets of differently coloured edges.

Decidable theories, I

R. Büchi

M. Rabin

The following were shown to be decidable by R. Büchi and M. Rabin respectively:

- The MSOL theory of words
- The MSOL theory of trees

We can use theorem C to show that the following MSOL theories are decidable.

- The MSOL-theory SP-graphs
- The MSOL-theory graphs of bounded tree width

Decidable theories, II

We can generalize this to

Theorem D:(Courcelle and M., 2001)

Let K be MSOL-inductive using disjoint unions, fusions and quantifier free MSOL-transductions.
Then $Th_{\text{MSOL}}(K)$ is decidable.

Proof idea:

One shows that an MSOL-inductive class K is always an MSOL-**transduction** of a class of trees.

Then one applies Rabin's theorem for trees.

Seese 1991 showed it for K of bounded tree width.

Representation of structures in MSOL-inductive classes

We can represent the structure $\mathfrak{A}$
by its **relational table**
or by a **parse term** $t_{\mathfrak{A}}$
which displays why it is in K.

In general, finding $t_{\mathfrak{A}}$ is $\mathbf{NP}$-complete.

Theorem E:(Courcelle and M., 2001)
Let K be an MSOL-inductive class of τ-structures and $\phi \in$ MSOL(τ).

Given a parse term $t_{\mathfrak{A}}$ for $\mathfrak{A}$, then the problem of deciding

$$\mathfrak{A} \models \phi$$

can be decided in linear time
(in the size of $t_{\mathfrak{A}}$).

Model checking, I

Model checking is the problem to check

$$\mathfrak{A} \models \phi$$

for $\mathfrak{A}$ a finite τ structure and $\phi \in \mathrm{SOL}(\tau)$.

We measure the problem in the size of $\mathfrak{A}$ and ϕ (combined case) or for specific ϕ.

Theorem:(M. and Pnueli, 1996)
Even for MSOL there are ϕ such that the
problem is arbitrarily high in the polynomial hierarchy.

Theorem:(Vardi, 1982)
The combined problem is $\mathbf{PSpace}$-complete
even for FOL.

Model checking, II

We want to do model checking on MSOL-inductive classes K.

We can represent the structure $\mathfrak{A}$ by its **relational table**
or by a **parse term** $t_{\mathfrak{A}}$ which displays why it is in K.

In general, finding $t_{\mathfrak{A}}$ is $\mathbf{NP}$-complete.

Theorem E:(Courcelle and M., 2001)
Let K be an MSOL-inductive class of τ-structures and $\phi \in$ MSOL(τ).

Given a parse term $t_{\mathfrak{A}}$ for $\mathfrak{A}$, then the problem of deciding

$$\mathfrak{A} \models \phi$$

can be decided in linear time
(in the size of $t_{\mathfrak{A}}$).

Proof of theorem E

$\mathrm{MSOL}^q(\tau)$ is finite for every finite relational τ.

If all the operations are effectively MSOL-smooth, there is an algorithm which computes for every ϕ the look-up table (using theorem C).

Otherwise, we don't have such an algorithm, but still for each ϕ the look-up table for complete $\mathrm{MSOL}^q(\tau)$-types is finite (using theorem B).

Now we can compute along $t_{\mathfrak{A}}$, in the style of *dynamic programming*.

Moshe Vardi

Bruno Courcelle

Yachin Pnueli