Revisiting the Institutional Approach to Herbrand’s Theorem

Ionuț Țuțu$^{1,2}$ José Luiz Fiadeiro$^{1}$

$^1$Department of Computer Science, Royal Holloway University of London

$^2$Simion Stoilow Institute of Mathematics of the Romanian Academy

6th Conference on Algebra and Coalgebra in Computer Science

Nijmegen, 2015
Herbrand’s Fundamental Theorem

• central result in proof theory
• deals with the reduction of provability in first-order logic to provability in propositional logic

\[ \exists \{x_1, \ldots, x_n\} \cdot \rho(x_1, \ldots, x_n) \text{ is valid} \]
if and only if
\[ \text{there is a sequence of terms } t_1, \ldots, t_n \]
\[ \text{such that } \rho(t_1, \ldots, t_n) \text{ is valid} \]
Herbrand’s Fundamental Theorem

- central result in proof theory
- deals with the reduction of provability in first-order logic to provability in propositional logic

\[
\exists \{x_1, \ldots, x_n\} \cdot \rho(x_1, \ldots, x_n) \text{ is valid}
\]

if and only if

there is a sequence of terms \(t_1, \ldots, t_n\)

such that \(\rho(t_1, \ldots, t_n)\) is valid

1929
Herbrand’s Fundamental Theorem

- central result in proof theory
- deals with the reduction of provability in first-order logic to provability in propositional logic

- difficulties in following the proof and errors reported by Bernays and Gödel
Herbrand’s Fundamental Theorem

- central result in proof theory
- deals with the reduction of provability in first-order logic to provability in propositional logic
- gaps and counterexamples found by Dreben, Andrews, and Aanderaa
- the publication of the first emended (and detailed) proof of the result

Investigations in proof theory: The properties of true propositions

Jacques Herbrand
The resolution inference rule

• introduced by Robinson
• well-suited for automation

$$\exists X \cdot Q \land g \quad \forall Y \cdot c \leftarrow H$$

$$\exists X' \cdot \theta(Q) \land \theta(H)$$

• led to the development of logic programming – PROLOG (Kowalski & Colmerauer)
The resolution inference rule

- introduced by Robinson
- well-suited for automation

\[
\exists X \cdot Q \land g \quad \forall Y \cdot c \leftarrow H
\]

\[
\exists X' \cdot \theta(Q) \land \theta(H)
\]

- led to the development of logic programming – PROLOG (Kowalski & Colmerauer)

Herbrand 1929  
Robinson 1965 1973
Foundations of logic programming

Given a logic program $\Gamma$, the answers to an existential query can be found simply by examining a term model – the least Herbrand model – instead of all the models that satisfy $\Gamma$.

1. $\Gamma \models_\Sigma \exists X \cdot \rho$
2. $\emptyset_{\Sigma, \Gamma} \models_\Sigma \exists X \cdot \rho$
3. There exists $\psi: X \rightarrow Y$ such that $\Gamma \models_\Sigma \forall Y \cdot \psi(\rho)$.

Q: How many Prolog programmers does it take to change a lightbulb?
A: Yes

$\exists \{x\} \cdot \text{“}x\text{ is a number”}$

$\land \text{“}X\text{ Prolog programmers can change a lightbulb”}$

Q: How many Prolog programmers does it take to change a lightbulb?
A: 1
Foundations of logic programming

Given a logic program $\Gamma$, the answers to an existential query can be found simply by examining a term model – the least Herbrand model – instead of all the models that satisfy $\Gamma$.

1. $\Gamma \models_{\Sigma} \exists X \cdot \rho$
2. $O_{\Sigma,\Gamma} \models_{\Sigma} \exists X \cdot \rho$
3. There exists $\psi: X \rightarrow Y$ such that $\Gamma \models_{\Sigma} \forall Y \cdot \psi(\rho)$.

Q: How many Prolog programmers does it take to change a lightbulb?
A: 1

$\exists \{x\} \cdot "x \text{ is a number}"
\wedge "x \text{ Prolog programmers can change a lightbulb}"

Q: How many Prolog programmers does it take to change a lightbulb?
A: Yes
Foundations of logic programming

Given a logic program $\Gamma$, the answers to an existential query can be found simply by examining a term model – the least Herbrand model – instead of all the models that satisfy $\Gamma$.

1. $\Gamma \vdash \Sigma \exists X \cdot \rho$
2. $\sigma, \Gamma \vdash \Sigma \exists X \cdot \rho$
3. There exists $\psi : X \rightarrow Y$ such that $\Gamma \vdash \Sigma \forall Y \cdot \psi(\rho)$.

Q: How many Prolog programmers does it take to change a lightbulb?
A: Yes

$\exists \{x\} \cdot \text{“}x\text{ is a number”}$
$\land \text{“}x\text{ Prolog programmers can change a lightbulb”}$

Q: How many Prolog programmers does it take to change a lightbulb?
A: 1

Herbrand 1929 Robinson 1965 1984
Given a logic program $\Gamma$, the answers to an existential query can be found simply by examining a term model – the least Herbrand model – instead of all the models that satisfy $\Gamma$.

1. $\Gamma \models_\Sigma \exists X \cdot \rho$

2. $\mathcal{O}_{\Sigma,\Gamma} \models_\Sigma \exists X \cdot \rho$

3. There exists $\psi: X \rightarrow Y$ such that $\Gamma \models_\Sigma \forall Y \cdot \psi(\rho)$.

**Foundations of logic programming**

Q: How many Prolog programmers does it take to change a lightbulb?
A: Yes

$\exists\{x\} \cdot \text{“} x \text{ is a number”}$

$\wedge \text{“} x \text{ Prolog programmers can change a lightbulb”}$

Q: How many Prolog programmers does it take to change a lightbulb?
A: 1
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \exists \{x, y\} \cdot \text{sorted}(2, 3, x, y, 5) \]
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \exists \{x: \text{Num}\} . \text{sorted}(2, 3, x) = T \]
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \exists \{ s : \text{List} \rightarrow B \} \cdot s[2, 3, 5] = T \]
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \exists \{ s : \text{Stream} \} \cdot s \sim \text{tail}(s) \]
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \langle \text{Sig}, \text{Sen}, \text{Mod}, \models \rangle \]

subject to a satisfaction condition:

- for every \( \varphi : \Sigma \to \Omega \), \( M \in |\text{Mod}(\Omega)| \), \( \rho \in \text{Sen}(\Sigma) \)
  \( M \upharpoonright \varphi \models_{\Sigma} \rho \iff M \models_{\Omega} \varphi(\rho) \)
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \exists \{x, y\} \cdot \text{sorted}(2, 3, x, y, 5) \]
\[ \chi: \langle F, P \rangle \mapsto \langle F \cup \{x, y\}, P \rangle \]

\[ \exists \chi \cdot \text{sorted}(2, 3, x, y, 5) \]
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming
A multitude of variants

- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming

\[ \Sigma \vdash \sum_{\Sigma}(X) \iff \text{Mod}_{\Sigma}(X) \]

\[ \exists X \cdot \rho \]
Institutions as functors

- Each institution $I = \langle \text{Sig}, \text{Sen}, \text{Mod}, \models \rangle$ can be identified with a functor $I : \text{Sig} \to \text{Room}$ where $I(\Sigma) = \langle \text{Sen}(\Sigma), \text{Mod}(\Sigma), \models_\Sigma \rangle$.

- Similarly, substitution systems can be defined as functors $S : \text{Subst} \to G / \text{Room}$.

Rooms and corridors

\[ \langle S, M, \models \rangle \xrightarrow{\alpha} \langle S', M', \models' \rangle \xrightarrow{\beta} \]

Timeline:

- Herbrand 1929
- Robinson 1965
- Lloyd 1984
- 2004
- 2014
From institutions to substitution systems

- let $Q$ be a class of signature morphisms of an institution $I : \text{Sig} \rightarrow \text{Room}$

For every $I$-signature $\Sigma$ we obtain a substitution system $\mathcal{SI}_Q^I : \text{Subst}_Q^I \rightarrow I(\Sigma) / \text{Room}$:

- the objects of $\text{Subst}_Q^I$ are signature morphisms $\chi : \Sigma \rightarrow \Sigma(\chi)$ belonging to $Q$
- a $\Sigma$-substitution $\psi : \chi_1 \rightarrow \chi_2$ is a corridor $\langle \text{Sen}_\Sigma(\psi), \text{Mod}_\Sigma(\psi) \rangle : I(\Sigma(\chi_1)) \rightarrow I(\Sigma(\chi_2))$
Quantification spaces

- for every subcategory $Q \subseteq \text{Sig}^\sim$, the functor $\text{dom}: Q \to \text{Sig}$ gives rise to a natural transformation $\iota_Q: (\_ / Q) \Rightarrow \text{dom}^{\text{op}} \circ (\_ / C)$

**Definition.** $Q$ is said to be a quantification space for an institution $J: \text{Sig} \to \text{Room}$ if

1. every arrow in $Q$ forms a pushout in $\text{Sig}$, and
2. $\iota_Q$ is a natural isomorphism.
Quantification spaces

- for every subcategory \(Q \subseteq \text{Sig}^-\), the functor \(\text{dom}: Q \to \text{Sig}\) gives rise to a natural transformation \(\iota_Q: (_/Q) \Rightarrow \text{dom}^{\text{op}}; (_/C)\)

**Definition.** \(Q\) is said to be a quantification space for an institution \(J: \text{Sig} \to \text{Room}\) if

1. every arrow in \(Q\) forms a pushout in \(\text{Sig}\), and
2. \(\iota_Q\) is a natural isomorphism.
Representable signature extensions

**Definition.** An extension $\chi: \Sigma \to \Sigma(\chi)$ is *representable* if there exist

- a $\Sigma$-model $M_\chi$ and
- an isomorphism of categories $i_\chi$

such that the following diagram commutes:

\[
\begin{array}{ccc}
\text{Mod}(\Sigma) & \xrightarrow{- \lvert_\chi} & \text{Mod}(\Sigma(\chi)) \\
\uparrow & & \downarrow \text{forgetful} \\
\text{Mod}(\Sigma(\chi)) & \xrightarrow{i_\chi} & M_\chi / \text{Mod}(\Sigma)
\end{array}
\]
Representable signature extensions

**Proposition.** The representation of signature extensions generalizes to a functor $R_{\Sigma}^Q : \text{Subst}_\Sigma^Q \to \text{Mod}(\Sigma)$, where

- for every $\chi : \Sigma \to \Sigma(\chi)$ in $|Q|$, $R_{\Sigma}^Q(\chi) = M_\chi$,
- for every substitution $\psi : \chi_1 \to \chi_2$, $R_{\Sigma}^Q(\psi) = (i_{\chi_2}^{-1} \circ \text{Mod}_\Sigma(\psi) \circ i_{\chi_1})(1_{M_\chi_2})$.

Moreover, for every $\Sigma$-substitution $\psi$, $\text{Mod}_\Sigma(\psi)$ is uniquely determined by $R_{\Sigma}^Q(\psi)$. 

Herbrand

1929

Robinson

1965

Lloyd

1984

2004

2014
**Proposition.** Every morphism of signatures \( \varphi : \Sigma \to \Sigma' \) gives rise to a functor \( \Psi_\varphi : \text{Subst}_\Sigma \to \text{Subst}_{\Sigma'} \) defined as follows:
Deriving generalized substitution systems

**Theorem.** Every institution $I: \text{Sig} \to \text{Room}$ equipped with

- an adequate quantification space $\mathcal{Q}$ of representable signature extensions and
- compatible categories $\text{Subst}_\Sigma$ of $\mathcal{Q}$-representable $\Sigma$-substitutions,

determines a generalized substitution system that has model amalgamation.
Herbrand’s theorem revisited

Let \( \langle \Sigma, \Gamma \rangle \) be a LP and \( \exists \chi \cdot \rho \) a query such that

- \( \Sigma \) and \( \langle \Sigma, \Gamma \rangle \) have initial models \( o_{\Sigma} \) and \( o_{\Sigma,\Gamma} \),
- \( M_\chi \) is projective with respect to the unique homomorphism \( \Gamma !: o_{\Sigma} \to o_{\Sigma,\Gamma} \), and
- the sentence \( \rho \) is basic.

The following statements are equivalent:

1. \( \Gamma \models_{\Sigma} \exists \chi \cdot \rho. \)
2. \( o_{\Sigma,\Gamma} \models_{\Sigma} \exists \chi \cdot \rho. \)
3. \( \exists \chi \cdot \rho \) admits a \( \Gamma \)-solution.
Thank you!
Further Reading


