Revisiting the Institutional Approach to Herbrand's Theorem

Ionuţ Ţuţu[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

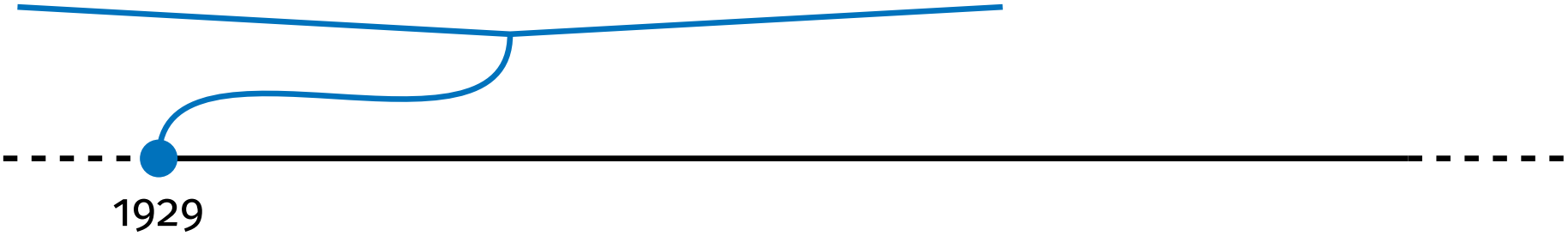
Investigations in proof theory:
The properties of true propositions
JACQUES HERBRAND

$\exists\{x_1, \ldots, x_n\} \cdot \rho(x_1, \ldots, x_n)$ *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*



Herbrand's Fundamental Theorem

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

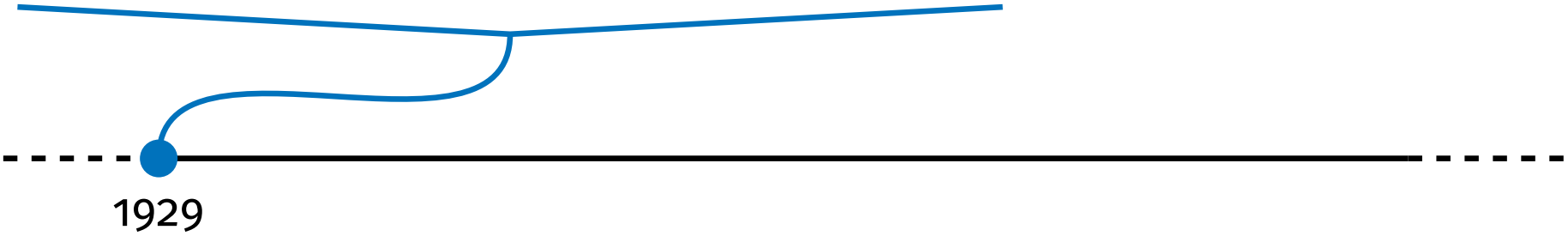
Investigations in proof theory:
The properties of true propositions
JACQUES HERBRAND

$\exists\{x_1, \ldots, x_n\} \cdot \rho(x_1, \ldots, x_n)$ *is valid*

if and only if

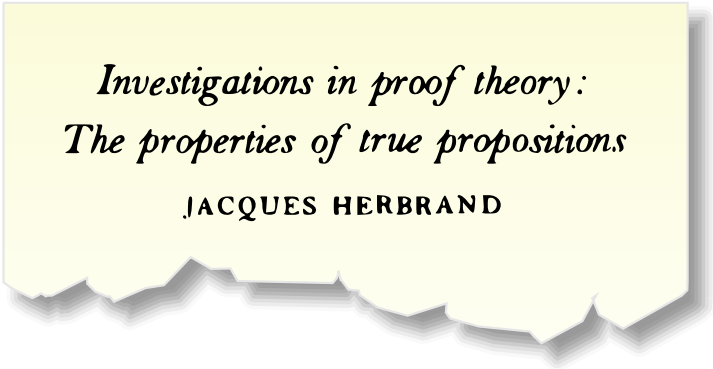
there is a sequence of terms $t_1, \ldots, t_n$

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

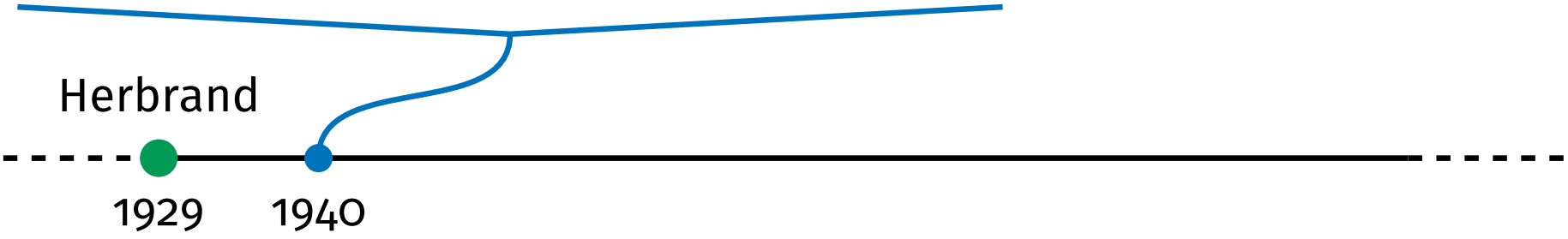


Herbrand's Fundamental Theorem

- 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

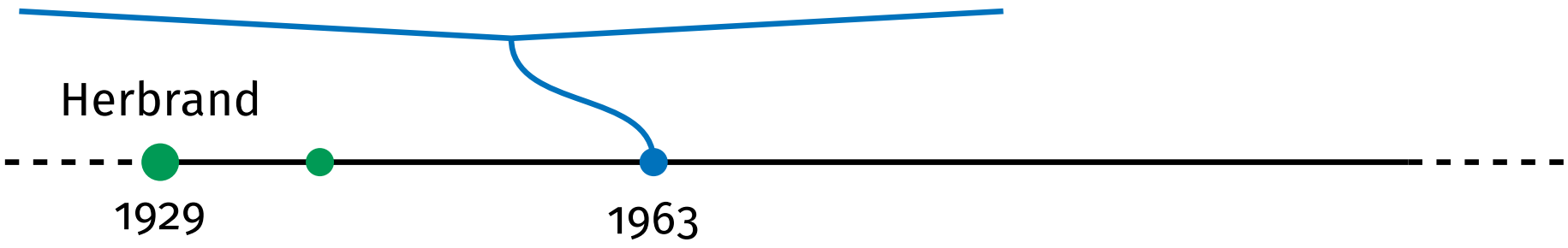


Herbrand's Fundamental Theorem

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

Investigations in proof theory:
The properties of true propositions
JACQUES HERBRAND

- gaps and counterexamples found by Dreben, Andrews, and Aanderaa
- the publication of the first emended (and detailed) proof of the result

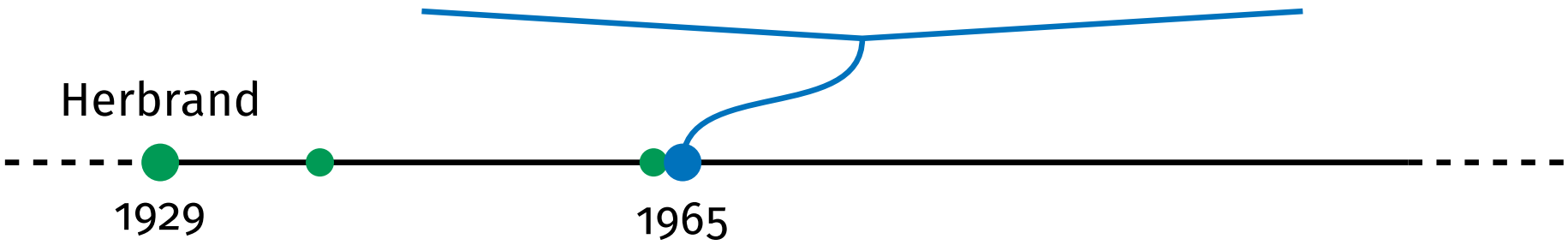


The resolution inference rule

- introduced by Robinson
- well-suited for automation

$$\frac{\exists X \cdot Q \wedge g \qquad \forall Y \cdot c \leftarrow H}{\exists X' \cdot \theta(Q) \wedge \theta(H)}\ \theta$$

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

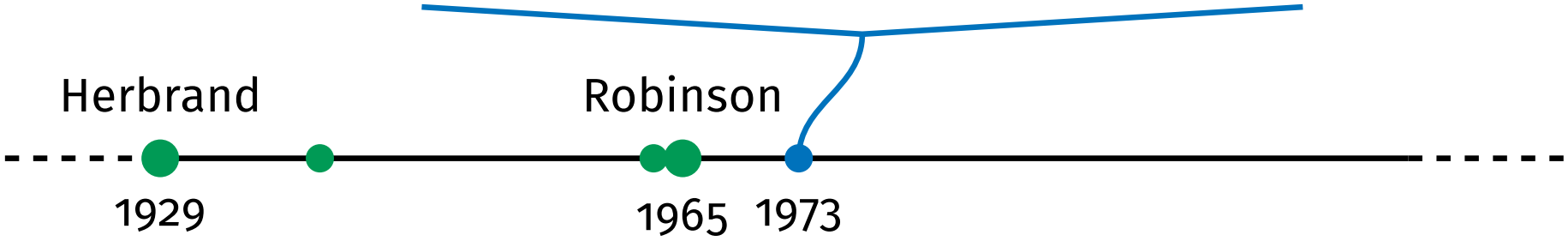


The resolution inference rule

- introduced by Robinson
- well-suited for automation

$$\frac{\exists X \cdot Q \wedge g \qquad \forall Y \cdot c \leftarrow H}{\exists X' \cdot \theta(Q) \wedge \theta(H)}\ \theta$$

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



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

$\exists\{x\} \cdot$ "x is a number"

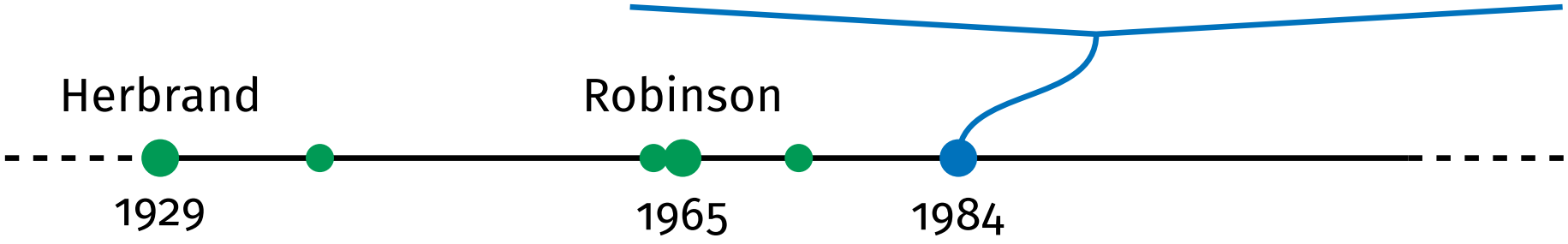
$\bigwedge$ "x Prolog programmers can change a lightbulb"

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

Foundations of logic programming

Given a logic program Γ, 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 Γ.

1. $\Gamma \vDash_\Sigma \exists X \cdot \rho$
2. $\mathrm{o}_{\Sigma,\Gamma} \vDash_\Sigma \exists X \cdot \rho$
3. There exists $\psi \colon X \to Y$ such that $\Gamma \vDash_\Sigma \forall Y \cdot \psi(\rho)$.



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

$\exists\{x\} \cdot$ "x is a number"

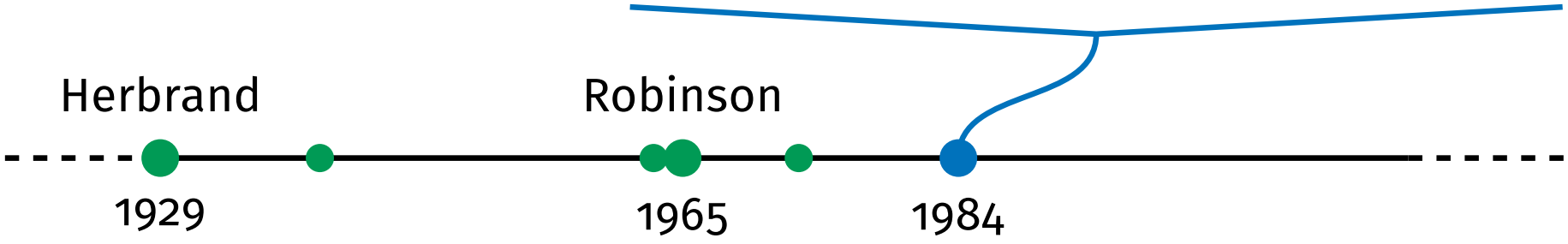
$\wedge$ "x Prolog programmers can change a lightbulb"

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

Foundations of logic programming

Given a logic program Γ, 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 Γ.

1. $\Gamma \vDash_\Sigma \exists X \cdot \rho$
2. $\mathrm{o}_{\Sigma,\Gamma} \vDash_\Sigma \exists X \cdot \rho$
3. There exists $\psi \colon X \to Y$ such that $\Gamma \vDash_\Sigma \forall Y \cdot \psi(\rho)$.



Foundations of logic programming

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

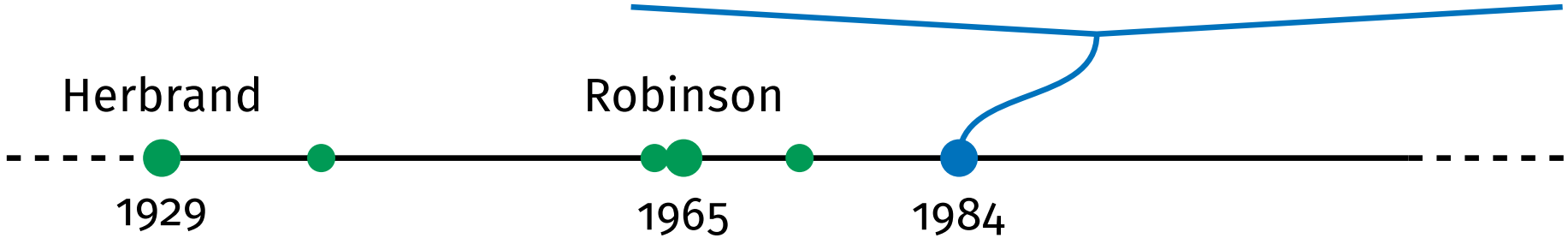
$\exists\{x\} \cdot$ "x is a number"

$\wedge$ "x Prolog programmers can change a lightbulb"

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

Given a logic program Γ, 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 Γ.

1. $\Gamma \vDash_\Sigma \exists X \cdot \rho$
2. $o_{\Sigma,\Gamma} \vDash_\Sigma \exists X \cdot \rho$
3. There exists $\psi \colon X \to Y$ such that $\Gamma \vDash_\Sigma \forall Y \cdot \psi(\rho)$.



Foundations of logic programming

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

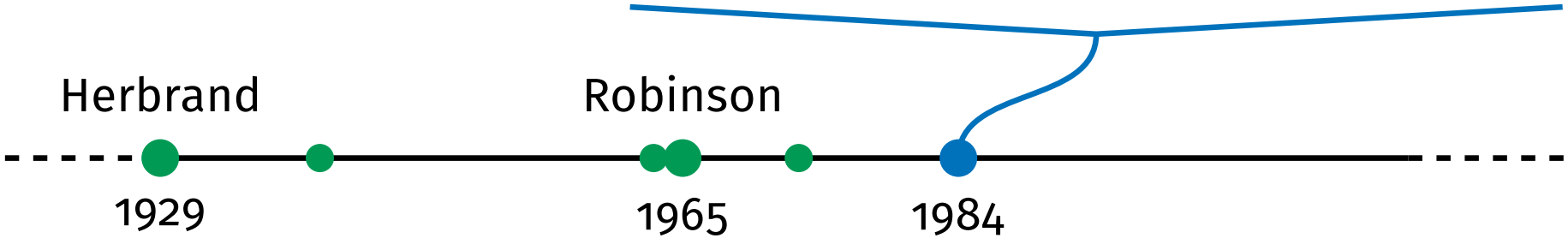
$\exists\{x\} \cdot$ "x is a number"

$\wedge$ "x Prolog programmers can change a lightbulb"

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

Given a logic program Γ, 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 Γ.

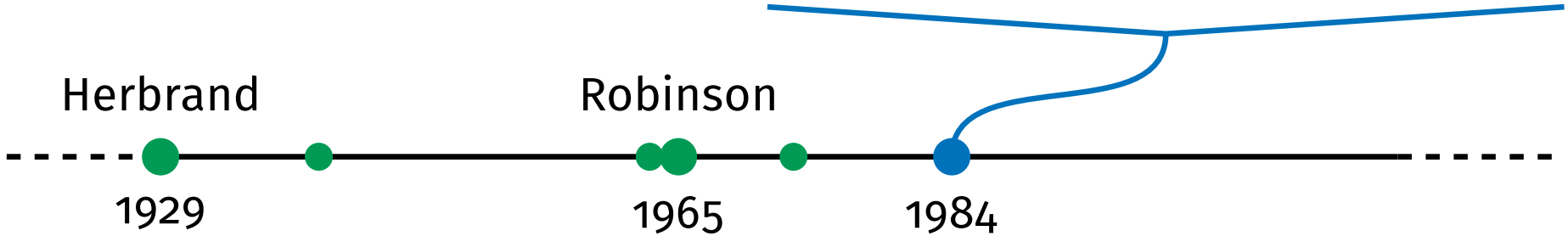
1. $\Gamma \vDash_\Sigma \exists X \cdot \rho$
2. $\mathrm{O}_{\Sigma,\Gamma} \vDash_\Sigma \exists X \cdot \rho$
3. There exists $\psi \colon X \to Y$ such that $\Gamma \vDash_\Sigma \forall Y \cdot \psi(\rho)$.



A multitude of variants

$\exists\{x, y\} \cdot \texttt{sorted}(2, 3, x, y, 5)$

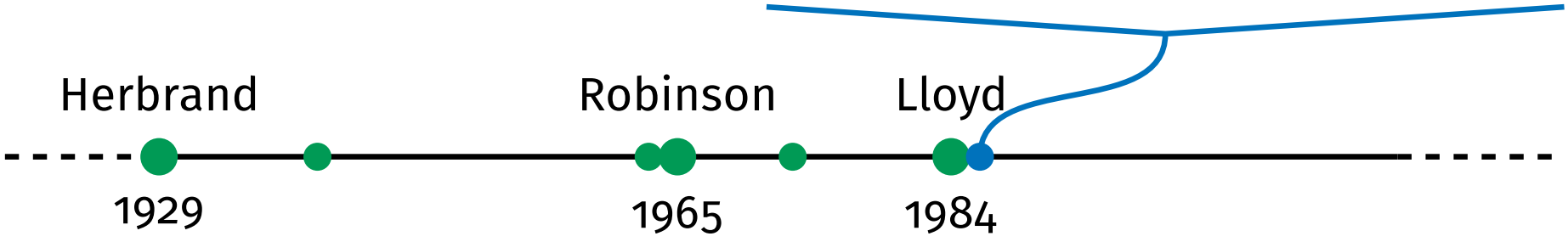
- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming



A multitude of variants

$$\exists\{x\colon \mathtt{Num}\}\cdot \mathtt{sorted}(2,3,x) = \mathsf{T}$$

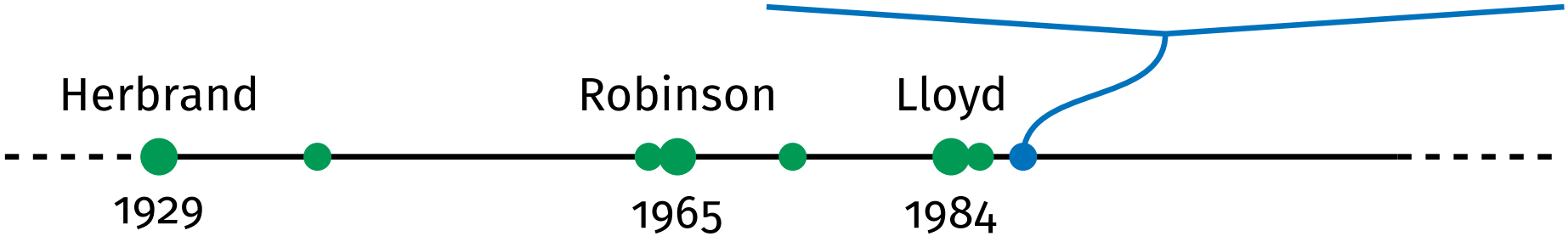
- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming



A multitude of variants

$$\exists\{s\colon \texttt{List} \to \mathrm{B}\} \cdot s\,[2,3,5] = \mathrm{T}$$

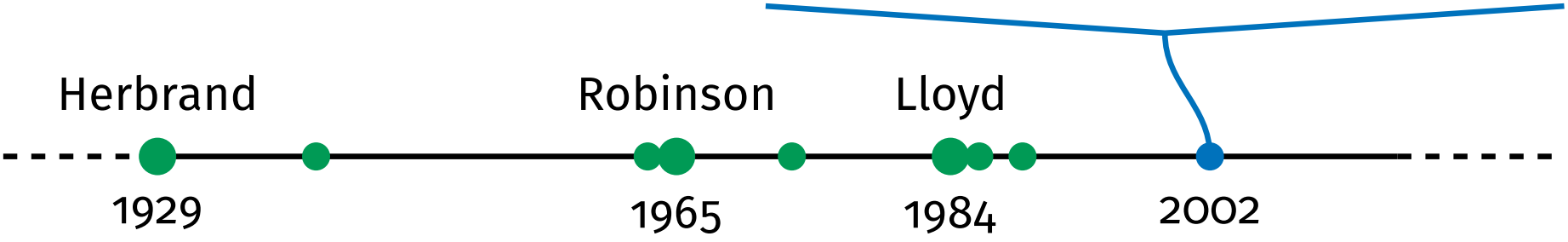
- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming



A multitude of variants

$\exists\{s\colon \mathtt{Stream}\}\cdot s \sim \mathtt{tail}(s)$

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

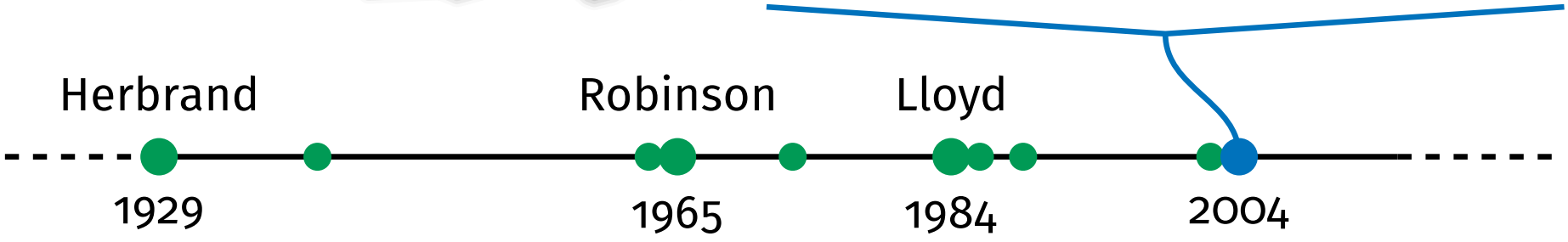


A multitude of variants

$$\langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$$

$$\begin{tikzcd}
 & \mathrm{Sen}(\Sigma) \arrow[dd, dash, "\models_\Sigma"] \\
\Sigma \arrow[ur, mapsto, "\mathrm{Sen}"] \arrow[dr, mapsto, "\mathrm{Mod}"'] & \\
 & \mathrm{Mod}(\Sigma)
\end{tikzcd}$$

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



A multitude of variants

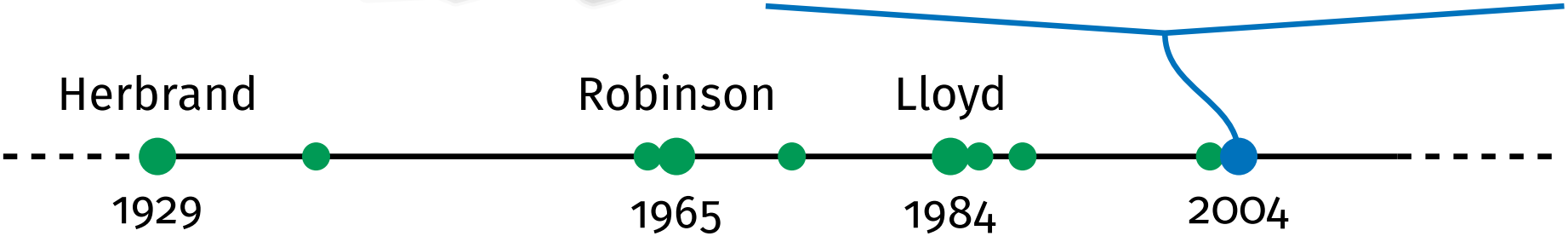
$\langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$

subject to a
satisfaction condition:

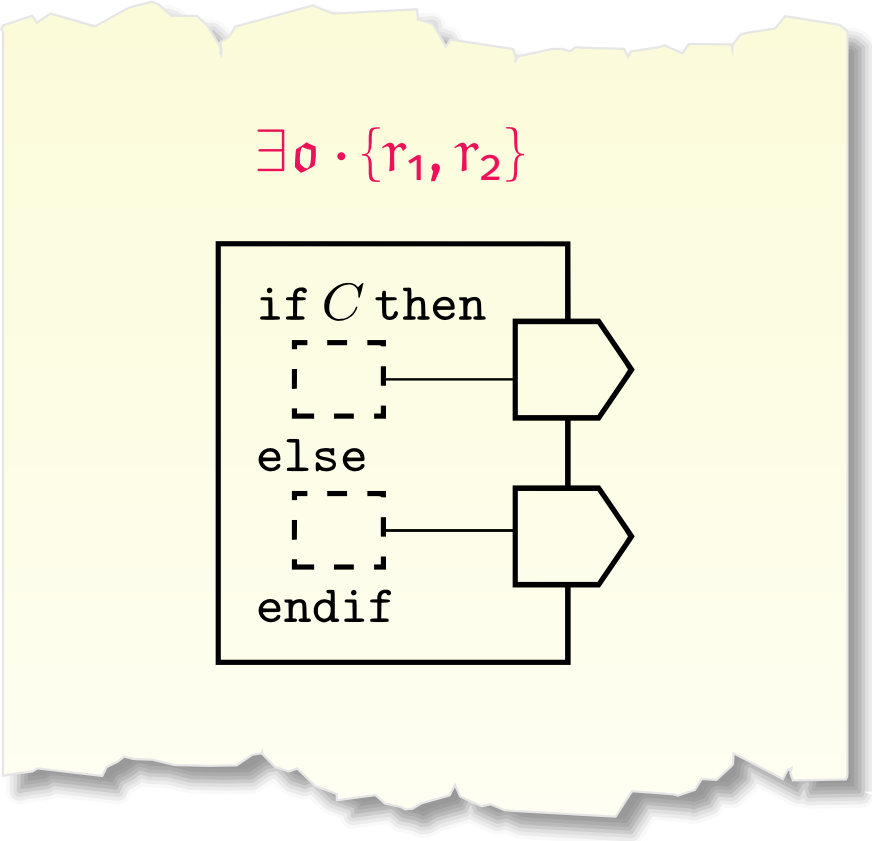
for every $\varphi \colon \Sigma \to \Omega$,
$M \in |\mathrm{Mod}(\Omega)|$, $\rho \in \mathrm{Sen}(\Sigma)$

$M\restriction_{\varphi} \models_{\Sigma} \rho$ *iff* $M \models_{\Omega} \varphi(\rho)$

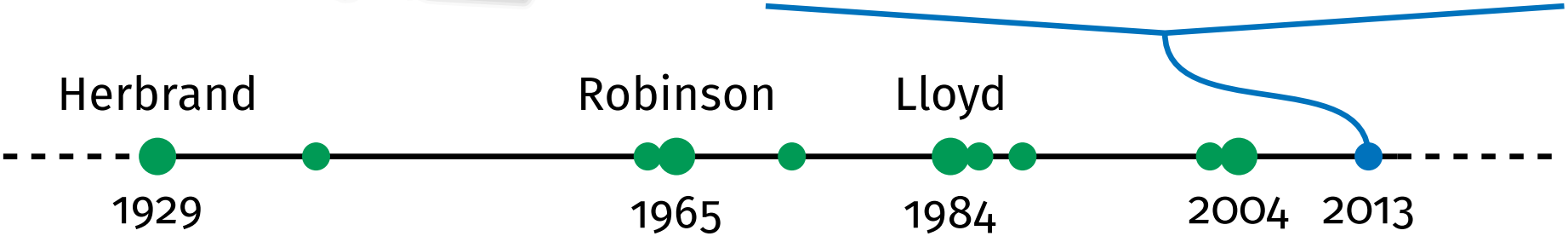
- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming



A multitude of variants



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

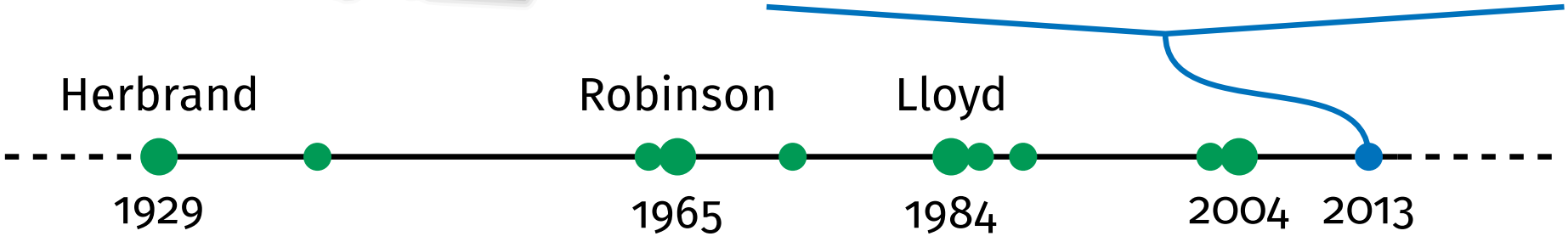


A multitude of variants

$$\exists \underbrace{\{x, y\}}_{\chi : \langle F,P\rangle \hookrightarrow \langle F\cup\{x,y\},P\rangle} \cdot \mathtt{sorted}(2, 3, x, y, 5)$$

$$\exists \chi \cdot \mathtt{sorted}(2, 3, x, y, 5)$$

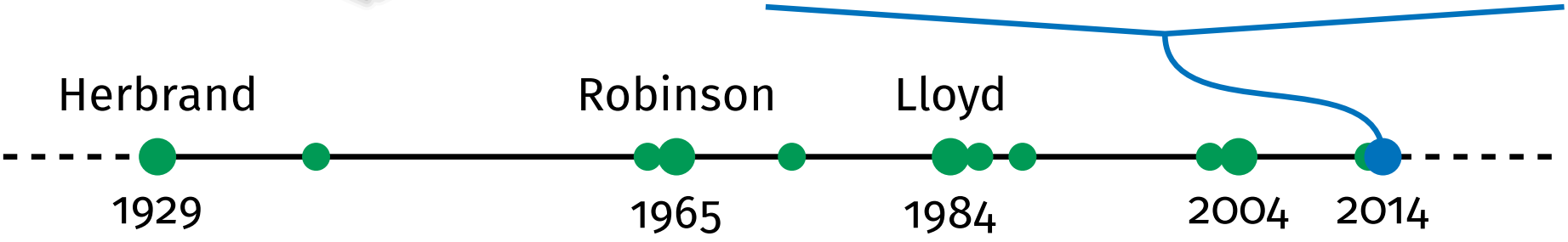
- relational first-order logic
- many-sorted equational logic
- higher-order logic
- hidden algebra
- institution-independent
- service-oriented
- abstract logic programming



A multitude of variants

$$
\begin{tikzcd}
 & \mathrm{Sen}_\Sigma(X) \arrow[d] \arrow[dr, dash, "\models"] & \\
\Sigma \arrow[d, "\varphi"'] & X \arrow[ur, mapsto] \arrow[dr, mapsto] \arrow[dd, mapsto] & \mathrm{Mod}_\Sigma(X) \arrow[d] \\
 & & \mathrm{Sen}_\Omega(X^\varphi) \arrow[dr, dash, "\models"] \\
\Omega & X^\varphi \arrow[ur, mapsto] \arrow[dr, mapsto] & \mathrm{Mod}_\Omega(X^\varphi) \arrow[uu] \\
 & &
\end{tikzcd}
$$

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

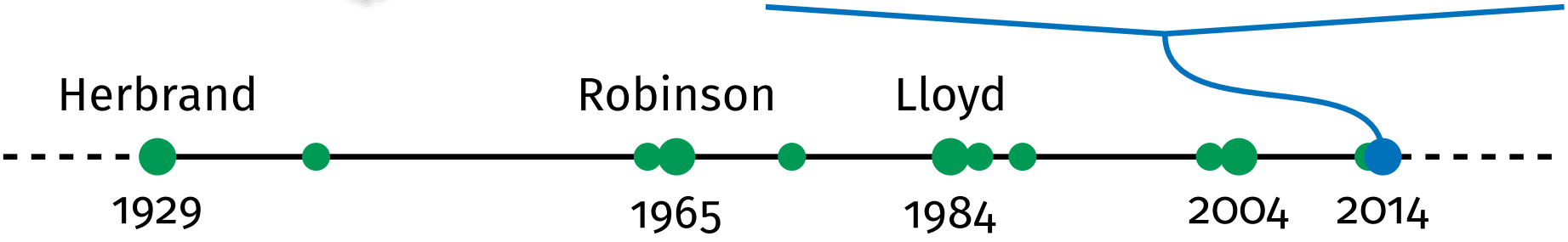


A multitude of variants

$$\begin{tikzcd} & \mathsf{Sen}_\Sigma(X) \arrow[dr, dash, "\models"] & \\ \Sigma \quad X \arrow[ur, mapsto] \arrow[rr, mapsto] & & \mathsf{Mod}_\Sigma(X) \end{tikzcd}$$

$\exists X \cdot \rho$

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



Institutions as functors

Rooms and corridors

$$\begin{tikzcd} \langle S, \mathbb{M}, \models \rangle \arrow[rr, bend left, "\alpha"] & & \langle S', \mathbb{M}', \models' \rangle \arrow[ll, bend left, "\beta"] \end{tikzcd}$$

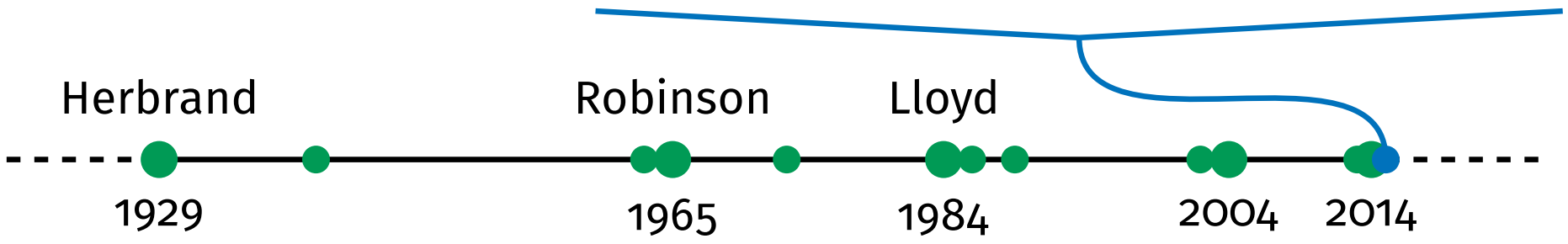
- each institution $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}, \mathsf{Sen}, \mathsf{Mod}, \models \rangle$ can be identified with a functor

$$\mathcal{I} \colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$$

where $\mathcal{I}(\Sigma) = \langle \mathsf{Sen}(\Sigma), \mathsf{Mod}(\Sigma), \models_\Sigma \rangle$

- similarly, substitution systems can be defined as functors

$$\mathcal{S} \colon \mathbb{S}\mathrm{ubst} \to \mathrm{G} \,/\, \mathbb{R}\mathrm{oom}$$



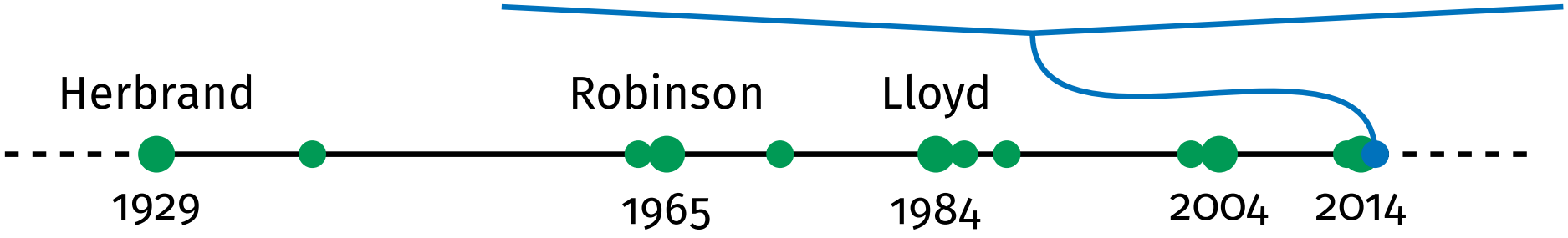
From institutions to substitution systems

$$\begin{tikzcd}
 & \mathcal{I}(\Sigma) \arrow[dl, "\mathcal{I}(\chi_1)"'] \arrow[dr, "\mathcal{I}(\chi_2)"] & \\
\mathcal{I}(\Sigma(\chi_1)) \arrow[rr, bend right, "\langle \mathsf{Sen}_\Sigma(\psi), \mathsf{Mod}_\Sigma(\psi) \rangle"'] & & \mathcal{I}(\Sigma(\chi_2))
\end{tikzcd}$$

- let $\mathcal{Q}$ be a class of signature morphisms of an institution $\mathcal{I}\colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$

  For every $\mathcal{I}$-signature Σ we obtain a substitution system $\mathcal{SI}^{\mathcal{Q}}_{\Sigma}\colon \mathbb{S}\mathrm{ubst}^{\mathcal{Q}}_{\Sigma} \to \mathcal{I}(\Sigma) \,/\, \mathbb{R}\mathrm{oom}$:

- *the objects of $\mathbb{S}\mathrm{ubst}^{\mathcal{Q}}_{\Sigma}$ are signature morphisms $\chi\colon \Sigma \to \Sigma(\chi)$ belonging to $\mathcal{Q}$*
- *a Σ-substitution $\psi\colon \chi_1 \to \chi_2$ is a corridor $\langle \mathsf{Sen}_\Sigma(\psi), \mathsf{Mod}_\Sigma(\psi) \rangle\colon \mathcal{I}(\Sigma(\chi_1)) \to \mathcal{I}(\Sigma(\chi_2))$*



Quantification spaces

$$\begin{tikzcd}
\Sigma \arrow[r, "\varphi"] \arrow[d, "\chi"'] & \Sigma' \arrow[d, "\chi'"] \\
\Omega \arrow[r, "\phi"'] & \Omega'
\end{tikzcd}$$

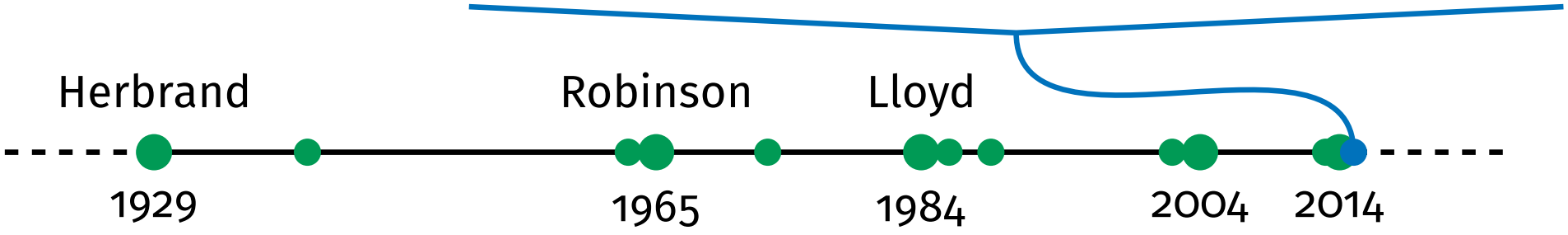
$$\downarrow \iota_{\mathbb{Q},\chi}$$

$$\Sigma \xrightarrow{\varphi} \Sigma'$$

- for every subcategory $\mathbb{Q} \subseteq \mathbb{S}\mathrm{ig}^{\rightarrow}$, the functor $\mathrm{dom}\colon \mathbb{Q} \to \mathbb{S}\mathrm{ig}$ gives rise to a natural transformation $\iota_{\mathbb{Q}}\colon (\_ \,/\, \mathbb{Q}) \Rightarrow \mathrm{dom}^{\mathrm{op}} \fatsemi (\_ \,/\, \mathbb{C})$

Definition. $\mathbb{Q}$ is said to be a quantification space for an institution $\mathcal{I}\colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ if

1. every arrow in $\mathbb{Q}$ forms a pushout in $\mathbb{S}\mathrm{ig}$, and
2. $\iota_{\mathbb{Q}}$ is a natural isomorphism.



Quantification spaces

$$
\begin{tikzcd}
\Sigma \arrow[r, "\varphi"] \arrow[d, "\chi"'] & \Sigma' \arrow[d, "\chi^{\varphi}"] \\
\Sigma(\chi) \arrow[r, "\varphi^{\chi}"'] & \Sigma'(\chi^{\varphi})
\end{tikzcd}
$$

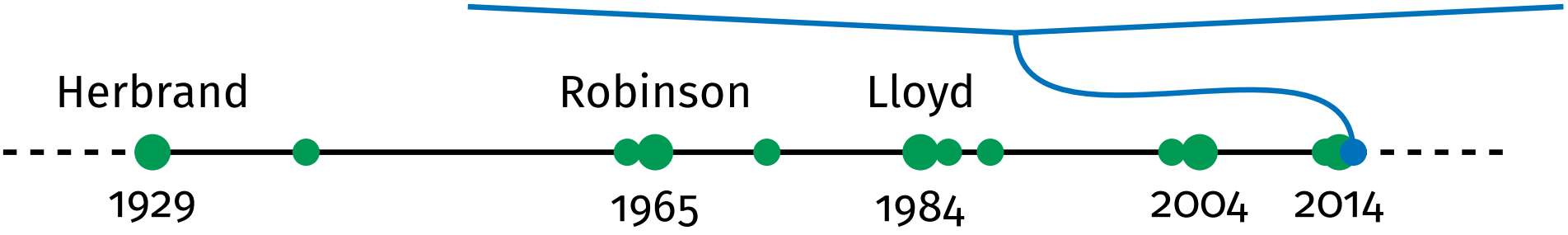
$$\downarrow \iota_{\mathbb{Q},\chi}$$

$$\Sigma \xrightarrow{\varphi} \Sigma'$$

- for every subcategory $\mathbb{Q} \subseteq \mathbb{S}\mathrm{ig}^{\rightarrow}$, the functor $\mathrm{dom}\colon \mathbb{Q} \to \mathbb{S}\mathrm{ig}$ gives rise to a natural transformation $\iota_{\mathbb{Q}}\colon (\_ \,/\, \mathbb{Q}) \Rightarrow \mathrm{dom}^{\mathrm{op}} \fatsemi (\_ \,/\, \mathbb{C})$

Definition. $\mathbb{Q}$ is said to be a quantification space for an institution $\mathcal{I}\colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ if

1. every arrow in $\mathbb{Q}$ forms a pushout in $\mathbb{S}\mathrm{ig}$, and
2. $\iota_{\mathbb{Q}}$ is a natural isomorphism.



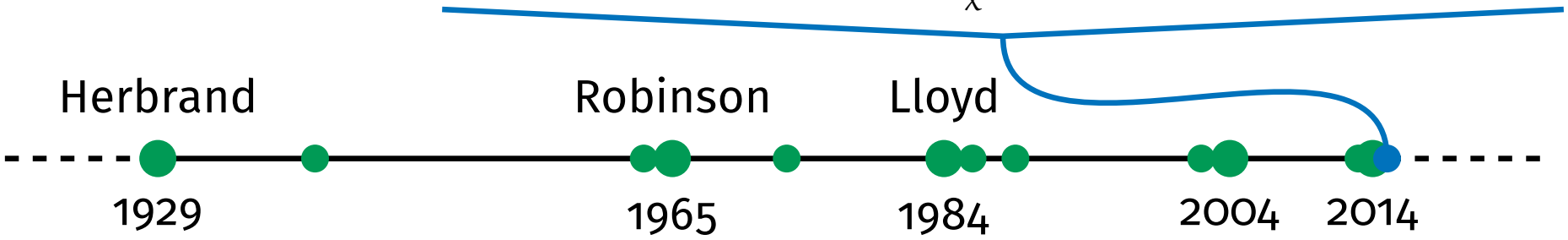
Representable signature extensions

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

- a Σ-model M_χ and
- an isomorphism of categories i_χ

such that the following diagram commutes:

$$\begin{tikzcd}
\mathsf{Mod}(\Sigma) & \\
\mathsf{Mod}(\Sigma(\chi)) \arrow[u, "-\restriction_\chi"] \arrow[r, "i_\chi"'] & M_\chi \,/\, \mathsf{Mod}(\Sigma) \arrow[ul, "\text{forgetful}"']
\end{tikzcd}$$

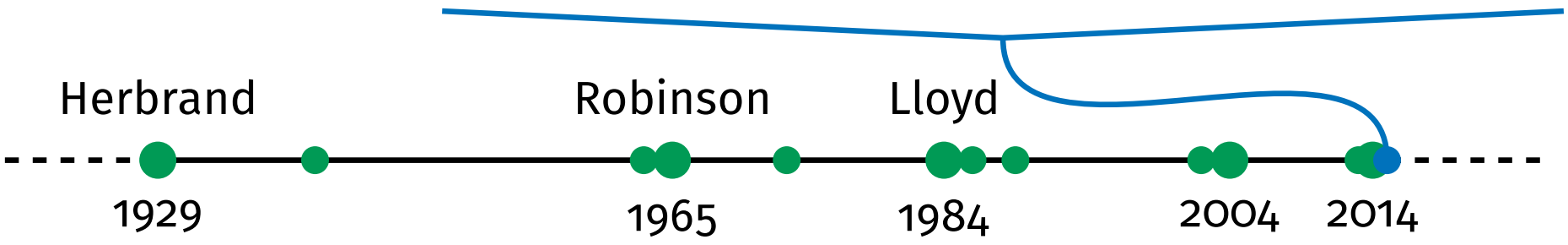


Representable signature extensions

Proposition. The representation of signature extensions generalizes to a functor $R^{\mathbb{Q}}_{\Sigma} \colon \mathbb{S}\mathrm{ubst}^{\mathbb{Q}}_{\Sigma} \to \mathsf{Mod}(\Sigma)$, where

- for every $\chi \colon \Sigma \to \Sigma(\chi)$ in $|\mathbb{Q}|$, $R^{\mathbb{Q}}_{\Sigma}(\chi) = M_{\chi}$,
- for every substitution $\psi \colon \chi_1 \to \chi_2$, $R^{\mathbb{Q}}_{\Sigma}(\psi) = (i_{\chi_2}^{-1} \fatsemi \mathsf{Mod}_{\Sigma}(\psi) \fatsemi i_{\chi_1})(1_{M_{\chi_2}})$.

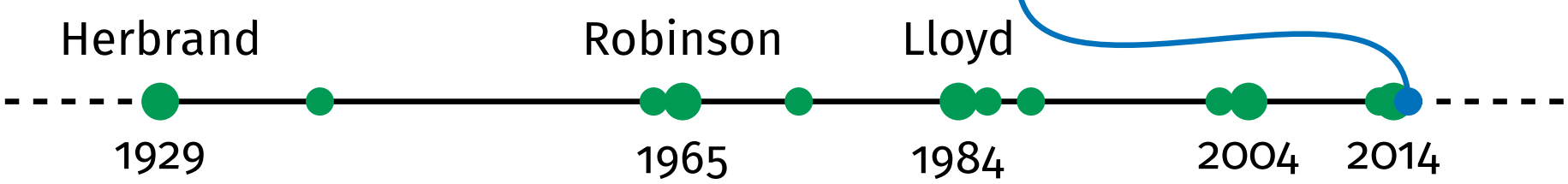
Moreover, for every Σ-substitution ψ, $\mathsf{Mod}_{\Sigma}(\psi)$ is uniquely determined by $R^{\mathbb{Q}}_{\Sigma}(\psi)$.



Representable signature extensions

Proposition. Every morphism of signatures $\varphi\colon \Sigma \to \Sigma'$ gives rise to a functor $\Psi_\varphi\colon \mathbb{S}\mathrm{ubst}_\Sigma \to \mathbb{S}\mathrm{ubst}_{\Sigma'}$ defined as follows:

$$\begin{tikzcd}
 & \mathcal{J}(\Sigma) \arrow[rr, "\varphi"] \arrow[dl, "\mathcal{J}(\chi_1)"'] \arrow[ddr, "\mathcal{J}(\chi_2)"] & & \mathcal{J}(\Sigma') \arrow[dl, "\mathcal{J}(\chi_1^\varphi)"'] \arrow[ddr, "\mathcal{J}(\chi_2^\varphi)"] & \\
\mathcal{J}(\Sigma(\chi_1)) \arrow[rr, "\mathcal{J}(\varphi^{\chi_1})"'] \arrow[dr, "\psi"'] & & & \mathcal{J}(\Sigma'(\chi_1^\varphi)) \arrow[dr, "\left(R^{\mathbb{Q}}_{\Sigma'}\right)^{-1}\left(R^{\mathbb{Q}}_{\Sigma}(\psi)^\varphi\right)"] & \\
 & & \mathcal{J}(\Sigma(\chi_2)) \arrow[rr, "\mathcal{J}(\varphi^{\chi_2})"'] & & \mathcal{J}(\Sigma'(\chi_2^\varphi))
\end{tikzcd}$$

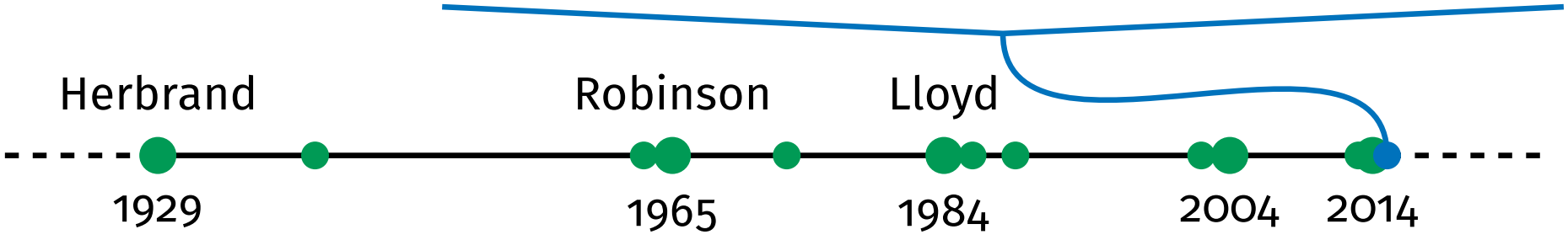


Deriving generalized substitution systems

Theorem. Every institution $\mathcal{I}\colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ equipped with

- an adequate quantification space $\mathbb{Q}$ of representable signature extensions and
- compatible categories $\mathbb{S}\mathrm{ubst}_\Sigma$ of $\mathbb{Q}$-representable Σ-substitutions,

determines a generalized substitution system that has model amalgamation.



Herbrand's theorem revisited

$$\begin{tikzcd} M_\chi \arrow[rr, dashed, "h"] \arrow[dr, "f"'] & & O_\Sigma \arrow[dl, "!_\Gamma"] \\ & O_{\Sigma,\Gamma} & \end{tikzcd}$$

$M \models \rho$

iff

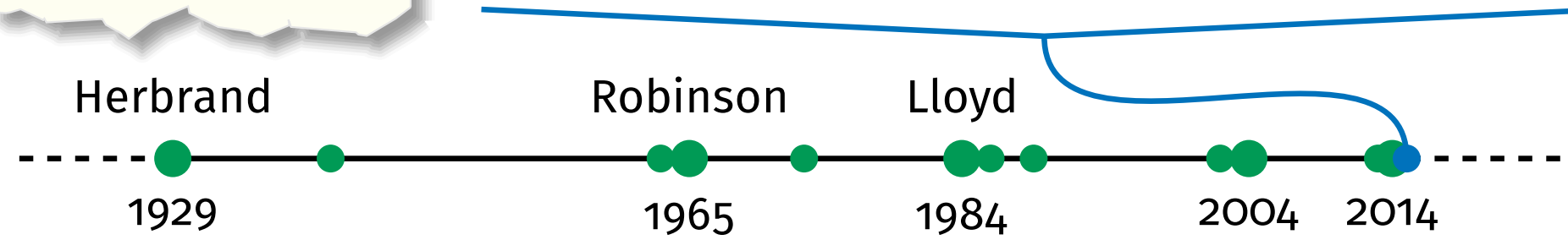
$M_\rho \to M$

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

- Σ and $\langle \Sigma, \Gamma \rangle$ have initial models O_Σ and $O_{\Sigma,\Gamma}$,
- M_χ is projective with respect to the unique homomorphism $!_\Gamma : O_\Sigma \to O_{\Sigma,\Gamma}$, and
- the sentence ρ 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 Γ-solution.



Thank you!

Further Reading

J. Herbrand.
Investigations in proof theory.
In: *From Frege to Gödel: A Source Book in Mathematical Logic*, HUP, 1967.

J. W. Lloyd.
Foundations of Logic Programming, 2nd Edition.
Springer, 1987.

R. Diaconescu.
Herbrand theorems in arbitrary institutions.
Information Processing Letters, Elsevier, 2004.

I. Ţuţu and J. L. Fiadeiro.
From conventional to institution-independent logic programming.
Journal of Logic and Computation, OUP, in press.