Talk #10: Interpretations

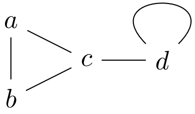
Semilattices are complex enough to encode graphs

Semilattices are complex enough to encode graphs

The graph

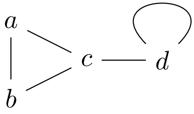
Semilattices are complex enough to encode graphs

The graph



Semilattices are complex enough to encode graphs

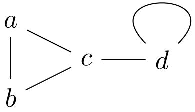
The graph



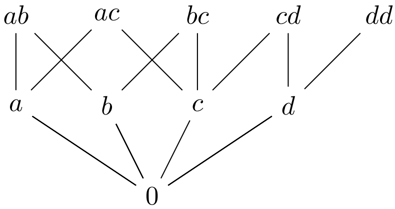
can be “encoded” into a semilattice:

Semilattices are complex enough to encode graphs

The graph

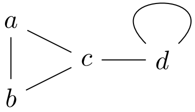


can be “encoded” into a semilattice:

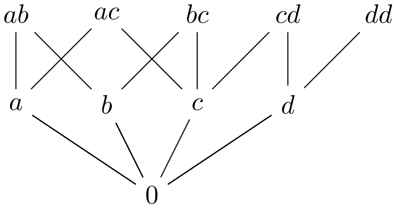


Semilattices are complex enough to encode graphs

The graph



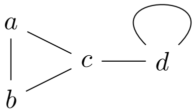
can be “encoded” into a semilattice:



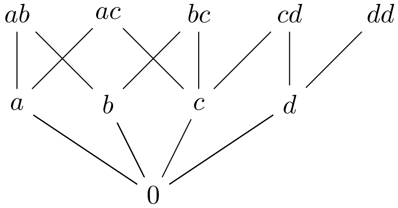
Vertices = atoms.

Semilattices are complex enough to encode graphs

The graph



can be “encoded” into a semilattice:

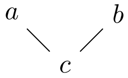


Vertices = atoms. Edge between $x, y \in V$ represented by a height-2 element that dominates both x and y. This IS a semilattice order, provided …

Graphs are complex enough to encode semilattices

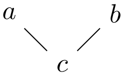
Graphs are complex enough to encode semilattices

The semilattice



Graphs are complex enough to encode semilattices

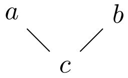
The semilattice



can be encoded as follows.

Graphs are complex enough to encode semilattices

The semilattice



can be encoded as follows. Given $\langle S; \wedge \rangle$,

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd} a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph

Graphs are complex enough to encode semilattices

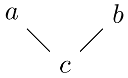
The semilattice

$$\begin{tikzcd}[column sep=small] a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques),

Graphs are complex enough to encode semilattices

The semilattice



can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex:

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd}[column sep=small] a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$,

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd} a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$,

Graphs are complex enough to encode semilattices

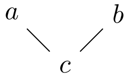
The semilattice

$$\begin{tikzcd}[column sep=small] a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$,

Graphs are complex enough to encode semilattices

The semilattice



can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1}$,

Graphs are complex enough to encode semilattices

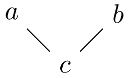
The semilattice

$$\begin{tikzcd} a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1}$, $L_{\wedge,2}$,

Graphs are complex enough to encode semilattices

The semilattice



can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd}[column sep=small] a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$ for instances of $s_1 \wedge s_2 = s_3$.

Graphs are complex enough to encode semilattices

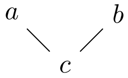
The semilattice

$$\begin{tikzcd}[column sep=small] a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$ for instances of $s_1 \wedge s_2 = s_3$. For each cell of the table for $\wedge$

Graphs are complex enough to encode semilattices

The semilattice



can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$ for instances of $s_1 \wedge s_2 = s_3$. For each cell of the table for $\wedge$ (e.g. $a \wedge b = c$)

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd} a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$ for instances of $s_1 \wedge s_2 = s_3$. For each cell of the table for $\wedge$ (e.g. $a \wedge b = c$) draw:

Graphs are complex enough to encode semilattices

The semilattice

$$\begin{tikzcd} a \arrow[dr, dash] & & b \arrow[dl, dash] \\ & c & \end{tikzcd}$$

can be encoded as follows. Given $\langle S; \wedge \rangle$, choose distinct, recognizable pieces of graph (e.g. large cliques), each with a distinguished vertex: G_s for element $s \in S$, $H_\wedge$ for symbol $\wedge$, $K_{\wedge,1},\ L_{\wedge,2},\ M_{\wedge,3}$ for instances of $s_1 \wedge s_2 = s_3$. For each cell of the table for $\wedge$ (e.g. $a \wedge b = c$) draw:

$$\begin{tikzcd}
G_a \arrow[dr, dash] & & & & G_b \arrow[dl, dash] \\
& K_{\wedge,1} \arrow[dr, dash] & & L_{\wedge,2} \arrow[dl, dash] & \\
& & H_\wedge \arrow[d, dash] & & \\
& & M_{\wedge,3} \arrow[d, dash] & & \\
& & G_c & &
\end{tikzcd}$$

Interpretations, 1

Interpretations, 1

Let K and L be first-order languages.

Let K and L be first-order languages. A **scheme for interpreting** K **in** L,

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K,

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K,

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K, an L-formula $\varphi_F(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}, \mathbf{z})$, and

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K, an L-formula $\varphi_F(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}, \mathbf{z})$, and

(5) for each k-ary relation symbol R in K, an L-formula $\rho_R(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{z})$.

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting** K **in** L, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K, an L-formula $\varphi_F(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}, \mathbf{z})$, and

(5) for each k-ary relation symbol R in K, an L-formula $\rho_R(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{z})$.

Interpretations, 1

Let K and L be first-order languages. A **scheme for interpreting K in L**, $\Sigma = (m, n, \Delta, \varepsilon, \varphi_1, \ldots, \rho_1, \ldots)$, consists of the following ingredients:

(1) integers $m \geq 1, n \geq 0$.

In the next items $\mathbf{x}$ and $\mathbf{y}$ denote m-tuples of variables and $\mathbf{z}$ denotes an n-tuple of variables.

(2) an L-formula $\Delta(\mathbf{x}, \mathbf{z})$ (called the domain formula),

(3) an L-formula $\varepsilon(\mathbf{x}, \mathbf{y}, \mathbf{z})$ (called the equivalence formula),

(4) for each k-ary operation symbol F in K, an L-formula $\varphi_F(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{y}, \mathbf{z})$, and

(5) for each k-ary relation symbol R in K, an L-formula $\rho_R(\mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{z})$.

Interpretations, 2

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$".

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \iff \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \iff \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \iff \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

A class $\mathcal{K}$ of K-structures is **interpretable by parameters**

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \iff \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

A class $\mathcal{K}$ of K-structures is **interpretable by parameters** (or **definable by parameters**)

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \iff \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \iff \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \iff \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

A class $\mathcal{K}$ of K-structures is **interpretable by parameters** (or **definable by parameters**) in a class $\mathcal{L}$ of L-structures by a scheme Σ

Interpretations, 2

An L-structure $\mathbf{B}$ and a tuple $\mathbf{b} \in B^n$ **admits** the scheme Σ if "$\mathbf{B}$ can be used to construct a K-structure using parameters $\mathbf{b}$". I.e., if there is a K-structure $\mathbf{A} = \langle A; F_1, \ldots, R_1, \ldots \rangle$ and a surjective mapping $h : \Delta(B, \mathbf{b}) \to A$ such that

(i) $h(\mathbf{u}) = h(\mathbf{v}) \Longleftrightarrow \varepsilon(\mathbf{u}, \mathbf{v}, \mathbf{b})$.

(ii) $F(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) = h(\mathbf{v}) \Longleftrightarrow \varphi_F(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{v}, \mathbf{b})$.

(iii) $R(h(\mathbf{u}_1), \ldots, h(\mathbf{u}_k)) \Longleftrightarrow \rho_R(\mathbf{u}_1, \ldots, \mathbf{u}_k, \mathbf{b})$.

A class $\mathcal{K}$ of K-structures is **interpretable by parameters** (or **definable by parameters**) in a class $\mathcal{L}$ of L-structures by a scheme Σ if for every $\mathbf{A} \in \mathcal{K}$ there is a $\mathbf{B} \in \mathcal{L}$ and $\mathbf{b} \in B^n$ such that $\mathbf{A}$ witnesses that $(\mathbf{B}, \mathbf{b})$ admits Σ.

Some theorems

Some theorems

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle \alpha, \beta \rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle \alpha, \beta \rangle$-minimal algebra of type $\mathbf{4}$.

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle \alpha, \beta \rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle \alpha, \beta \rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations)

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations) If $\mathcal{K}$ is interpretable in $\mathcal{L}$ and $\mathcal{L}$ is interpretable in $\mathcal{M}$, then $\mathcal{K}$ is interpretable in $\mathcal{M}$.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations) If $\mathcal{K}$ is interpretable in $\mathcal{L}$ and $\mathcal{L}$ is interpretable in $\mathcal{M}$, then $\mathcal{K}$ is interpretable in $\mathcal{M}$.

Theorem.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations) If $\mathcal{K}$ is interpretable in $\mathcal{L}$ and $\mathcal{L}$ is interpretable in $\mathcal{M}$, then $\mathcal{K}$ is interpretable in $\mathcal{M}$.

Theorem. If $\mathcal{V}$ is a variety and some finite member has a type $\mathbf{4}$ or type $\mathbf{5}$ congruence covering,

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations) If $\mathcal{K}$ is interpretable in $\mathcal{L}$ and $\mathcal{L}$ is interpretable in $\mathcal{M}$, then $\mathcal{K}$ is interpretable in $\mathcal{M}$.

Theorem. If $\mathcal{V}$ is a variety and some finite member has a type $\mathbf{4}$ or type $\mathbf{5}$ congruence covering, then any class of structures in any finite language is interpretable in $\mathcal{V}$.

Some theorems

Theorem. Any class of structures in a finite language is interpretable in the class of graphs. (Copy the idea for interpreting semilattices into graphs.)

Theorem. The class of graphs is interpretable in the variety of semilattices and also the variety of distributive lattices.

Theorem. The variety of semilattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{5}$. The variety of distributive lattices is interpretable in any variety generated by an $\langle\alpha, \beta\rangle$-minimal algebra of type $\mathbf{4}$.

Theorem. If $\mathbf{A}$ is a finite algebra and U is a neighborhood of A, then the variety $\mathcal{V}(\mathbf{A}|_U)$ is interpretable in $\mathcal{V}(\mathbf{A})$.

Theorem. (Transitivity of interpretations) If $\mathcal{K}$ is interpretable in $\mathcal{L}$ and $\mathcal{L}$ is interpretable in $\mathcal{M}$, then $\mathcal{K}$ is interpretable in $\mathcal{M}$.

Theorem. If $\mathcal{V}$ is a variety and some finite member has a type $\mathbf{4}$ or type $\mathbf{5}$ congruence covering, then any class of structures in any finite language is interpretable in $\mathcal{V}$.

Decidable varieties

Decidable varieties

Let L be a finite language.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ,

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes,

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps,

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)
3. (Alfred Tarski, 1940/1949)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)
3. (Alfred Tarski, 1940/1949)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)
3. (Alfred Tarski, 1940/1949)
   Every axiomatizable class of Boolean algebras is decidable.

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)
3. (Alfred Tarski, 1940/1949)
   Every axiomatizable class of Boolean algebras is decidable. (Type **3** examples.)

Decidable varieties

Let L be a finite language. A class $\mathcal{V}$ of L-structures is **decidable** if the first-order theory of $\mathcal{V}$ computable. According to the Church-Turing thesis, decidability of $\mathcal{V}$ means that there is a Turing machine that accepts a first-order sentence σ, computes, halts after finitely many steps, and outputs a correct answer to the question:

Question. Is σ true throughout $\mathcal{V}$?

Theorems. The following classes are decidable:

1. (Andrzej Ehrenfeucht, 1959)
   The class of all monounary algebras is decidable. (Type **1** examples.)
2. (Wanda Szmielew, 1955)
   Every axiomatizable class of abelian groups is decidable. (Type **2** examples.)
3. (Alfred Tarski, 1940/1949)
   Every axiomatizable class of Boolean algebras is decidable. (Type **3** examples.)

Hereditary undecidability

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)
   Any class that interprets a herditarily undecidable class in a finite language is also herditarily undecidable.

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)
   Any class that interprets a herditarily undecidable class in a finite language is also herditarily undecidable.

Theorem.

Hereditary undecidability

An axiomatizable class $\mathcal{K}$ of L-structures is **hereditarily undecidable** if every axiomatizable class $\mathcal{K}' \supseteq \mathcal{K}$ is undecidable.

Theorem.

1. (Igor Lavrov, 1963)
   The class of finite graphs is hereditarily undecidable.
2. (Michael Rabin, 1965)
   Any class that interprets a herditarily undecidable class in a finite language is also herditarily undecidable.

Theorem. A locally finite variety that is not hereditarily undecidable must satisfy $\operatorname{typ}(\mathcal{V}) \subseteq \{\mathbf{1}, \mathbf{2}, \mathbf{3}\}$.

Local structure in decidable varieties, 1

Tails.

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1**

Local structure in decidable varieties, 1

Tails.

Any variety that admits type **1** (or type **5**)

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails.

Tails.

Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem.

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

The proof is accomplished by interpreting the hereditarily undecidable class of Boolean pairs into $\mathcal{V}$.

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

The proof is accomplished by interpreting the hereditarily undecidable class of Boolean pairs into $\mathcal{V}$. A **Boolean pair** on X is a structure $(B_0, B_1, \subseteq)$ where

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

The proof is accomplished by interpreting the hereditarily undecidable class of Boolean pairs into $\mathcal{V}$. A **Boolean pair** on X is a structure $(B_0, B_1, \subseteq)$ where

$$\mathcal{P}_{\text{fin}}(X) \leq B_0 \leq B_1 \leq \mathcal{P}(X)$$

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

The proof is accomplished by interpreting the hereditarily undecidable class of Boolean pairs into $\mathcal{V}$. A **Boolean pair** on X is a structure $(B_0, B_1, \subseteq)$ where

$$\mathcal{P}_{\text{fin}}(X) \leq B_0 \leq B_1 \leq \mathcal{P}(X)$$

and $\subseteq$ is inclusion.

Local structure in decidable varieties, 1

Tails.
Any variety that admits type **1** (or type **5**) minimal sets must admits instances where the minimal sets of type **1** (or **5**) have nonempty tails. Nevertheless, we have this

Theorem. (David Hobby-Ralph McKenzie, 1988)
If $\mathcal{V}$ is a locally finite variety that admits types **2** or **3** minimal sets with nonempty tails, then $\mathcal{V}$ is hereditarily undecidable.

The proof is accomplished by interpreting the hereditarily undecidable class of Boolean pairs into $\mathcal{V}$. A **Boolean pair** on X is a structure $(B_0, B_1, \subseteq)$ where

$$\mathcal{P}_{\text{fin}}(X) \leq B_0 \leq B_1 \leq \mathcal{P}(X)$$

and $\subseteq$ is inclusion. Burris and McKenzie proved that the class of graphs interprets into the class of Boolean pairs.

Local structure in decidable varieties, 2

Transfer principles.

Transfer principles.

The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

Local structure in decidable varieties, 2

Transfer principles.

The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise.

Local structure in decidable varieties, 2

Transfer principles.

The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j}\rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j}\rangle$ and the $\langle \mathbf{j}, \mathbf{i}\rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3}\rangle$-transfer principle,

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise.

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$.

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$. Let $U = e(A) \in \mathrm{Min}_{\mathbf{A}}(0, \beta)$ and $V = \{0, 1\} \in \mathrm{Min}_{\mathbf{A}}(\beta, \gamma)$.

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$. Let $U = e(A) \in \mathrm{Min}_{\mathbf{A}}(0, \beta)$ and $V = \{0, 1\} \in \mathrm{Min}_{\mathbf{A}}(\beta, \gamma)$. Choose distinct $a, b \in U$ in a $\langle 0, \beta \rangle$-trace N.

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$. Let $U = e(A) \in \mathrm{Min}_{\mathbf{A}}(0, \beta)$ and $V = \{0, 1\} \in \mathrm{Min}_{\mathbf{A}}(\beta, \gamma)$. Choose distinct $a, b \in U$ in a $\langle 0, \beta \rangle$-trace N. By the transfer failure, we must have $(a, b) \in \mathrm{Cg}(0, 1)$,

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$. Let $U = e(A) \in \mathrm{Min}_{\mathbf{A}}(0, \beta)$ and $V = \{0, 1\} \in \mathrm{Min}_{\mathbf{A}}(\beta, \gamma)$. Choose distinct $a, b \in U$ in a $\langle 0, \beta \rangle$-trace N. By the transfer failure, we must have $(a, b) \in \mathrm{Cg}(0, 1)$, so there exists a nonconstant polynomial $f \colon V \to N$.

Local structure in decidable varieties, 2

Transfer principles.
The $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principle states that a 3-element congruence interval

$$\alpha \overset{\mathbf{i}}{\prec} \beta \overset{\mathbf{j}}{\prec} \gamma$$

does not arise. If both the $\langle \mathbf{i}, \mathbf{j} \rangle$ and the $\langle \mathbf{j}, \mathbf{i} \rangle$-transfer principle hold for $\mathcal{V}$, then types $\mathbf{i}$ and $\mathbf{j}$ are "independent" of each other.

To prove, e.g., that a decidable variety $\mathcal{V}$ satisfies the $\langle \mathbf{2}, \mathbf{3} \rangle$-transfer principle, assume otherwise. There must exist $\mathbf{A} \in \mathcal{V}$ with congruences $0 \overset{\mathbf{2}}{\prec} \beta \overset{\mathbf{3}}{\prec} \gamma$. Let $U = e(A) \in \mathrm{Min}_{\mathbf{A}}(0, \beta)$ and $V = \{0, 1\} \in \mathrm{Min}_{\mathbf{A}}(\beta, \gamma)$. Choose distinct $a, b \in U$ in a $\langle 0, \beta \rangle$-trace N. By the transfer failure, we must have $(a, b) \in \mathrm{Cg}(0, 1)$, so there exists a nonconstant polynomial $f \colon V \to N$. Finish the argument by showing that the class of structures $\langle \overline{V}, \overline{U}, \overline{f} \rangle$ where $\overline{V}$ is a BA, $\overline{U}$ is a vector space, and $\overline{f} \colon \overline{V} \to \overline{U}$ is nonconstant is herditarily undecidable and interpretable into $\mathcal{V}$.

Local structure in decidable varieties, 3

Labelled congruence lattices.

Local structure in decidable varieties, 3

Labelled congruence lattices.

If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$,

Local structure in decidable varieties, 3

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Local structure in decidable varieties, 3

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

$$\mathbf{A} = \mathbf{A_1} \times \mathbf{A_2} \times \mathbf{A_3}, \quad \mathrm{typ}\{\mathbf{A_i}\} = \{\mathbf{i}\}.$$

Local structure in decidable varieties, 3

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

$$\mathbf{A} = \mathbf{A_1} \times \mathbf{A_2} \times \mathbf{A_3}, \quad \mathrm{typ}\{\mathbf{A_i}\} = \{\mathbf{i}\}.$$

Additional work is needed to prove that locally type-**i** algebras form a subvariety.

Local structure in decidable varieties, 3

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

$$\mathbf{A} = \mathbf{A_1} \times \mathbf{A_2} \times \mathbf{A_3}, \quad \mathrm{typ}\{\mathbf{A_i}\} = \{\mathbf{i}\}.$$

Additional work is needed to prove that locally type-**i** algebras form a subvariety. Here one needs to show that a subalgebra of a type-**i** algebra is type **i**.

Local structure in decidable varieties, 3

Labelled congruence lattices.
If one shows that all $\langle \mathbf{i}, \mathbf{j} \rangle$-transfer principles hold for $\mathbf{i} \neq \mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A}) \cong L_{\mathbf{1}} \times L_{\mathbf{2}} \times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

$$\mathbf{A} = \mathbf{A_1} \times \mathbf{A_2} \times \mathbf{A_3}, \quad \mathrm{typ}\{\mathbf{A_i}\} = \{\mathbf{i}\}.$$

Additional work is needed to prove that locally type-**i** algebras form a subvariety. Here one needs to show that a subalgebra of a type-**i** algebra is type **i**.
Additional work is needed to prove that the subvarieties are independent.

Labelled congruence lattices.
If one shows that all $\langle\mathbf{i},\mathbf{j}\rangle$-transfer principles hold for $\mathbf{i}\neq\mathbf{j}$, then all congruence lattices in $\mathcal{V}$ factor as products of labelled lattices:

$$\mathrm{Con}(\mathbf{A})\cong L_{\mathbf{1}}\times L_{\mathbf{2}}\times L_{\mathbf{3}}.$$

Additional work is needed to prove that this congruence lattice decomposition comes from an algebra decomposition:

$$\mathbf{A}=\mathbf{A_1}\times\mathbf{A_2}\times\mathbf{A_3},\quad \mathrm{typ}\{\mathbf{A_i}\}=\{\mathbf{i}\}.$$

Additional work is needed to prove that locally type-**i** algebras form a subvariety. Here one needs to show that a subalgebra of a type-**i** algebra is type **i**.
Additional work is needed to prove that the subvarieties are independent.

Local structure in decidable varieties, 4

Theorem.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety. (Finitely generated case is fully known.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety. (Finitely generated case is fully known. $\mathcal{S}$ = "generalized Boolean algebras".)

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety. (Finitely generated case is fully known. $\mathcal{S}$ = "generalized Boolean algebras".)

Moreover, it is impossible to interpret the class of graphs into any of $\mathcal{S}, \mathcal{A}, \mathcal{D}$.

Local structure in decidable varieties, 4

Theorem. Let $\mathcal{V}$ be a locally finite variety. If it is impossible to interpret the class of graphs into $\mathcal{V}$, then there is a unique decomposition

$$\mathcal{V} = \mathcal{S} \otimes \mathcal{A} \otimes \mathcal{D}$$

where

1. $\mathcal{S}$ is a type-**1** strongly abelian variety. (The structure of $\mathcal{S}$ is fully known. $\mathcal{S}$ = "generalized monounary algebras".)
2. $\mathcal{A}$ is a type-**2** affine variety. (Much is known. $\mathcal{A}$ = "generalized modules over a finite ring".)
3. $\mathcal{D}$ is a type-**3** discriminator variety. (Finitely generated case is fully known. $\mathcal{S}$ = "generalized Boolean algebras".)

Moreover, it is impossible to interpret the class of graphs into any of $\mathcal{S}, \mathcal{A}, \mathcal{D}$.