Computation with Atoms
Szymon Toruńczyk, University of Warsaw
includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...
Atoms

a fixed *underlying* logical structure
Atoms

a fixed underlying logical structure

Examples:

\[(\mathbb{N}, =)\] – pure set

\[(\mathbb{Q}, \leq)\] – dense order

\[(\mathbb{R}, +, \times, 0, 1)\] – field of reals

\[(\mathbb{N}, +, \leq)\] – Presburger arithmetic
Atoms

a fixed underlying logical structure

Examples:

\((\mathbb{N}, =)\) – pure set ← main example in this talk

\((\mathbb{Q}, \leq)\) – dense order

\((\mathbb{R}, +, \times, 0, 1)\) – field of reals

\((\mathbb{N}, +, \leq)\) – Presburger arithmetic
Hereditarily definable set
Hereditarily definable set

5
\{a: a \in \text{Atoms}\}
\{a: a \in \text{Atoms}, a \neq 7 \land a \neq 5\}
\{\{a:a \in \text{Atoms}, a \neq b\}: b \in \text{Atoms}\}
\{\{b: b \in \text{Atoms}, a < b \land b < c\}: a, c \in \text{Atoms}, a < c\}

if 5 \in \text{Atoms}

if 5, 7 \in \text{Atoms}

if \text{Atoms} = (\mathbb{Q}, \leq)
Hereditarily definable set

5
\{a: a \in \text{Atoms}\} \quad \text{if } 5 \in \text{Atoms}

\{a: a \in \text{Atoms}, a \neq 7 \land a \neq 5\} \quad \text{if } 5,7 \in \text{Atoms}

\{\{a:a \in \text{Atoms}, a \neq b\}: b \in \text{Atoms}\} \quad \text{if } \text{Atoms} = (\mathbb{Q}, \leq)

\{\{b: b \in \text{Atoms}, a < b \land b < c\}: a,c \in \text{Atoms}, a < c\}

\text{Examples}

\text{Syntax}

\text{hdef ::= variable | parameter from Atoms}

\{ hdef : variable,\ldots, variable \in \text{Atoms}, \text{ first order formula} \}
\text{in language of Atoms, with parameters}

\text{hdef} \cup \text{hdef}
Hereditarily definable set

5
{a: a ∈ Atoms} if 5∈Atoms

{a: a ∈ Atoms, a≠7 ∧ a≠5} if 5,7∈Atoms

{{a:a ∈ Atoms, a≠b}: b ∈ Atoms}

{{b: b ∈ Atoms, a<b ∧ b<c}: a,c ∈ Atoms, a<c} if Atoms = (Q,≤)

\{x,y\} ^{\text{def}} = \{x\} ∪ \{y\} \quad (x,y) ^{\text{def}} = \{x, \{x,y\}\}

Syntax

tdef ::= variable | parameter from Atoms

| \{ tdef : variable,...,variable ∈ Atoms, first order formula \} in language of Atoms, with parameters

| tdef ∪ tdef
Hereditarily definable sets have finite descriptions

e.g. \[\{a : a \in \text{Atoms}, a \neq b\} : b \in \text{Atoms}\]

→ can be input and processed by algorithms
Hereditarily definable sets have finite descriptions

e.g. \( \{ \{a:a \in \text{Atoms}, a\neq b\}: b \in \text{Atoms}\} \)

\( \rightarrow \) can be input and processed by algorithms

equality of sets is decidable \( \iff \) the theory of Atoms is decidable
Hereditarily definable $X$

graphs  automata  Turing machines
graphs
graphs

A pair \((V,E)\) of hereditarily definable sets with \(E \subseteq \binom{V}{2}\)
infinite clique

vertices: \{ a : a \in \text{Atoms} \}
edges: \{\{a, b\} : a, b \in \text{Atoms}, a \neq b\}
Johnson graph

vertices: \( \{ \{a, b\} : a, b \in \text{Atoms}, a \neq b \} \)
edges: \( \{ \{\{a, b\}, \{b, c\}\} : a, b, c \in \text{Atoms}, a \neq b \wedge b \neq c \wedge a \neq c \} \)
some other graph

vertices: \{(a,b) : a,b \in \text{Atoms}, a \neq b \}\nedges: \{\{(a,b),(b,c)\} : a,b,c \in \text{Atoms}, a \neq b \land b \neq c \land a \neq c\}
graphs

A pair \((V,E)\) of hereditarily definable sets with \(E \subseteq \binom{V}{2}\)

- **infinite clique**
  - vertices: \(\{a : a \in \text{Atoms}\}\)
  - edges: \(\{(a,b) : a, b \in \text{Atoms}, a \neq b\}\)

- **Johnson graph**
  - vertices: \(\{(a,b) : a, b \in \text{Atoms}, a \neq b\}\)
  - edges: \(\{\{(a,b),(b,c)\) : a, b, c \in \text{Atoms}, a \neq b \neq c, a \neq c\}\)

- **some other graph**
  - vertices: \(\{(a,b) : a, b \in \text{Atoms}, a \neq b\}\)
  - edges: \(\{\{(a,b),(b,c)\) : a, b, c \in \text{Atoms}, a \neq b \neq c, a \neq c\}\)
graphs

A pair $(V,E)$ of hereditarily definable sets with $E \subseteq \binom{V}{2}$

infinite clique
- vertices: $\{a : a \in \text{Atoms}\}$
- edges: $\{(a,b) : a,b \in \text{Atoms}, a \neq b\}$

Johnson graph
- vertices: $\{(a,b) : a,b \in \text{Atoms}, a \neq b\}$
- edges: $\{\{(a,b),(b,c) : a,b,c \in \text{Atoms}, a \neq b \lor b \neq c \neq a\}$

some other graph
- vertices: $\{(a,b) : a,b \in \text{Atoms}, a \neq b\}$
- edges: $\{\{(a,b),(b,c) : a,b,c \in \text{Atoms}, a \neq b \lor b \neq c \neq a\}$

decision problems:
- connectedness
- 3-colorability
- homomorphism
- isomorphism
- ...

graphs

A pair \((V,E)\) of hereditarily definable sets with \(E \subseteq \binom{V}{2}\)

- **infinite clique**
  - vertices: \(\{a : a \in \text{Atoms}\}\)
  - edges: \(\{(a,b) : a,b \in \text{Atoms}, a \neq b\}\)

- **Johnson graph**
  - vertices: \(\{(a,b) : a,b \in \text{Atoms}, a \neq b\}\)
  - edges: \(\{\{(a,b),(b,c)\} : a,b,c \in \text{Atoms}, a \neq b \neq c \neq a\}\)

- **some other graph**
  - vertices: \(\{(a,b) : a,b \in \text{Atoms}, a \neq b\}\)
  - edges: \(\{\{(a,b),(b,c)\} : a,b,c \in \text{Atoms}, a \neq b \neq c \neq a\}\)

**decision problems:**
- connectedness
- 3-colorability
- homomorphism
- isomorphism
- ...

\[\text{discussed later}\]
Hereditarily definable $X$

graphs

pair $(V, E)$ of hereditarily definable sets with $\mathbb{E}C\left(\frac{V}{2}\right)$

decision problems:
- connectedness
- 3-colorability
- homeomorphism
- isomorphism
- ...
Hereditarily definable $X$

graphs

pair $(V,E)$ of hereditarily definable sets with $E \subseteq \binom{V}{2}$

decision problems:
- connectedness
- 3-colorability
- homomorphism
- isomorphism
- ...

automata

Turing machines
automata
automata

A tuple of hereditarily definable sets

\((\text{States, Alphabet, Initial, Accepting, } \delta)\)

where \(\text{Initial, Accepting} \subseteq \text{States}\) and \(\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States}\)
Atoms are \((Q, \leq)\).

An automaton can accept sequences

\[ q_1 \ q_2 \ q_3 \ q_4 \ldots \ q_n \]

such that \(q_1 < q_2 < q_3 < q_4 < \ldots < q_n\)
Atoms are $(\mathbb{N},=)$.

An automaton can accept sequences

$$a_1 \ a_2 \ a_3 \ a_4 \ \ldots \ a_n$$

such that $a_i = a_j$ for some $i \neq j$
Atoms are \((\mathbb{N},=)\).

An automaton can accept sequences

\[
a_1 \ a_2 \ a_3 \ a_4 \ldots \ a_n
\]

such that \(a_i=a_j\) for some \(i\neq j\)

Deterministic automata \(\neq\) nondeterministic automata
Atoms are \((\mathbb{N}, \cdot)\).

An automaton can accept sequences

\[
\{a_1, a_2\} \quad \{a_3, a_4\} \quad \{a_5, a_6\} \quad \ldots \quad \{a_{2n-1}, a_{2n}\}
\]
automata

A tuple of hereditarily definable sets

\[(\text{States, Alphabet, Initial, Accepting, } \delta)\]

where \ Initial, Accepting \subseteq \text{States} \text{ and } \delta \subseteq \text{States} \times \text{Alphabet} \times \text{States}

Atoms are \(\mathbb{Q}, \leq\).

An automaton can accept sequences
\[q_1 \ q_2 \ q_3 \ q_4 \ldots \ q_n\]

such that \(q_1 < q_2 < q_3 < q_4 < \ldots < q_n\)

Atoms are \(\mathbb{N}, \prec\).

An automaton can accept sequences
\[a_1 \ a_2 \ a_3 \ a_4 \ldots a_n\]

such that \(a_m < a_{m+1}\) for some \(m\)

Atoms are \(\mathbb{N}, =\).

An automaton can accept sequences:
\[\{a_1, a_2\} \{a_3, a_4\} \{a_5, a_6\} \ldots \{a_{2n-1}, a_{2n}\}\]

deterministic automata ≠ nondeterministic automata
automata

A tuple of hereditarily definable sets

\[(\text{States}, \text{Alphabet}, \text{Initial}, \text{Accepting}, \delta)\]

where \(\text{Initial}, \text{Accepting} \subseteq \text{States}\) and \(\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States}\)

Atoms are \((Q,\leq)\).

An automaton can accept sequences

\[q_1 q_2 q_3 q_4 ... q_n\]

such that \(q_1 \leq q_2 \leq q_3 \leq q_4 \leq ... \leq q_n\)

Atoms are \((N,\rightarrow)\).

An automaton can accept sequences

\[a_1 a_2 a_3 a_4 ... a_n\]

such that \(a_i \rightarrow a_j\) for some \(i < j\)

Atoms are \((N,\leq)\).

An automaton can accept sequences

\[\{a_1, a_2\} \{a_3, a_4\} \{a_5, a_6\} ...\]

deterministic automata ≠ nondeterministic automata

Can model some infinite-state systems with restricted data access
e.g. register automata (Kaminsky–Francez) etc.
automata

A tuple of hereditarily definable sets

\[(\text{States, Alphabet, Initial, Accepting, } \delta)\]

where Initial, Accepting \(\subseteq\) States and \(\delta \subseteq\) States×Alphabet×States

Atoms are \((Q,\delta)\).

An automaton can accept sequences

\[q_1 q_2 q_3 q_4 \ldots q_n\]

such that \(q_1 \leq q_2 \leq q_3 \leq q_4 \leq \ldots \leq q_n\)

Atoms are \((N,\leq)\).

An automaton can accept sequences

\[a_1 a_2 a_3 a_4 \ldots a_n\]

such that \(a_n \leq a_{n+1}\) for some \(i\)

Atoms are \((N,\leq)\).

An automaton can accept sequences

\[\{a_1, a_2\} \{a_3, a_4\} \{a_5, a_6\} \ldots \{a_{2k-1}, a_{2k}\}\]

deterministic automata ≠ nondeterministic automata

Can model some infinite-state systems with restricted data access

e.g. register automata (Kaminsky–Francez) etc.

computational problems:

- emptiness
- language equality
- minimization
- ...

Turing machines
Turing machines

A tuple of hereditarily definable sets

\[(\text{States}, \text{Alphabet}, \delta)\]

where \(\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States} \times \text{Alphabet} \times \{\leftarrow, \rightarrow\}\)
when $\text{Atoms} = (Q, \leq)$:

deterministic $=$ nondeterministic
when $\text{Atoms} = ((\mathbb{Z}/2\mathbb{Z})\omega, +)$:

\[ P \neq NP \]

separating language:

sequences of vectors $v_1, v_2, v_3, v_4, \ldots, v_n$

which are linearly dependent
when Atoms = (N,=):

\[ P \neq NP \]

deterministic \(\neq\) nondeterministic
when Atoms = (\(\mathbb{N},=\)):

\[ P \neq NP \]

deterministic \neq \text{nondeterministic}

separating language:

\[ a_1 a_2 a_3 \ldots a_n b_1 b_2 b_3 \ldots b_n \]

of elements of \( A = \{\{a,b,c\},\{d,e,f\} : a,b,c,d,e,f \in \mathbb{N} \} \)
when Atoms = (N,=):

$$P \neq NP$$

deterministic $\neq$ nondeterministic

separating language:

sequences  \[ a_1 a_2 a_3 \ldots a_n b_1 b_2 b_3 \ldots b_n \]

of elements of \( A = \{\{a, b, c\}, \{d, e, f\} : a, b, c, d, e, f \in \mathbb{N}\} \)

such that for some bijection \( \pi : \mathbb{N} \rightarrow \mathbb{N} \)

\[ (b_1 b_2 \ldots b_n) = (\pi(a_1)\pi(a_2)\ldots\pi(a_n)) \]
when Atoms = (N,=):

\[ P \neq NP \]

deterministic \neq \text{nondeterministic}

separating language:

sequences \[ a_1 a_2 a_3 \ldots a_n b_1 b_2 b_3 \ldots b_n \]

of elements of \[ A = \{\{\{a, b, c\}, \{d, e, f\}\}: a, b, c, d, e, f \in N\} \]

such that for some bijection \[ \pi: N \rightarrow N \]

\[ (b_1 b_2 \ldots b_n) = (\pi(a_1) \pi(a_2) \ldots \pi(a_n)) \]

related to:

- Cai-Furer-Immermann graphs
- universal algebra
- \( A \) is not homogeneous in a finite relational/functional language
Turing machines

A tuple of hereditarily definable sets

\[(\text{States}, \text{Alphabet}, \delta)\]

where \(\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States} \times \text{Alphabet} \times \{\leftarrow, \rightarrow\}\)

when \(\text{Atoms} = (\mathbb{Q}, \cdot)\):

deterministic = nondeterministic

when \(\text{Atoms} = (\mathbb{Z}/2\mathbb{Z})^\times, +)\):

\[P \neq NP\]

when \(\text{Atoms} = (\mathbb{N}, =)\):

\[P \neq NP\]

deterministic \neq\text{ nondeterministic}

separating language

sequences \(a_1a_2a_3...a_nb_1b_2b_3...b_n\)

of elements of \(\mathcal{A} = \{(a,b), (d,e), (a,c,d,e,f)\in\mathbb{N}\}\)

such that for some bijection in \(\mathcal{N} \times \mathcal{N}\)

\((b_1b_2...b_n)-(\pi(a_1)\pi(a_2)\pi(a_3))\)

related to:
• Cai-Furer-Immermann graph
• universal algebra
• \(\mathcal{A}\) is not homogeneous in a finite relational/functional language
Hereditarily definable $X$

graphs

Pair $(V, E)$ of hereditarily definable sets with $E \subseteq \binom{V}{2}$

decision problems:
- connectedness
- 3-colorability
- homomorphism
- isomorphism
...

automata

A tuple of hereditarily definable sets

$(\text{States}, \text{Alphabet}, \text{Initial}, \text{Accepting}, \delta)$

where $\text{Initial}, \text{Accepting} \subseteq \text{States}$ and $\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States}$

Can model some infinite-state systems with restricted data access e.g., register automata (Kaminsky–Francez) etc.

computational problems:
- emptiness
- language equality
- minimization
...

Turing machines

A tuple of hereditarily definable sets

$(\text{States}, \text{Alphabet}, \delta)$

where $\delta \subseteq \text{States} \times \text{Alphabet} \times \text{States} \times \text{Alphabet} \times \{\text{\textsc{s}}, \text{\textsc{e}}\}$
1. In set theory, Fraenkel and Mostowski studied sets constructed on top of an underlying set of urelementa or atoms. "Hereditarily definable sets" are a special case of these, and have finite syntax.


3. Bojańczyk et al. (2011) rediscovered these sets in the case of homogeneous atoms, in the context of automata theory and called them orbit-finite sets with atoms.

4. Up to isomorphism, a structure is hereditarily definable \(\iff\) it interprets in Atoms.
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms
* Real underlying logical structure:
  - main example in this talk:
    - $\mathbb{N}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{N}^+$, $\mathbb{N}$ - initial algebra

Hereditarily definable set
Examples:
- $\{a : a \in \text{Atoms} \land a \neq b\}$: $b \in \text{Atoms}$
- $(a, b) : a, b \in \text{Atoms}$

Syntax
- $\text{Def} \equiv \text{variable} | \text{parameter from Atoms}$
- $\text{Def} \equiv \text{Def} \land \text{Def}$

Hereditarily definable $X$

Hereditarily definable $X$

equality of sets is decidable $\iff$ the theory of Atoms is decidable

History
1. In 1971, M. Pohlers proposed a general framework for defining definability in a wide class of structures, including $\omega$-incomplete structures.
2. Definability in $\omega$-incomplete structures is a key concept in the study of definability in $\omega$-incomplete structures. The main results of this paper are obtained in this framework.
3. The paper is organized into three main sections: definability in $\omega$-complete structures, definability in $\omega$-incomplete structures, and definability in $\omega$-incomplete structures.
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms
- Real underlying logical structure:
  - Examples:
    - (N, =) - part set
    - (Q, ≤) - dense order
    - (R, +, x, 0, 1) - field of reals
    - (N, +, ≤) - Presburger arithmetic

Hereditarily definable set
- Examples:
  - \[ \{ x : a \in \text{Atoms}, b \in \text{Atoms} \} \]

Syntax
- Variables: \( v \)
- Parameters: \( p \)
- Formulas: \( \phi(x) \)
- Equations: \( x = y \)
- Theories: \( T(X) \)

Hereditarily definable \( X \)
- Graphs
- Automata
- Turing machines

Hereditarily definable sets have finite descriptions
- e.g. \( \{ x : a \in \text{Atoms}, a \neq b \} \cap \text{Atoms} \)

Equality of sets is decidable if the theory of \( \text{Atoms} \) is decidable
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms
- Real underlying logical structure:
  - Examples:
    - $\langle N, = \rangle$ - part set
    - $\langle Q, \leq \rangle$ - dense order
    - $\langle R, +, \cdot, 0, 1 \rangle$ - field of reals
    - $\langle N, +, \leq \rangle$ - Presburger arithmetic

Hereditarily definable set
- Examples:
  - $\langle 5, \mathbb{N}, +, \cdot, <, = \rangle$
  - $\langle \mathbb{R}, +, \cdot, 0, 1 \rangle$

Syntax
- $\text{label} := \text{variable} | \text{parameters from Atoms}$
- $\langle \text{label} = \text{label}, \text{variable} \in \text{Atoms} \rangle$

Hereditarily definable $X$
- $\langle x, x \in X, x \notin X \rangle$

Hereditarily definable sets have finite descriptions
- e.g. $\langle \{a \in \text{Atoms}, a+b\} : b \in \text{Atoms} \rangle$

* can be input and processed by algorithms

equality of sets is decidable $\iff$ the theory of Atoms is decidable
Computational Problems

definable sets can be presented as input to algorithms
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3-colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem
Reachability

Input: a hereditarily definable graph $G=(V,E)$, vertices $s,t \in V$

Decide: is $t$ reachable from $s$?

decidable when atoms are $(\mathbb{N},=)$ or $(\mathbb{Q},\leq)$
Reachability

**Input:** a hereditarily definable graph $G=(V,E)$, vertices $s,t \in V$

**Decide:** is $t$ reachable from $s$?

decidable when atoms are $(\mathbb{N},=)$ or $(\mathbb{Q},\leq)$

undecidable for $\text{Atoms} = (\mathbb{N},+1)$  
  ($\sim$Minsky machines)
Algorithm pseudocode

```
function reachability(V,E,s,t)

R₀ := \{s\};
n=0;
repeat
    R_{n+1} := Rₙ \cup \{w \mid (v,w)\in E, v\in Rₙ, w\in V\};
    n := n+1;
until (Rₙ=Rₙ₋₁);
return (t\in Rₙ)
```
Algorithm pseudocode

function reachability(V,E,s,t)

R₀ := {s};
n=0;
repeat
    Rₙ₊₁ := Rₙ U {w | (v,w)∈E, v∈Rₙ, w∈V};
    n := n+1;
until (Rₙ=Rₙ₋₁);
return (t∈Rₙ)

Terminates for ω-categorical Atoms
Examples: $(\mathbb{N},=), (\mathbb{Q},\leq)$, Rado graph

**Theorem** [Ryll-Nardzewski, Engeler, Svenonius]
a structure Atoms is \(\omega\)-categorical if and only if for all \(n \in \mathbb{N}\)
Examples: \((\mathbb{N},=),(\mathbb{Q},\leq), \text{Rado graph}\)

**Theorem** [Ryll-Nardzewski, Engeler, Svenonius]

A structure \(\text{Atoms}\) is \(\omega\)-categorical if and only if for all \(n \in \mathbb{N}\)

\(\text{Atoms}^n\) has finitely many orbits under the action of \(\text{Aut}(\text{Atoms})\)
initely many orbits under the action of

\[ \text{Atoms} = (Q, \leq) \]

\( Q^3 \)
initely many orbits under the action of

$$\text{Atoms} = (\mathbb{Q}, \leq)$$

$$\mathbb{Q}^3$$
infinitely many orbits under the action of

\[ \text{Atoms} = (\mathbb{Q}, \leq) \]
Definitely many orbits under the action of $\text{Atoms} = (\mathbb{Q}, \leq)$
initely many orbits under the action of

\[ \text{Atoms} = (\mathbb{Q}, \leq) \]
initely many orbits under the action of \( Q \)

\[
\text{Atoms } = (Q, \leq)
\]
Nitely many orbits under the action of

\[ \text{Atoms} = (\mathbb{Q}, \leq) \]

\[ \begin{array}{cccccccc}
q_1 < q_2 < q_3 & q_2 < q_1 < q_3 & q_3 < q_2 < q_1 & \cdots & q_1 = q_2 < q_3 & q_1 = q_3 < q_2 \\
(1,4,6) & (3.5,2,11) & (6,4,1) & (4,1,6) & (4,4,6) & (2,11,2) \\
(2,3,5,11) & (7,4,100) & (100,7,4) & (3.5,2,11) & (7,4,100) & (4,100,4) \\
(4,7,100) & (6,3,15) & (15,6,3) & (6,3,15) & (3,3,15) & (3,15,3) \\
(2,3,15) & (40,7,100) & (100,40,7) & (40,7,100) & (7,100,7) & (7,100,7) \\
(12,3,15) & (100,45,7) & (2,11,2) & (2,11,2) & (2,11,2) & (2,11,2) \\
(18,12,3) & (12,3,15) & (7,4,100) & (4,4,15) & (4,4,4) \\
(12,3,15) & & (12,3,15) & (3,3,17) \\
\end{array} \]

\[ \mathbb{Q}^3 \]
There are infinitely many orbits under the action of

\[ \text{Atoms} = (Q, \leq) \]

\[ \mathbb{Q}^3 \]
Examples: \((\mathbb{N},=)\), \((\mathbb{Q},\leq)\), Rado graph

**Theorem** [Ryll-Nardzewski, Engeler, Svenonius]

A structure Atoms is \(\omega\text{-categorical}\) if and only if for all \(n \in \mathbb{N}\),\n
\[\text{Atoms}^n \text{ has finitely many orbits under the action of } \text{Aut}(\text{Atoms})\]
Algorithm pseudocode

\textbf{function} \text{reachability}(V, E, s, t)

\begin{align*}
R_0 & := \{s\}; \\
n & := 0; \\
\text{repeat} & \\
R_{n+1} & := R_n \cup \{w \mid (v, w) \in E, \ v \in R_n, \ w \in V\}; \\
n & := n + 1; \\
\text{until} & \ (R_n = R_{n-1}); \\
\text{return} & \ (t \in R_n)
\end{align*}

\textbf{terminates for} \omega\text{-categorical} \text{ Atoms: } R_0 \subseteq R_1 \subseteq R_2 \subseteq R_3 \subseteq \ldots \subseteq V \text{ are invariant subsets of } V, \text{ which has finitely many orbits.}

\textbf{Examples:} (\mathbb{N}, =), (\mathbb{Q}, <), \text{ Rado graph}

\textbf{Theorem} [Ryll-Nardzewski, Engeler, Svenonius]

a structure \text{Atoms} is \omega\text{-categorical if and only if for all } n \in \mathbb{N}

\text{Atoms}^n \text{ has finitely many orbits under the action of } \text{Aut}(\text{Atoms})

\text{Atoms} = (\mathbb{Q}, <)

\begin{tikzpicture}
\filldraw[fill=white!50!black, rounded corners=10pt] (-3.2, -1.8) rectangle (3.2, 1.8);
\draw[thick, -latex] (-3.1, 0) -- (3.1, 0);
\draw[thick, -latex] (0, -3.1) -- (0, 3.1);
\node at (0, 0) {$V$};
\draw[thick, dotted] (-3, -3) rectangle (3, 3);
\end{tikzpicture}
Algorithm pseudocode

function reachability(V, E, s, t)

R₀ := {s};
n = 0;
repeat
    Rₙ₊₁ := Rₙ ∪ {w | (v, w) ∈ E, v ∈ Rₙ, w ∈ V};
    n := n + 1;
until (Rₙ = Rₙ₋₁);
return (t ∈ Rₙ)

minimizes for ω-categorical Atoms: \( R₀ ⊆ R₁ ⊆ R₂ ⊆ R₃ ⊆ \ldots \subseteq V \) are invariant subsets of \( V \), which has finitely many orbits.

Examples: \((\mathbb{N}, =), (\mathbb{Q}, =)\), Rado graph

Theorem [Ryll-Nardzewski, Erdős, Svenonius]
a structure Atoms is ω-categorical if and only if for all \( n ∈ \mathbb{N} \)

Atomsⁿ has finitely many orbits under the action of \( \text{Aut}(\text{Atoms}) \)
Algorithm pseudocode

function reachability(V, E, s, t)

R₀ := {s};
n = 0;
repeat
    Rₙ₊₁ := Rₙ ∪ \{w \mid (v, w) \in E, v \in Rₙ, w \in V\};
    n := n + 1;
until (Rₙ = Rₙ₋₁);
return (t \in Rₙ)

Minimizes for \(\omega\)-categorical Atoms: \(R₀ \subseteq R₁ \subseteq R₂ \subseteq R₃ \subseteq ... \subseteq V\) are invariant subsets of \(V\), which has finitely many orbits.

Examples: (\(\mathbb{N}, =\)), (\(\mathbb{Q}, \leq\)), Rado graph

Theorem [Ryll-Nardzewski, Engeler, Svenonius]
a structure Atoms is \(\omega\)-categorical if and only if for all \(n \in \mathbb{N}\)

Atomsⁿ has finitely many orbits under the action of Aut(Atoms)
Algorithm pseudocode

\begin{verbatim}
function reachability(V,E,s,t)

R_0 := \{s\};
n=0;
repeat
    R_{n+1} := R_n \cup \{w \mid (v,w) \in E, v \in R_n, w \in V\};
n := n+1;
until (R_n=R_{n-1});
return (t \in R_n)
\end{verbatim}

ominates for \textit{\omega-categorical} Atoms: \(R_0 \subseteq R_1 \subseteq R_2 \subseteq R_3 \subseteq \ldots \subseteq V\) are invariant subsets of \(V\), which has finitely many orbits.

Examples: \((\mathbb{N},=),(\mathbb{Q},<)\), Rado graph

\textbf{Theorem} [Ryll-Nardzewski, Engeler, Svenonius]
a structure Atoms is \textit{\omega-categorical} if and only if for all \(n \in \mathbb{N}\)

Atoms
\textit{\textsuperscript{n}} has finitely many orbits under the action of \(\text{Aut}(\text{Atoms})\).
Algorithm pseudocode

function reachability(V,E,s,t)

R₀ := {s};
n=0;
repeat
    Rₙ₊₁ := Rₙ U \{w | (v,w) ∈ E, v ∈ Rₙ, w ∈ V\};
n := n+1;
until (Rₙ=Rₙ₋₁);
return (t ∈ Rₙ)

Equivalences for \(ω\)-categorical Atoms: \(R₀ ⊆ R₁ ⊆ R₂ ⊆ R₃ ⊆ \ldots \subseteq V\) are invariant subsets of \(V\), which has finitely many orbits.

Examples: \((\mathbb{N},=),(\mathbb{Q},≤)\), Rado graph

Theorem [Ryll-Nardzewski, Engeler, Svenonius]
a structure Atoms is \(ω\)-categorical if and only if for all \(n ∈ \mathbb{N}\)

Atoms\(^n\) has finitely many orbits under the action of \(\text{Aut}(\text{Atoms})\)
Algorithm pseudocode

\begin{verbatim}
function reachability(V, E, s, t)

R_0 := \{s\};
n = 0;
repeat
    R_{n+1} := R_n \cup \{w \mid (v, w) \in E, v \in R_n, w \in V\};
    n := n + 1;
until (R_n = R_{n-1});
return (t \in R_n)
\end{verbatim}

Terminates for \textit{\(\omega\)-categorical} Atoms: \(R_0 \subseteq R_1 \subseteq R_2 \subseteq R_3 \subseteq \ldots \subseteq V\) are invariant subsets of \(V\), which has finitely many orbits under the action of \(\text{Aut}(\text{Atoms})\).
Reachability

Input: a hereditarily definable graph $G=(V,E)$, vertices $s, t \in V$

Decide: is $t$ reachable from $s$?

decidable when atoms are $(\mathbb{N},=)$ or $(\mathbb{Q},\leq)$

undecidable for Atoms = $(\mathbb{N},+1)$ (Minsky machines)

Algorithm pseudocode

function reachability($V, E, s, t$)

$R_0 := \{s\}$;
$n := 0$;
repeat
  $R_{n+1} := R_n \cup \{w \mid (v,w) \in E, \forall e \in R_n, \exists \forall w \in V\}$;
  $n := n + 1$;
until ($R_n = R_{n-1}$);
return ($t \in R_n$)

Terminates for $\omega$-categorical Atoms: $R_0 \subseteq R_1 \subseteq R_2 \subseteq \ldots \subseteq V$ are invariant subsets of $V$, which has finitely many orbits.
Reachability

Input: a hereditarily definable graph $G=(V,E)$, vertices $s,t \in V$

Decide: is $t$ reachable from $s$?

decidable when atoms are $(\mathbb{N}=)$ or $(\mathbb{Q},\leq)$

undecidable for Atoms $= (\mathbb{N},+1)$ (~Minsky machines)

Algorithm pseudocode

function reachability($V,E,s,t$)

$R_0 := \{s\}$; $n=0$;

repeat

$R_{n+1} := R_n \cup \{w \mid (v,w) \in E, veR_n, weV\}$;

$n := n+1$;

until ($R_n=\emptyset$);

return $t \in R_n$;

Terminates for $\omega$-categorical Atoms: $R_0 \subseteq R_1 \subseteq R_2 \subseteq R_3 \subseteq \ldots \subseteq V$ are invariant subsets of $V$, which has finitely many orbits.
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3-colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

\[ \checkmark \text{Graph reachability}\]

- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3-colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3-colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
  - Graph planarity
  - Graph isomorphism
  - Graph 3-colorability
  - Solvability of systems of equations over finite field
  - Satisfiability of sets of clauses
  - Constraint Satisfaction Problems over finite template
  - Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
  - Graph planarity
  - Graph isomorphism
  - Graph 3–colorability
  - Solvability of systems of equations over finite field
  - Satisfiability of sets of clauses
  - Constraint Satisfaction Problems over finite template
  - Homomorphism problem
Input: A graph $G$ definable over Atoms = $(\mathbb{N},=)$
Decide: Is $G$ 3-colorable?
Is this problem decidable?
Example

Is the following graph 3-colorable?

\[ V = \{ (a, b) : a, b \in \text{Atoms}, a \neq b \} \]

\[ E = \{ \{(a, b), (b, c)\} : a, b, c \in \text{Atoms}, a \neq b \land b \neq c \land a \neq c \} \]
Example

Is the following graph 3-colorable?

\[ V = \{ (a,b) : a, b \in \text{Atoms, } a \neq b \} \]
\[ E = \{ ((a,b),(b,c)) : a, b, c \in \text{Atoms, } a \neq b \land b \neq c \land a \neq c \} \]
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?
**Input:** A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

**Decide:** Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]
[ Klin, Kopczyński, Ochremiak, T.15 ]
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N},=)$
Decide: Is $G$ 3-colorable?
Is this problem decidable?

Example

Is the following graph 3-colorable?

\[\forall x \in \mathbb{N} \exists y \in \mathbb{N} \neq x : x + y \in E\]

\[E = \left\{ (x,y) : x + y, x < y \right\}\]

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]
[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as $\text{Atoms}$. 
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N},=)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q},\leq)$ on a nonempty compact space has a fixpoint.
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N},=)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Example]

\[
\begin{array}{c}
\text{Example} \\
\text{Is the following graph 3-colorable?} \\
\forall \in \{a,b,c\} \in \text{Atoms}, a \\n\exists \in \{\{a,b\}, \{b,c\}, \{a,c\}\} \in \text{Atoms}, a \neq b, a \neq c, b \neq c \\
\end{array}
\]

[Example]

\[
\begin{array}{c}
\text{Example} \\
\text{Is the following graph 3-colorable?} \\
\forall \in \{a,b,c\} \in \text{Atoms}, a \\n\exists \in \{\{a,b\}, \{b,c\}, \{a,c\}\} \in \text{Atoms}, a \neq b, a \neq c, b \neq c \\
\end{array}
\]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q},\leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1,2,3\}^{V(G)}$
**Input:** A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

**Decide:** Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13] [Klin, Kopczyński, Ochremiak, T.15]

**Proof.** Instead of $(\mathbb{N}, =)$ use $(\mathbb{Q}, \leq)$ as $\text{Atoms}$.

**Theorem** (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q}, \leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1, 2, 3\}^{V(G)}$ compact
Input: A graph $G$ definable over Atoms = $(\mathbb{N},=)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as Atoms.

Theorem (Pestov). Any continuous action of Aut$(\mathbb{Q},\leq)$ on a nonempty compact space has a fixpoint.

Let $Col = \{\text{proper colorings of } G\} \subseteq \{1,2,3\}^{|V(G)|}$

closed compact
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N}, =)$ use $(\mathbb{Q}, \leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q}, \leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1, 2, 3\}^{\text{V}(G)}$.
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N}, =)$ use $(\mathbb{Q}, \leq)$ as $\text{Atoms}$.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q}, \leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1,2,3\}^{V(G)}$
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N},=)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Example

Is the following graph 3-colorable?

\[ V = \{a,b,c,d,e\} \quad \text{and} \quad E = \{\{a,b\},\{a,c\},\{a,d\}\} \]

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q},\leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1,2,3\}^{\text{V}(G)}$

By Pestov:

\text{if there is a 3-coloring of } G \text{ then there is an invariant one.}
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N}, =)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13]

[Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N}, =)$ use $(\mathbb{Q}, \leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q}, \leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1, 2, 3\}^{V(G)}$

By Pestov:

if there is a 3-coloring of $G$ then there is an invariant one.

Check 3-colorability of the finite graph $G/\text{Aut}(\mathbb{Q}, \leq)$ of orbits of $G$. ■
Input: A graph $G$ definable over $\text{Atoms} = (\mathbb{N},=)$

Decide: Is $G$ 3-colorable?

Is this problem decidable?

Proof of decidability uses Ramsey theory [Bodirsky, Pinsker, Tsankov'13] [Klin, Kopczyński, Ochremiak, T.15]

Proof. Instead of $(\mathbb{N},=)$ use $(\mathbb{Q},\leq)$ as Atoms.

Theorem (Pestov). Any continuous action of $\text{Aut}(\mathbb{Q},\leq)$ on a nonempty compact space has a fixpoint.

Let $\text{Col} = \{\text{proper colorings of } G\} \subseteq \{1,2,3\}^{\text{V}(G)}$

By Pestov:

if there is a 3-coloring of $G$ then there is an invariant one.

Check 3-colorability of the finite graph $G/\text{Aut}(\mathbb{Q},\leq)$ of orbits of $G$. ■

Also works when Atoms are a Ramsey structure (KPT theorem).
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
  - Graph planarity
  - Graph isomorphism
  - Graph 3–colorability
  - Solvability of systems of equations over finite field
  - Satisfiability of sets of clauses
  - Constraint Satisfaction Problems over finite template
  - Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
  - Graph planarity
  - Graph isomorphism
  - Graph 3–colorability
  - Solvability of systems of equations over finite field
  - Satisfiability of sets of clauses
  - Constraint Satisfaction Problems over finite template
  - Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- √ Graph reachability
- √ Deterministic automata minimisation
- √ Context-free grammar emptiness
- √ Tree/pushdown automata emptiness
  - Graph planarity
  - Graph isomorphism
- √ Graph 3–colorability
- √ Solvability of systems of equations over finite field
- √ Satisfiability of sets of clauses
- √ Constraint Satisfaction Problems over finite template
  - Homomorphism problem
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3-colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem [Bodirsky, Pinsker, Tsankov'13]
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity
- Graph isomorphism
- Graph 3–colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem [Bodirsky, Pinsker, Tsankov'13]
Computational Problems

definable sets can be presented as input to algorithms

- Graph reachability
- Deterministic automata minimisation
- Context-free grammar emptiness
- Tree/pushdown automata emptiness
- Graph planarity

? – Graph isomorphism
- Graph 3–colorability
- Solvability of systems of equations over finite field
- Satisfiability of sets of clauses
- Constraint Satisfaction Problems over finite template
- Homomorphism problem [Bodirsky, Pinsker, Tsankov'13]
Summary

1. Definable sets are a data structure for representing certain infinite sets.
Summary

1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,...
Summary

1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the *same code* as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,...

3. Model theory helps prove termination/correctness.
Summary

1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,...

3. Model theory helps prove termination/correctness.
Summary

1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the *same code* as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,…

3. Model theory helps prove termination/correctness.

Not in this talk:
- Non-$\omega$-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
\[ X = \emptyset; \]

\[
\text{for } a \text{ in Reals} \\
\quad \text{for } b \text{ in Reals} \\
\quad \quad \text{for } c \text{ in Reals} \\
\quad \quad \quad \text{for } x \text{ in Reals} \\
\quad \quad \quad \quad \text{if } (a \times x \times x + b \times x + c \times x = 0) \\
\quad \quad \quad \quad \quad X = X \cup \{(a, b, c)\}; \\
\]

\text{return } X;
X = ∅;

for a in Reals
    for b in Reals
        for c in Reals
            for x in Reals
                if (a*x*x+b*x+c*x=0)
                    X=X∪{(a,b,c)};

return X;

output: \{(a,b,c): a,b,c∈\text{Reals}, b^2-4ac≥0\}
X = ∅;

for a in Reals
  for b in Reals
    for c in Reals
      for x in Reals
        if (a*x*x+b*x+c*x=0)
          X=XU{(a,b,c)};

return X;

output: \{ (a,b,c): a,b,c \in \text{Reals}, b^2-4ac \geq 0 \}

function reachable(V,E,s,t)

R = \{ s \}
P = ∅

while (R≠P)
  P=R;
  for v in P
    for w in V
      if \{ v,w \} \in E
        R=RU{w}

return (t \in R);
1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the *same code* as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,...

3. Model theory helps prove termination/correctness.

Not in this talk:
- Non-$\omega$-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
Summary

1. Definable sets are a data structure for representing certain infinite sets.

2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,...

3. Model theory helps prove termination/correctness.

Not in this talk:
- Non-$\omega$-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
- notions of computational tractability ("PTime"), relation to CPT of Blass, Gurevich, Shelah.
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms
- Real underlying logical structure:
  \[ (\mathbb{N}, =) \] - part set
  \[ (\mathbb{Q}, \leq) \] - dense order
  \[ (\mathbb{R}, +, \times, 0, 1) \] - field of reals
  \[ (\mathbb{N}, +, \leq) \] - Presburger arithmetic

Hereditarily definable set

Examples:
- \[ \{ a, a \in \text{Atoms} \} \]
- \[ \{ a \in \text{Atoms}, a \neq \emptyset \} \]
- \[ \{ a, b \in \text{Atoms}, a \neq b \} \]

Syntax:
- \( \text{def} := \text{variable} | \text{parameter from Atoms} \)
- \( \text{def} \in \text{def} \)

Hereditarily definable sets have finite descriptions
- e.g. \( \{ \{ a, a \in \text{Atoms}, a \neq b \} : b \in \text{Atoms} \} \)
- \( * \) can be input and processed by algorithms

Summary
1. Definable sets are a data structure for representing certain infinite sets.
2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata-expression-grammar conversions,
3. Model theory helps prove termination/correctness.

Not in this talk:
- Non-\( \omega \)-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
- Notions of computational tractability ("PTime"), relation to CPT of Blass, Gurevich, Shelah.
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Example:
\( (\mathbb{N}, =) \) - part set
\( (\mathbb{Q}, \leq) \) - dense order
\( (\mathbb{R}, +, *, 0, 1) \) - field of reals
\( (\mathbb{N}, +, \leq) \) - Presburger arithmetic

Hereditarily definable sets have finite descriptions

\[ \{ (a, a \in \text{Atoms}, a + b) : b \in \text{Atoms} \} \]

* can be input and processed by algorithms

Summary
1. Definable sets are a data structure for representing certain infinite sets.
2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tape automata emptiness,
   - automata-regular-expression-grammar conversions,...
3. Model theory helps prove termination/correctness.

Note in this talk:
- Non-u-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
- notions of computational tractability ("PTime"), relation to CPT of Bläsić, Gurevich, Shelah.
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms

* Real underlying structure:

  \( (N, \geq) \) – part set
  \( (Q, \leq) \) – dense order
  \( (R, +, \times, 0, 1) \) – field of real numbers
  \( (N, +, \leq) \) – Presburger arithmetic

Hereditarily definable set

Examples:

\[
\begin{align*}
\text{if } a \in \text{Atoms} & \quad \text{if } b \in \text{Atoms} \\
\{a \in \text{Atoms} \mid a \in a\} & \quad \{b \in \text{Atoms} \mid b \in a\}
\end{align*}
\]

Syntax

\[
\begin{align*}
\text{let } x \text{ in } & \quad \text{variable} \\
\text{let } f(x) \text{ in } & \quad \text{f(x) in order formula}
\end{align*}
\]

Hereditarily definable

\( X \)

Summary

1. Definable sets are a data structure for representing certain infinite sets.
2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata–expressions–grammar conversion...
3. Model theory helps prove termination/correctness.

Not in this talk:

- Non-w-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
- Notions of computational tractability ("PTime"), relation to CPT of Blass, Gurevich, Shelah.

Thank you for your attention!
Computation with Atoms
Szymon Toruńczyk, University of Warsaw

includes work with Bojańczyk, Klin, Kopczyński, Lasota, Ochremiak,...

Atoms

- Real underlying logical structure:
  \( (\mathbb{N}, =) \) - part set
  \( (\mathbb{Q}, \leq) \) - dense order
  \( (\mathbb{R}, +, \cdot, 0, 1) \) - field of reals
  \( (\mathbb{N}, +, \leq) \) - Presburger arithmetic

Hereditarily definable set

- Examples:
  \( \emptyset \)
  \( \{a \in \text{Atoms} : a > 2\} \)
  \( \{a \in \text{Atoms} : a < 2\} \)
  \( \{b \in \text{Atoms} : a < b\} \)
  \( \{a, b \in \text{Atoms} : a < b\} \)

- Syntax:
  \( \text{Var} \)
  \( \lambda \text{Var} \) - variable
  \( \exists \text{Var} \) - quantifier
  \( \text{Atoms} \)
  \( \text{formula} \)
  \( \text{whole formula} \)

Hereditarily definable sets have finite descriptions

- e.g. \( \{a : a \in \text{Atoms}, a+b \} \)

+ can be input and processed by algorithms

Summary

1. Definable sets are a data structure for representing certain infinite sets.
2. Some classical algorithms can be lifted to definable structures using the same code as normally, e.g.,
   - automata reachability,
   - automata minimisation,
   - pushdown/tree automata emptiness,
   - automata expressions, grammar conversions,

3. Model theory helps prove termination/correctness.

Not in this talk:
- Non-w-categorical Atoms, application to timed automata
- A programming language with loops over infinite sets
- notions of computational tractability ("PTime"), relation to CPT of Blass, Gurevich, Shelah.

Thank you for your attention!