Set theory in Dedukti

Deducteam Seminar – September 2023

Claude Stolze – Thomas Traversié

Introduction

- Naive set theory is very popular among mathematicians
- ZFC is often used as a foundation for mathematics
- Set theory is also used in frameworks such as B or TLA+
- We present an encoding of B in Dedukti as well as an encoding of IZst as pointed graphs

Encoding the B theory in Dedukti

Encoding B in Dedukti

The B language uses an unusual type theory using set types. In our encoding, we use four categories of terms:

- types T, which type objects
- objects M, which are programs and data (integers, functions, records...)
- kinds K, which types predicates
- predicates, which are either 0-ary (of type Prop), or unary (of type $T \to \mathrm{Prop}$), for some type T

Syntax

$$
\begin{array}{lll}
T & ::= & b \mid T \times T \mid \mathrm{Set}(T) \mid \mathrm{struct}(l_1 : T_1, \ldots, l_n : T_n) \\
M & ::= & x \mid c(M_1, \ldots, M_n) \mid h(\lambda x : T.(P|M)) \mid \mathrm{bool}(P) \mid \{P\} \\
 & & \mid \{M_1, \ldots, M_n\} \mid [M_1, \ldots, M_n] \mid (M_1, M_2) \mid \mathrm{pr}_i(M) \\
 & & \mid \mathrm{rec}(l_1 : M_1, \ldots, l_n : M_n) \mid M.l \\
K & ::= & \mathrm{Prop} \mid T \to \mathrm{Prop} \\
P & ::= & M_1 \in M_2 \mid M_1 = M_2 \mid P\,M \mid \forall x : T, P \mid \exists x : T, P \mid P\,op\,P \\
 & & \mid \lambda x : T, P \mid \neg P \mid \mathrm{true}
\end{array}
$$

A note on second-order constructors of syntax $h(\lambda x : T.(P|M))$:

- $\%_{T_1,T_2}(\lambda x : T_1.(P|M))$ is a function that, for any x s.t. P, gives M, i.e. it encodes $\lambda x \in \{x : T_1|P\}.M$
- $\Sigma_T(\lambda x : T.(P|M))$ is the sum of all the M, for any x s.t. P, i.e.

$$\sum_{x \in \{x:T|P\}} M$$

- ...

Typing terms

$$\frac{(x : T) \in \Gamma}{\Gamma \vdash x : T}$$

$$\frac{(c : (T_1, \ldots, T_n) \to T) \in \Sigma \quad \Gamma \vdash M_1 : T_1 \quad \cdots \quad \Gamma \vdash M_n : T_n}{\Gamma \vdash c(M_1, \ldots, M_n) : T}$$

$$\frac{\Gamma \vdash P : \mathrm{Prop}}{\Gamma \vdash \mathrm{bool}(P) : \mathbb{BOOL}}$$

$$\frac{(h : (T_1 \to T_2) \to T_3) \in \Sigma \quad \Gamma, x : T_1 \vdash P : \mathrm{Prop} \quad \Gamma, x : T_1 \vdash M : T_2}{\Gamma \vdash h(\lambda x : T_1.(P|M)) : T_3}$$

$$\frac{\Gamma \vdash P : T \to \mathrm{Prop}}{\Gamma \vdash \{P\} : \mathrm{Set}(T)}$$

$$\frac{\Gamma \vdash M_1 : T \quad \cdots \quad \Gamma \vdash M_n : T}{\Gamma \vdash \{M_1, \ldots, M_n\} : \mathrm{Set}(T)}$$

$$\frac{\Gamma \vdash M : T_1 \times T_2}{\Gamma \vdash \mathrm{pr}_i(M) : T_i}$$

$$\frac{\Gamma \vdash M_1 : T \quad \cdots \quad \Gamma \vdash M_n : T}{\Gamma \vdash [M_1, \ldots, M_n] : \mathrm{Set}(\mathbb{Z} \times T)}$$

$$\frac{\Gamma \vdash M_1 : T_1 \quad \Gamma \vdash M_2 : T_2}{\Gamma \vdash (M_1, M_2) : T_1 \times T_2}$$

$$\frac{\Gamma \vdash M : \mathrm{struct}(l_1 : T_1, \ldots, l_n : T_n)}{\Gamma \vdash M.l_i : T_i}$$

$$\frac{\Gamma \vdash M_1 : T_1 \quad \cdots \quad \Gamma \vdash M_n : T_n}{\Gamma \vdash \mathrm{rec}(l_1 : M_1, \ldots, l_n : M_n) : \mathrm{struct}(l_1 : T_1, \ldots, l_n : T_n)}$$

Typing predicates

$$\frac{\Gamma \vdash M_1 : T \quad \Gamma \vdash M_2 : \mathrm{Set}(T)}{\Gamma \vdash M_1 \in M_2 : \mathrm{Prop}} \qquad \frac{\Gamma, x : T \vdash P : \mathrm{Prop}}{\Gamma \vdash \forall x : T, P : \mathrm{Prop}}$$

$$\frac{\Gamma \vdash M_1 : T \quad \Gamma \vdash M_2 : T}{\Gamma \vdash M_1 = M_2 : \mathrm{Prop}} \qquad \frac{\Gamma, x : T \vdash P : \mathrm{Prop}}{\Gamma \vdash \exists x : T, P : \mathrm{Prop}}$$

$$\frac{\Gamma \vdash P_1 : \mathrm{Prop} \quad \Gamma \vdash P_2 : \mathrm{Prop}}{\Gamma \vdash P_1 \; op \; P_2 : \mathrm{Prop}} \qquad \frac{\Gamma \vdash P : \mathrm{Prop}}{\Gamma \vdash \neg P : \mathrm{Prop}}$$

$$\frac{\Gamma, x : T \vdash P : \mathrm{Prop}}{\Gamma \vdash \lambda x : T.P : T \to \mathrm{Prop}} \qquad \frac{}{\Gamma \vdash \mathrm{true} : \mathrm{Prop}}$$

$$\frac{\Gamma \vdash P : T \to \mathrm{Prop} \quad \Gamma \vdash M : T}{\Gamma \vdash P \; M : \mathrm{Prop}}$$

First-order encoding

Second-order constructors are encoded in the first-order encoding by creating a fresh variable and a specific axiom for that variable. For instance:

$\%_{T_1,T_2}(\lambda x : T_1.(P(x)|f(x))$ is encoded by creating a fresh atomic set s as the function graph, and, assuming $P(x)$ and $f(x)$ have been encoded properly, we add the accompanying axiom:

$$\forall z : T_1 \times T_2.z \in s \Leftrightarrow \exists x : T_1.P(x) \wedge z = (x, f(x))$$

Encoding using pointed graphs

Why encoding sets using pointed graphs?

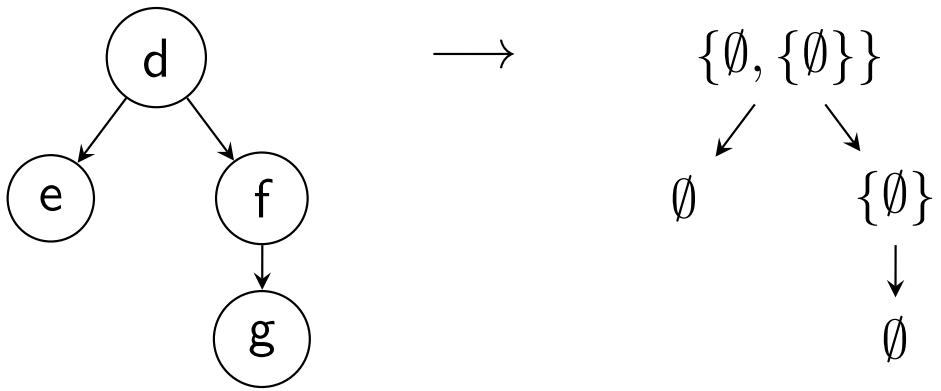
- Implementation in Dedukti
  - State each axiom
  - Define each axiom as a rewriting rule [Crabbé, 1974]
  - Encode sets using *pointed graphs* [Dowek-Miquel, 2007]

- Cut-elimination property

Theory of pointed graphs IZmod

- Pointed graph: directed graph with a root [Aczel, 1988]

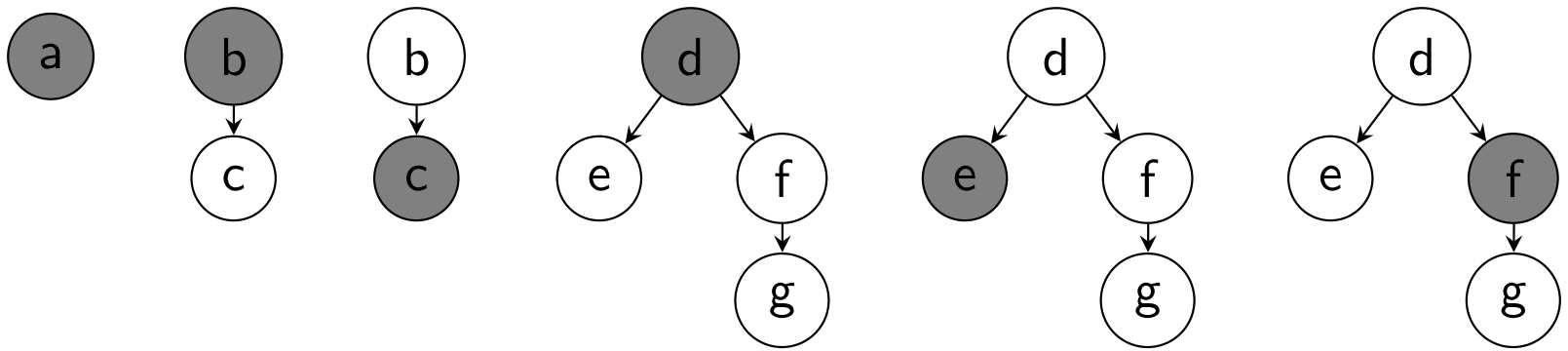
- Interpretation depends on the location of the root



Root at e or g: $\emptyset$
Root at f: $\{\emptyset\}$
Root at d: $\{\emptyset, \{\emptyset\}\}$

Theory of pointed graphs IZmod

- Examples of representation of sets by pointed graphs



$\emptyset$ $\{\emptyset\}$ $\emptyset$ $\{\emptyset, \{\emptyset\}\}$ $\emptyset$ $\{\emptyset\}$

A same graph can represent different sets depending on the root

A same set can be represented by different graphs

Theory of pointed graphs IZmod

- Definitions

$x\ \eta_a\ y$	edge from y to x in pointed graph a
a/x	changes the root of pointed graph a to be node x
$\mathrm{root}(a)$	returns the root of pointed graph a

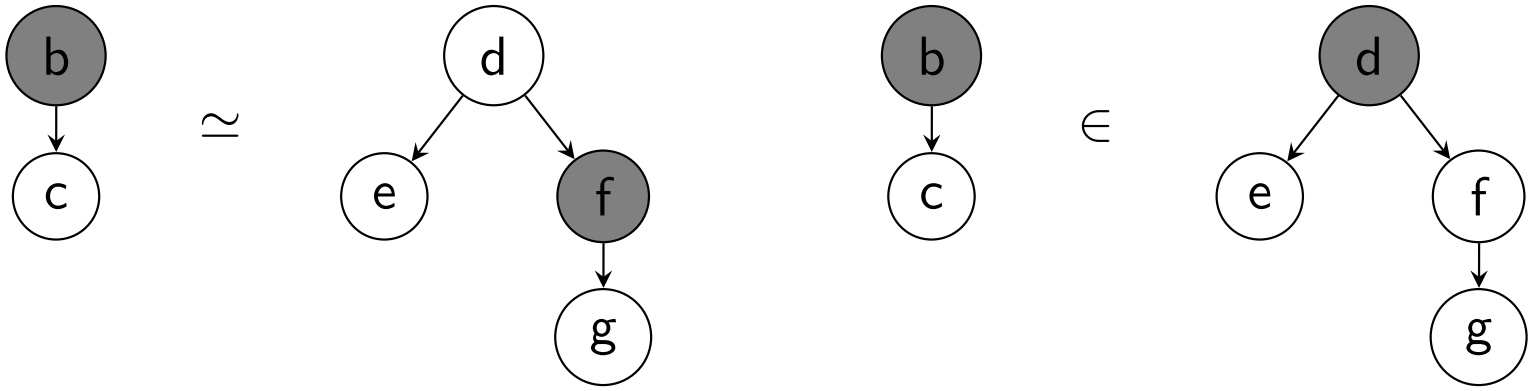
- Rewriting rules

$$x\ \eta_{a/z}\ y \longrightarrow x\ \eta_a\ y$$

$$\mathrm{root}(a/x) \longrightarrow x$$

$$(a/x)/y \longrightarrow a/y$$

Relations between pointed graphs



- Bisimilarity

$$a \simeq b \longrightarrow \exists r,\ r\ root(a)\ root(b)$$
$$\wedge\ \forall x \forall x'\ \forall y\ (x'\ \eta_a\ x\ \wedge\ r\ x\ y\ \Rightarrow\ \exists y'\ (y'\ \eta_b\ y\ \wedge\ r\ x'\ y'))$$
$$\wedge\ \forall y \forall y'\ \forall x\ (y'\ \eta_b\ y\ \wedge\ r\ x\ y\ \Rightarrow\ \exists x'\ (x'\ \eta_a\ x\ \wedge\ r\ x'\ y'))$$

- Membership relation

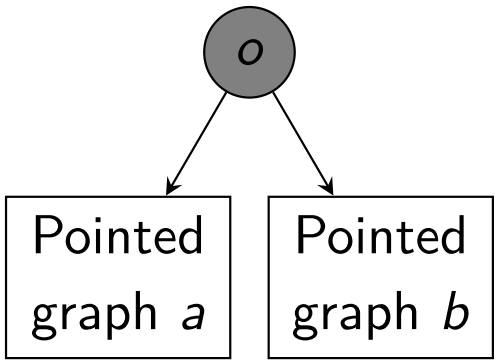
$$a \in b \longrightarrow \exists x\ (x\ \eta_b\ root(b)\ \wedge\ a \simeq (b/x))$$

Constructions

- For each axiom of set theory, we need a constructor defined by rewriting rules

- Pairing: $\forall a \forall b \exists c \forall x \ (x \in c \Leftrightarrow (x \simeq a \vee x \simeq b))$

  Creation of $c = \{a, b\}$



Nodes of $a \neq$ nodes of $b \neq o$

Constructions

- Disjoint injections i, j such that o is not in their images

$$i'(i(x)) \longrightarrow x \qquad I(i(x)) \longrightarrow \top$$
$$j'(j(x)) \longrightarrow x \qquad J(j(x)) \longrightarrow \top$$

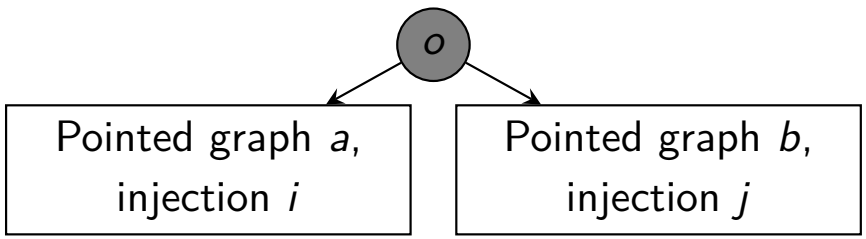
$$I(j(x)) \longrightarrow \bot \qquad I(o) \longrightarrow \bot$$
$$J(i(x)) \longrightarrow \bot \qquad J(o) \longrightarrow \bot$$

with inverses i', j' and images I, J

Constructions

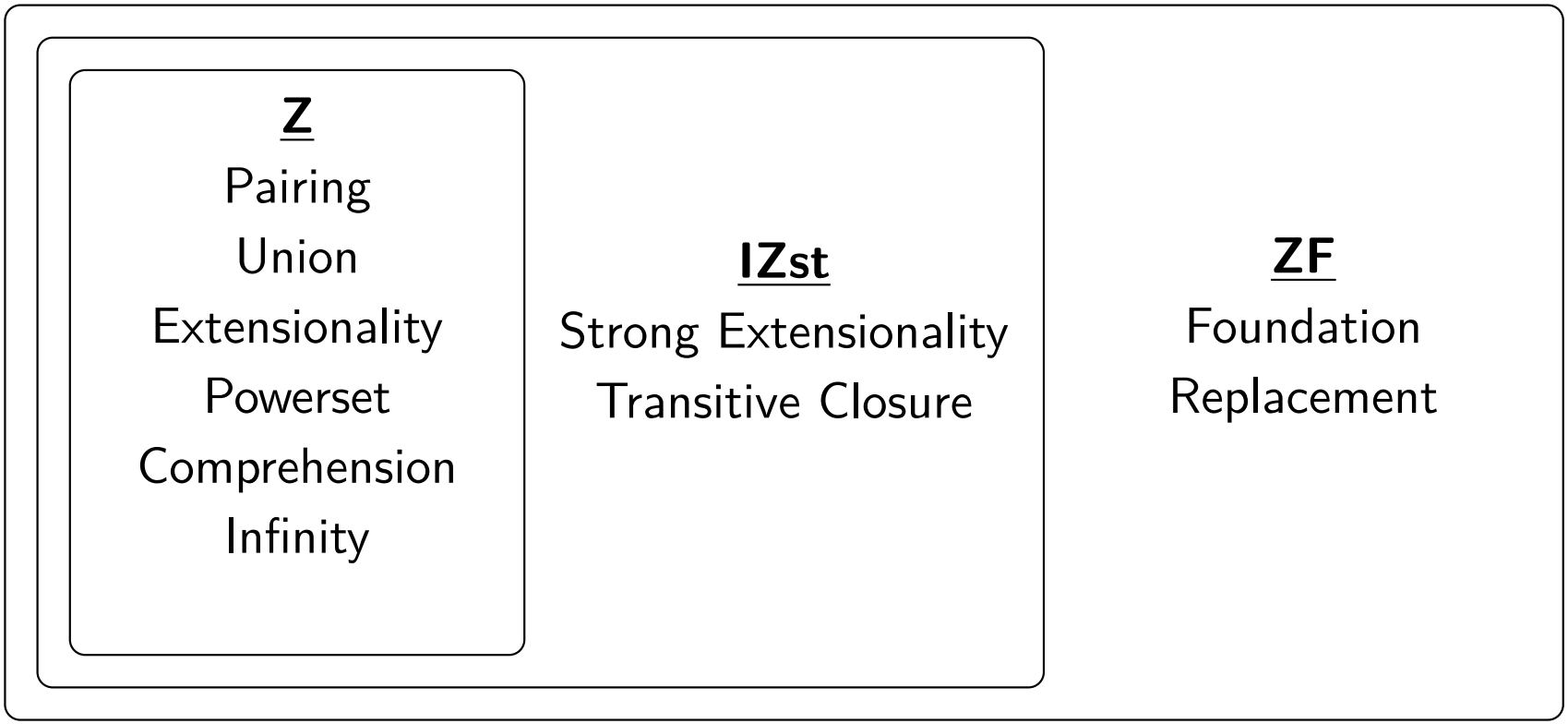
- Pairing: $\mathsf{root}(\{a, b\}) \longrightarrow o$

$$
\begin{aligned}
x\ \eta_{\{a,b\}}\ x' \longrightarrow\ & (\exists y \exists y'\ (x = i(y) \wedge x' = i(y') \wedge y\ \eta_a\ y')) \\
& \vee\ (\exists y \exists y'\ (x = j(y) \wedge x' = j(y') \wedge y\ \eta_b\ y')) \\
& \vee\ (x = i(\mathsf{root}(a)) \wedge x' = o) \\
& \vee\ (x = j(\mathsf{root}(b)) \wedge x' = o)
\end{aligned}
$$



- Similar constructions for the other axioms

Set theories



Set theories

- Strong Extensionality axiom

$$\forall x_1 ... \forall x_n \forall a \forall b \ (R(a,b) \\ \wedge \ \forall x \forall x' \forall y \ (x' \in x \wedge R(x,y) \Rightarrow \exists y' \ (y' \in y \wedge R(x',y'))) \\ \wedge \ \forall y \forall y' \forall x \ (y' \in y \wedge R(x,y) \Rightarrow \exists x' \ (x' \in x \wedge R(x',y'))) \\ \Rightarrow a \simeq b)$$

where $R(a,b)$ is a formula with free variables $x_1, ..., x_n$

- Transitive Closure axiom

$$\forall a \exists e \ (a \subseteq e \wedge \underbrace{\forall x \forall y \ (x \in y \wedge y \in e \Rightarrow x \in e)}_{\text{transitive set}})$$

IZmod and IZst

- The theory of pointed graphs IZmod validates IZst [pen and paper proof, Dowek-Miquel, 2007]

- Each axiom of IZst is a lemma in IZmod
  + Intermediary lemmas on the structure of pointed graphs
  = 53 lemmas necessary

  Pairing (lemma 43): $\forall x\ (x \in \{a, b\} \Leftrightarrow (x \simeq a \vee x \simeq b))$
  ↼ Lemma 36: $(\{a, b\}/i(\text{root}(a))) \simeq a$
  ↼ Lemma 37: $(\{a, b\}/j(\text{root}(b))) \simeq b$

Implementation

- Implementation in Dedukti: formal proofs of the 53 lemmas [Blot-Dowek-Traversié, 2022]

```
constant symbol graph : Set;
constant symbol node : Set;
```

- User POV: import the files and use the "axioms" as usual

Conclusion

Sum up

Encoding of B

Sets as types

Encoding in Dedukti and Why3

Encoding with pointed graphs

Sets as pointed graphs

Theory between Z and ZF

Conjecture: cut-elimination property (ongoing/future work)