Simple extensions
3 Simple extensions
3.1 Expressions with numbers

Expressions with numbers
3 Simple extensions
3.1 Expressions with numbers

**Numerals**

- Terms and values (blackboard).
- New type $\textbf{Nat}$ (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
3 Simple extensions
3.1 Expressions with numbers

Arithmetic expressions

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
3 Simple extensions
3.1 Expressions with numbers

Boolean expressions

• Terms and values (blackboard).
• New type \texttt{Bool} (blackboard).
• Typing rules (blackboard).
• Evaluation rules (blackboard).
Examples

• Find a derivation of the typing assertion

\[ \vdash \text{if isZero 0 then 0 else 2 + 3} : \text{Nat} \]

(blackboard).

• Find a derivation of the typing assertion \( \vdash (\forall x : \text{Bool}. x) \text{true} : \text{Bool} \) (blackboard).

• Let \( t = \forall x : \text{Bool}. \text{if } x \text{ then false else } f x \). Find a derivation of the typing assertion \( f : \text{Bool} \to \text{Bool} \vdash t : \text{Bool} \to \text{Bool} \) (oneself).
Properties

• Erasure (oneself).
• Permutation (oneself).
• Substitution (oneself).
• Uniqueness (blackboard).
• Progress (blackboard).
• Preservation (blackboard).
• Normalization (blackboard).
3 Simple extensions
3.2 Product types

Product types
3 Simple extensions
3.2 Product types

**Unit type**

- Terms and values (blackboard).
- New type `Unit` (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
Pair types

- Terms and values (blackboard).
- New types of the form $T \times U$ (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
Examples

• For $t = \text{fst}(4 - 1, \text{if } \text{true } \text{then } \text{false } \text{else } \text{false})$, find a type $T$ such that $\vdash t : T$ is derivable and an evaluation sequence reducing $t$ to a value (blackboard).

• For $t = (\forall x : \text{Nat} \times \text{Nat}. \text{snd} x)(5 - 4, 4 + 5)$, find a type $T$ such that $\vdash t : T$ is derivable and an evaluation sequence reducing $t$ to a value (oneself).
Properties

• Erasure (oneself).

• Permutation (oneself).

• Substitution (oneself).

• Uniqueness (blackboard).

• Progress (blackboard).

• Preservation (blackboard).

• Normalization (blackboard).
3 Simple extensions
3.3 Sum types
3 Simple extensions
3.3 Sum types

**Bottom type**

- New type **Zero** (blackboard).
### Variant types

- Terms and values (blackboard).
- New types of the form $T + U$ (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
Properties

- Erasure (oneself).
- Permutation (oneself).
- Substitution (oneself).
- Ambiguity (oneself).
- Progress (blackboard).
- Preservation (blackboard).
- Normalization (blackboard).
3 Simple extensions
3.4 Syntactic sugar

Syntactic sugar
Annotations

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
- Translation with correctness (blackboard).
Sequencing

• Terms (blackboard).

• Typing rules (blackboard).

• Evaluation rules (blackboard).

• Translation with correctness (blackboard).
Wildcard pattern

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
- Translation with correctness (blackboard).
Generalized tuple types

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
- Translation with correctness (blackboard).
3 Simple extensions
3.4 Syntactic sugar

Generalized variant types

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
- Translation with correctness (blackboard).
Exercises

• Exercise 11.9.1 (oneself).

• Redefine the syntax of and the rules concerning variant types in such a way that uniqueness holds (blackboard).
3 Simple extensions
3.4 Syntactic sugar

Let bindings

- Terms (blackboard).
- Typing rules (blackboard).
- Evaluation rules (blackboard).
- Translation with correctness (blackboard).
- Exercise 11.5.2 (oneself).
3 Simple extensions
3.5 Recursion

Recursion
3 Simple extensions
3.5 Recursion

Extensions

• Terms (blackboard).

• Typing rules (blackboard).

• Evaluation rules (blackboard).
Example

• For

\[
  t = \lambda f : \text{Nat} \to \text{Bool}. \lambda x : \text{Nat}. \quad \text{if } x == 0 \text{ then } \text{true} \text{ else } \neg(f(x - 1)),
\]

find a type \( T \) such that \( \vdash \text{fix}\, t : T \) is derivable and find an evaluation sequence of \textbf{let} \( e \, \text{=} \, \text{fix}\, t \text{ in } e \, 3 \) that ends with a value (blackboard).
Exercises

• Exercise 11.11.1 (oneself).

• For

\[ t = \lambda p : (\text{Nat} \rightarrow \text{Bool}) \times (\text{Nat} \rightarrow \text{Bool}). \]

\[ (\lambda x : \text{Nat}. \text{if } x = 0 \text{ then } \text{true } \text{ else } \text{snd} p(x - 1), \]

\[ \lambda x : \text{Nat}. \text{if } x = 0 \text{ then } \text{false } \text{ else } \text{fst} p(x - 1)), \]

find a type \( T \) such that \( \vdash \text{fix} t : T \) is derivable and find an evaluation sequence of \textbf{let} \( e = \text{fix} t \text{ in } \text{fst} e \ 3 \) that ends with a value (oneself).
Properties

• Erasure (oneself).
• Permutation (oneself).
• Substitution (oneself).
• Uniqueness (oneself).
• Progress (blackboard).
• Preservation (blackboard).
• Non-termination: occurrence of bottom in all types (blackboard).
3 Simple extensions
3.5 Recursion

Syntactic sugar

• Terms (blackboard).

• Typing rules (oneself).

• Evaluation rules (oneself).

• Rewrite the example definition two slides back equivalently using the new syntactic sugar and redo the exercises (oneself).