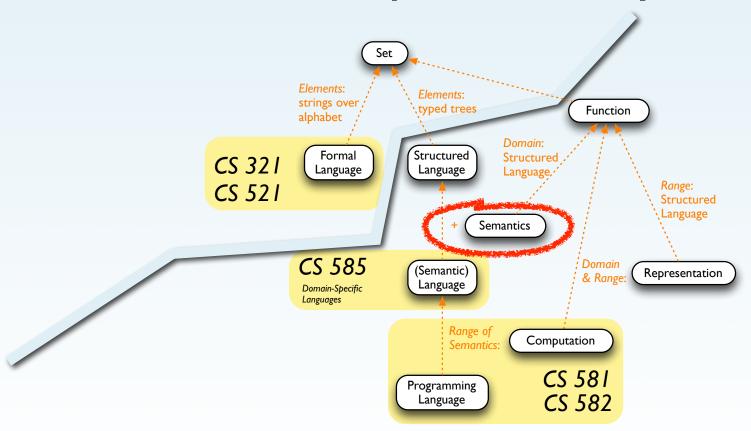
I Introduction



Before paper and scissors

PL Concept Hierarchy



Languages: The What and How

Language Concepts values

values
operations
names
functions
data types
state
typing
objects

Grammars
Rule systems

Haskell Idris

•••

Language Aspects syntax (form)
semantics (meaning)
properties
paradigm (feature sets)

Metalanguages

Language Processing

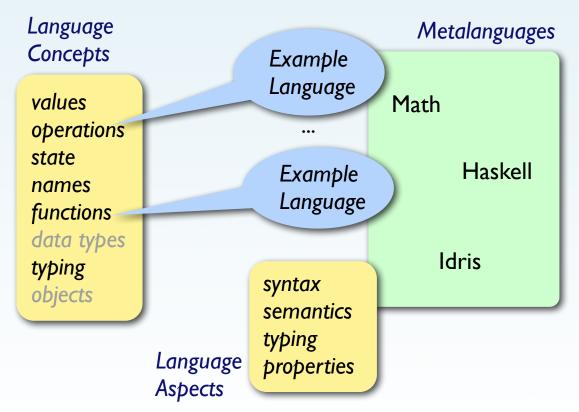
parsing type checking compiling interpreting

("Engineering", CS 480)

Programming Language Descriptions

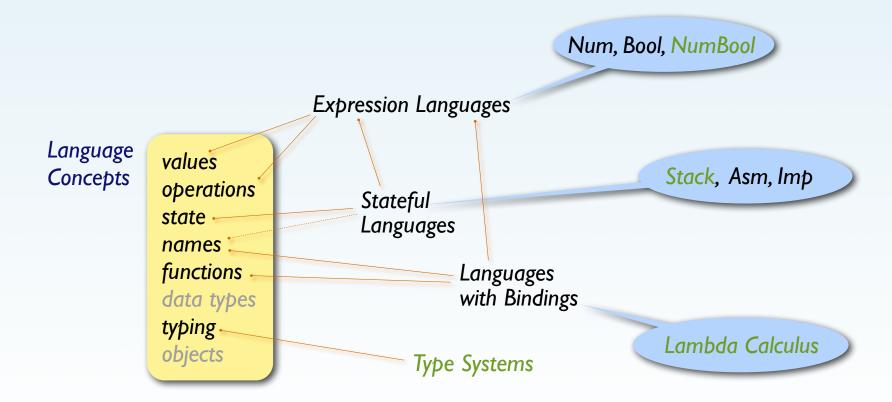
		Math	Haskell	Idris
Prog	(Abstract) Syntax	Grammar	Data Type	Data Type
Programming	Semantics			
ning	Denotation	al Function	Function	Function
	Operation	Rule System	-	Dependent Type
Language	Type System	m Rule System	Function	Dependent Type
uction	Properties Properties Properties	Theorem & Proof	_	Dep.Type & Tree

Teaching Approach

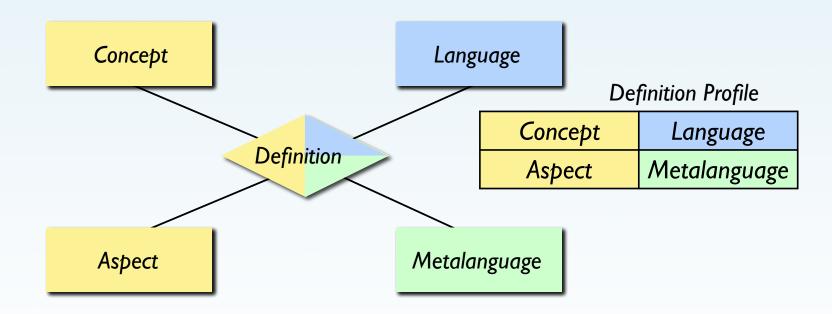


Illustrate a concept by defining its aspects as part of an example language using a metalanguage

Example Languages



Language Definitions



Some Example Definitions

```
Conditional NumBool
                     Conditional NumBool
                                                                       Syntax
                              Grammar
                      Syntax
t \in Term ::=
                                              data Term = ...
                if t then t else t
                                                             If Term Term Term
```

```
Conditional NumBool
                                  Syntax
                                            Idris
data Term : Type where ...
  If : Term → Term → Term → Term
```

```
Conditional
                                  lava
                                Grammar
                        Syntax
e \in Expr ::=
s \in Stmt ::=
               if (e) {s} [else {s}]
```

```
Conditional
                                 lava
                      Syntax
                               Haskell
data Expr = ...
data Stmt = ...
    If Expr Stmt (Maybe Stmt)
```

CS 582 • Introduction

Haskell

More Definitions

Denotational
Semantics

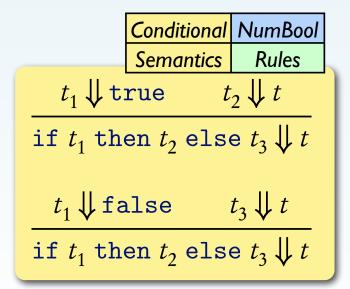
Conditional NumBool
Semantics Haskell

data Value = B Bool | I Int

sem :: Term → Value
sem (If c t e) | sem c==B True = sem t
| otherwise = sem e

Big-Step Operational Semantics Conditional NumBool Semantics Rules $t_1 \Downarrow \texttt{true} \qquad t_2 \Downarrow t$ if t_1 then t_2 else $t_3 \Downarrow t$ if $t_1 \Downarrow \texttt{then} \ t_2 \iff t_3 \Downarrow t$ if $t_1 \Downarrow \texttt{then} \ t_2 \iff t_3 \Downarrow t$

More Definitions



10

Crucial Representations

