I Introduction

Due to his grammar mistake, Wilbur found a position. It just wasn't the one he wanted.

Will work as food
I Introduction

The role of PLs in CS
How to study PLs?
Topics & Schedule
Science and Engineering

Understand & Explain
Science

Builds on
Engineering

enables

Physics
Chemistry
Theoretical CS

ME, EE, ...
CE, ...
SE, ...

Introduction
What is Computation?

**Systematic Transformation of Representation**

- **Systematic**
  - Intensional
  - Description

- **Transformation**
  - Function

- **Representation**
  - Abstraction that preserves particular features

Description given in a Language
Facilitating Computation by Machines

Computing Languages

Programming Language: Formal Description of Computation

Many users/domains + many machines ⇒ Many PLs
Impact of Programming & Language Design

Multiplier

Design a Language

Empower people to ...

Multiplier

Develop a Program

Empower people to ...

Use a Program

Faster and more reliable problem solving
I Introduction

The role of PLs in CS
How to study PLs?
Topics & Schedule
Language Diversity

Perl
Ruby
Python
Fortran
Haskell
Java
Scheme
Racket

Introduction
Exploring Languages

Comparing by programming?

Describing common concepts
PL Concept Hierarchy

- **Set**
  - Elements: strings over alphabet

- **Function**
  - Domain: Structured Language
  - Range: Structured Language
  - Range & Range: Representation

- **Semantics**
  - Domain & Range: Computation
  - Range of Semantics: Programming Language

- **Structured Language**
  - Elements: typed trees
  - Domain: Structured Language

- **Formal Language**
  - CS 321
  - CS 521

- **Domain-Specific Language**
  - CS 585

- **Programming Language**
  - CS 581
  - CS 582
Languages Landscape

**Language Concepts**
- values
- operations
- names
- functions
- data types
- state
- typing
- objects

**Grammars**
- Rule systems
- ...

**Haskell**
- Idris
- ...

**Syntax (form)**
- properties
- paradigm (feature sets)

**Metalanguages**

**Language Aspects**

**Language Processing**
- parsing
- type checking
- compiling
- interpreting

(“Engineering”, CS 480)
Programming Language Descriptions

<table>
<thead>
<tr>
<th>Math</th>
<th>Haskell</th>
<th>Idris</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Abstract) Syntax</td>
<td>Data Type</td>
<td>Data Type</td>
</tr>
<tr>
<td>Syntax</td>
<td>Function</td>
<td>Function</td>
</tr>
<tr>
<td>Semantics</td>
<td>Rule System</td>
<td>–</td>
</tr>
<tr>
<td>Denotational</td>
<td>Rule System</td>
<td>Dependent Type</td>
</tr>
<tr>
<td>Operational</td>
<td>Property</td>
<td>Dependent Type</td>
</tr>
<tr>
<td>Type System</td>
<td>Theorem &amp; Proof</td>
<td>Dep. Type &amp; Tree</td>
</tr>
</tbody>
</table>

Introduction
I Introduction

The role of PLs in CS

How to study PLs?

Topics & Schedule
Illustrate a concept by defining its aspects as part of an example language using a metalanguage.
Example Languages

Language Concepts
- values
- operations
- state
- names
- functions
- data types
- typing
- objects

Expression Languages
- Num, Bool, NumBool

Stateful Languages
- Stack, Asm, Imp

Languages with Bindings

Type Systems
- Lambda Calculus
Language Definitions

Definition Profile

<table>
<thead>
<tr>
<th>Concept</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect</td>
<td>Metalanguage</td>
</tr>
</tbody>
</table>

Introduction
Some Example Definitions

\[ t \in \text{Term} ::= \cdots \mid \text{if } t \text{ then } t \text{ else } t \]

\[ \text{data Term} = \cdots \mid \text{If Term Term Term Term} \]

\[ e \in \text{Expr} ::= \cdots \]
\[ s \in \text{Stmt} ::= \cdots \mid \text{if } (e) \{s\} \ [\text{else } \{s\}] \]

\[ \text{data Expr} = \cdots \]
\[ \text{data Stmt} = \cdots \mid \text{If Expr Stmt (Maybe Stmt)} \]
More Definitions

<table>
<thead>
<tr>
<th>Denotational Semantics</th>
<th>Conditional Semantics</th>
<th>Haskell</th>
</tr>
</thead>
<tbody>
<tr>
<td>data Value = B Bool</td>
<td>I Int</td>
<td></td>
</tr>
<tr>
<td>sem :: Term ( \rightarrow ) Value</td>
<td>sem (If c t e)</td>
<td>sem c==B True = sem t</td>
</tr>
<tr>
<td></td>
<td>otherwise</td>
<td>= sem e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditional Semantics</th>
<th>NumBool Semantics</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big-Step Operational Semantics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_1 \downarrow \text{true} )</td>
<td>( t_2 \downarrow t )</td>
<td></td>
</tr>
<tr>
<td>( \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \downarrow t )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( t_1 \downarrow \text{false} )</td>
<td>( t_3 \downarrow t )</td>
<td></td>
</tr>
<tr>
<td>( \text{if } t_1 \text{ then } t_2 \text{ else } t_3 \downarrow t )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conditional Typing</th>
<th>NumBool Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 : \text{Bool} )</td>
<td>( t_2 : T )</td>
</tr>
<tr>
<td>( t_3 : T )</td>
<td></td>
</tr>
<tr>
<td>( \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T )</td>
<td></td>
</tr>
</tbody>
</table>
More Definitions

data ('=>) : Term → Term → Type where
...
IfT :
  t1 => Tru → t2 => t →
  -----------------------------
  (If t1 Then t2 Else t3) => t

IfF :
  t1 => Fls → t3 => t →
  -----------------------------
  (If t1 Then t2 Else t3) => t
Crucial Representations

- Syntax
- Data Type
- Rules & Semantics
- Dependent Type
- Theorems

Haskell & Idris

Idris
Schedule & Grading

Tentative Lecture Syllabus

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Thu Quiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, Idris</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>Dependent Types, Proofs</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>Idris Practice</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>Inference Rules</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>Operational Semantics</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>Operational Semantics</td>
<td>Midterm</td>
</tr>
<tr>
<td>7</td>
<td>Operational Semantics</td>
<td>✓</td>
</tr>
<tr>
<td>8</td>
<td>Language Properties</td>
<td>✓</td>
</tr>
<tr>
<td>9</td>
<td>Type Systems &amp; Polymorphism</td>
<td>–</td>
</tr>
<tr>
<td>10</td>
<td>Type Inference</td>
<td>–</td>
</tr>
</tbody>
</table>

Important Dates and Times

<table>
<thead>
<tr>
<th>Quiz Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 17</td>
<td>Thu, 10:00am – 10:20am</td>
</tr>
<tr>
<td>Jan 31</td>
<td></td>
</tr>
<tr>
<td>Feb 14</td>
<td>Thu, 10:00am – 11:00am</td>
</tr>
<tr>
<td>Feb 21</td>
<td>Thu, 10:00am – 10:20am</td>
</tr>
<tr>
<td>Feb 28</td>
<td></td>
</tr>
<tr>
<td>Mar 20</td>
<td>Wed, 9:30am – 11:20am</td>
</tr>
</tbody>
</table>

Grading

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Component</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>Class participation</td>
<td>(during class and on Piazza)</td>
</tr>
<tr>
<td>20%</td>
<td>Quizzes</td>
<td></td>
</tr>
<tr>
<td>30%</td>
<td>Midterm Exam</td>
<td></td>
</tr>
<tr>
<td>40%</td>
<td>Final Exam</td>
<td></td>
</tr>
</tbody>
</table>