Once Upon a Time in CS 581 …

Abstract syntax
Denotational semantics
Types
Lambda calculus

Tools for Language Definition
Domain Analysis
Beacon of Simplicity

Introduction
Context for this Class

Design a language at least once in your life!

*Project focused* (very unlike CS 581 or CS 582)

Caution! DSLs: *evolving, unstructured area* (unlike CS 581/582)

Syntax/Tools vs. *Semantics*

Modelware vs. *Grammarware*
1 Introduction

\[ F = ma \]
\[ E = mc^2 \]

\[ 2H_2 + O_2 \rightarrow 2H_2O \]

Andante très expressif

Piano

\[ F = ma \]
\[ E = mc^2 \]

\[ 2H_2 + O_2 \rightarrow 2H_2O \]
I Introduction

What is a DSL?

How to Develop a DSL?

Course Content
PL Concept Hierarchy

Elements: strings over alphabet

Elements: typed trees

Domain: Structured Language

Range: Structured Language

Range & Range:

Domain & Range:

Computation

Programming Language

CS 585
Domain-Specific Languages

Set

Structure Language

Semantics

Function

Representation

CS 321
CS 521
What is a DSL?

<table>
<thead>
<tr>
<th>L</th>
<th>Formal Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>Structured Language</td>
</tr>
<tr>
<td>D</td>
<td>Semantic Domain</td>
</tr>
<tr>
<td>⟦ ⋅ ⟧</td>
<td>Semantic Function</td>
</tr>
</tbody>
</table>

Set... of strings
Set... of trees

Semantic Language

Turing-complete computation
Other computation
Non-computation

(General-Purpose) PL

S ⊕ ⟦ ⋅ ⟧ : S → D

S → D

Short form
Two Notions of Domain-Specificity

Methodological DSLs
- Spreadsheets
- SQL
- Venn Diagrams
- Dataflow

Topical DSLs
- Haskore
- Latex
- PFP
- QuickCheck
- Gene Translation
- Chemical Reactions
- Yacc
- Make
- Gene Translation

Computation
Structure
Application Domain
Example Domain: Chess

**Algebraic**
1. e4 e5  
2. Nf3 Nf6  
3. Bb5 a6  
4. Bxc6 dxc6

**Descriptive**
1. P-K4 P-K4  
2. N-KB3 N-QB3  
3. B-N5 P-QR3  
4. BxN QPxB

**ICCF**
1. 5254 5755  
2. 7163 2836  
3. 6125 1716  
4. 2536 4736

Many Notations ...
Example Domain: Music

Many Notations ...

Edim9
C#m7+
I Introduction

What is a DSL?

How to Develop a DSL?

Course Content
Syntax & Semantics

Many different notations for one domain

Most DSL research focuses on notation

Syntax A
Syntax B
Syntax C

Semantic Domain

S → D
3 Methodological Levels

Topical
- **Understanding**
  - Interview experts,
  - Collect examples,
  - Identify goals,
  - Study existing notations, ...

Technical
- **Modeling**
  - Syntax vs. Semantics First
- **Implementing**
  - Metalanguage
    - Haskell
  - Degree of Completeness,
    - External vs. Internal DSL,
    - Shallow vs. Deep Embedding, ...
DSL Development

Understanding

Creation

Modeling

Implementing

Interview experts, Collect examples, Identify goals, Study existing notations, ...

Elegance Expressiveness Usability

Semantics First in Haskell Prototype as a Shallow Internal DSL

Evaluation
I Introduction

What is a DSL?

How to Develop a DSL?

Course Content