

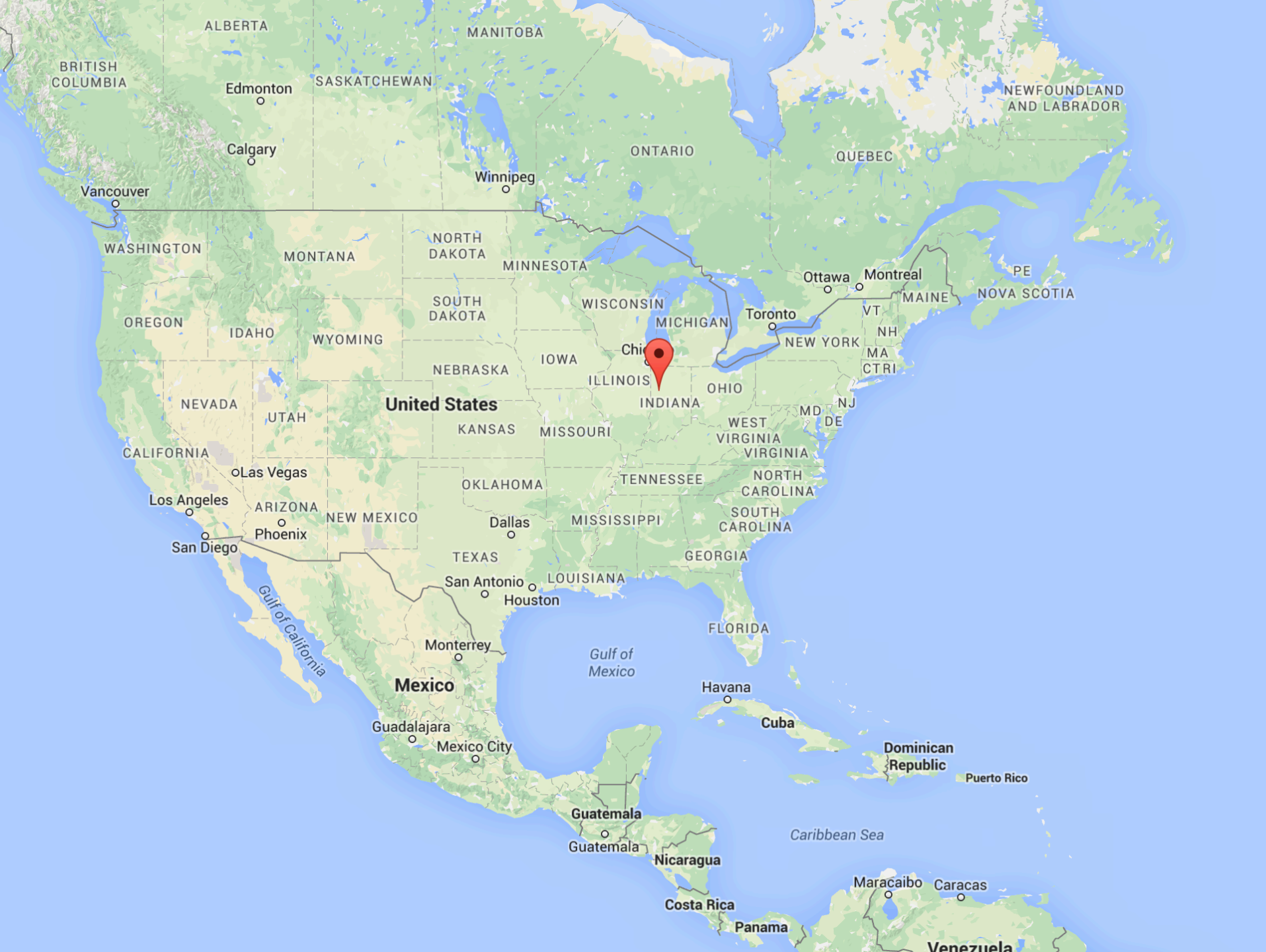
# **DSL PLATFORMS**

**DSL SUMMER SCHOOL 2015**

**TIARK ROMPF, PURDUE UNIVERSITY**



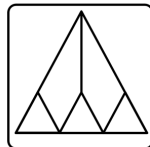
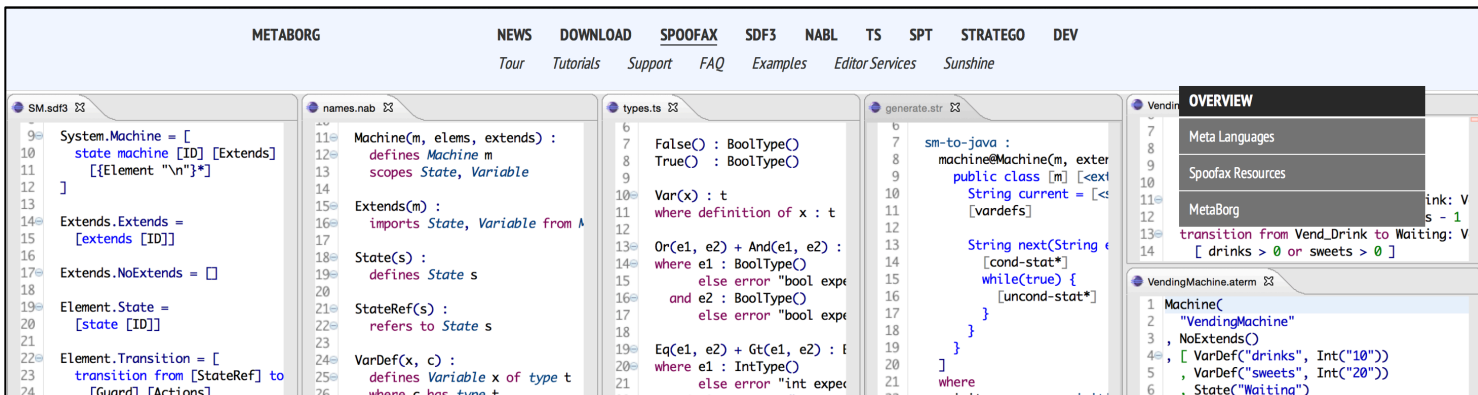




# **DSL PLATFORMS**

# SPOOFAX: A LANGUAGE WORKBENCH

Eelco Visser, Monday 11:15



[Edit on GitHub](#)

## The SpooFax Language Workbench

SpooFax is a platform for developing textual domain-specific languages with full-featured [Eclipse](#) editor plugins.

With the SpooFax language workbench, you can write the grammar of your language using the high-level SDF grammar formalism. Based on this grammar, basic editor services such as syntax highlighting and code folding are automatically provided. Using high-level descriptor languages, these services can be customized. More sophisticated services such as error marking and content completion can be specified using rewrite rules in the Stratego language.

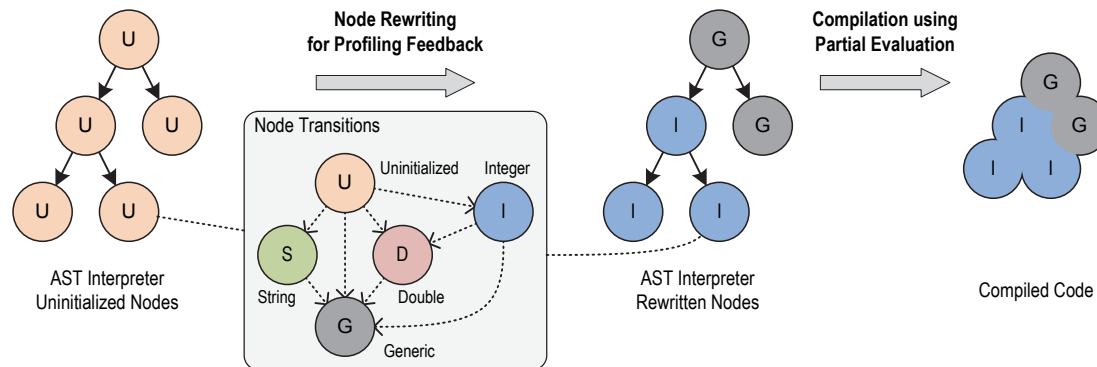
## Meta Languages

Language definitions in SpooFax are constructed using the following meta-languages:

- The [SDF3](#) syntax definition formalism
- The [NaBL](#) name binding language
- The [TS](#) type specification language
- The [Stratego](#) transformation language

# TRUFFLE: JUST IN TIME COMPILERS

Thomas Würthinger, Friday 9:15



T. Würthinger, C. Wimmer, A. Wöß, L. Stadler, G. Duboscq, C. Humer, G. Richards, D. Simon, and M. Wolczko. One VM to rule them all. In Proceedings of Onward!, 2013.

# QDSL: QUOTED DSLS

Philip Wadler, Monday 15:45

How does one integrate a Domain-Specific Language  
and a host language?

Quotation (McCarthy, 1960)

Normalisation (Gentzen, 1935)

# DSL EMBEDDING IN SCALA

Tiark Rompf, Tuesday 9:15

## **LMS** = Lightweight Modular Staging

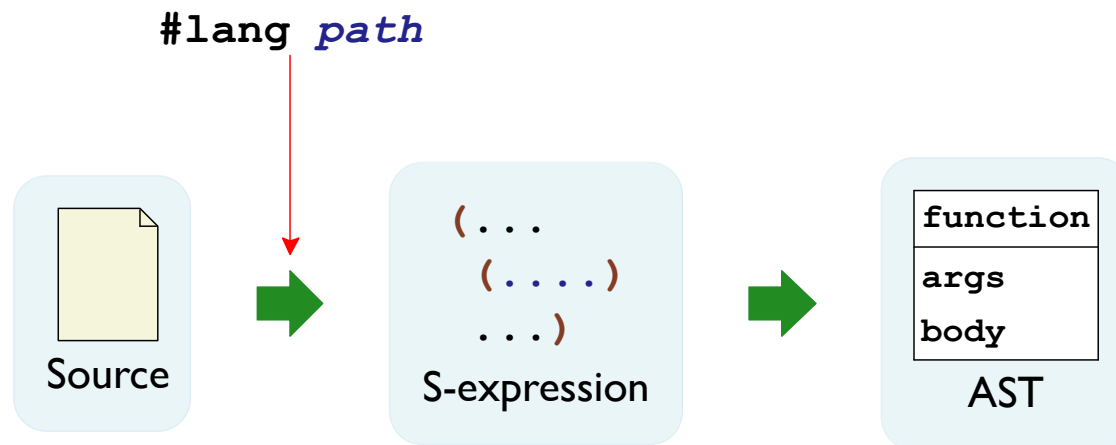
- Int, String, T
  - "execute now"
- Rep[Int], Rep[String], Rep[T]
  - "generate code to execute later"
- if (c) a else b -> \_\_ifThenElse(c,a,b)
  - "language virtualization"
- Extensible IR, transformers, loop fusion, ...
- "Batteries included"



# DSL EMBEDDING IN RACKET

Matthew Flatt, Tuesday 13:45

## DSLs via Macros in Racket



- `#lang` line determines concrete syntax parser implemented by module at *path*
- Result is a parenthesized module

# DSL EMBEDDING IN HASKELL

Ryan Newton, Wednesday 9:15

## Haskell features for DSL Construction

Front-end

- \*Template Haskell: typed and untyped splices
- \*Rebindable syntax
- \*Type-safe observable sharing
- \*Alternate “Prelude”s
- \*Type classes + overloaded literals

Middle-end

### Compiler construction technologies

- \*Scrap-your-boilerplate (SYB)
- \*Syntactic library
- \*GADT ASTs for type preservation
- \*Nanopass tooling
- \*Finally-tagless abstract syntax

Back-end

- \*Quasiquoters: typed syntax blocks
- \*Type-safe backends (e.g. LLVM)
- \*Finally-tagless for mixed shallow/deep embedded exec.

# STANZA: NEW HORIZONS

Johnathan Bachrach, Friday 13:45

## Stanza\* Essence

1

- best of scripting and production languages
  - easy to understand and powerful to use
  - gradual types -> easy parameteric types
- simple orthogonal concepts
  - functions, objects, pipes, and namespace separated
  - use concepts in unlimited ways – serendipity
  - entire language including optimizing native compiler in 20K LOC
- powerful macros for conventional syntax
  - almost entire stanza syntax written as macros
  - better DSL hosting language



\* developed by Patrick Li @ EECS Berkeley