

Implementing MyFPL with LionWeb

Meinte Boersma (DSL Consultancy)

Why MyFPL?

- Implement a FPL with projectional editing
- Showcase LionWeb
- Do some things differently from established traditions
- “Blogging excuse”
- Abandoned GenFPL:
 - “Gen<X>” got a bad rep
 - Broke the “*Rule Of Three*”



Overview of MyFPL

1. Blog series
2. Tech stack
3. Aspects of a language
4. Organization of the metamodel/structure (with MetaRel)
5. Implementation of other aspects
 1. Reducing interpreter
 2. Constraints and type system
 3. Syntax
6. Highlights
 1. Function invocation per argument identity
 2. Unknowns
 3. Faults
7. Pros and cons of projectional editing for a general FPL



Blog series

- Links:

- <https://dsl-consultancy.ghost.io/myfpl-toc/>
- <https://codeberg.org/dslmeinte/my-fpl/src/branch/main/blogs/index.adoc>

- Every week a new post



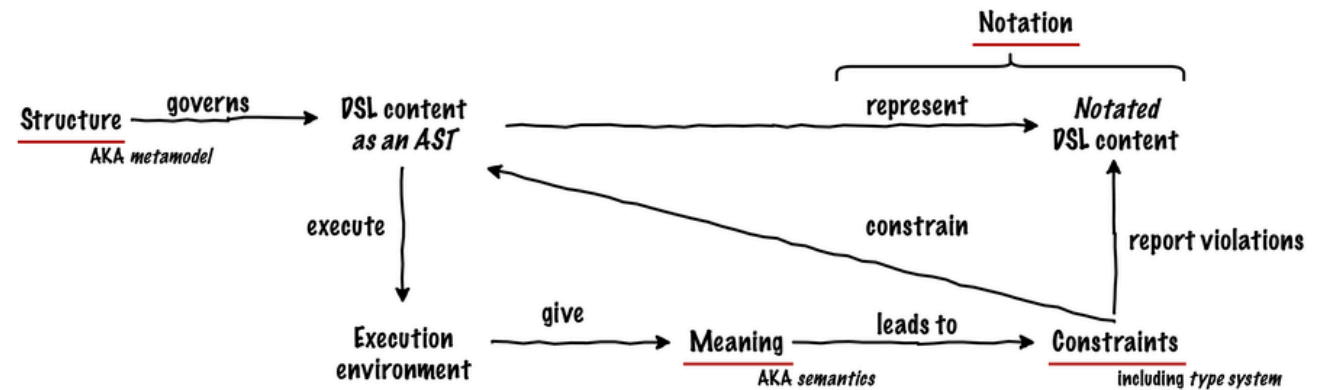
Tech stack

- TypeScript → JavaScript
- Running in the browser and on Node.js
- NPM dependencies:
 - `@lionweb/{core, class-core[-generator]}`
 - React & `{T|J}SX`



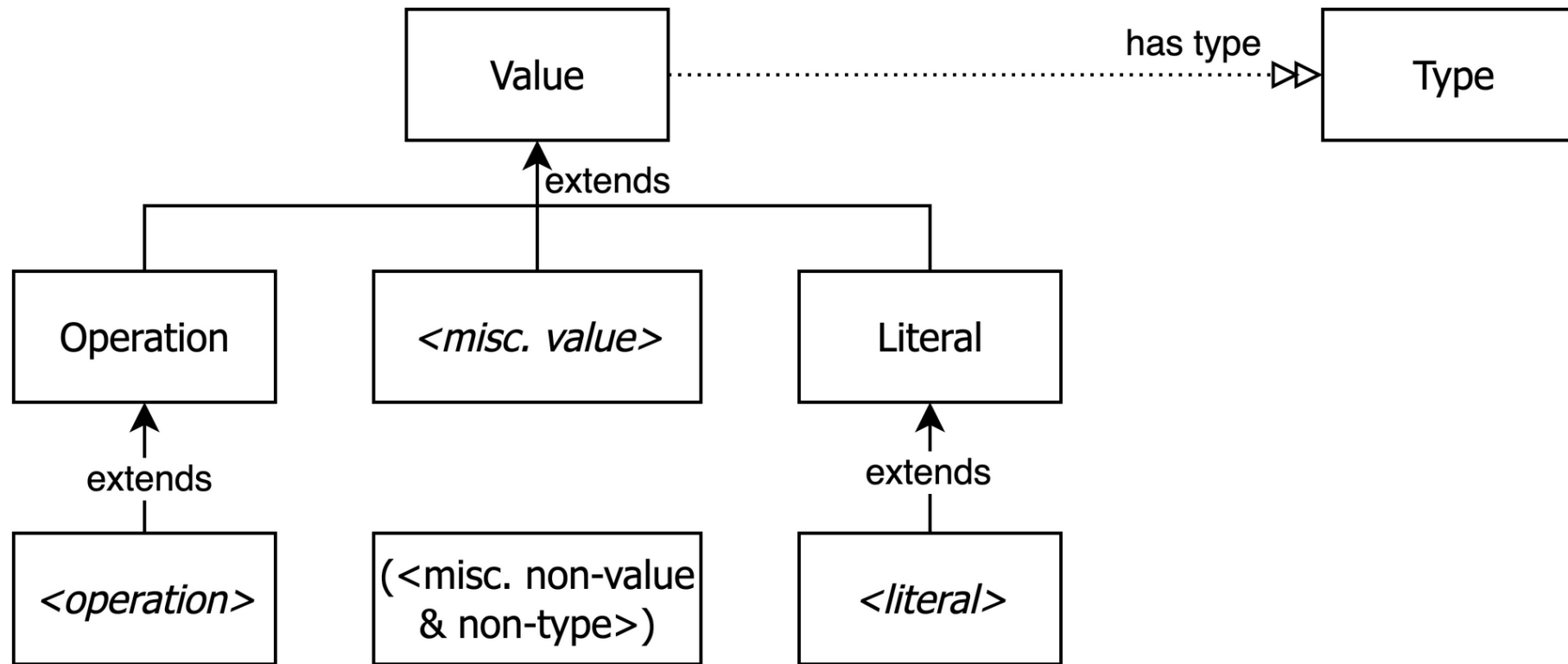
Aspects of a language

- Metamodel (AKA Structure)
- Semantics (AKA Meaning)
- Constraints (includes type system)
- Syntax (AKA Notation)

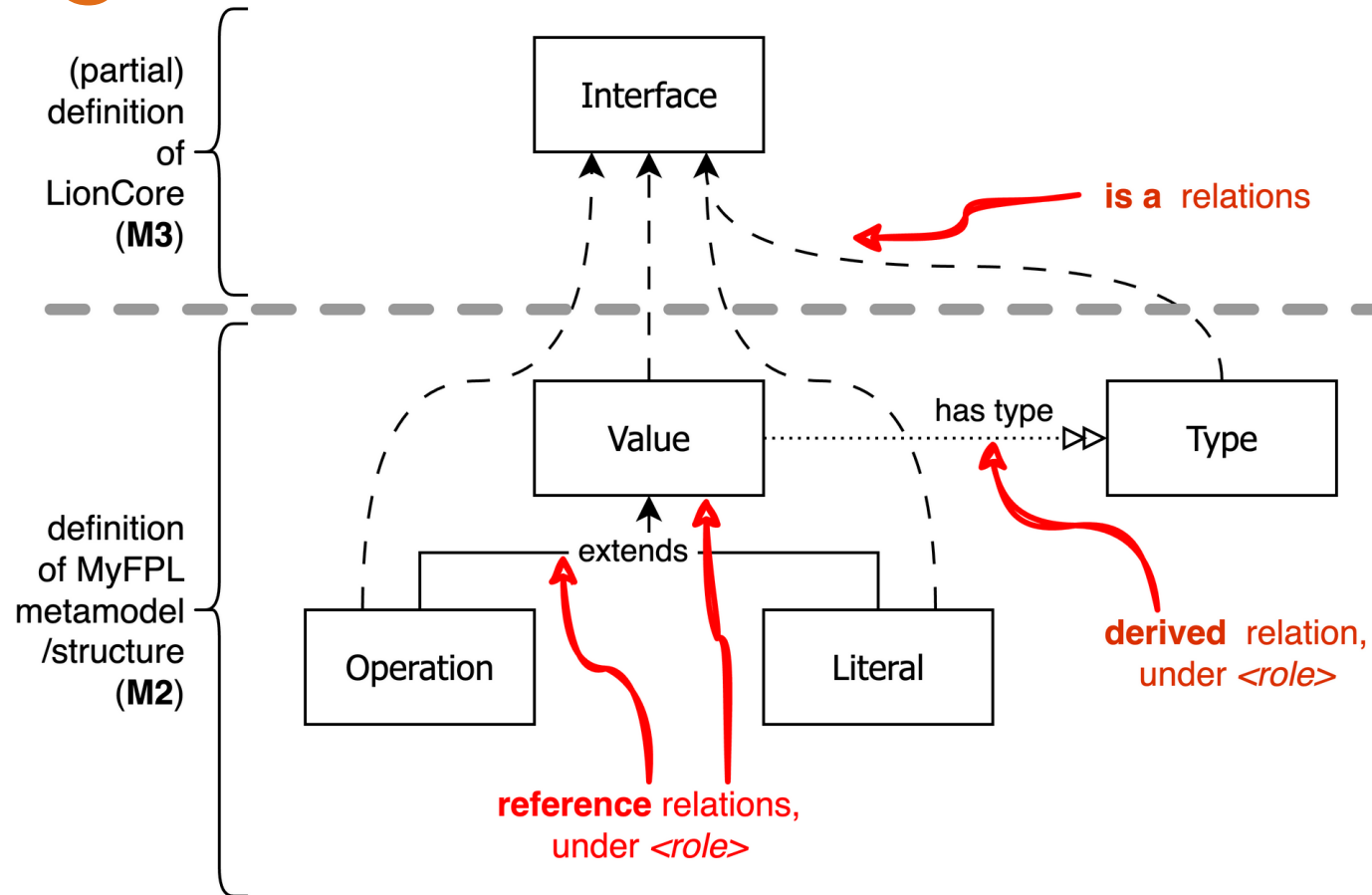


© Meinte Boersma, for Manning Publications

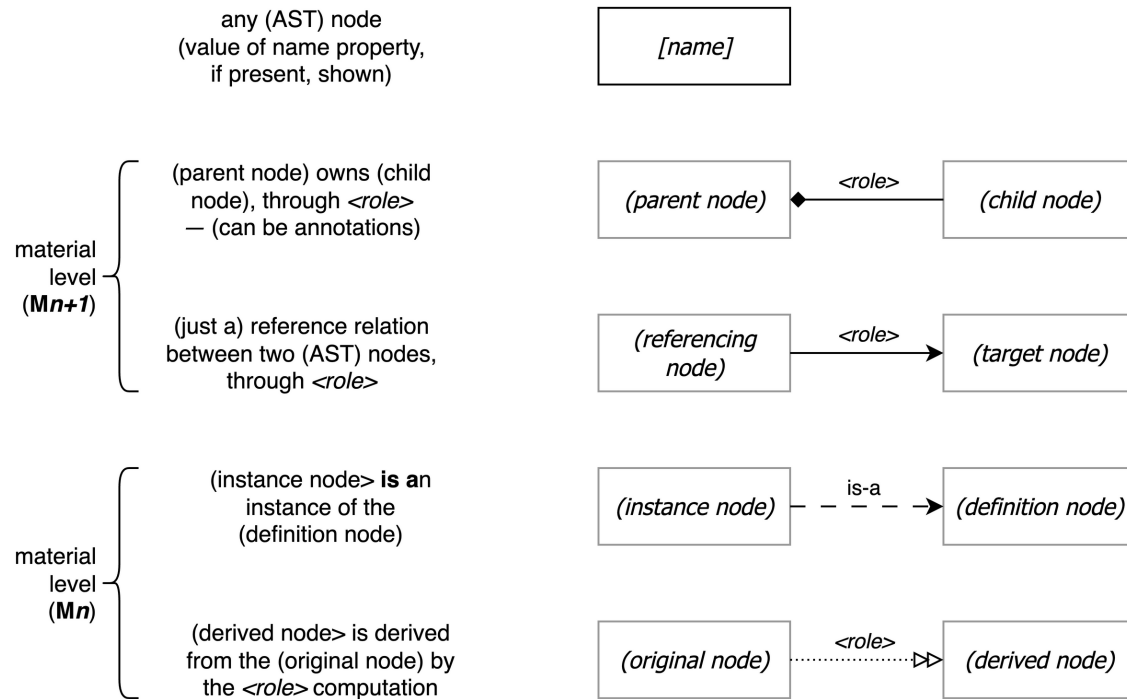
Meta-organization



Meta-organization (with *MetaRel*)

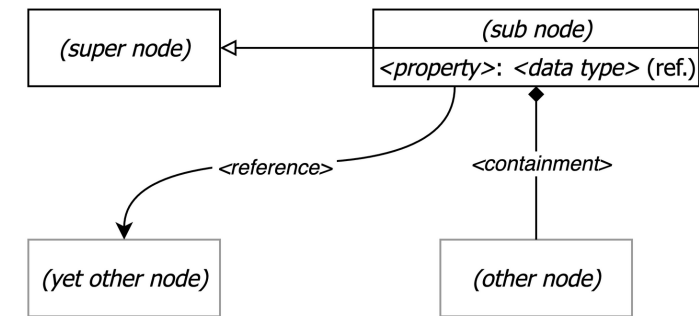


MetaRel legenda



LionCore shorthand

Classifier (sub node)
is a specialization of
Classifier (super node)



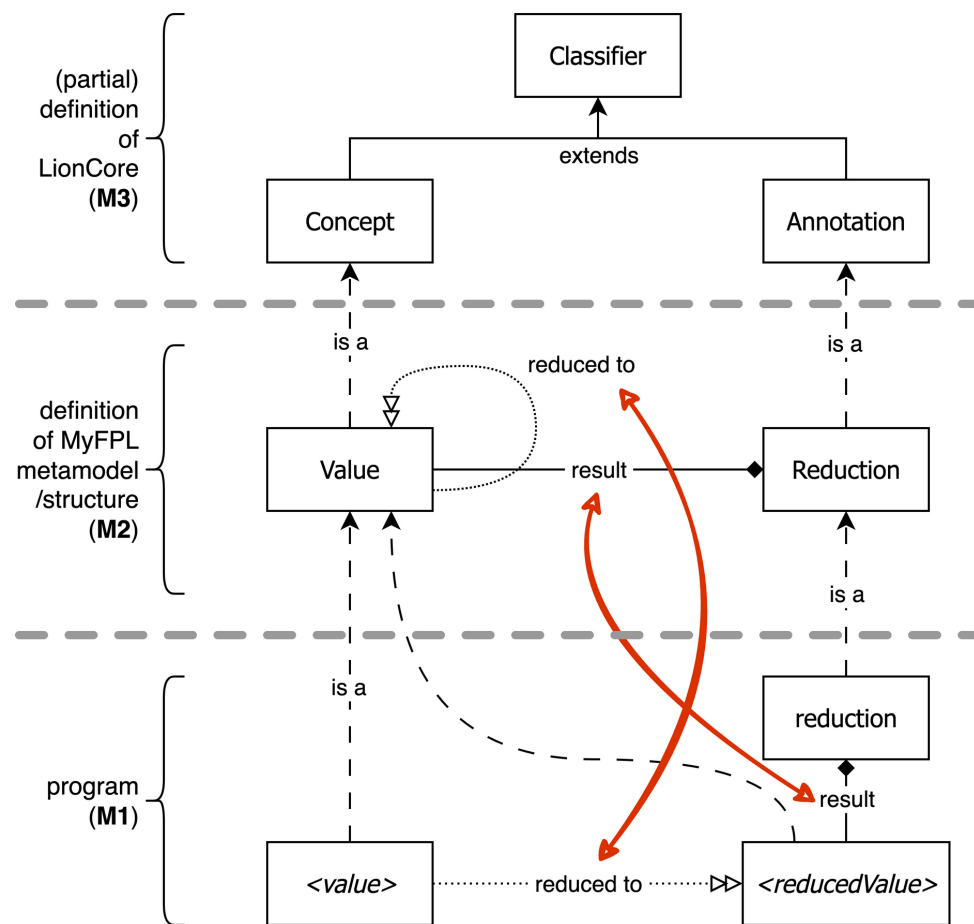
Implementation of syntax

- Syntax (AKA Notation): project to HTML — minimal UI



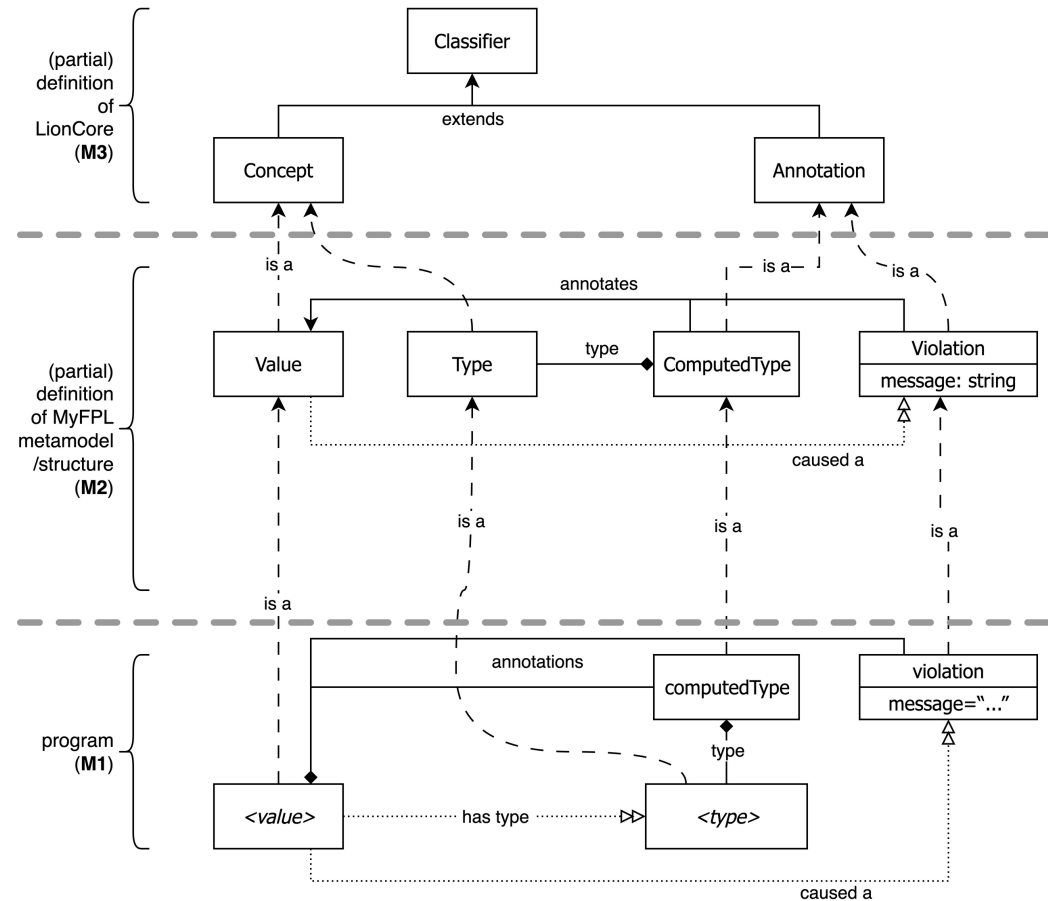
Implementation of semantics

- Semantics (AKA Meaning):
reduce Value,
add Reduction annotation
- Produces a *reduced* program
- Akin to partial evaluation

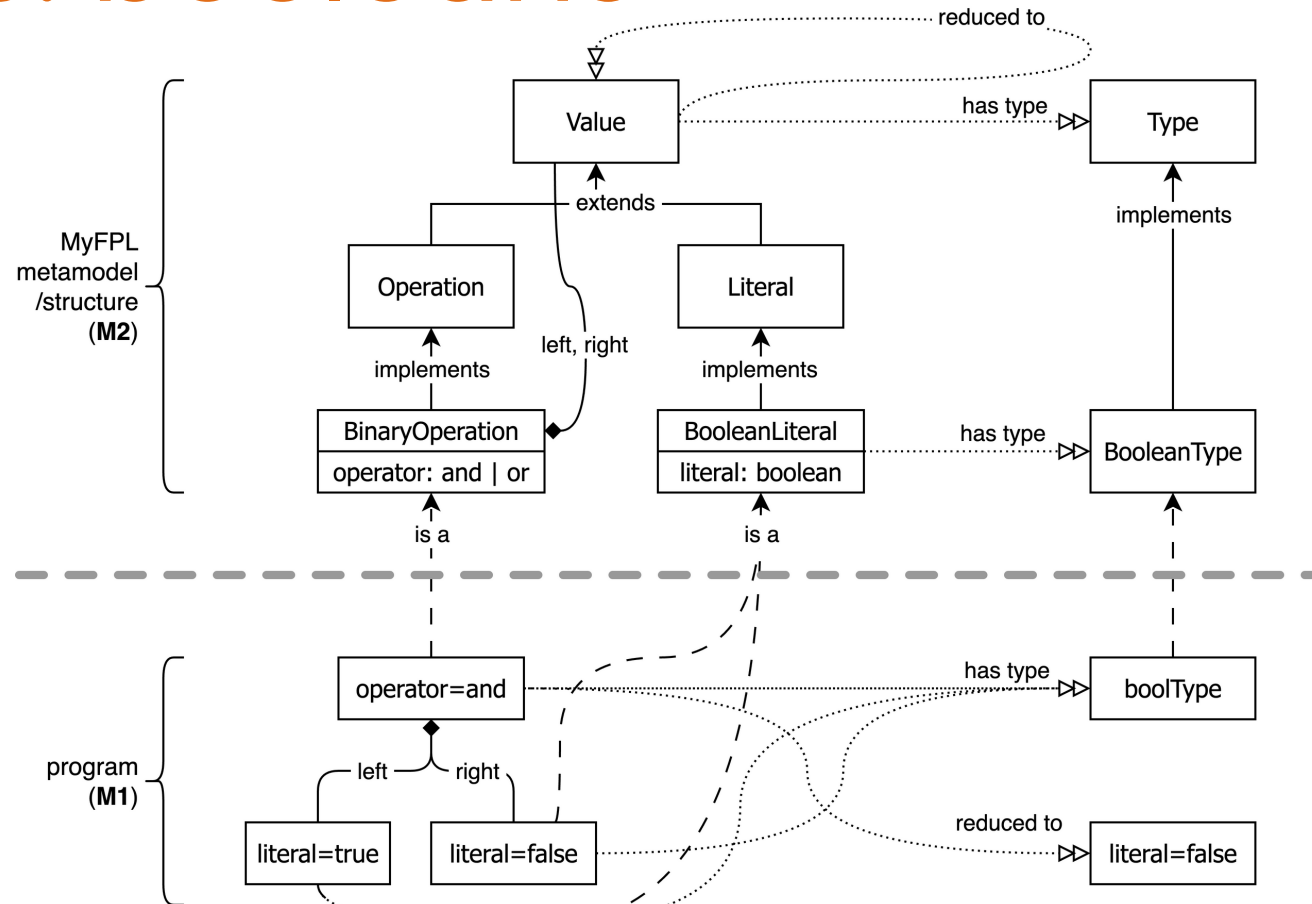


Implementation of constraints

- Constraints (with type system):
add ComputedType
and Violation annotations



Example: booleans



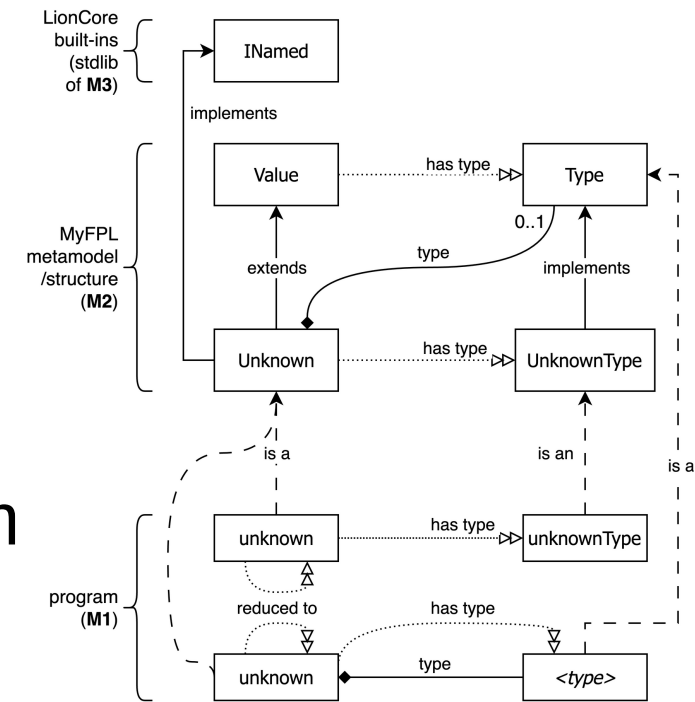
Highlights: function invocation

- Traditionally: *parameters* are **positionally** satisfied by *arguments*
- MyFPL: argument binding *refers* to a parameter
- Pros:
 - Order not important
 - Feel of *Parameter Object Pattern*
 - Not all parameters have to be satisfied
- (“Super-Schönfinkelization” was a misnomer)



Highlights: unknowns

- An (instance of) Unknown is a Value that reduces to itself
 - Type is given optionally
 - Examples:
 - unknown **or** true → true
 - false **and** unknown → false
 - true **and** unknown → unknown
 - Future work: unknowns with predicates



Highlights: faults

- No meaningful reduction $\Rightarrow$ Fault
 - Pathological example: 1 / 0
- Fault reduces to Fault
- Like an exception, but without stopping the whole program



Pros and cons of projectional editing

- Pros:

- No grammar wrangling penalties, due to
 - Limitations of parser tech used
 - Number of rules impacts performance or ambiguity
- Nice syntax, modern UI

- Cons:

- No piggybacking off a text editor $\Rightarrow$ UI is more work to implement
- No standard differ that integrates with projection
- Integration with text-based tools is not for free



Pros and cons of LionWeb

- Pros:
 - Lots of infrastructure for free
- Cons/future work:
 - No *derived models* yet — had to use annotations instead



Questions

<https://dsl-consultancy.ghost.io/myfpl-toc/>

<https://codeberg.org/dslmeinte/my-fpl/src/branch/main/blogs/index.adoc>