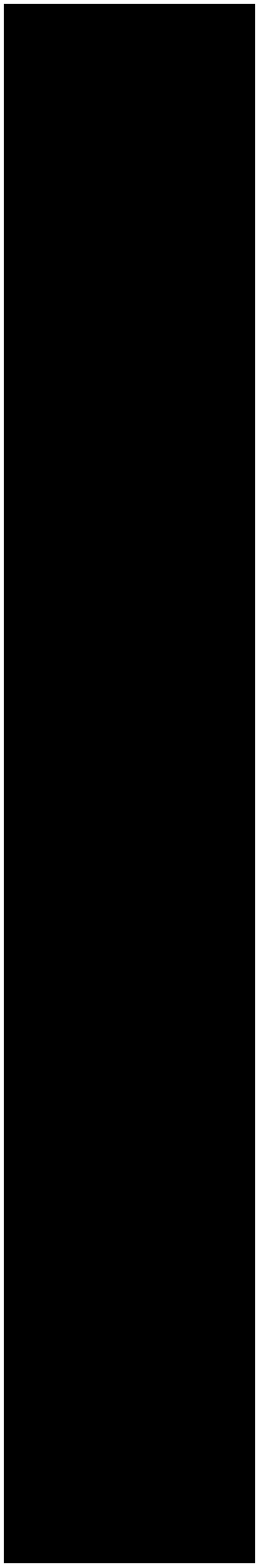

Goals

- Relationships

Introduction: Language

- *Language*, in the way that we use this term, is some formal way of communication.

- Two basic elements of a language are its *syntax* and its *semantics*.

Introduction: Reflection

- *Reflection* in a language means that we can have syntactic constructs that have syntax as its semantics.

- So, to have reflection, the first thing we need is to have some way of representing syntax: *quotation*.
  - Make the set of syntax constructs a subset of the represented values domain.
  - Should have some syntax forconstr

Programming Languages: Reflection

- The next step would be an evaluator for quoted syntax.
- This could be implemented in the language itself.

s48f*1516799 0 Td(v)Tj 9.719

Pr

Scheme: Syntax

From the Scheme Revised^5 Report:

> *Scheme*

Scheme: Reflection

- Reflection in Scheme is simple due to two important facts:
  - Syntactic structures are part of the Scheme value domain
  - The `eval` function is available to user programs.
- As said, implementing an evaluation function can be done by:
  - Using a syntactic fixed point, or,
  - Exposing the internal evaluation to the user-level: breaking the abstr

Procedural Reflection

All this is called “procedural reflection”, and is useful in the context of many other substrate systems besides a programming language:

- object systems (CLOS: a Meta Object Protocol),
- operating systems (kernel modules) and file systems,
- databases (trigger functions & meta data),
- . . . and logical systems.

Nuprl: Quotations So Far,uote f at96 0 TAitken'l: