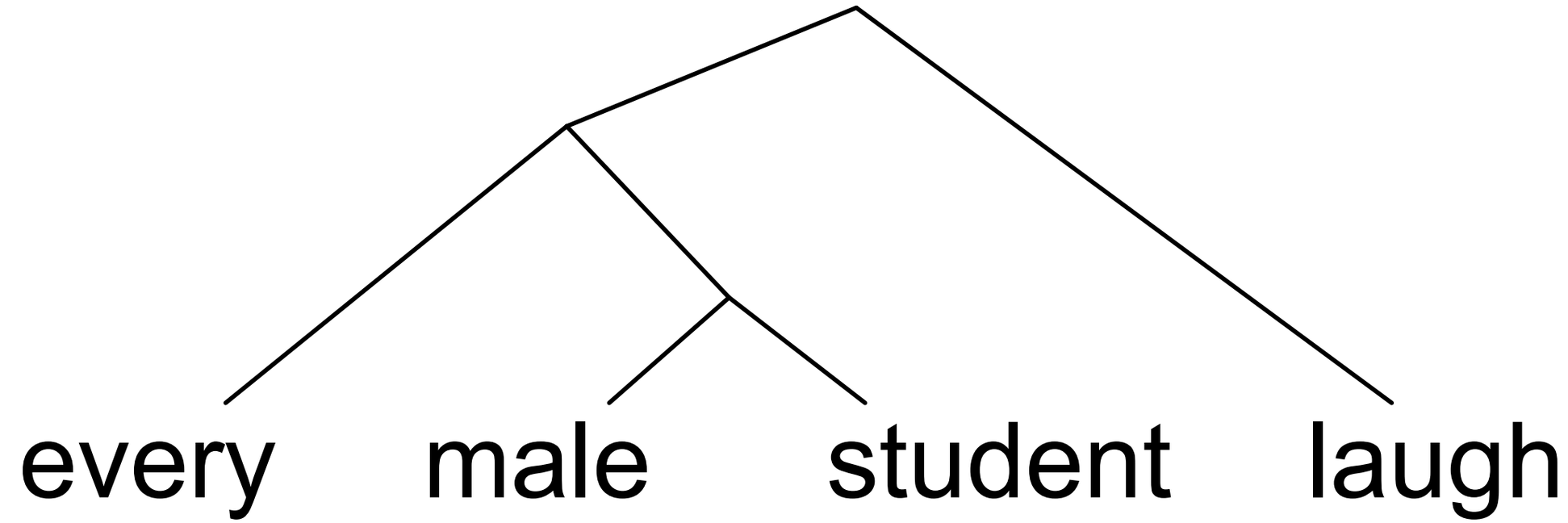
Model-Theory Semantics implemented in Clojure

@jimtyhurst
jim@tyhurst.com
github.com/jimtyhurst
Lightning Talk at Lambda Lounge
St. Louis, MO
April 7, 2011

Compositional Semantics

Every male student laughed



Compositional Semantics

Every male student laughed.

```
(((m "every")
   ((m "male") (m "student")))
      (m "laugh"))
```

Compositional Semantics

Who laughed.
((m "who") (m "laugh"))

Which student laughed.
(((m "which") (m "student")) (m "laugh"))

Points of interest

- Use sets for intuitive implementation of word meaning.
- Use functions to build phrase meaning compositionally.
- Composition entails functions that build functions.

Model

- Universe of discourse E = set of "entities"
- Meaning function m
- Properties: 1-place predicates = E → T = subset of E
- Relations: 2-place predicates = E x E → T
- Notation: Power set of E = P(E) = subsets of E
- Adjectives: P(E) → P(E)
- Quantifiers: P(E) x P(E) → T
- Individuals: P(E) → T

Sample Model

- github.com/jimtyhurst/generalized-quantifiers
  See model.clj
- universe = set of keywords
- lexicon = map of [word, denotation]
- denotations:
  - set for: noun, intransitive verb
  - 2-place relation for transitive verb
  - function for: individual, quantifier, adjective

Generalized Quantifiers

every(p,q) iff $p \subseteq q$

some(p,q) iff $p \cap q \neq \{\}$

no(p,q) iff $p \cap q = \{\}$

most(p,q) iff $|p \cap q| > |p - q|$

at-least-3(p,q) iff $|p \cap q| \geq 3$

Generalized Quantifiers

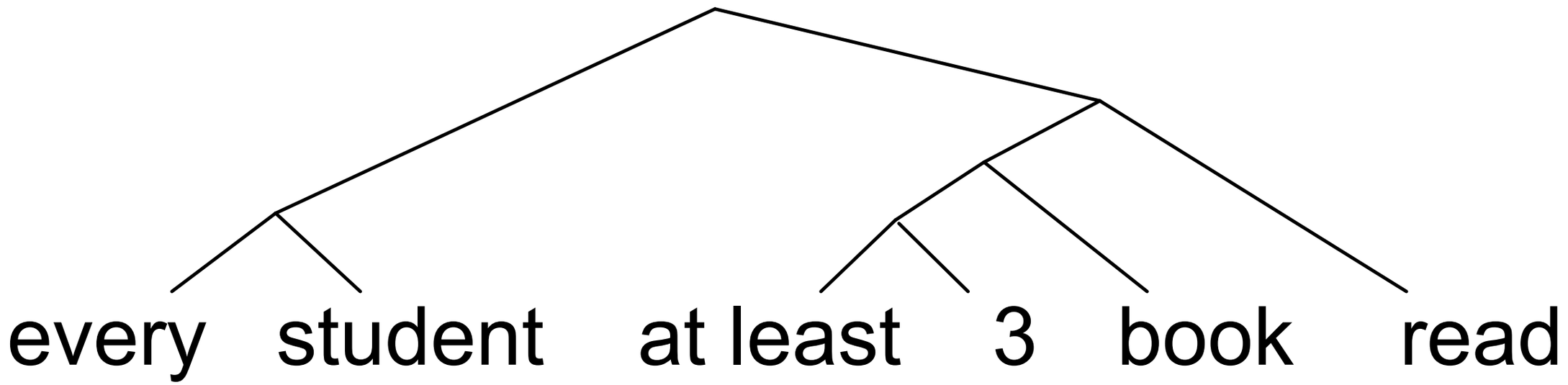
"more than 3 but less than 10"
(f and g)(p,q) iff f(p,q) $\wedge$ g(p,q)

"less than 100 or more than 200"
(f or g)(p,q) iff f(p,q) $\vee$ g(p,q)

"not more than 10"
(not f)(p,q) iff $\neg$(f(p,q))

Compositional Semantics

Every student read at least 3 books



Compositional Semantics

Every student read at least 3 books.

```
(((m "every") (m "student"))
   ((((m "at least") (m 3)) (m "book"))
      (m "read")))
```

Remaining work

(theory is well-known; I have not implemented it yet)

- Logical Form (LF) as a structural layer derived from Surface Structure:
  - captures constraints on scope ambiguity
  - enables statement of meaning function in terms of semantic structure, rather than implementation of model
- Quantifier Phrase (QP) binds argument of transitive verb:
  - enables object-wide scope ambiguity
  - enables WH-movement from object position
- Relative clauses
- Boolean combinations of syntactic units

Scope ambiguity represented in Logical Form

Every student read at least 3 books.