Model theory – A formalization of a “database”

Properties
Curried multi-argument functions

\[
\llbracket \text{respect} \rrbracket = \llbracket \lambda y. \lambda x. \text{respect}(x, y) \rrbracket = \begin{bmatrix}
  f & \mapsto & 0 \\
  f & \mapsto & 1 \\
  k & \mapsto & 1 \\
  b & \mapsto & 0 \\
  f & \mapsto & 1 \\
  k & \mapsto & 0 \\
  b & \mapsto & 0
\end{bmatrix}
\]

\[
\llbracket \lambda x. \lambda y. \text{respect}(y)(x)(b)(f) \rrbracket = 1
\]
Quiz question

- Which individuals are the red things in Palo Alto?
- Who respects kathy (k)?
Adding more complex NPs

NP: A man $\Rightarrow \exists x.\text{man}(x)$
S: A man loves Mary
    $\Rightarrow * \text{love}(\exists x.\text{man}(x),\text{mary})$

- How to fix this?
A disappointment

Our first idea for NPs with determiner didn’t work out:

“A man” \( \rightarrow \exists z. \text{man}(z) \)

“A man loves Mary” \( \rightarrow \star \text{love}(\exists z. \text{man}(z),\text{mary}) \)

But what was the idea after all?
Nothing!

\( \exists z. \text{man}(z) \) just isn’t the meaning of “a man”.

If anything, it translates the complete sentence

“There is a man”

Let’s try again, systematically…
A solution for quantifiers

What we want is:

“A man loves Mary” ~> ∃z(man(z) ∧ love(z,mary))

What we have is:

“man” ~> λy.man(y)
“loves Mary” ~> λx.love(x,mary)

How about:  ∃z(λy.man(y)(z) ∧ λx.love(x,mary)(z))

Remember: We can use variables for any kind of term.

So next:

λP(λQ.∃z(P(z) ∧ Q(z)))  ~< “A”