

Ling 5801: Problem Set 6

Due via Carmen dropbox at 11:59 PM 12/15.

1. [5 pts.] Complete the following:

$$\begin{bmatrix} 0 & 2 & 4 \\ 1 & 3 & 5 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} (- \cdot -) + (- \cdot -) + (- \cdot -) \\ (- \cdot -) + (- \cdot -) + (- \cdot -) \end{bmatrix}$$

2. [5 pts.] Complete the following:

$$\cos\left(\begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 6 \end{bmatrix}\right) = \left(\frac{-}{\sqrt{-^2 + -^2}} \cdot \frac{-}{\sqrt{-^2 + -^2}}\right) + \left(\frac{-}{\sqrt{-^2 + -^2}} \cdot \frac{-}{\sqrt{-^2 + -^2}}\right)$$

3. [10 pts.] Write a lambda calculus expression that means *Most on-duty professors are advisors*, using predicates **OnDuty**, **Prof** and **Advisor** of type $(d) \rightarrow p$, generalized quantifiers as defined in Lecture Notes 20 Section 4, and conjunctions as defined in Lecture Notes 20 Section 5. **Draw a tree** above your expression showing how the expression was derived, as shown in Lecture Notes 20 Section 4. Make sure each rule in this derivation conforms to the format of one of the lambda calculus composition rules shown in Lecture Notes 20 Section 3 and Section 5.
4. [10 pts.] Write a lambda calculus expression that means *Most professors who are on duty teach at least two courses*, using predicates **Prof**, **Course** and **OnDuty** of type $(d) \rightarrow p$, **Teach** of type $(d) \rightarrow (d) \rightarrow p$, generalized quantifiers as defined in Lecture Notes 20 Section 4, and conjunctions as defined in Lecture Notes 20 Section 5. **Draw a tree** above your expression showing how the expression was derived, as shown in Lecture Notes 20 Section 4. Make sure each rule conforms to the format of one of the lambda calculus composition rules shown in Lecture Notes 20 Section 3 and Section 5.
5. [10 pts.] PROGRAMMING: Using only the syntax and modules described in the lecture notes, write a program to read a list of entities (on lines beginning with **E**) and predicates (on lines beginning with **P**) in the following format:

```
E Kim
E Pat
E Sam
E History101
E Physics102
P Prof Kim
P Prof Pat
P Course History101
P Course Physics102
P Teach Kim History101
P Teach Pat Physics102
```

And implement the following lambda calculus expression:

$\text{Most } (\lambda_{x:d} \text{ Prof } x) (\lambda_{x:d} \text{ Some } (\lambda_{y:d} \text{ Course } y) (\lambda_{y:d} \text{ Teach } x y))$

using the Python `lambda` operator as described in Lecture Notes 19. You may write named functions for the predicates `Prof`, `Course` and `Teach`, and the quantifiers `Most` and `Some`.