Lab 5 (Properties of Relations): **25 Points Total**

- This is an **individual** assignment. Work through the following lab.
- In this lab assignment, you will work experimentally with the DrRacket language on **Properties of relations**.
- Keep in mind that in addition to this lab, there are Racket and Scheme resources linked in the syllabus if you need help.

Implement the following Racket functions:

1. **Reflexive?**
   Input: a list of pairs, **L** and a list **S**. Interpreting **L** as a binary relation over the set **S**, **Reflexive?** returns **#t** if **L** is a reflexive relation over the set **S** and **#f** otherwise.
   Examples:
   ```
   (display "Reflexive?\n")
   (Reflexive? '((a a) (b b) (c c)) '(a b c)) ---> #t
   (Reflexive? '((a a) (b b)) '(a b c)) ---> #f
   (Reflexive? '((a a) (a s) (b b) (c c)) '(a b c)) ---> #f
   (Reflexive? '() '()) ---> #t
   ```

2. **Symmetric?**
   Input: a list of pairs, **L**. Interpreting **L** as a binary relation, **Symmetric?** returns **#t** if **L** is a symmetric relation and **#f** otherwise.

   Examples:
   ```
   (display "Symmetric?\n")
   (Symmetric? '((a a) (a b) (b a) (b c) (c b))) ---> #t
   (Symmetric? '((a a) (a b) (a c) (c a))) ---> #f
   (Symmetric? '((a a) (b b))) ---> #t
   (Symmetric? '()) ---> #t
   ```

3. **Transitive?**
   Input: a list of pairs, **L**. Interpreting **L** as a binary relation, **Transitive?** returns **#t** if **L** is a transitive relation and **#f** otherwise.
   Examples:
   ```
   (display "Transitive? \n")
   (Transitive? '((a b) (b c) (a c)))    ---> #t
   (Transitive? '((a a) (b b) (c c))) ---> #t
   (Transitive? '((a b) (b a)))  ---> #f
   (Transitive? '((a b) (b a) (a a))) ---> #f
   (Transitive? '((a b) (b a) (a a) (b b))) ---> #t
   (Transitive? '())---> #t
   ```

You must use recursion, and not iteration. You may not use side-effects (e.g. set!).

The solutions will be turned in by posting a single Racket program (lab05.rkt) containing a definition of all the functions specified.