PR 2 – Logic Programming – First Term 2009/2010
Deadline: December 10th, 2009, 23:55 WIB (uploaded on SCeLE)
Filename: `LP_PR3_YourName_YourNPM.pl`
Late submission is not allowed.
Note: to trace the execution in Prolog, use `trace` command. See the documentation.

1. Define a predicate `expand(Term, Argument, NewTerm)` which expands the given `Term` with the new argument `Argument`. You can use the operator `'=..'`. Example:

   ```
   ?- expand(p(a(x),b,c), f(y), NewTerm).
   NewTerm = p(a(x),b,c,f(y))
   ```

2. Define a predicate `instantiated(Term)` which is true if each variable inside the term `Term` is already instantiated. You are **not** allowed to use the built-in predicate `ground`. Example:

   ```
   ?- instantiated(p(a(X),b,c)).
   No.

   ?- X=d, instantiated(p(a(X),b,c)).
   Yes.
   ```

3. Define a predicate `breakupterm(Term, VarList, AtomList, NumberList, StructList)` that is true when each of the following holds:

   - `VarList` contains all variables occurring in the term `Term`
   - `AtomList` contains all atoms occurring in the term `Term`
   - `NumberList` contains all numbers occurring in the term `Term`
   - `StructList` contains all structures occurring in the term `Term` which includes `Term` itself if it is a compound term.

   For example:

   ```
   ?- breakupterm(f(f(a,A), 1,c), VL, AL, NumL, SL).
   VL = [A]
   AL = [a,c]
   NumL = [1]
   SL = [f(f(a,A), 1,c), f(a,A)]
   ```

   Note: in the input, it is guaranteed that the variables representing `VarList`, `AtomList`, `NumberList`, and `StructList` does **not** occur in `Term`.

4. Define a predicate `foldl(List, Pred, Result)` which is true when `Result` is obtained by applying `Pred` successively to elements of `List`. Note that `Pred` must be a ternary predicate that is left-associative. For example:

   ```
   ?- foldl([2,3,5], plus, R).
   R = 10.
   ```

   Note that `plus` is a ternary predicate which simulates addition (and also, you have to define `plus` in order to make the previous query works). In the example, $10 = (2 + 3) + 5$, emphasiszing the left-associativity. Another example, suppose `kali/3` is a ternary predicate that simulates multiplication. Then, we have the following:

   ```
   ?- foldl([1,2,3,4,5],kali,R).
   R = 120.
   ```

   Another example, consider `minus` that simulates substraction. Then,

```
?- foldl([8,3,4],minus,R).
R = 1.
```

Notice that the result $R = 1$ because $1 = (8 - 3) - 4 \neq 8 - (3 - 4)$.