Heinrich Heine Universität Düsseldorf

Software Engineering and Programming Languages
Prof. Dr. Michael Leuschel
David Geleßus, Fabian Vu

Introduction to Logic Programming – WS 2023
Exercise Sheet 6

1 Exercises

There are unit tests for each programming exercise in the ILIAS.

The exercises will be discussed on 28th November 2023.

Exercise 1 (Lecture – Inverted Classroom)
Watch the lecture videos *11 LP Unification Algorithm*[1], *12 CNF FOL*[2], and *13 SLD FOL*[3] in the HHU Mediathek. The corresponding slides are uploaded in ILIAS: 7_SLD.pdf

The complete playlist is available at: `https://mediathek.hhu.de/playlist/691`.

Note: you have to log in with your HHU account (Uni-Kennung) to see the lecture videos!

Exercise 2 (Unification)
Find the most general unificator (mgu) σ for each of the following pairs of terms if they can be unified. Do *not* use a Prolog interpreter to solve the tasks. It is sufficient to state the results for σ.

a) `[]` and `[]`

b) `[H|T]` and `[1,2,[3]]`

c) `[X,Y]` and `[c|[[a,b]]]`

d) `2` and `1+1`

e) `r(a,X)` and `r(Y,r(a,b))`

f) `f(X,Y)` and `f(Y,f(X))`

g) `n(a,b)` and `f(X,Y)`

h) `f(a,b)` and `f(X,X)`

i) `[1,2|E]-E` and `[X,Y,F|G]-[a,b,c]`

[1] `https://mediathek.hhu.de/watch/d842296a-45a6-45f6-9ec6-4dd31b64e784`
[2] `https://mediathek.hhu.de/watch/6c8a4e75-7a11-439f-8808-0b7286df41c9`
[3] `https://mediathek.hhu.de/watch/d5ab1ee4-79c9-4659-8a8d-2a5a96089696`

Exercise 3 (Unification)
Decide for each pair of substitutions (unificators) which substitution is more general.

Use the following definition: The substitution Θ is more general than Φ if there exists σ such that $\Phi = \Theta\sigma$.

a) `{X/a}` and `{X/a,Y/a}`

b) `{X/Y}` and `{X/a,Y/a}`

c) `{X/Y,Z/a}` and `{X/a,Y/a,Z/a}`

d) `{X/1,Y/Z}` and `{X/1,Y/2,Z/3}`

e) `{X/a}` and `{X/b}`

Exercise 4 (Permutations)
Implement a predicate `mypermutation(+L, -P)` which calculates all permutations of a list `L`. It should be possible to enumerate all permutations using backtracking.

Example:

```
1 ?- mypermutation([1,2,3], X).
2  X = [1,2,3] ;
3  X = [1,3,2] ;
4  ...
```

It is not important in which order the permutations are found.

Hint: Implement a predicate which inserts an element at each position in a list when backtracking. Call this predicate for each element of the input list and the permutated tail of the list (i.e., after the recursive call).

Exercise 5
Implement a predicate `drop(+L, +N, -NL)` which drops every `N`-th element from `L` and returns the resulting list in `NL`.

Example:

```
1 ?- drop([a,b,c,d,e,f,g,h,i,k], 3, NL).
2 NL = [a,b,d,e,g,h,k]
```