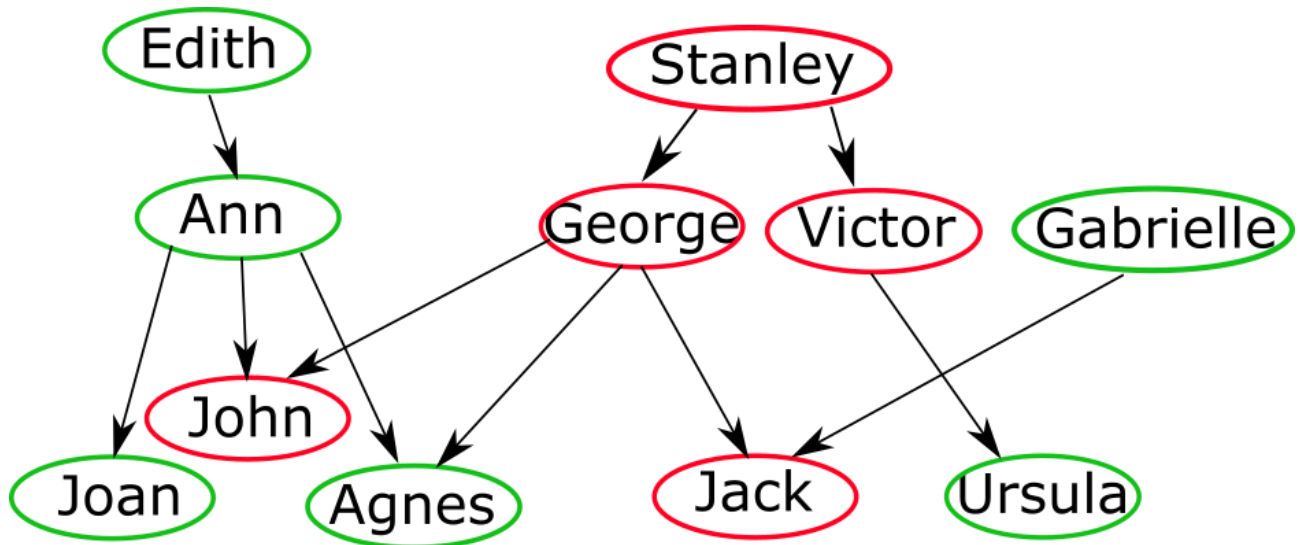


# Programming languages – ProLog

## Laboratory instruction A (2021/22)

T. Goluch

Exercises should be carried out in the given order. After completing each exercise, please present lecturer result and save the function/program code in the solution file. If any of exercises seems too difficult, you can skip them and move on to the next one.



### 1. Exercise (0.5 pts.)

**Implement the following 11 facts (for each person separately):**

- Ann, Joan, Agnes, Gabrielle, Edith and Ursula are women.
- Jack, George, John i Victor are Stanley are men.

**Execute 3 queries:**

- Is Ann a woman?
- Is Stanley a woman?
- Who's the man?

### 2. Exercise (0.5 pts.)

**Implement the following 11 facts (for each two-person relation separately):**

- Ann is the mother of Joan, Agnes and John.
- Edith is the mother of Ann.
- Gabrielle is the mother Jack
- George is the father of Jack, Agnes and John.
- Victor is the father of Ursula.
- Stanley is the father of George and Victor.

**Execute 4 queries:**

- Is Ann Joan's mother?

- Is Joan the mother of Ann?
- Who's dad Joan?
- Whose dad is George?

### 3. Exercise (0.5 pts.)

**Please implement the following 3 relations:**

- being siblings,
- being a sister, being a brother.

**Execute 7 queries:**

- Are Joan and Agnes siblings?
- Are Joan and Jack Jan siblings?
- Who's John siblings?
- Is Agnes the sister of Jack?
- Is Joan the sister of Jack?
- Is John the sister of Joan?
- What brothers does Joan have?

### 4. Exercise (0.5 pts.)

**Please implement the following relation:**

- being a partner

**Execute 3 queries:**

- Are Ann and George partners?
- Who's partner?
- Who's George partner?

### 5. Exercise (0.5 pts.)

**Please implement the following relation:**

- being a grandfather

**Execute 3 queries:**

- Is Stanley the grandfather of Jack?
- Is Gabrielle the grandfather of Agnes?
- Who's George grandfather?

### 6. Exercise (0.5 pts.)

**Implement the following 3 facts:**

- Jack is 21 years old.
- John is 18 years old.
- Ann is 17 years old.

**Please implement the following 2 relation:**

- A person 18 years or older is of legal age.
- A person of legal age may drink beer.

**Execute 3 queries:**

- Is Jack of legal age?
- Can Ann drink beer?
- Who can drink beer?

## 7. Exercise (1 pts.)

Implement the following 2 facts:

- the bulb in room 21 is lit.      la lampadina nella stanza 21 è accesa.
- the blinds in the window of room 20 are exposed.      le persiane della finestra della stanza 20 sono a vista.

Please implement the following relation:

- in room X is light if a light bulb in room X is on or blinds are exposed and it is bright outside. The relationship should take two variables: room number and a logical variable determining whether it is bright outside. The report should use previous facts.

Execute queries:

- Is it light in room 19 if it's bright outside?
- Is it light in room 19 if it's dark outside?
- Is it light in room 20 if it's bright outside?
- Is it light in room 20 if it's dark outside?
- Is it light in room 21 if it's bright outside?
- Is it light in room 21 if it's dark outside?
- which room is it light if it's light outside?
- which room is it light if it's dark outside?

## 8. Exercise (1 pts.)

Please implement the `is_sorted(List, Res)` predicate taking a list of natural numbers and returning the answer `Yes` if the list is sorted and `No` otherwise.

## 9. Exercise (1 pts.)

Please implement the `remove_from_list(List, Char, Rem_list)` taking the character list (as the first parameter), the character that is the character pattern to be removed from the list (as the second parameter). The program returns (as the third parameter) the list passed in the first parameter with all occurrences of the character given as the second parameter removed.

For example, called: `remove_from_list(['a','y','f','g','h','t','a','a','d','b'], 'a', Rem_list)` should return the result: `Rem_list = ['y','f','g','h','t','d','b']`.

## 10. Exercise (1 pts.)

Please implement the `add_opr(L, R)` predicate inserting four operators: `+`, `-`, `*` and `/` between `L` and `R`. The division will be returned only if `R` is different from zero.

For example, called: `add_opr (1, 2, EQ).` should return the result:

`EQ = 1 + 2`

`Yes (maybe more)`

`EQ = 1 - 2`

`Yes (maybe more)`

`EQ = 1 * 2`

`Yes (maybe more)`

`EQ = 1 / 2`

and called: `add_opr (7, 0, EQ).`:

`EQ = 7 + 0`

`Yes (maybe more)`

`EQ = 7 - 0`

`Yes (maybe more)`

`EQ = 7 * 0`

## 11.Exercise (0.5 pts.)

Please implement the `join(L, L1, L2)` predicate which, just like the built-in `append/3` concatenates two lists into one. For example, called: `join([1, 2], [3, 4], L).` should return the result:

```
L = [1, 2, 3, 4]
```

What will be the result of executing such a query: `join(L1, L2, [1, 2, 3]).`?

## 12.Exercise (0.5 pts.)

Please implement the `split(L, L1, L2)` predicate taking an n-element list and returns all n-1 possible divisions into two lists.

For example, called: `split([1, 2, 3], L1, L2).` should return the result:

```
L1 = [1]
```

```
L2 = [2, 3]
```

```
Yes (maybe more)
```

```
L1 = [1, 2]
```

```
L2 = [3]
```

```
Yes
```

## 13. Exercise (bonus exercise) (1 pts.)

Please implement the `all_equat(L, EQ)` taking a n-element list of integers and returning all available combinations of the four operators: `+`, `-`, `*`, `/` inserted between the numbers of lists. Remember to avoid division by zero.

For example, called: `all_equat([1, 2, 0], EQ).` should return the result:

```
EQ = 1 + 2 + 0
```

```
Yes (maybe more)
```

```
EQ = 1 + 2 - 0
```

```
Yes (maybe more)
```

```
EQ = (1 + 2) * 0
```

```
Yes (maybe more)
```

```
EQ = 1 - 2 + 0
```

```
Yes (maybe more)
```

```
...
```

```
EQ = 1 * (2 * 0)
```