The project is written in Python and composed by two module:

- **policy.py**
  This file contains two classes: PolicyModel and Policy.
  The first, Policy model, contains all the information to represent a policy:
  - **policy_name**: symbolic name for the current policy;
  - **users**: the set of all users;
  - **roles**: the set of all possible roles;
  - **ua**: the initial user-role assignments;
  - **cr**: the list of can-revoke rules
  - **ca**: the list of can assign rules;
  - **goal**: the target role;

  This object is a model object that has no methods.
  The second, Policy object, contains the PolicyModel class and has all the methods and logic to simulate a policy and try to solve the role reachability problem.
  The only public method that can be invoked is the:

```
def simulate_policy(self, patience: int = 5000):
    """
    Simulate the execution of a policy to find a valid path to reach the target (goal) role

    :param patience: number oc iterations for convergence
    :return:
        - 1 - the target role is reachable
        - 0 - the target role is unreachable
    """
```

  This method try to solve the reachability problem applying to the current steps:
  1. Apply a backward_slicing step;
  2. Apply a forward_slicing step;
  3. Loop on the inference rules to transit from the initial set of user-role assignments to an UR*set that contains a tuple (u, $r_{goal}$)

  All the methods are upper bounded by a fixed number of tentatives (parameter **patience**) to avoid endless loops.
  The analyser sometimes give a wrong prediction with the policy1.
  Example of correct run of the algorithm:

```
DEBUG:root:Policy policy2.arbac checked in 1.807s, target_reachable=0
DEBUG:root:Policy policy5.arbac checked in 2.753s, target_reachable=0
DEBUG:root:Policy policy8.arbac checked in 2.722s, target_reachable=0
DEBUG:root:Total execution time: 7.286s
```

- **arbac_analyser.py**
  This file contains the main routine and all the accessory methods to read a policy file and create an instance of a Policy object.