MISIO lab 1
Jakub Tomczak, 127083
14 marca 2019

1. Jak oceniani są agenci?
Agenci sa oceniani przy pomocy punktów. Agent otrzymuje 10 pkt jeżeli wyczyści lokalizacje, która jest zabrudzona, natomiast za zmiane lokalizacji jest odejmowany 1 punkt.
2. Jakie cechy ma to środowisko?
Środowisko posiada nastepujace cechy: cześciowo obserwowalne, deterministyczne, sekwencyjne, statyczne, dyskretne, jednoagentowe, znane.
3. Jeden z agentów okazał sie dużo lepszy. Dlaczego?
ModelBasedVacuumAgent nie wykonuje żadnej operacji jeżeli wie, że nie ma co robić, natomiast gorszy agent - ReflexVacuumAgent zawsze wykona jakaś operacje, dlatego traci wiecej punktów.
4. Czy agenty ReflexVacuumAgent lub ModelBasedVacuumAgent sa racjonalne? Uzasadnij.
Pierwszy agent - ReflexVacuumAgent nie jest agentem racjonalnym, ponieważ zawsze wykonuje jakiś ruch (jeżeli lokalizacja jest brudna to ja czyści, inaczej przeskakuje do drugiej). Natomiast agent ModelBased- VacuumAgent działa racjonalnie (wg swojej najlepszej wiedzy). Korzysta on ze wszelkiej wiedzy jaka ma nt. środowiska, wiec wg swojej wiedzy jest graczem racjonalnym maksymalizującym jakość.
5. Czy dla tego środowiska istnieje racjonalny agent odruchowy? Uzasadnij.
Nie, w tym środowisku tylko agent odruchowy ale z modelem świata może działać racjonalnie. Agent odruchowy nie korzystając z pamięci nie potrafi dobrze określić kolejnego ruchu tak dobrze jak agent z pamięcią.
6. Jakie ma ono cechy?
Środowisko posiada nastepujace cechy: cześciowo obserwowalne, stochastyczne, sekwencyjne, statyczne, dyskretne, jednoagentowe, znane.
7. Napisz program agenta, który bedzie (średnio) lepszy dla tego środowiska (50 kroków) niż ModelBaseVacuumAgent oraz ReflexVacuumAgent.

```
number_of_no_op = 0
no_op_limit = 9
oper = 0
model = {loc_A: None, loc_B: None}

def MyAgent(test = False):
    def program(percept):
        location, status = percept
        global number_of_no_op
        global oper
        global no_op_limit
        oper+=1
        model[location] = status

            if model[loc_A] == model[loc_B] == 'Clean' and
number_of_no_op < no_op_limit:
            number_of_no_op += 1
            return 'NoOp'
        elif status == 'Dirty':
            return 'Suck'
        elif oper <= no_op_limit:
            number_of_no_op += 1
            return 'NoOp'
        elif location == loc_A:
            number_of_no_op = 0
            return 'Right'
        elif location == loc_B:
            number_of_no_op = 0
            return 'Left'

    if not test:
        return program
    else:
        return Agent(program)
```

8. Oblicz statystyki
Wartość oczekiwana 44.93, odchylenie standardowe 18.54.
Przedział ufności na poziomie 95% - <44.82; 45.05>