

Sprawozdanie - Laboratorium 2

Inteligencja Obliczeniowa w Analizie Danych Cyfrowych

Autorzy:

Jakub Kubicki

Eryk Mikołajek

Marcin Zub

1. Cel ćwiczenia

- Zapoznanie się z dziedziną STRIPS
- Implementacja problemów w dziedzinie STRIPS oraz ich rozwiązanie z wykorzystaniem metody *Forward Planning*
- Analiza czasu wykonania się poszczególnych problemów i podproblemów oraz wykorzystanie heurystyk

2. Opis problemu

Problem polega na znalezieniu rozwiązania (w tym przypadku metodą *Forward Planning*) dla kilku problemów.

Są one zdefiniowane poprzez:

1. Domenę
2. Opis stanu początkowego
3. Opis stanu końcowego

Naszą dziedzinę (**domenę**) stanowią klocki ułożone na stole. Każdy z nich ma przypisaną dla siebie unikalną etykietę. **Stan początkowy** definiujemy i rozumiemy jako sytuację, w której każdy klocek leży na stole oraz nie ma na sobie żadnego innego obiektu. **Stan końcowy** (czyli ten, który chcemy osiągnąć poprzez użycie solvera) jest zdefiniowany analogicznie. Przedstawia on szczególną sytuację ułożenia klocków na stole, do której dążymy.

Do pierwszego z problemów zaproponowaliśmy następującą heurystykę: Dla każdego stanu heurystyka sprawdza czy ten stan znajduje się także w stanie końcowym (celu). Za każdy stan który nie zgadza się ze stanem końcowym, do heurystyki dodawana jest wartość 1. Ostatecznie, daje to efekt, że heurystyka zlicza ile bloków jest nie na swoich miejscach. Heurystyka ta może pomóc, ponieważ program będzie wiedział w którą stronę się kierować aby dotrzeć do celu - zminimalizować liczbę bloków nie na swoich miejscach.

Dzięki wyżej opisanej heurystyce zaobserwowaliśmy znaczną poprawę czasu poszukiwań rozwiązania.

3. Realizacja rozwiązania

Projekt został zrealizowany w języku Python. Do implementacji zadań z laboratorium wykorzystaliśmy bibliotekę *AI Python*. Znajdują się w niej funkcje, które pomogły stworzyć i opisać każdy z problemów. Jest tam też solver, którego użyliśmy do wygenerowania rozwiązań do poszczególnych problemów.

4. Podjęte próby rozwiązania

- Zdefiniowaliśmy, a następnie zaimplementowaliśmy własne dziedziny dla poszczególnych problemów. Jedną składającą się z 6 klocków, a drugą składającą się z 9 klocków
- Zdefiniowaliśmy, a następnie zaimplementowaliśmy 3 problemy poprzez podanie wartości stanów początkowych oraz końcowych
- Dla w.w problemów zdefiniowaliśmy własną heurystykę
- Dla każdego z problemów dodaliśmy dwa pod-cele
- Policzyliśmy czas, w którym solver znalazł rozwiązanie dla każdego z problemów (zarówno z heurystyką, jak i bez heurystyki)
- Zdefiniowaliśmy oraz zaimplementowaliśmy 3 dodatkowe problemy, które wymagały co najmniej 20 instancji akcji.

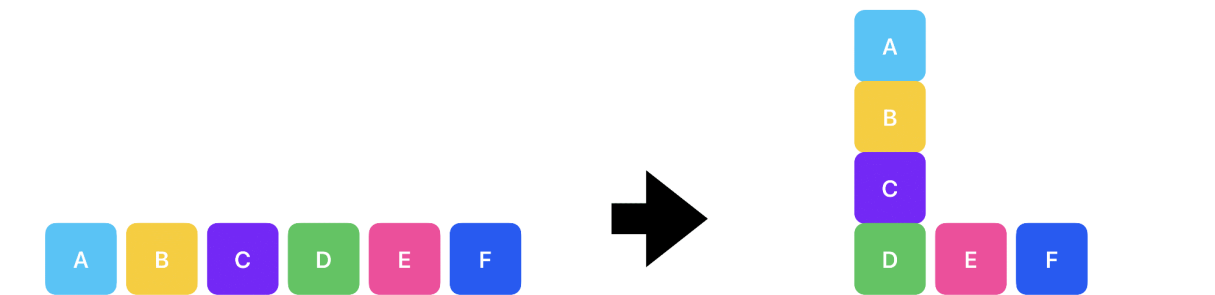
5. Statystyki

Problem 1:

Zmierzony czas bez heurystyki: **72.42 [s]**

Zmierzony czas z heurystyką: **0.0014 [s]**

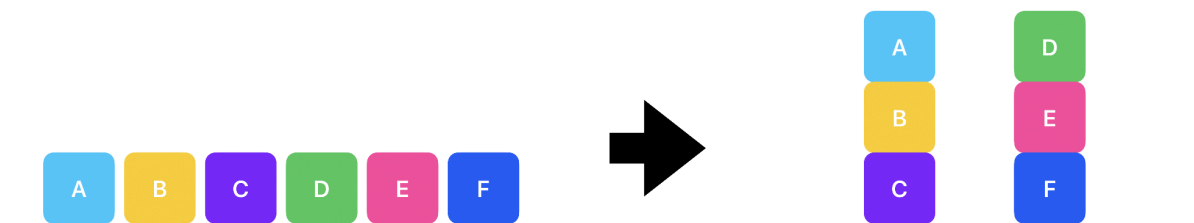
Zmierzony czas z podcelami: **39.12 [s]**



Problem 2:

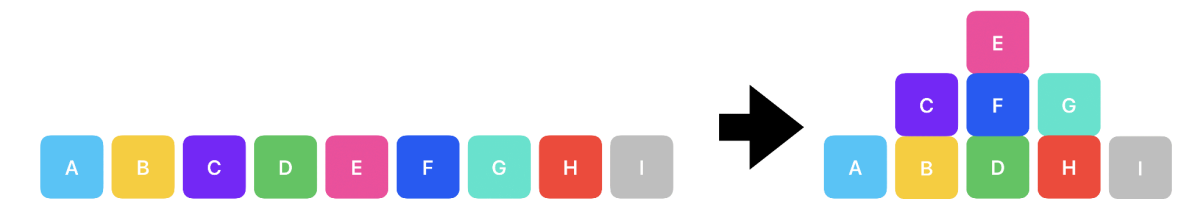
Zmierzony czas bez heurystyki: **491.01 [s]**

Zmierzony czas z heurystyką: **0.002 [s]**
Zmierzony czas z podcelami: **> 412 [s]**



Problem 3:

Zmierzony czas bez heurystyki: **> 10 [min]**
Zmierzony czas z heurystyką: **0.005 [s]**
Zmierzony czas z podcelami: **> 10 [min]**



Dodatkowe problemy:

```
1 initial_state_4 = {on('a'): 'table', clear('a'): True,
2                   on('b'): 'table', clear('b'): True,
3                   on('c'): 'table', clear('c'): True,
4                   on('d'): 'table', clear('d'): True,
5                   on('e'): 'table', clear('e'): True,
6                   on('f'): 'table', clear('f'): True,
7                   on('g'): 'table', clear('g'): True,
8                   on('h'): 'table', clear('h'): True,
9                   on('i'): 'table', clear('i'): True,
10                  on('j'): 'table', clear('j'): True,
11                  on('k'): 'table', clear('k'): True,
12                  on('l'): 'table', clear('l'): True}
13
14 goal_4 = {clear('a'): True, on('a'): 'table',
15           clear('b'): False, on('b'): 'table',
16           clear('c'): False, on('c'): 'b',
17           clear('d'): True, on('d'): 'c',
18           clear('e'): False, on('e'): 'd',
19           clear('f'): False, on('f'): 'e',
20           clear('g'): True, on('g'): 'f',
21           clear('h'): True, on('h'): 'g',
22           clear('i'): True, on('i'): 'h',
23           clear('j'): False, on('j'): 'i',
24           clear('k'): False, on('k'): 'j',
25           clear('l'): True, on('l'): 'k'}
```

```

1 initial_state_5 = initial_state_4.copy()
2 initial_state_5.update({on('m'): 'table', clear('m'): True,
3                        on('n'): 'table', clear('n'): True,
4                        on('o'): 'table', clear('o'): True,
5                        on('p'): 'table', clear('p'): True,
6                        on('q'): 'table', clear('q'): True,
7                        on('r'): 'table', clear('r'): True,
8                        on('s'): 'table', clear('s'): True})
9
10 goal_5 = {clear('m'): True, on('m'): 'table',
11           clear('n'): False, on('n'): 'table',
12           clear('o'): True, on('o'): 'n',
13           clear('p'): False, on('p'): 'o',
14           clear('q'): True, on('q'): 'p',
15           clear('r'): False, on('r'): 'q',
16           clear('s'): True, on('s'): 'r'}

```

```

1 initial_state_6 = initial_state_5.copy()
2 initial_state_6.update({clear('t'): True, on('t'): 'table',
3                        clear('u'): True, on('u'): 'table',
4                        clear('v'): True, on('v'): 'table',
5                        clear('w'): True, on('w'): 'table',
6                        clear('x'): True, on('x'): 'table',
7                        clear('y'): True, on('y'): 'table',
8                        clear('z'): True, on('z'): 'table'})
9
10 goal_6 = {clear('t'): False, on('t'): 's',
11           clear('u'): False, on('u'): 't',
12           clear('v'): False, on('v'): 'u',
13           clear('w'): False, on('w'): 'v',
14           clear('x'): False, on('x'): 'w',
15           clear('y'): False, on('y'): 'x',
16           clear('z'): True, on('z'): 'y'}

```

6. Wnioski

Występuje bardzo duża różnica, pomiędzy czasem znalezienia rozwiązania przez solver z wykorzystaniem heurystyki, a czasem znalezienia rozwiązania bez wykorzystania heurystyki. Wraz ze zwiększaniem się liczby klocków, problemy stają się coraz bardziej rozbudowane. Nawet z zastosowaniem heurystyk, czas ich rozwiązania znacznie się zwiększa.