

Inteligencja obliczeniowa w analizie danych cyfrowych

mgr inż. Damian Płóciennik

dplociennik@agh.edu.pl

Katedra Biocybernetyki i Inżynierii Biomedycznej
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

Centrum Doskonałości Sztucznej Inteligencji AGH

18 marca 2025

- Mam przyjemność poprowadzić państwu zajęcia 3, 4, 5 (STRIPS) oraz 11 i 12 (uczenie ze wzmocnieniem w przestrzeniach ciągłych).
- Projekty realizujemy w grupach dwuosobowych (lub samodzielnie).
- Projekty należy przesłać na MS Teams przed ostatnimi zajęciami z danego tematu, na których odbędzie się ich oddawanie. W przypadku STRIPS jest to ok. 31.03.
- Konsultacje odbywają się po wcześniejszym umówieniu zdalnie (MS Teams) lub stacjonarnie. Proszę o kontakt poprzez wiadomość e-mail dplociennik@agh.edu.pl lub MS Teams.

Część I

STRIPS

Definicja

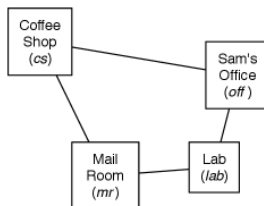
STRIPS (Stanford Research Institute Problem Solver) to formalny język używany do wyrażania problemów planowania, który pierwotnie został zaprojektowany do sterowania działaniami robota w manipulowalnym środowisku. Jego głównym celem jest automatyczne generowanie planów, czyli sekwencji działań, które przeprowadzają system z początkowego stanu do pożądanego stanu docelowego.

Kluczowe elementy STRIPS



- **States** - określane przez zbiór logicznych propozycji.
- **Goals** - określane jako zbiór warunków opisujących pożądany wynik.
- **Actions** - każda akcja jest opisywana przez:
 - **Preconditions** - warunki, które muszą być spełnione, aby działanie mogło zostać wykonane.
 - **Effects** - zmiany w stanie systemu, które zachodzą w wyniku wykonania działania.

Przykładowa domena



Właściwości

- RLoc - Rob's location
- RHC - Rob has coffee
- SWC - Sam wants coffee
- MW - Mail is waiting
- RHM - Rob has mail

Akcje

- mc - move clockwise
- mcc - move counterclockwise
- puc - pickup coffee
- dc - deliver coffee
- pum - pick up mail
- dm - deliver mail

Pick-up coffee (puc):

- **precondition:** $RLoc = cs$, $RHC = False$. Robot musi być w coffee shop i nie trzymać kawy, aby móc podnieść kawę.
- **effect:** $RHC = True$. Po tym jak robot podniesie kawę, będzie trzymać kawę. Nic więcej się nie zmienia.

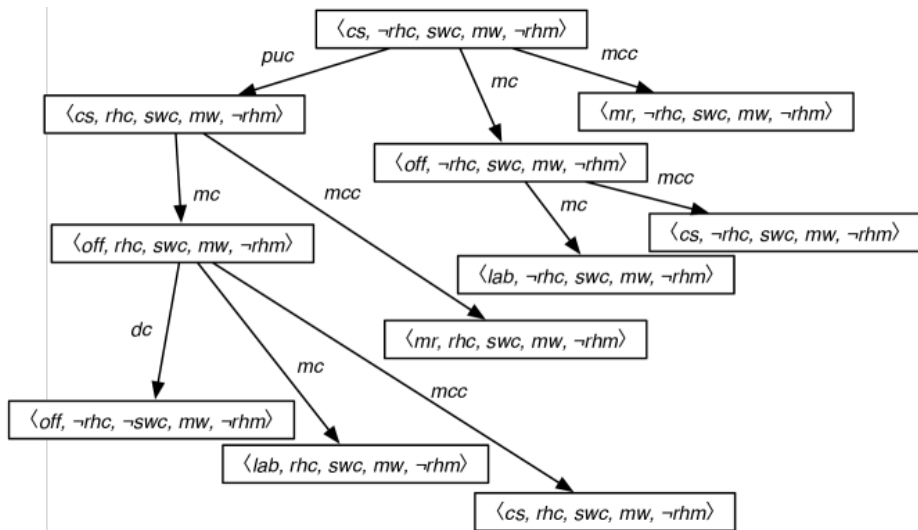
Deliver coffee (dc):

- **precondition:** $RLoc = off$, $RHC = True$. Robot musi być w biurze i trzymać kawę, aby móc dostarczyć kawę.
- **effect:** $RHC = False$, $SWC = False$. Po tym jak robot dostarczy kawę, przestaje trzymać kawę, a Sam więcej nie chce kawy. Nic więcej się nie zmienia.

Definicja

Forward planning to metoda eksploracji przestrzeni stanów, w której zaczynamy od stanu początkowego i poprzez zastosowanie akcji staramy się dotrzeć do stanu docelowego.

Forward planning



Definicja

Heurystyka to funkcja oceny, która pomaga przewidzieć, jak blisko jesteśmy rozwiązania.

Przykład

Dla planu robota dostarczającego, jeśli wszystkie działania mają koszt równy 1, możliwa dopuszczalna funkcja heurystyczna dla określonego celu to maksimum z:

- odległości od lokalizacji robota w stanie s do lokalizacji celu, jeśli taka istnieje,
- odległości od lokalizacji robota w stanie s do kawiarni powiększonej o trzy (ponieważ robot musi przynajmniej dotrzeć do kawiarni, odebrać kawę i dostarczyć ją do biura), jeśli cel zawiera $SWC = \text{false}$, natomiast stan s zawiera $SWC = \text{true}$, $RHC = \text{false}$.

Slajdy utworzono na podstawie materiałów dostępnych na stronie
<https://artint.info/AIPython/>.

- Fragmenty kodu AI Python z opisami:
<https://artint.info/AIPython/aipython/aipython.pdf>
- Archiwum .zip z kodem AIPython:
<https://artint.info/AIPython/aipython.zip>
- Prezentacje (chapter 6):
<https://artint.info/3e/slides/index.html>
- Treść książki, opisy teoretyczne:
<https://artint.info/3e/html/ArtInt3e.Ch6.html>