



I007 Osnove umjetne inteligencije

Tema: Klasično planiranje

14. 4. 2021.



1 Klasično planiranje





Klasično planiranje

- pretpostavljamo da je okruženje u kojem radimo jednoagentno, potpuno vidljivo, determinističko i statičko
- za prikazivanje svijeta koristimo faktorizirani prikaz znanja: stanje svijeta predstavljamo pomoću skupa varijabli
- za to koristimo jezik poznat pod imenom PDDL (eng. Planning domain definition language) uz pomoću kojeg definiramo probleme pretraživanja: početno stanje, akcije koje se mogu provoditi u određenom stanju, rezultat primjene akcija i testiranje cilja
- svako stanje je predstavljeno konjukcijom fluent (temeljni, bezfunkcijski atomi)
- koristi se semantika baza podataka: pretpostavka zatvorenog svijeta znači da je fluent koji nije spomenut netočan (lažan), te imamo pretpostavku o jedinstvenosti imena





Klasično planiranje

- akcije se opisuju skupom shema akcija kojima se eksplicitno opisuju funkcije “Akcija(s)” i “Rezultat(s, a)”
- skup temeljnih (bez varijabli) akcija može se predstaviti jednom shemom akcije
- shema se sastoji od imena akcije, liste svih varijabli koje se koriste u shemi, preduvjeta i efekata
- shema akcije se pretvara u logičke rečenice uz pretpostavku da su sve varijable univerzalno kvantificirane, te proizvoljno pridružujemo vrijednosti koje pridružujemo varijablama
- kažemo da je akcija a primjenjiva u stanju s ako s zadovoljava sve preuvjete
- početno stanje se definira kao konjunkcija fluenata
- ciljevi se opisuju konjunkcijom pozitivnih ili negativnih fluenata.
Problem je riješen kada nađemo niz akcija koje za rezultat imaju stanje koji sadrži sve fluente cilja (a možda i još neke dodatne)





Primjer 1.

Akcija(Letjeti(p, od, do),

Preduvjeti: $Na(p, od) \wedge Zrakoplov(p) \wedge ZračnaLuka(od)$
 $\wedge ZračnaLuka(do)$

Efekti: $\neg Na(p, od) \wedge Na(p, do)$

Akcija(Letjeti(p₁, ZLO, ZLP),

Preduvjeti: $Na(p_1, ZLO) \wedge Zrakoplov(p_1) \wedge ZračnaLuka(ZLO)$
 $\wedge ZračnaLuka(ZLP)$

Efekti: $\neg Na(p_1, ZLO) \wedge Na(p_1, ZLP)$





Primjer 1.

Akcija(Letjeti(p, od, do),

Preduvjeti: $Na(p, od) \wedge Zrakoplov(p) \wedge ZračnaLuka(od)$
 $\wedge ZračnaLuka(do)$

Efekti: $\neg Na(p, od) \wedge Na(p, do)$

Akcija(Letjeti(p₁, ZLO, ZLP),

Preduvjeti: $Na(p_1, ZLO) \wedge Zrakoplov(p_1) \wedge ZračnaLuka(ZLO)$
 $\wedge ZračnaLuka(ZLP)$

Efekti: $\neg Na(p_1, ZLO) \wedge Na(p_1, ZLP)$





Primjer 2. Transporta zrakoplovnog tereta

Početno stanje($\text{Na}(c_1, \text{ZLO}) \wedge \text{Na}(c_2, \text{ZLP}) \wedge \text{Na}(p_1, \text{ZLO})$
 $\wedge \text{Na}(p_2, \text{ZLP}) \wedge \text{Teret}(c_1) \wedge \text{Teret}(c_2)$
 $\wedge \text{Zrakoplov}(p_1) \wedge \text{Zrakoplov}(p_2)$
 $\wedge \text{ZračnaLuka}(\text{ZLO}) \wedge \text{ZračnaLuka}(\text{ZLP}))$

Ciljno stanje($\text{Na}(c_1, \text{ZLP}) \wedge \text{Na}(c_2, \text{ZLO})$)

Akcija($\text{Utovariti}(c, p, a)$,

Preduvjeti: $\text{Na}(c, a) \wedge \text{Na}(p, a) \wedge \text{Teret}(c) \wedge \text{Zrakoplov}(p)$
 $\wedge \text{ZračnaLuka}(a)$

Efekti: $\neg \text{Na}(c, a) \wedge \text{U}(c, p)$)

Akcija($\text{Istovariti}(c, p, a)$,

Preduvjeti: $\text{U}(c, p) \wedge \text{Na}(p, a) \wedge \text{Teret}(c) \wedge \text{Zrakoplov}(p)$
 $\wedge \text{ZračnaLuka}(a)$

Efekti: $\neg \text{U}(c, p) \wedge \text{Na}(c, a)$)

Akcija($\text{Letjeti}(p, \text{od}, \text{do})$,

Preduvjeti: $\text{Na}(p, \text{od}) \wedge \text{Zrakoplov}(p) \wedge \text{ZračnaLuka}(\text{od})$
 $\wedge \text{ZračnaLuka}(\text{do})$

Efekti: $\neg \text{Na}(p, \text{od}) \wedge \text{Na}(p, \text{do})$)





Primjer 2. Transporta zrakoplovnog tereta

Plan koji daje rješenje transportnog problema je sljedeći niz akcija

[Utovariti(c_1, p_1, ZLO), Letjeti(p_1, ZLO, ZLP), Istovariti(c_1, p_1, ZLP),
Utovariti(c_2, p_2, ZLP), Letjeti(p_2, ZLP, ZLO), Istovariti(c_2, p_2, ZLO)]





Klasično planiranje

- algoritmi planiranja mogu se promatrati kao algoritmi pretraživanja prostora
- na najopćenitijem nivou dijelimo ih na progresije (pretraživanje unaprijed) i regresije (pretraživanje unatrag)
- kod planiranja unaprijed, razmatramo moguće efekte akcija, dok se kod pretraživanja unatrag pitamo koja bi akcija dovela do željenog rezultata
- i kod planiranja je moguće koristiti heuristike; npr. jedna od mogućih heuristika bi bila ona koja zanemaruje preduvjete problema
- račun situacija: korištenje zaključivanja logike prvog reda u svrhu planiranja





Vrijeme, raspored i sredstva

- klasično planiranje govori o tome što napraviti i u kojem redosljedu, no ne kaže ništa o vremenu: koliko dugo se akcija izvodi ili kada se izvodi
- s tim dijelom se bavi raspoređivanje
- u stvarnom svijetu postoje brojna ograničenja s resursima koja moramo zadovoljiti (broj osoblja, ista osoba ne može u jednom trenutku biti na više mjesta i sl.)
- ukoliko problemi planiranja sadrže vremensko ograničenje i ograničenje resursa, tada moramo koristiti drugačiji pristup
- problem dijelimo na dvije faze planiranja: prvo se provede klasično planiranje, a zatim raspoređivanje
- također se koristi višerazinsko (hijerarhijsko) planiranje: planiramo na apstraktnom nivou, a u trenutku kada se sama akcija treba izvesti radimo određena profinjenja plana





Zadatak 1.

Zadatak za vježbu

Promatramo problem mijenjanja probušene gume. Cilj je ispravnu rezervnu gumu propisno montirati na osovinu automobila, pri čemu je početno stanje ono s probušenom gumom na kotaču na osovini i sa ispravnom gumom u prtljažniku. Napišite PDDL opis problema i navedite barem jedno rješenje.





Zadatak 2.

Zadatak za vježbu

Problem majmuna i banana je problem u kojem je majmun u laboratoriju u kojem sa stropa vise banane koje su mu izvan dohvata. Majmunu je dostupna kutija koja će mu omogućiti da dohvati banane ako se popne na nju. Na početku je majmun na poziciji A , banane su na poziciji B , a kutija je na poziciji C . Majmun i kutija su na poziciji *Nisko*, ali ako se majmun popne na kutiju, imat će poziciju *Visoko* na kojoj su i banane. Akcije koje majmun može napraviti su: *Idi* od jednog mjesta do drugog, *Gurni* objekt od jednog mjesta do drugog, *PopniSe* ili *Sidi* s nekog objekta, i *Dohvati* ili *Pusti* neki objekt. Rezultat akcije *Dohvati* je da majmun drži objekt, ali samo ako su majmun i objekt na istoj poziciji i na istoj visini. Napišite PDDL opis problema i navedite barem jedno rješenje.

