

Mesterséges intelligencia - 8. előadás

Cselekvéstervezés

Cselekvéstervezési feladatok

- Logisztikai hálózatok, műholdak, szerelési művelettervezés
- cselekvéssorozat adott kiindulási állapotból célállapotba, elérhető akciók sorozatának végrehajtásával
- speciális feladatoknak már van dedikált megoldásuk

Feltevések

- csak determinisztikus cselekvéstervezés (környezet tökéletesen ismert és megfigyelhető)
- tervekészítés offline, végrehajtás csukott szemmel
- zárt világ feltevés → a világ véges, teljes egészében megismerhető

Tervezési feladat leírása

- **világ leírása:** kezdeti állapot (teljesen specifikált), célállapot (teljesen vagy részben specifikált)
 - elsőrendű logikával → atomi elsőrendű logikai kifejezések konjunkciójával
 - pl: `airplane(Plane1) ∧ at(Plane1, BUD) ∧ ...`
 - nem megengedett: változók (`at(Plane2, x)`), függvény (`Plane3, BasisOf(Plane3)`), diszjunkció (`∨`)
 - **célfeltétel:** részlegesen definiált, ponált és negált literálok is megengedettek
 - célállapot = tartalmazza a célfeltétel összes ponált literálját, és nem tartalmazza egyik negált literálját se!
- **lehetőséges akciók:** cselekvés sémák (adott helyzetben mit csinálhatunk, milyen hatása lesz)
 - cselekvés séma: akció neve, paraméter lista, előfeltételek, hatások
 - pl:

```
(:action fly
  :parameters(?p ?s ?d)
  :precondition (and (airplane ?p)
                     (airport ?s)
                     (airport ?d)
                     (at ?p ?s)
                     (not (=s ?d)))
  :effect (and (at ?p ?d)
               (not (at ?p ?s))))
```

- ilyen nyelv pl: PDDL
- feladatpéldány definíciója: entitások, kezdeti állapot és célfeltétel definiálása
 - vannak **statikus predikátumok** → ami fix, nem változik
 - vannak **dinamikus predikátumok** → amik változnak, pl hogy hol van a repülő vagy teherautó
- **terv**

- cselekvések (részben) rendezett halmaza
 - (drive BUD-truck BUD-airport BUD-post BUD)
 - (load-truck package1 BUD-truck BUD-post)
 - (drive BUD-truck BUD-post BUD-airport BUD)
 - (unload package1 BUD-truck BUD-airport)
 - (load-plane package1 airplane1 BUD-airport)
- - (fly airplane1 BUD-airport NAP-airport)
 - (unload package1 airplane1 NAP-airport)
 - (load-truck package1 NAP-truck NAP-airport)
 - (drive NAP-truck NAP-airport NAP-post NAP)
 - (unload package1 NAP-truck NAP-post)

Tervkészítő algoritmusok

Megközelítések

- **szituációkalkulus** → elsőrendű logika kiterjesztése szituáció tényezőkkel, következés relációt is felvesszünk, már elavult
- **keresés az állapottérben** → csomópont = világállapot, egymást követő állapotok = egy terv
- **keresés a tervtérben** → csomópont = egy részleges terv, egy következő állapot = egy tovább finomított terv
- **egyéb technikák** → gráf-alapú tervkészítés, fordítás kielégíthetőségi feladat (SAT)

Keresés állapottérben - előreláncoló keresés

- ötlet:
 - 1 csomópont = 1 világállapot
 - gyökér = kezdeti állapot
 - egy állapotban valamilyen cselekvés sémát alkalmazunk → így jutunk következő állapotba
 - addig ismételjük, amíg meg nem érkezünk egy célállapotba
 - terv = egy sorozat a gyökekből egy megtalált célállapotig
- milyen keresést használjunk?
 - tetszőleges fakesés használható
 - szélességi és legjobbat-először → exponenciális lesz :(
 - **mélyégi, mohó, IDA*** → memóriaigény lineáris, teljesekek :)
- milyen heurisztika alapján?
 - kézenfekvő ötletek: kiegészítetlen célfeltételek száma → NEM jó: alulbecsülhet (egy cselekvés törli a másik hatását), túlbecsülhet (egy cselekvés több feltételt is előállíthat)
 - **FastForward** heurisztika (FF) → elfogadható (pl A*-hoz)
 - minden csomópontban oldjunk meg egy egyszerűsített cselekvéstervezési feladatot
 - egyszerűsítés = csak a számunkra kedvező hatást nézzük → At() ne teljesüljön helyett NotAt() teljesüljön! (nincs negatív következmény)
 - jóhír: mindig alábecsüli!

Keresés a tervek terében

- ötlet:
 - állapotterres csúnya redundanciákat tudott tartalmazni!
 - cselekvések minden sorrendjét végigpróbáltuk
 - 1 csomópont = 1 részleges terv
 - gyökér = üres terv
 - ezt finomítjuk, míg teljesen specifikált tervet nem kapunk
- részben rendezett tervkészítő
 - cselekvések, változók értékének lekötése, cselekvés-párok közötti sorrendiség korlát
 - sorrend → folytonos nyíl
 - okozati kapcsolat → szaggatott nyíl (előállította a következő feltételét)
 - okozati kapcsolatból sorrendi korlát is következik!
- Hiányosság:
 - **kielégítetlen előfeltétel** → ezeket kell megszüntetnünk adott cselekvés megtalálásával (valami olyannal ami biztosítja ezen előfeltételt, és leköt bizonyos változókat)
 - ezek közt okozati kapcsolat lesz!
 - **okozati kapcsolat fenyegetése** → ha egy a cselekvés biztosítja b cselekvés p előfeltételét, ÉS egy másik c cselekvés képes törölni p -t → fenyegetés megszüntetése új korlát által
 - **precedencia: $b \rightarrow c$** , azaz biztosítjuk, hogy a fenyegető cselekvés később fog megtörténni, mint az a cselekvés, aminek előfeltételét veszélyezteti
 - **precedencia: $c \rightarrow a$** , azaz biztosítjuk, hogy a fenyegető cselekvés előbb fog megtörténni, mint az a cselekvés ami a veszélyeztetett előfeltételt biztosítja
 - **változók lekötése** → lekötjük úgy a változókat, hogy megszűnjön a veszély
- **tervtérbeli keresés (PSP)**

```
PSP( $\pi$ )
   $flaws \leftarrow \text{OpenGoals}(\pi) \cup \text{Threats}(\pi)$ 
  if  $flaws = \emptyset$  then return( $\pi$ )
  select any flaw  $\phi \in flaws$ 
   $resolvers \leftarrow \text{Resolve}(\phi, \pi)$ 
  if  $resolvers = \emptyset$  then return(failure)
  nondeterministically choose a resolver  $\rho \in resolvers$ 
   $\pi' \leftarrow \text{Refine}(\rho, \pi)$ 
  return(PSP( $\pi'$ ))
end
```

Heurisztikák a tervek terében

- nehezebb jó heurisztikát találni, mint állapottérben
- lehetséges heurisztika: nyitott előfeltételek száma
 - új operátor több előfeltételt kielégíthet (felülbecslés, mert mást el is ronthat)
 - interferencia különböző operátorok hatásai között (alulbecslés, mert segíthetik is egymást)

Tervtér vs állapottér → melyik hatékonyabb?

- kevesebb redundancia (tervtér) vs jobb heurisztikák (állapottér)
- állapotteres vezet, a jobb heurisztikák miatt

Cselekvéstervezés korlátai

- **optimális tervek kellenének** → bonyolult költségfüggvények (nem alkalmasak az eddig nézettek, pl numerikus képességeik sincsenek)
- legnépszerűbb cselekvéstervezők még a legrövidebb tervet se mindig találják meg!
- számítási hatékonyság → kénytelenek leszünk dedikált megoldót készíteni :(
- tervkészítés bizonytalan környezetben → eddig bemutatott dolgok ilyen NEM kezeltek!