
Mesterséges Intelligencia Cselekvéstervezés

Kovács András

MTA SZTAKI, akovacs@sztaki.hu

2018.10.11.



Cselekvéstervezési feladatok

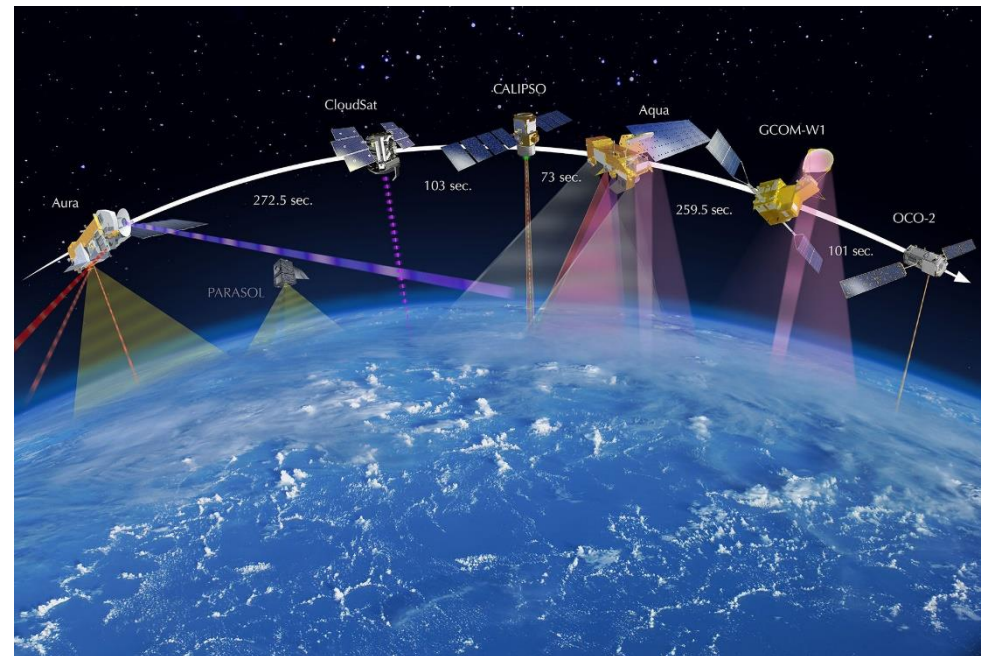
■ Logisztikai hálózatok

- Küldeményeket juttatnánk el feladási helyükről célállomásukra
- Adott járműveket használhatunk (pl. repülő, teherautó)
- Hogyan csináljuk?



Cselekvéstervezési feladatok (2)

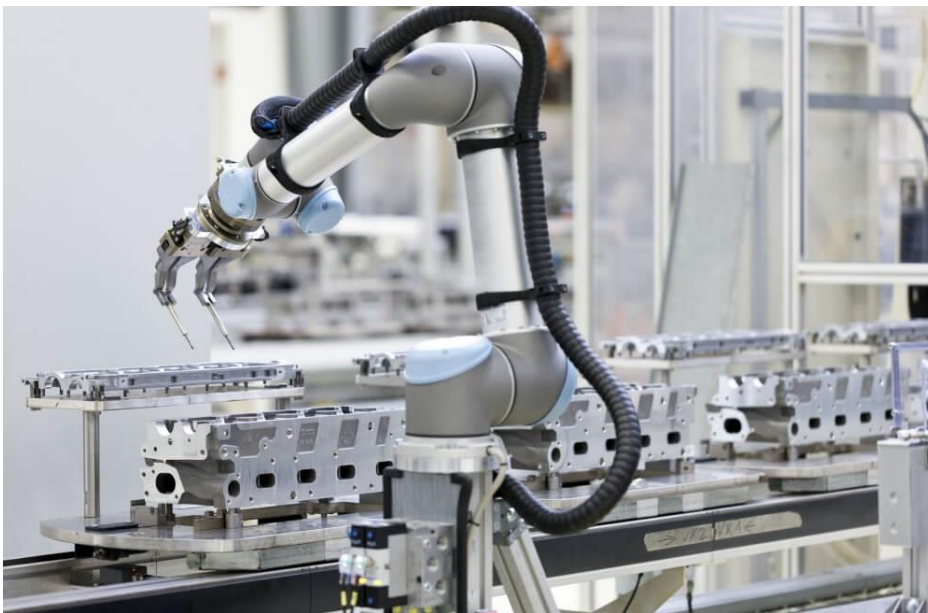
- Műholdas megfigyelések tervezése
 - Adott megfigyelési műveletek (hely, időablak, eszközigény, stb.)
 - Adott műholdak (pálya, eszközök, energiaforrás, stb.)
 - Hogyan csináljuk?



Cselekvéstervezési feladatok (3)

■ Szerelési művelettervezés

- Alkatrészekből szerelnénk össze egy adott terméket
- Adott eszközökkel (pl. robotok, szerszámok)
- A technológiai korlátok figyelembe vételével
- **Hogyan csináljuk?**



Cselekvéstervezési feladat definíciója

- Olyan cselekvéssorozat meghatározása, amely biztosítja az ágens céljainak elérését
 - Egy adott kiindulási állapotból
 - Egy célállapotba
 - Valamely, az ágens számára elérhető akciók sorozatának végrehajtásával

- Dedikált vs. általános megoldó

- Mi kell ehhez?
 - Egy nyelv, amivel leírjuk a tervezési feladatot
 - Egy algoritmus, ami megoldja

Feltevések

■ Determinisztikus cselekvéstervezés

- A környezet tökéletesen megfigyelhető
- Az ágens cselekvéseinek kimenete determinisztikus
- Más ágensek nem hatnak a környezetre

■ Tervkészítés *offline*, majd a terv végrehajtása *csukott szemmel*

- Nincs szükség valós időben érzékelésre és döntéshozatalra

■ Zárt világ feltevés (*closed world assumption*)

- A világ teljes egészében megismerhető
- Amiről nem tudjuk, hogy igaz, az hamis

A tervezési feladat leírása

■ Világállapot leírása

- Kezdeti állapot: teljesen specifikált
- Célállapot: teljesen vagy részben specifikált (célfeltétel)

■ Lehetséges akciók: cselekvés sémák

- Milyen feltétele van az alkalmazásuknak?
- Milyen hatásuk van a világállapotra?

Világállapot leírása

■ Atomi elsőrendű logikai kifejezések konjunkciója

- Kötött és függvénymentes
- Megengedett: $\text{airplane}(\text{Plane1}) \wedge \text{at}(\text{Plane1}, \text{BUD}) \wedge \dots$
- Nem megengedett: $\text{at}(\text{Plane2}, x), \text{at}(\text{Plane3}, \text{BasisOf}(\text{Plane3})),$
 $\text{at}(\text{Truck3}, \text{BUD}) \vee \text{at}(\text{Truck3}, \text{NAP})$

■ Ezt a formalizmust használjuk

- A kezdeti állapot leírására
- A célfeltétel leírására (kis kiegészítéssel: negált literálok)
- (Tetszőleges állapot belső reprezentációjára egy állapottérbeli tervező algoritmus által)

Célállapot/célfeltétel

■ A célfeltétel egy részlegesen definiált állapot

- Elsőrendű logikai literálok konjunkciója
- Ponált és negált literálok megengedettek

■ Egy állapot célállapot, ha

- Tartalmazza a célfeltétel összes ponált literálját, és
- Nem tartalmazza a cél egyik negált literálját sem

■ Pl:

- Célfeltétel: $\text{at}(\text{Package1}, \text{BUD}) \wedge \text{at}(\text{Plane2}, \text{NAP}) \wedge \neg \text{at}(\text{Truck3}, \text{BUD_BME})$
- Állapot 1: $\text{at}(\text{Package1}, \text{BUD}) \wedge \text{at}(\text{Plane2}, \text{NAP}) \wedge \text{at}(\text{Package8}, \text{CDG}) \wedge \dots$ ✓
- Állapot 2: $\text{at}(\text{Package1}, \text{BUD}) \wedge \text{at}(\text{Package8}, \text{CDG})$ ✓ ✗
- Állapot 3: $\text{at}(\text{Package1}, \text{BUD}) \wedge \text{at}(\text{Plane2}, \text{NAP}) \wedge \text{at}(\text{Truck3}, \text{BUD_BME}) \wedge \dots$ ✗

Cselekvés leírása

■ Cselekvés séma

- Akció neve
- Paraméter lista
- Előfeltételek
- Hatások

```
(: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))))
```

Cselekvés séma alkalmazása

■ Paraméterlista behelyettesítése olyan módon, hogy

- A cselekvés összes előfeltétele kielégíthető
- A világállapot leírásában található literálokkal

Világállapot:

$(\text{airport BUD}) \wedge (\text{airport NAP}) \wedge (\text{airport CDG}) \wedge (\text{at plane1 BUD}) \wedge \dots$

Cselekvés séma:

`(:action fly`

`:parameters (?p ?s ?d)`

`:precondition (and (airplane ?p)`

`(airport ?s)`

`(airport ?d)`

`(at ?p ?s)`

`(not (= ?s ?d)))`

`:effect ...`

Lehetséges cselekvések:

`fly(plane1 BUD NAP)`

`fly(plane1 BUD CDG)`

`...`

Cselekvés séma alkalmazása (2)

■ Világállapot frissítése a cselekvés hatásaival

- A megfelelő literálok hozzáadása/eltávolítása a világállapotból

(airport BUD) $\wedge$ (airport NAP) $\wedge$ (airport CDG)
 $\wedge$ (at plane1 BUD) $\wedge$...

fly(plane1 BUD NAP)

(airport BUD) $\wedge$ (airport NAP) $\wedge$ (airport CDG)
 $\wedge$ (at plane1 NAP) $\wedge$...

```
(: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))))
```

A teljes tervezési feladat leírása

■ Planning Domain Definition Language, PDDL

- A szakterület (domain) definíciója
- A feladatpéldány definíciója

■ Szakterület definíció

- Fejléc
- Predikátumok
- Cselekvés sémák

```
(define (domain logistics-strips)
  (:requirements :strips :equality)
  (:predicates (obj ?o)
               (truck ?t)
               (at ?o ?l)
               (in ?o ?t)
               (airplane ?p)
               (airport ?s)
               (in-city ?s ?city)
               (city ?c)
               (location ?l))
  ...)
```

} Fejléc

} Predikátumok

A teljes tervezési feladat leírása (2)

■ Szakterület definíció, folyt.

■ Cselekvés sémák: repülő, ill. teherautó mozgatása

(: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))))

(:action drive

:parameters (?truck ?s ?d ?city)

:precondition (and (truck ?truck)

(at ?truck ?s)

(in-city ?s ?city)

(in-city ?d ?city)

(not (= ?s ?d)))

:effect (and (at ?truck ?d)

(not (at ?truck ?s))))

A teljes tervezési feladat leírása (2)

■ Szakterület definíció, folyt.

■ Cselekvés sémák: repülő, ill. teherautó mozgatása

(: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))))

(:action drive

:parameters (?truck ?s ?d ?city)

:precondition (and (truck ?truck)

(at ?truck ?s)

(in-city ?s ?city)

(in-city ?d ?city)

(not (= ?s ?d)))

:effect (and (at ?truck ?d)

(not (at ?truck ?s))))

A teljes tervezési feladat leírása (3)

■ Szakterület definíció, folyt.

■ Cselekvés sémák: repülő, ill. teherautó berakodása, kirakodása

(:action load-truck

:parameters (?o ?truck ?loc)

:precondition (and (obj ?o) (truck ?truck)
(at ?o ?loc) (at ?truck ?loc))

:effect (and (not (at ?o ?loc))
(in ?o ?truck)))

(:action load-plane

:parameters (?o ?p ?loc)

:precondition (and (obj ?o) (airplane ?p)
(at ?o ?loc) (at ?p ?loc))

:effect (and (not (at ?o ?loc))
(in ?o ?p)))

(:action unload

:parameters (?o ?v ?loc)

:precondition (and (in ?o ?v)
(at ?v ?loc))

:effect (and (at ?o ?loc)
(not (in ?o ?v))))

A teljes tervezési feladat leírása (3)

■ Szakterület definíció, folyt.

■ Cselekvés sémák: repülő, ill. teherautó berakodása, kirakodása

(:action load-truck

:parameters (?o ?truck ?loc)

:precondition (and (obj ?o) (truck ?truck)

(at ?o ?loc) (at ?truck ?loc))

:effect (and (not (at ?o ?loc))
(in ?o ?truck)))

(:action load-plane

:parameters (?o ?p ?loc)

:precondition (and (obj ?o) (airplane ?p)

(at ?o ?loc) (at ?p ?loc))

:effect (and (not (at ?o ?loc))
(in ?o ?p)))

(:action unload

:parameters (?o ?v ?loc)

:precondition (and (in ?o ?v)

(at ?v ?loc))

:effect (and (at ?o ?loc)
(not (in ?o ?v))))

A teljes tervezési feladat leírása (4)

■ A feladatpéldány definíciója

- Entitások
- Kezdeti állapot
- Célfeltétel

■ Pl. egy csomag eljuttatása a Budapesti postáról a nápolyi postára

```
(define (problem log0)
  (:domain logistics-strips)
  (:objects package1 BUD-truck NAP-truck airplane1
    BUD-post NAP-post BUD-airport NAP-airport
    BUD NAP)
  (:init (OBJ package1)
    (TRUCK BUD-truck)
    (TRUCK NAP-truck)
    (AIRPLANE airplane1)
    (LOCATION BUD-post)
    (LOCATION NAP-post)
    (LOCATION BUD-airport)
    (LOCATION NAP-airport)
    (AIRPORT BUD-airport)
    (AIRPORT NAP-airport)
    (CITY BUD)
    (CITY NAP)
    (IN-CITY BUD-post BUD)
    (IN-CITY BUD-airport BUD)
    (IN-CITY NAP-post NAP)
    (IN-CITY NAP-airport NAP)
    (at package1 BUD-post)
    (at airplane1 BUD-airport)
    (at BUD-truck BUD-airport)
    (at NAP-truck NAP-airport))
  (:goal (and (at package1 NAP-post)))
)
```

Statikus predikátumok

Dinamikus predikátumok

A 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 algoritmus: megközelítések

■ Szituációkalkulus

- Az elsőrendű logikai kiterjesztése szituáció változókkal
- Történetileg érdekes

■ Keresés az állapottérben

- A keresési csomópont egy világállapot
- Egymást követő állapotok sorozata meghatároz egy tervet



■ Keresés a tervtérben

- A keresési csomópont egy részleges terv
- Finomítjuk, mígnem egy teljesen definiált, megengedett tervet kapunk



■ Egyéb technikák, pl.

- Gráf-alapú tervkészítés (GraphPlan)
- Fordítás kielégíthetőségi feladatra (SAT)

Keresés állapottérben: előre láncoló keresés

■ Általános (nemdeterminisztikus) séma

Forward-search(O, s_0, g)

$s \leftarrow s_0$

$\pi \leftarrow$ the empty plan

loop

if s satisfies g then return π

$applicable \leftarrow \{a \mid a \text{ is a ground instance of an operator in } O, \text{ and } \text{precond}(a) \text{ is true in } s\}$

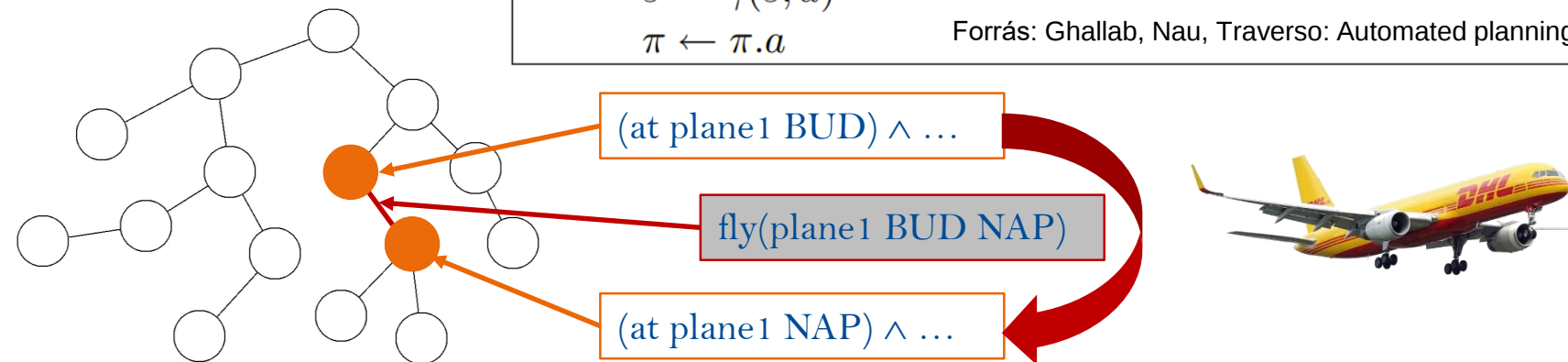
if $applicable = \emptyset$ then return failure

nondeterministically choose an action $a \in applicable$

$s \leftarrow \gamma(s, a)$

$\pi \leftarrow \pi.a$

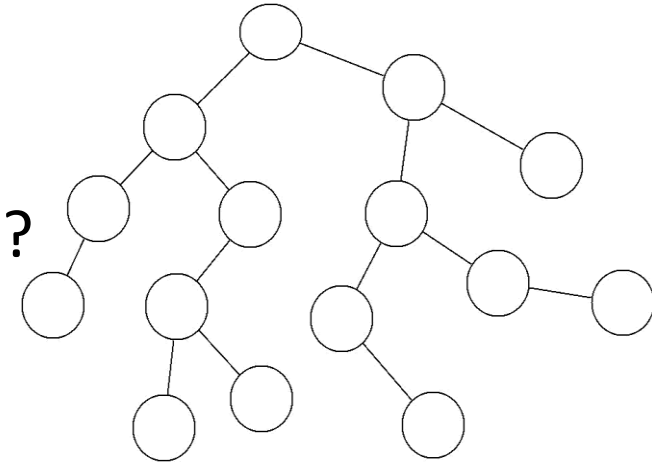
Forrás: Ghallab, Nau, Traverso: Automated planning – Theory and practice



Keresés állapot térben

■ Milyen sorrendben járjuk be az állapotteret?

- Keresési eljárás
- Heurisztika a kifejtendő csomópont kiválasztására



■ Elvileg tetszőleges ismert fa-keresési eljárás alkalmazható

- Mélységi keresés
- Szélességi keresés
- Legjobbat-először keresés
 - Mohó
 - A*
 - ...

Keresés állapot térben (2)

- Szélességi és legjobbat-először: helyes és teljes
 - Az exponenciális memóriaigény viszont problémás
- Mélységi, mohó, IDA*
 - A memóriaigény lineáris a terv és az állapotleírás hosszában
 - Teljesek?
 - Az eddig vizsgált tervezési feladatok (AKA klasszikus tervezés) végesek
 - A lehetséges állapotok száma véges
 - A keresési eljárások ciklus-ellenőrzéssel teljessé tehetők

Csomópont választási heurisztikák

■ Kézenfekvő heurisztikák

- Pl. kielégítetlen célfeltételek száma
- Alulbecsühet: cselekvések közötti kölcsönhatások
- Túlbecsülhet: egy cselekvés több célfeltételt előállít

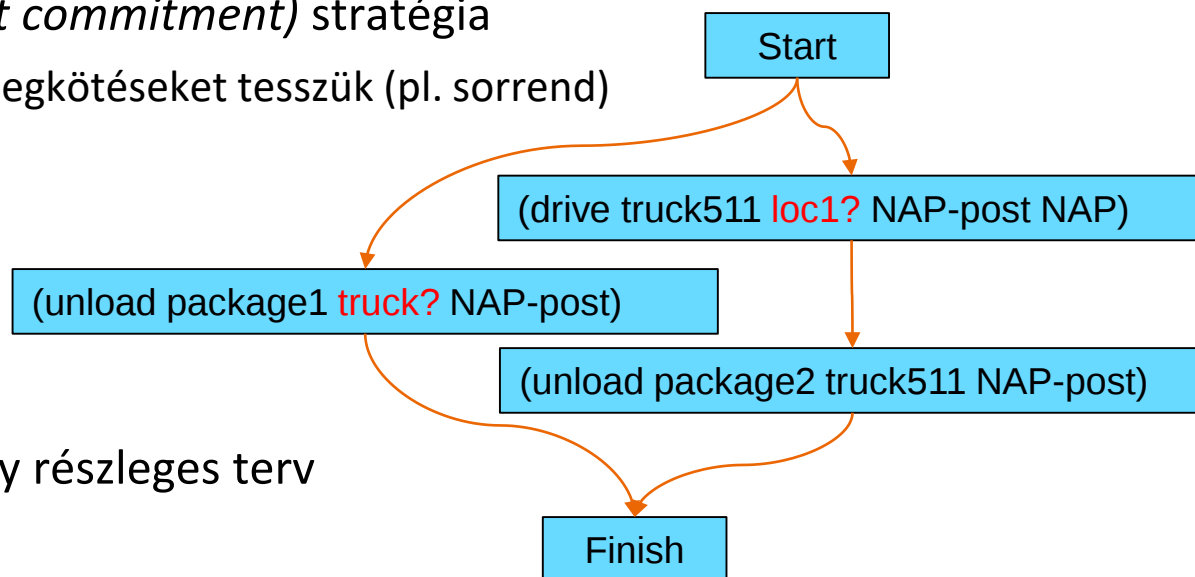
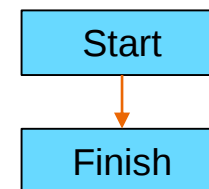
■ Egy elfogadható (*admissible*) heurisztika (pl. A*-hoz)

- Relaxált feladat: töröljük minden cselekvés negált hatásait (Empty-delete-list)
- Negált cél/előfeltételek esetén új „negált” atom, pl. [At\(.\)](#) → [NotAt\(.\)](#)
- Egyszerű tervező algoritmus futtatásával számítható (GraphPlan)
- Előre és hátrafelé láncoló keresésnél is használható
- Egy legendás cselekvéstervező rendszer alapja: FastForward (FF)

Keresés a tervek terében

■ Motiváció

- Az állapottérben való keresés redundáns
 - A cselekvések minden sorrendjét végigpróbálja
 - Mielőtt rájön, hogy az adott ágon nincs megengedett terv
- Legkisebb elkötelezettség (*least commitment*) stratégia
 - Csak olyan a legszükségesebb megkötéseket tesszük (pl. sorrend)



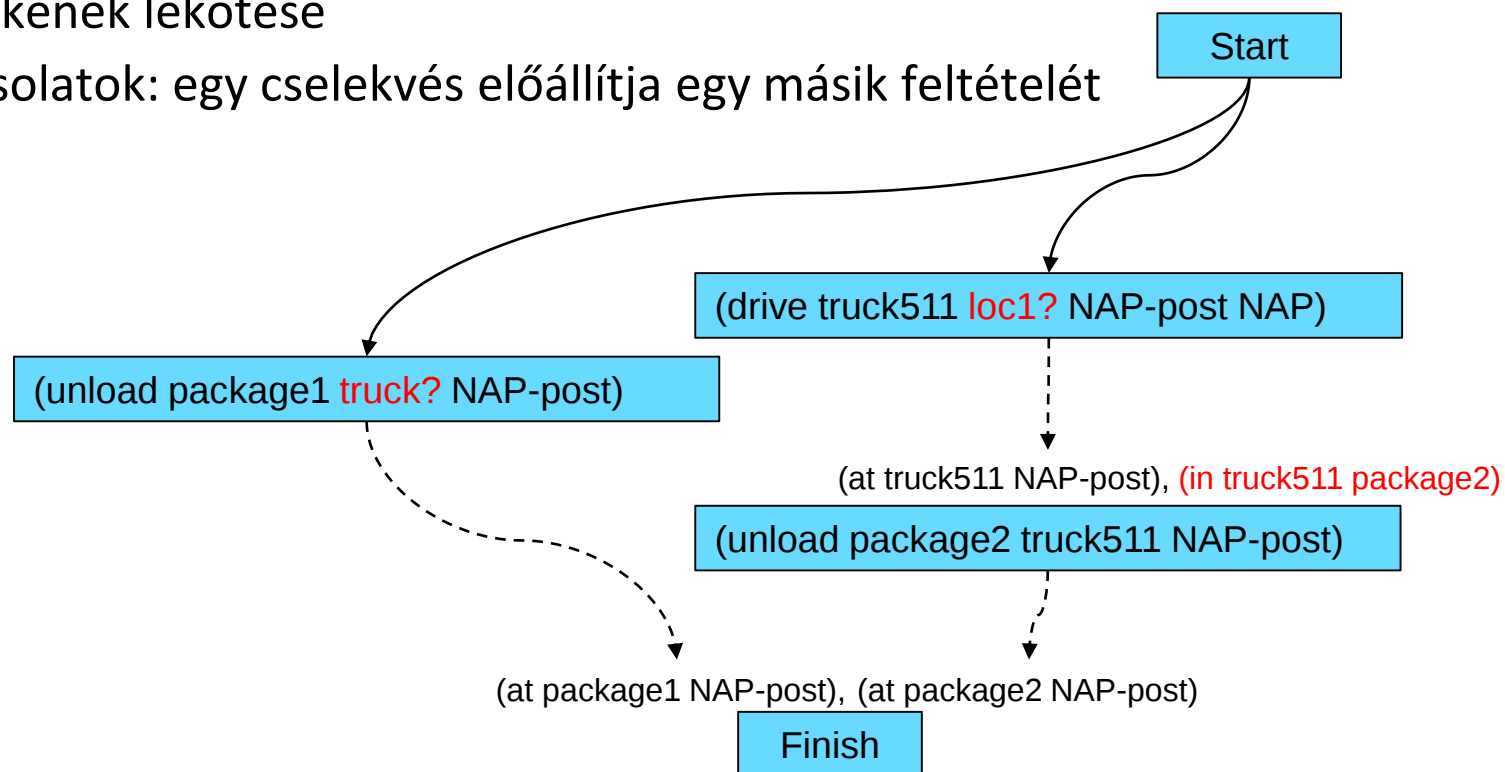
■ Keresés a tervek terében

- Minden keresési csomópont egy részleges terv
- Kezdetben az üres terv
- Ezt finomítjuk, mígnem elérünk egy teljesen specifikált tervet

Részben rendezett tervekészítő

■ A részben rendezett terv

- Cselekvések
- Cselekvés-párok közötti sorrendiség korlát (precedencia)
- Változók értékének lekötése
- Okozati kapcsolatok: egy cselekvés előállítja egy másik feltételét



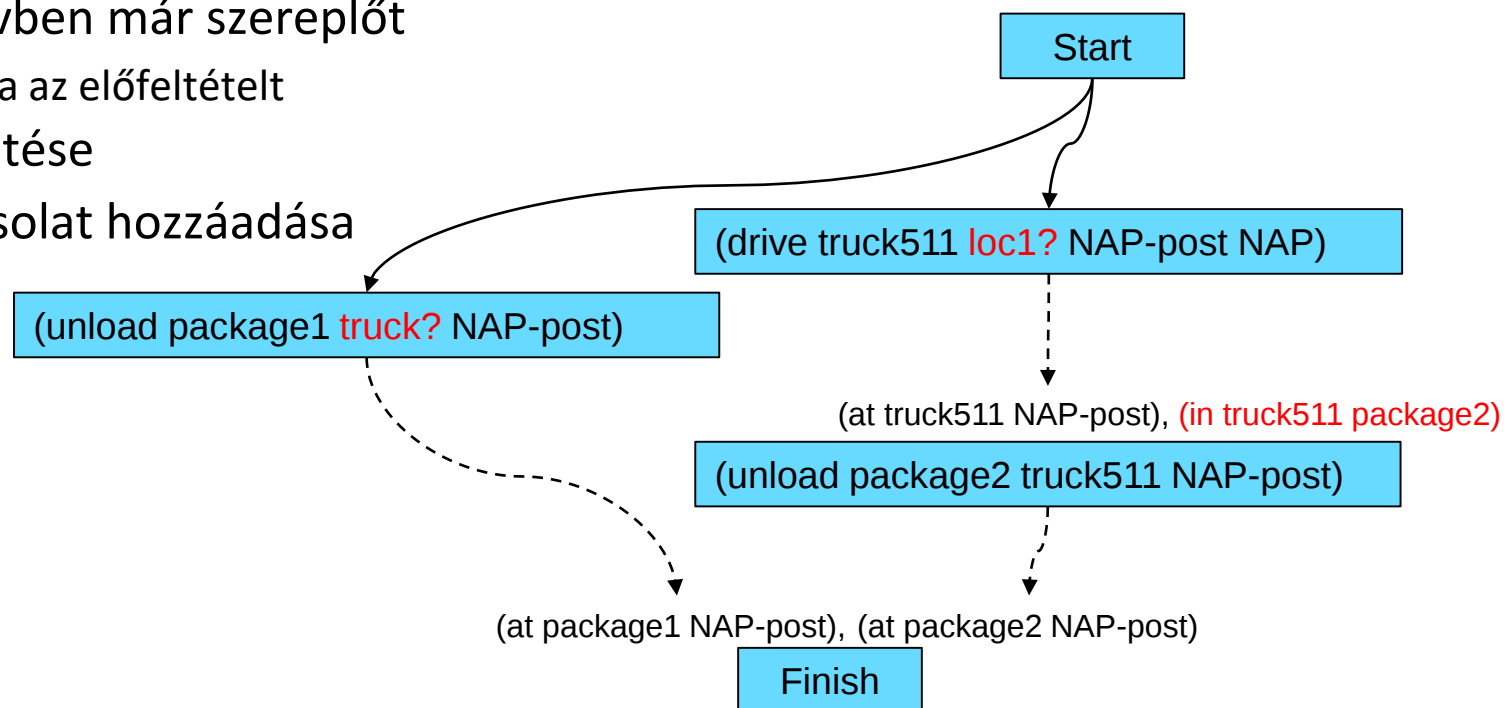
Hiányosságok a részleges tervben

■ Hiányosság: kielégítetlen előfeltétel

- Pl: (in truck511 package2)

■ Orvoslása: keressünk egy cselekvést

- Újat vagy tervben már szereplőt
 - Ami biztosítja az előfeltételt
- Változók lekötése
- Okozati kapcsolat hozzáadása



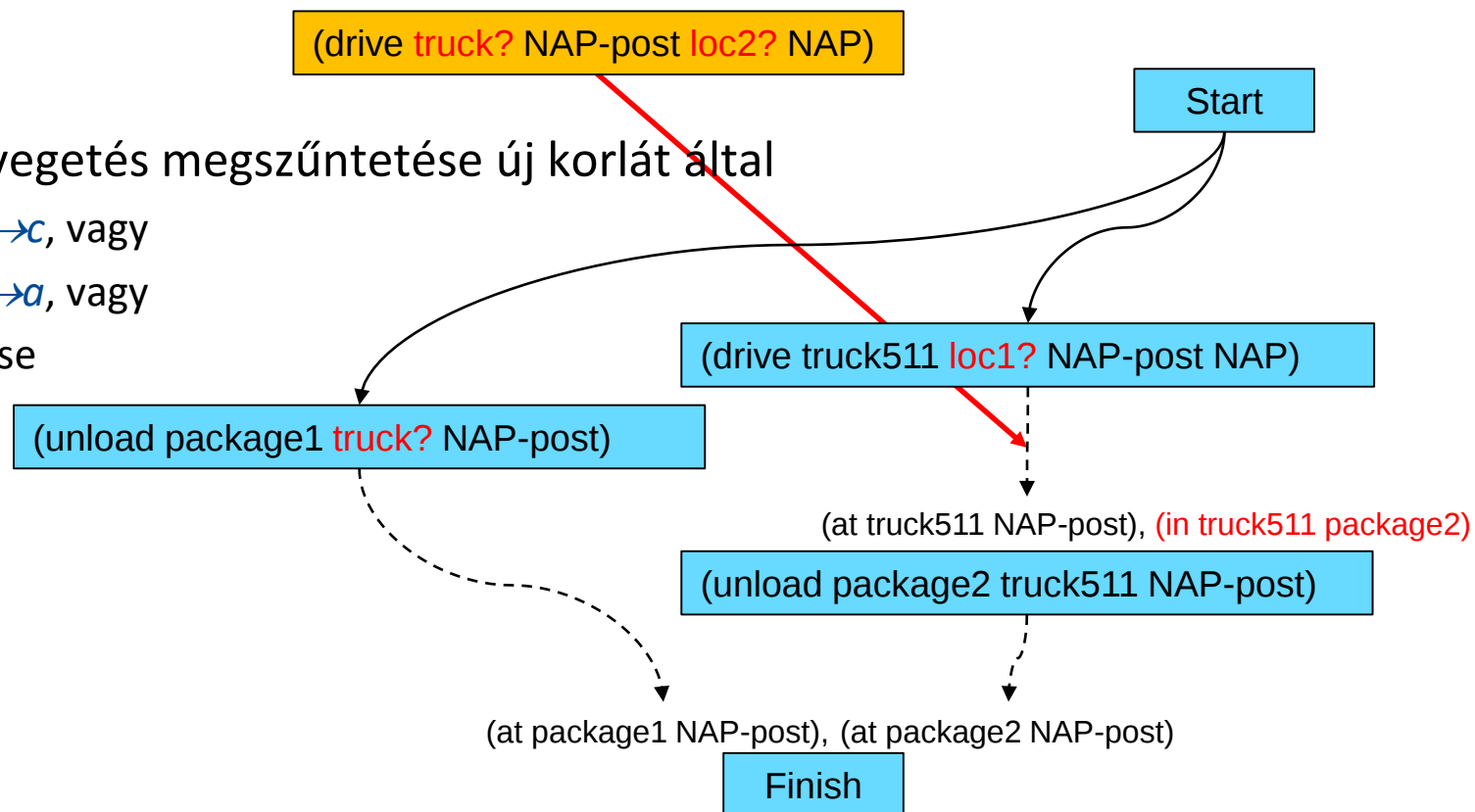
Hiányosságok a részleges tervben

■ Hiányosság: okozati kapcsolat fenyegetése

- Egy a cselekvés biztosítja b cselekvés p előfeltételét
- Egy cselekvés c képes törölni p -t

■ Orvoslása: fenyegetés megszüntetése új korlát által

- Precedencia: $b \rightarrow c$, vagy
- Precedencia: $c \rightarrow a$, vagy
- Változók lekötése



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
```

■ A keresés operátorai (*resolvers*)

- Cselekvés hozzáadása
- Precedencia korlát hozzáadása
- Változó értékének lekötése

PSP példa

- „Bevásárlás” feladat: szerezzünk fűrógépet (kapható a szerszámboltban, HWS), tejet és banánt (szupermarketben, SM)

- Cselekvések

- Start

- Hatás: At(Home) , sells(HWS,Drill) , Sells(SM,Milk) , Sells(SM,Banana)

- Finish

- Előfeltétel: Have(Drill) , Have(Milk) , Have(Banana) , At(Home)

- $\text{Go}(l,m)$

- Előfeltétel: $\text{At}(l)$

- Hatás: $\text{At}(m)$, $\neg\text{At}(l)$

- $\text{Buy}(p,s)$

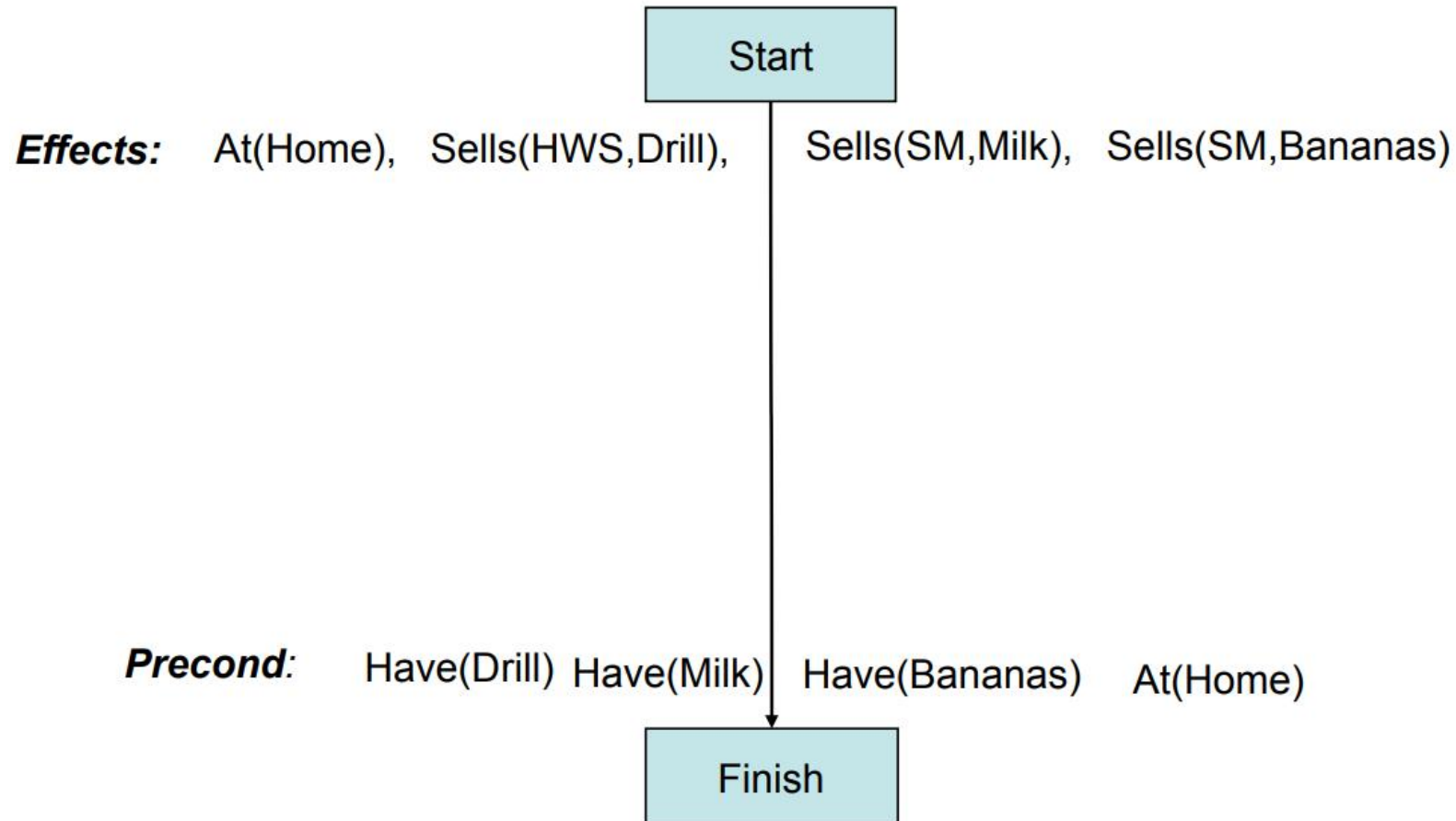
- Előfeltétel: $\text{At}(s)$, $\text{Sells}(s,p)$

- Hatás: $\text{Have}(p)$

Forrás: Russel & Norvig: AIMA; D. Nau: Automated planning: Theory and practice, UMD

PSP példa (2)

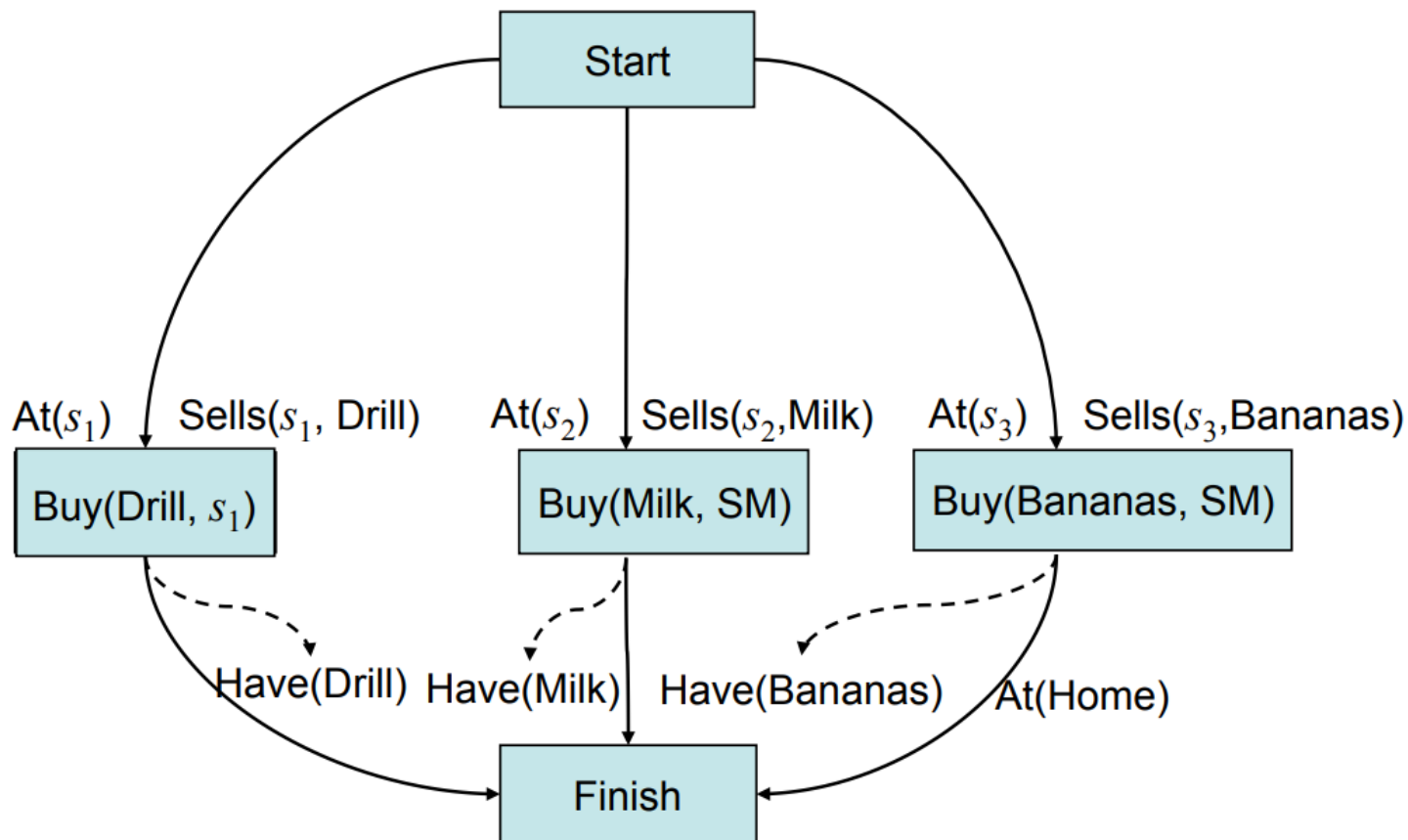
- Kezdeti terv: Start, Finish, köztük precedencia korlát



PSP példa (3)

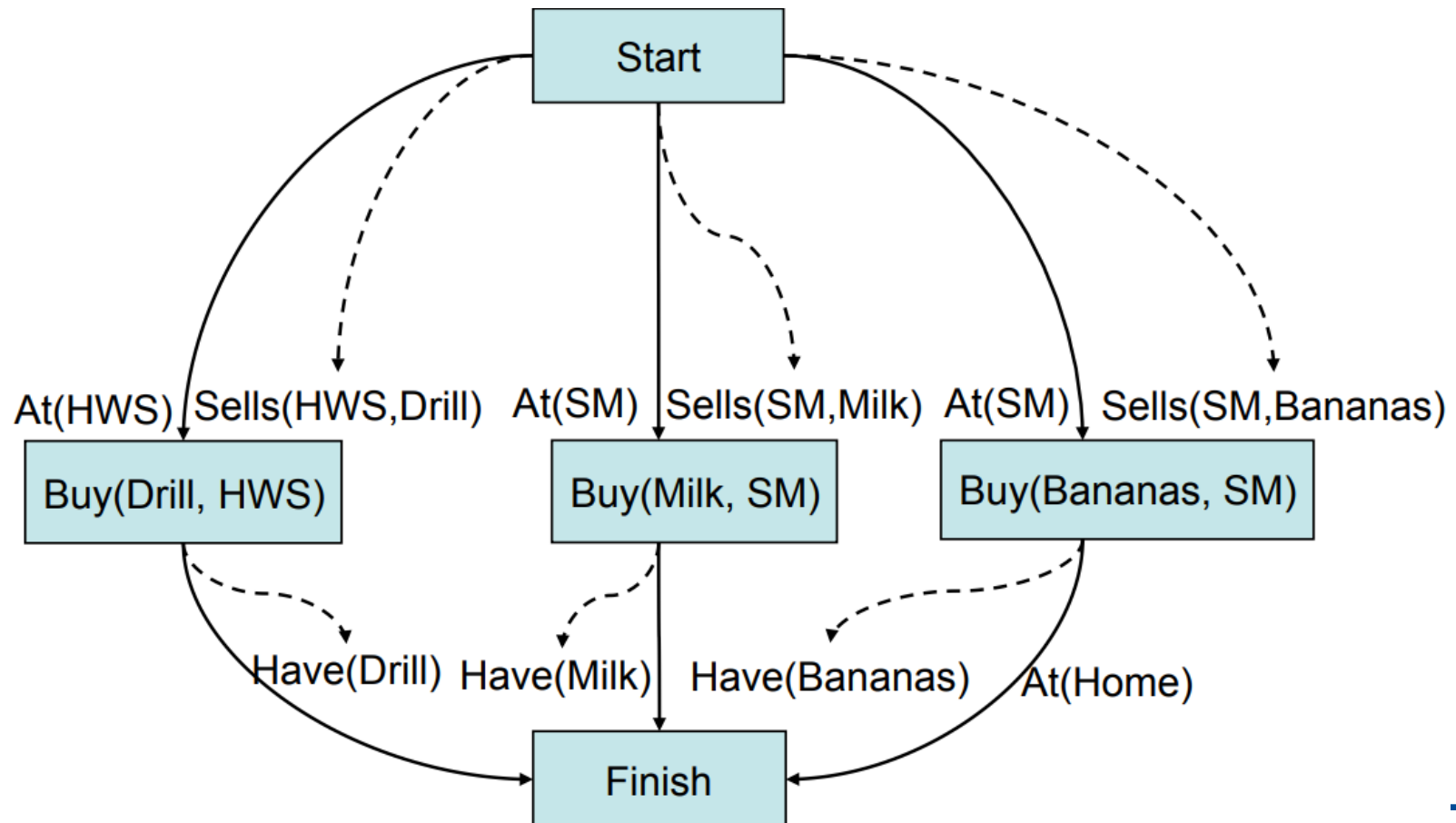
■ Finish-ben a Have(.) előfeltételek biztosítása egyértelmű

- Három lépésben három Buy cselekvés



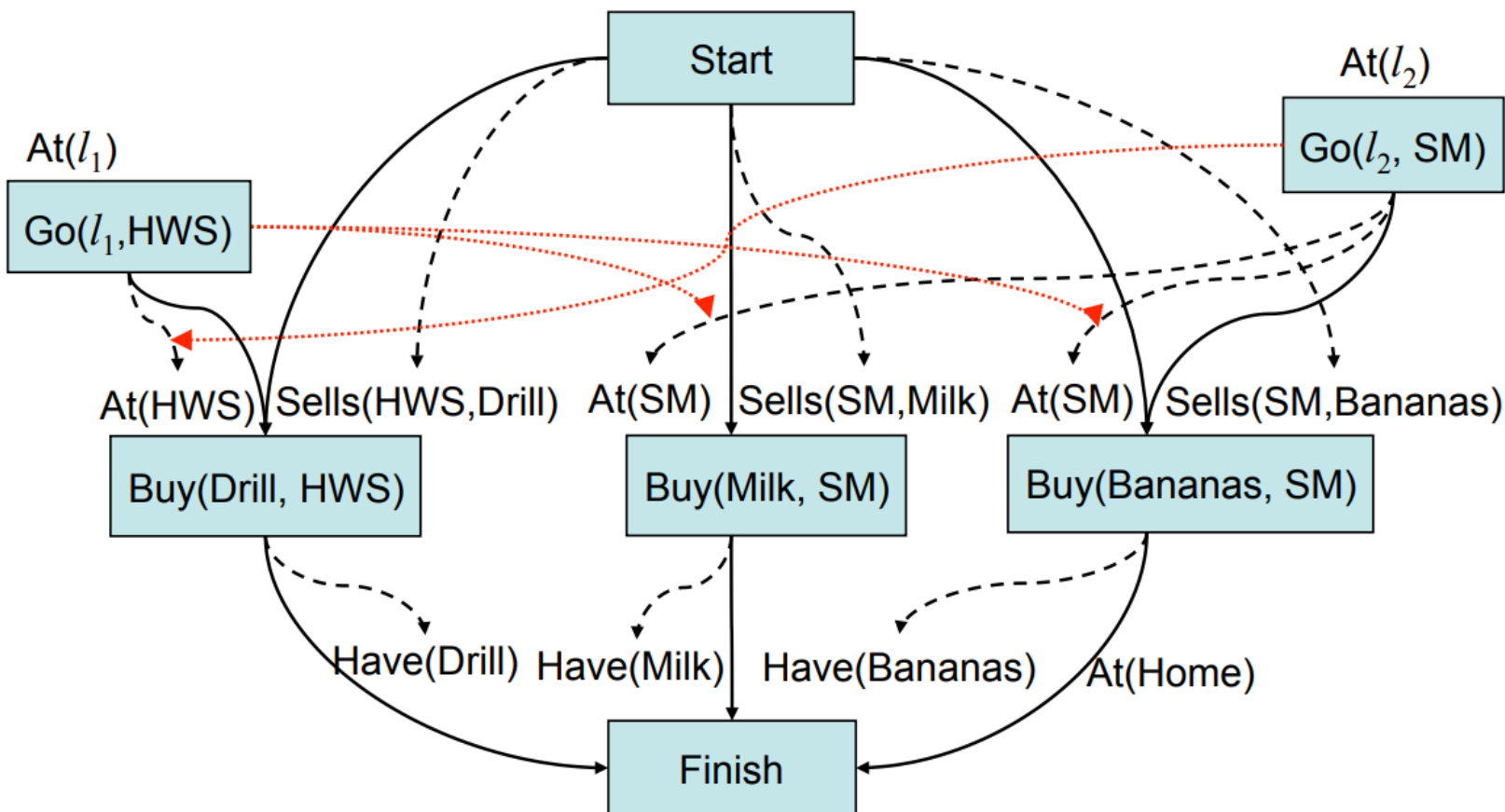
PSP példa (4)

- A Sells előfeltételek csak a Start-ból biztosíthatók



PSP példa (5)

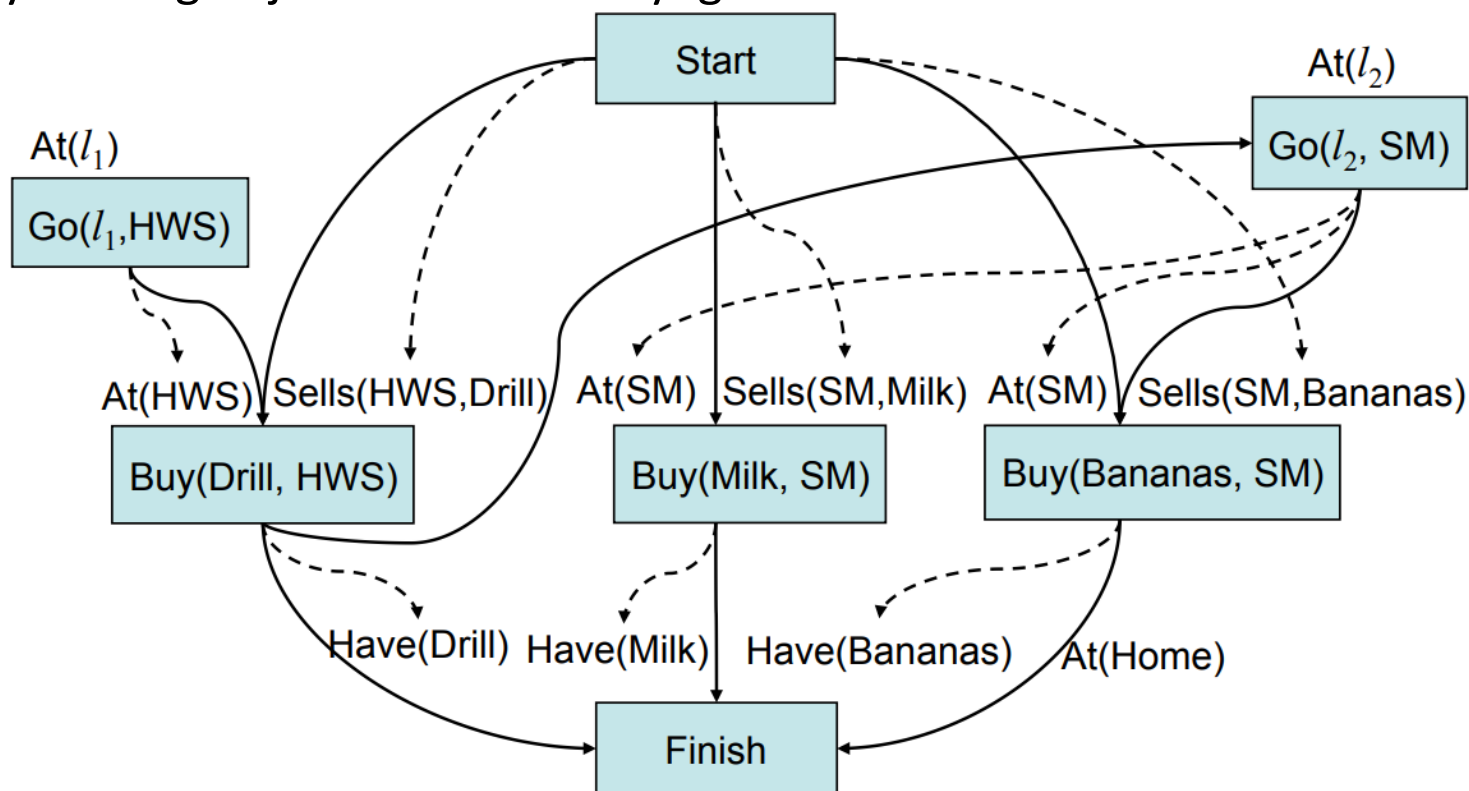
- Az $At(HWS)$ és $At(SM)$ csak $Go(.)$ cselekvéssel biztosítható
- Több okozati kapcsolat fenyegetve van!



PSP példa (6)

■ Hogyan orvosoljuk az $At(HWS)$ fenyegetettségét?

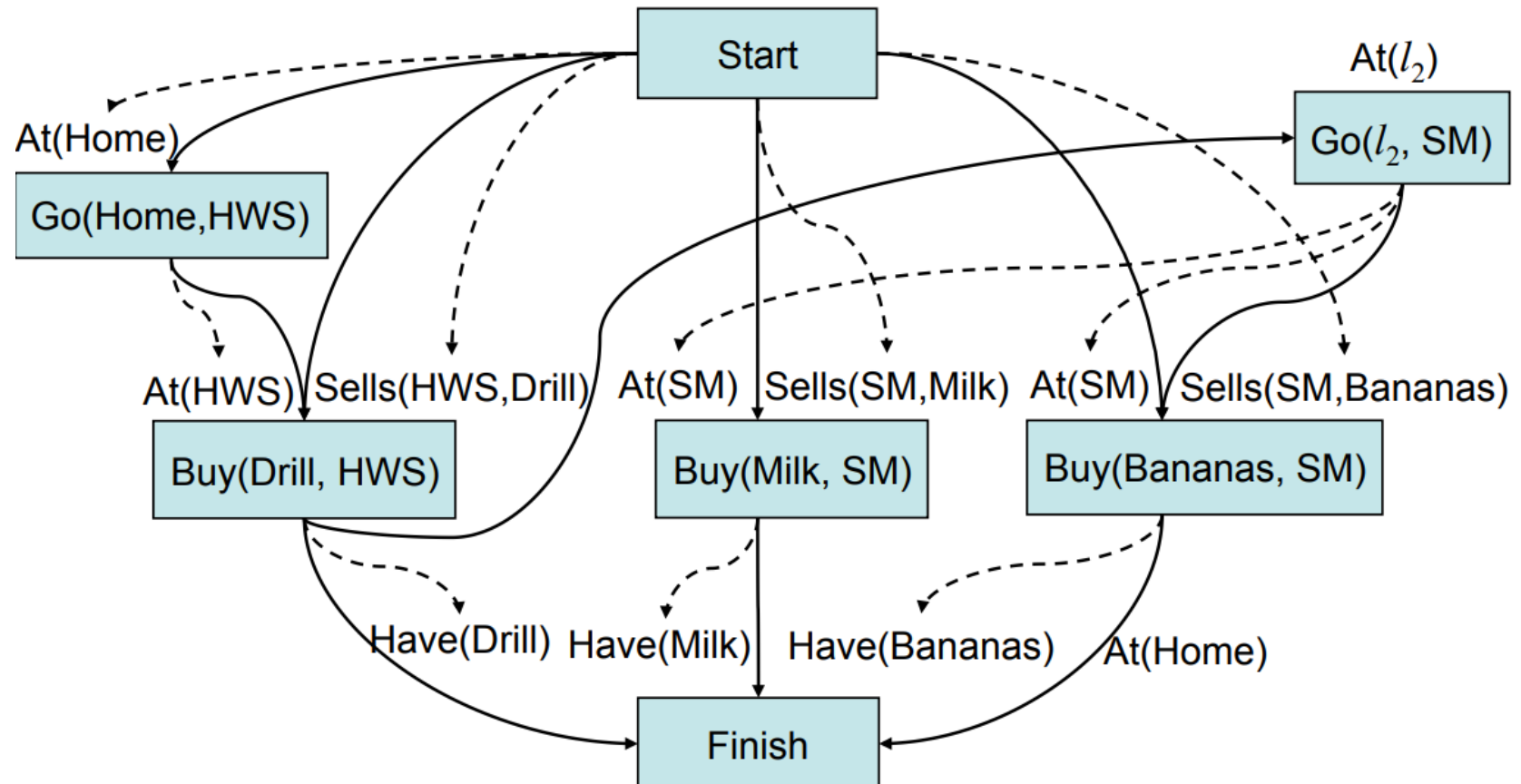
- Heurisztikus döntés: $Buy(Drill) \rightarrow Go(l_2, SM)$ precedencia
- Ez egyből megoldja a másik két fenyegetést is



PSP példa (7)

■ Hogyan biztosítjuk az $At(l_1)$ előfeltételt?

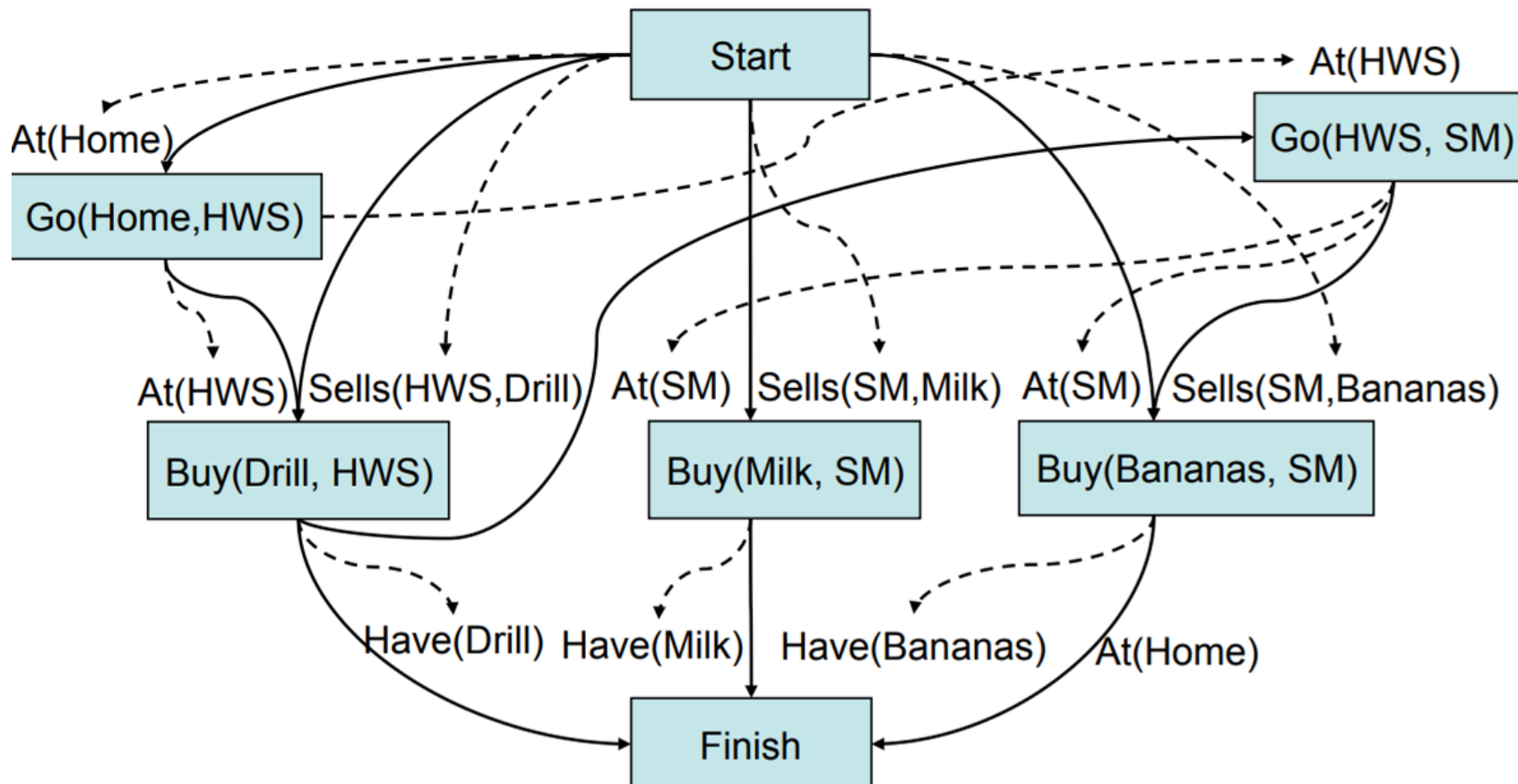
- A Start-ból, $l_1=Home$ változó lekötés hozzáadásával



PSP példa (8)

■ Hogyan biztosítjuk az $At(l2)$ előfeltételt?

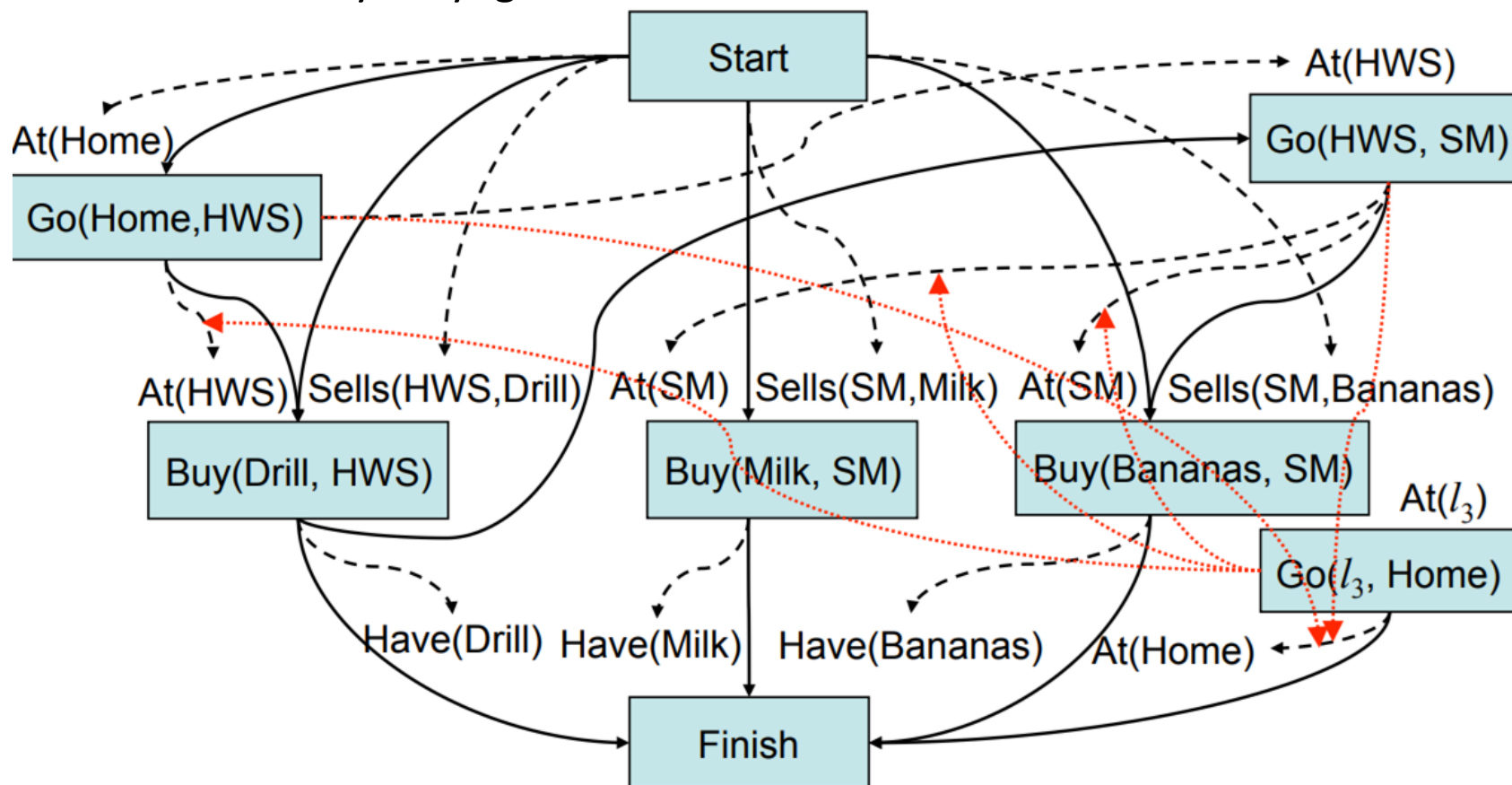
- A $Go(Home, HWS)$ -ból, $l2=HWS$ változó lekötés hozzáadásával



PSP példa (9)

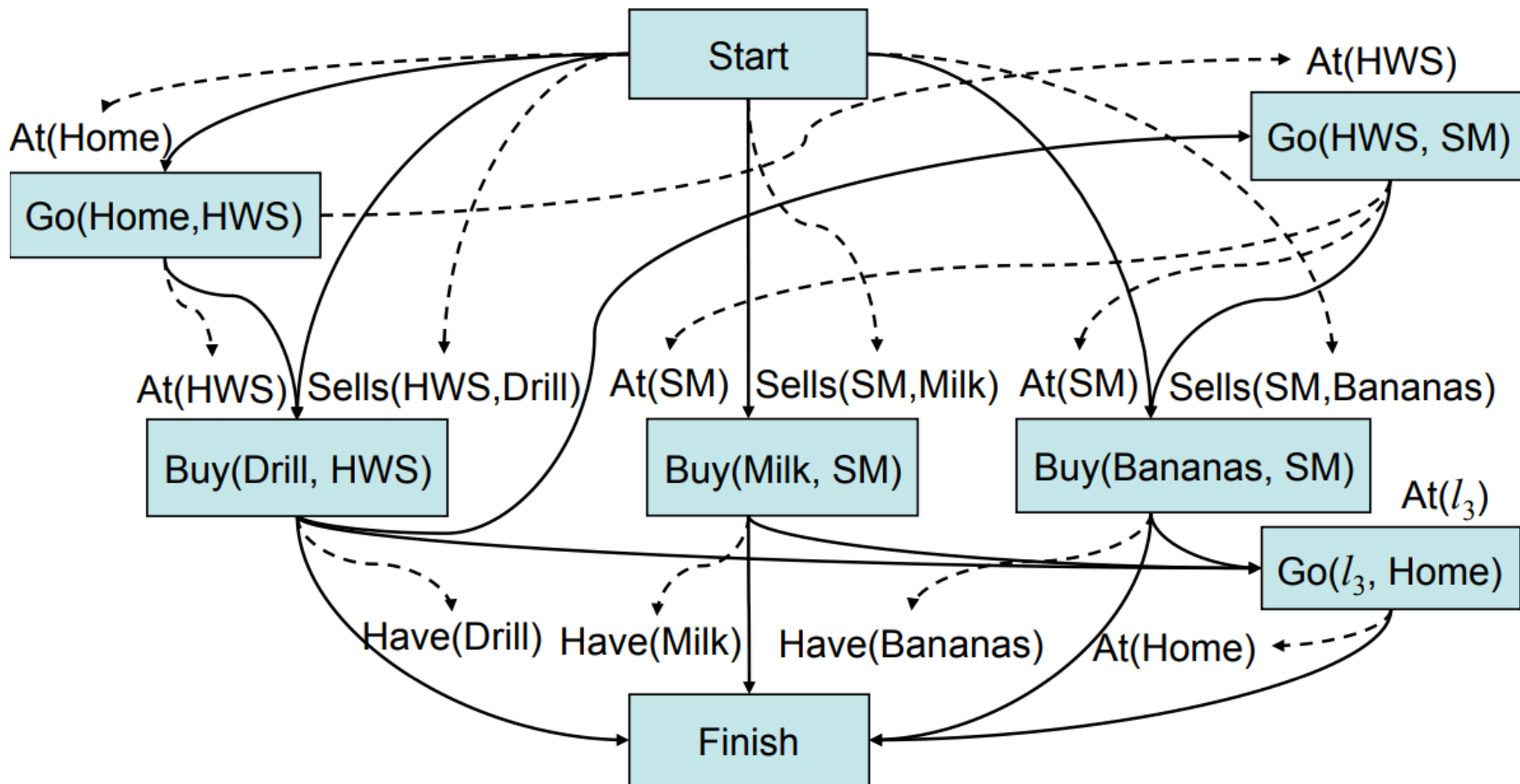
■ A Finish $\text{At}(\text{Home})$ előfeltételét új $\text{Go}(\cdot)$ cselekvéssel biztosítjuk

■ Ez létrehoz néhány fenyegetést



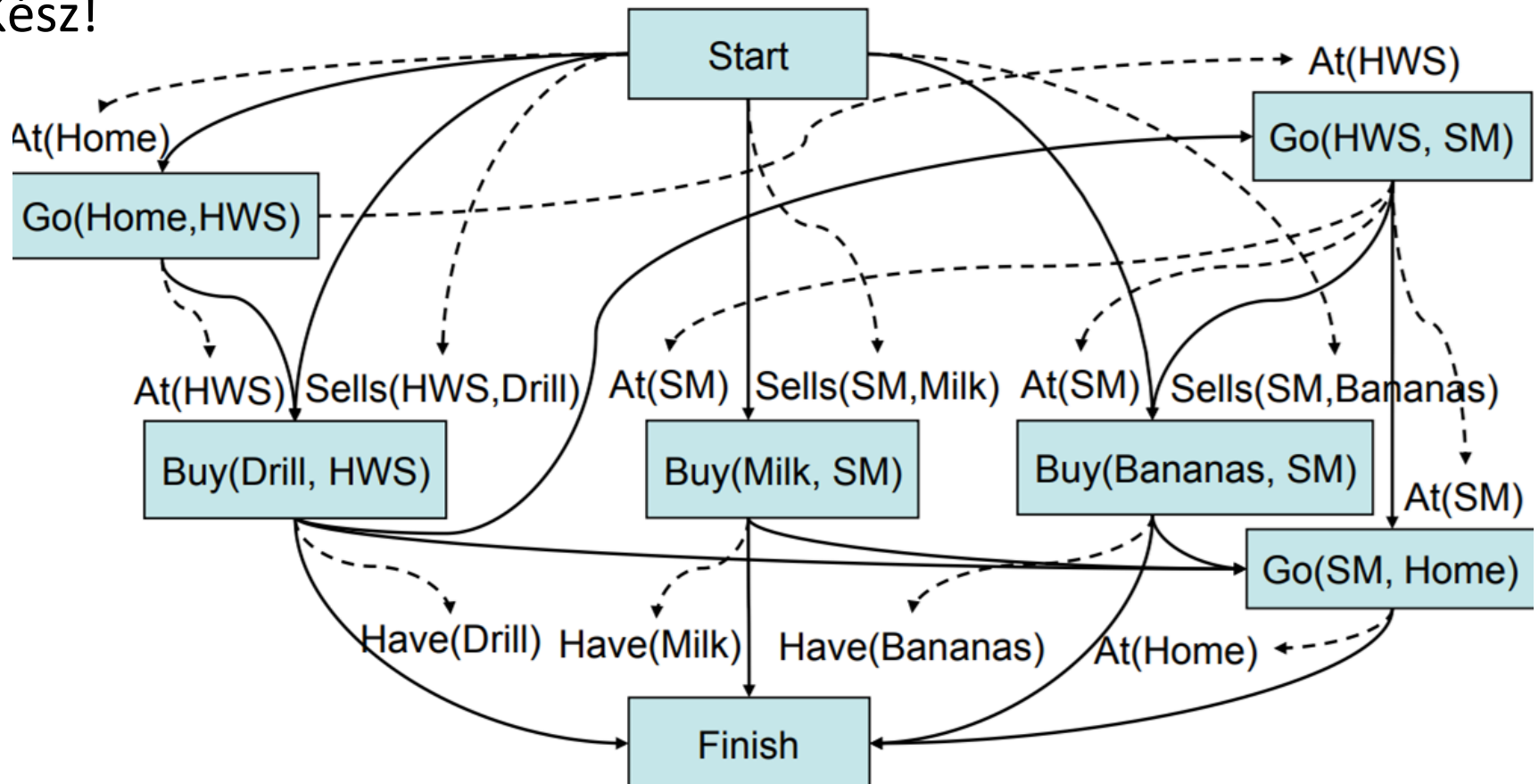
PSP példa (10)

- Az At(HWS) és At(SM) fenyegetettségét új precedenciákkal orvosoljuk



PSP példa (10)

- Az $At(l3)$ előfeltételt az $l3=SM$ lekötés biztosítja a $Go(HWS, SM)$ -ből
- Kész!



Heurisztikák a tervek terében

■ Heurisztikák a tervek terében

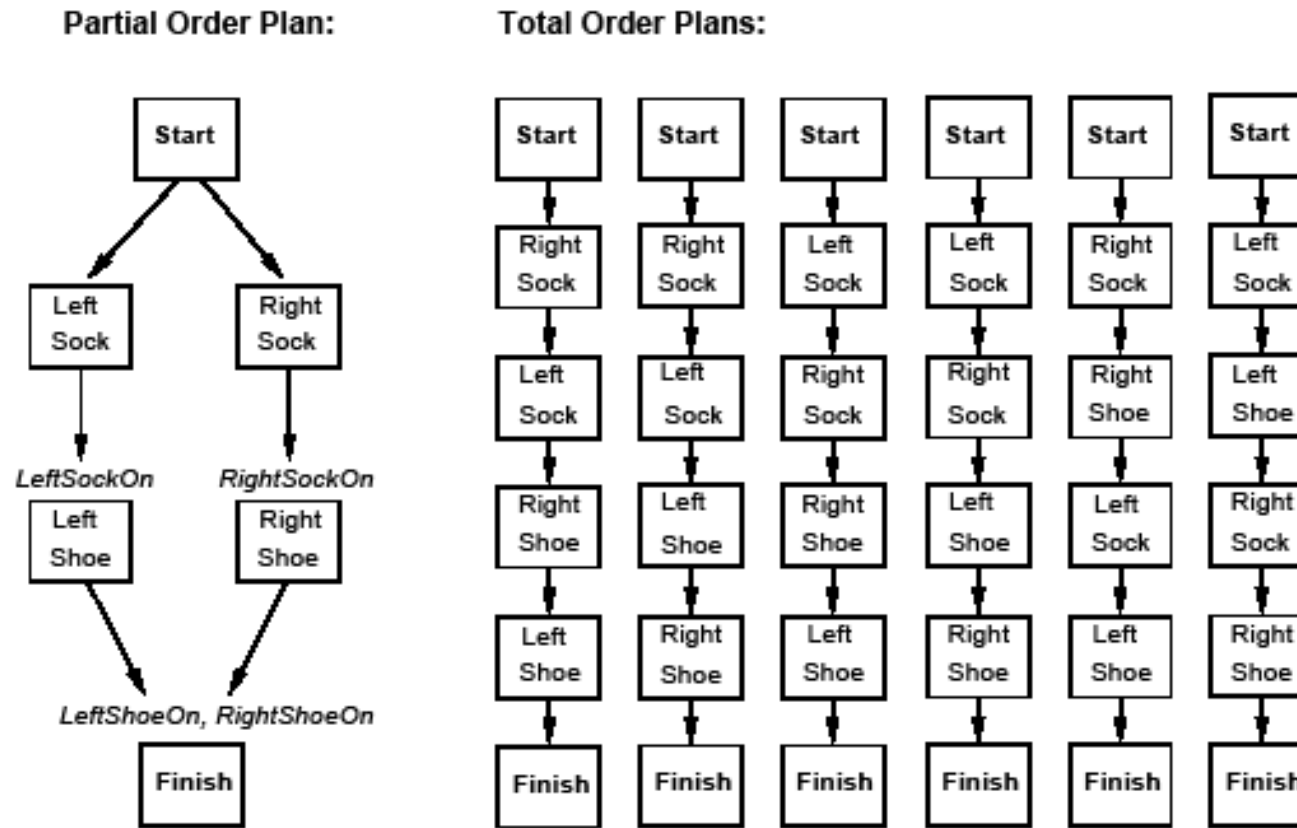
- Nehezebb jó heurisztikát találni, mint állapot térben
- Egy lehetséges heurisztika: nyitott előfeltételek száma
 - Egy új operátor több előfeltételt kielégíthet (felülbecslés)
 - Interferencia különböző operátorok hatásai között (alulbecslés)

■ Tervtér vs. állapot tér: a tervezés hatékonysága

- Sokáig nyitott kérdés: kevesebb redundancia vs. jobb heurisztikák
- Az utóbbi időben az állapot térbeli tervezés erősen nyeresre áll, a jobb heurisztikák miatt

Részben rendezett tervkészítés

■ Tervek a zokni-cipő problémájára



A PDDL cselekvéstervezés korlátai

- Optimalizálás, numerikus képességek
 - Alapvetően a cél egy megengedett terv
 - Korlátozott lehetőségek az optimalizálásra, pl.
 - Klasszikus tervezésben a terv hossza
 - Egyéb egyszerűek, pl. átfutási idő (PDDL :durative-actions)
- Numerikus változók egyáltalán nem megengedettek
- Korlátozott lehetőségek az numerikus függvények kezelésére az operátorokban és a célfüggvényben
 - PDDL :fluents kiterjesztés

Found Plan (output)

(fly-airplane apn2 apt7 apt1)
(fly-airplane apn1 apt8 apt1)
(fly-airplane apn1 apt1 apt2)
(fly-airplane apn1 apt2 apt3)
(fly-airplane apn1 apt3 apt4)
(fly-airplane apn1 apt4 apt5)
(fly-airplane apn1 apt5 apt6)
(fly-airplane apn2 apt1 apt2)
(fly-airplane apn2 apt2 apt3)
(fly-airplane apn2 apt3 apt5)
(fly-airplane apn2 apt5 apt6)
(drive-truck tru8 pos8 apt8 cit8)

apt8 → apt1 → apt 2 → apt3 → apt4 → apt5 → apt6 **???**!

A PDDL cselekvéstervezés korlátai (2)

■ Számítási hatékonyság

- Komplex, nagyméretű feladatokon önmagában nem hatékony
- Nehezen integrálható feladat-specifikus heurisztikákkal és keresési technikákkal
- Ilyen feladathoz jobban megéri dedikált megoldót fejleszteni

■ Tervkészítés bizonytalan környezetben

- A bemutatott tervezők nem kezelik a bizonytalanságot
- Jelentős eredmények bizonytalan környezet esetére is
 - Eshetőségi tervekészítés, Markov modellek, stb.

Összegzés

- Általános, alkalmazásfüggetlen cselekvéstervezés
 - Egy letisztult, az elsőrendű logikán alapuló nyelven
- Deklaratív megközelítés
 - Nincs szükség algoritmus leprogramozására
- Gyors fejlesztés
 - Egyszerű vagy kisméretű feladatok megoldására
 - „Rapid prototyping”
- De számottevő gyakorlati korlátok
 - A nyelv kifejezőképessége
 - Számítási hatékonyság

Ajánlott eszközök, olvasmányok

Online, felhő-alapú PDDL cselekvéstervező:

<http://solver.planning.domains/>

Korszerű Linux-os PDDL cselekvéstervező:

<http://www.fast-downward.org/>

Olvasmány egy legendás heurisztikus cselekvéstervező algoritmusról:

J. Hoffmann, B. Nebel, The FF planning system: Fast plan generation through heuristic search, *Journal of Artificial Intelligence Research* 14:253-302, 2001.

<https://arxiv.org/abs/1106.0675>

Dana Nau (University of Maryland) cselekvéstervezés kurzusának anyagai:

<https://www.cs.umd.edu/~nau/planning/slides/>

Kovács András
akovacs@sztaki.hu