Artificial intelligence -Project 3

-Planning Domain Definition Language -

Dunca Denisa Mihaela

10/01/2022

**Cuprins**

[1. Introducere 3](#_Toc92746126)

[2. Domeniul problemei 3](#_Toc92746127)

[3. Prima problemă - First House 4](#_Toc92746128)

[4. A doua problemă - Second House 5](#_Toc92746129)

[5. A treia problemă - Third House 6](#_Toc92746130)

[6. Concluzii 7](#_Toc92746131)

[7. Referințe 7](#_Toc92746132)

# Introducere

În cadrul acestui proiect am implementat o problema specifică limbajului PDD, problemă care reușește să scoată în evidență modul prin care acest tip de limbaj funcționează. În cele ce urmează vor fi prezentate domeniul problemei de tip grid și cele 3 aplicații ale acestuia.

Problema este despre un robot ”cleaner” care are ca si goal principal să scoată tot gunoiul din case atunci când o curăță. Acesta are la dispoziție două unelte pe care le poate utiliza, aspiratorul pentru a curăța camerele prăfuite și cheile pentru a deschide camerele încuiate astfel încât să ajungă la gunoi. Odată ajuns la gunoi, roboțelul de curățat îl pune în sac și îl duce afară la reciclat.

# Domeniul problemei

Domeniul numit ”messy\_house” prezintă predicatele și acțiunile puse la dispoziție in problemă. Predicatele sunt:

* at – pentru a arăta unde se află un obiect
* next – pentru a arăta lângă cine se află un obiect
* holding – pentru a vedea ce ține cleaner-ul în mână (aspirator sau chei)
* empty-hands – verifică dacă nu are nimic în mână
* half-open – verificare pentru cameră dacă este deschisă
* door-between – arată că este perete între două camere
* dust-in – arată ca e praf în cameră și trebuie aspirată
* cleaner – roboțelul care curăță
* trash – goal-ul, gunoiul care trebuie dat afară
* key – cheile cu care se deschid camerele
* vacuum – aspiratorul cu care se curăță camerele prăfuite
* place – arată un loc
* safe-place – locul sigur unde trebuie dus goal-ul
* recycle – când gunoiul este scos afară este reciclat
* open – arată că e deschisă camera
* empty-sack – golește sacul cu gunoi
* in-sack – se află în sac

În cele ce urmează se va discuta fiecare acțiune în parte.

* move: actiune care primește ca parametrii cleaner-ul, locația actuala a acestuia și destinația. Trebuie ca să existe un loc unde acesta să se mute și de asemenea să nu fie zid sau praf.
* recycle-trash: primește cleaner-ul, locul unde trebuie dus gunoiul și gunoiul. Are ca precondiții cei trei parametrii să existe și gunoiul să se afle în sac.
* grab\_tool: parametrii sunt cleaner-ul, un loc și tool-ul care poate fii chei sau aspirator, are ca precondiții cleaner-ul să aibă mâinile goale pentru a putea să ia o unealtă noua.
* drop\_tool: la fel ca si la grab\_tool, dar se schimbă precondiția, roboțelul trebuie sa aibă o unealtă în mână pentru a putea să îi dea drop.
* put\_in\_sack: primește un cleaner, un loc și gunoi. Sacul trebuie să fie gol și are ca rezultat punerea în sac a gunoiului de către cleaner.
* open\_room: primește un cleaner, un parametru care arată unde se află cleane-ul când vrea să deschidă ușa, un parametru care arată unde intrăm după ce deschidem ușa și un parametru care arată că are cheile la el. Are ca precondiții să existe o ușă între cele două camere, aceasta să nu fie deschisă și cleaner-ul să aibă în mână cheia. Ca efect deschide ușa dintre cele două camere.
* vacuum\_dust: primește ca și parametri cleaner-ul, locul unde acesta se află, camera care trebuie curățată și aspiratorul. Ca precondiții trebuie să fie prăfuită camera din al treilea parametru și cleaner-ul să aibă în mână aspiratorul. Are ca rezultat curățarea camere, nu mai e prăfuită.

# Prima problemă - First House

În prima problemă sunt date ca și obiective principale cele două variabile trash1 și trash2 și un grid în care se află diferite camere prăfuite, uși închise, un aspirator și o cheie. Scena ar arăta ca în figura1.

O imagine care conține text, carte de vizită

Descriere generată automat

Figură 1 Grid pentru problema 1

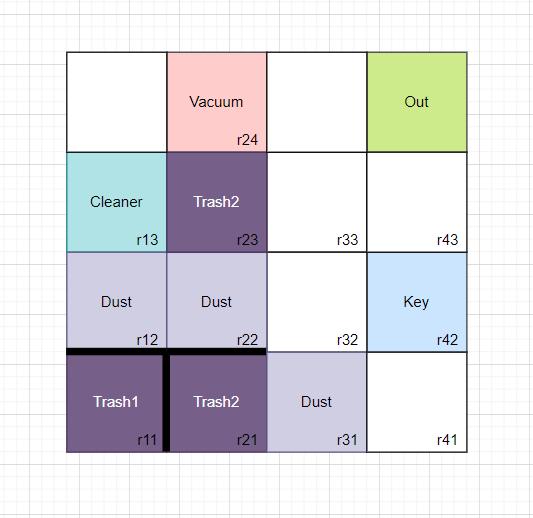
Prima dată este declarat domeniul pe care este bazată problema, apoi obiectele (cleaner, trash1, trash2, key, vacuum, ieșirea și coordonatele gridului). În partea de init se oferă informații legate de unde se află fiecare obiect important, camerele murdare, cum sunt aranjate coordonatele matricei pe care lucrăm și între ce camere sunt uși. Goal-ul este să se recicleze trash1 și trash2.

Planul generat respectă pe scurt următorii pași:

* Cleaner-ul se îndreaptă spre cheie
* Deschide camera în care se află trash1
* Pune trash1 în sac și îl duce afară la reciclat
* Lasă cheia și ia aspiratorul pentru a curăța camera prăfuită
* Apoi ia cheia și deschide ușa în care se află trash2
* Pune trash2 în sac și îl duce afară la reciclat

# A doua problemă - Second House

În a doua problemă sunt date ca și obiective principale cele trei variabile trash1, trash2 și trash3 și un grid în care se află diferite camere prăfuite, uși închise, un aspirator și o cheie. Scena ar arăta ca în figura2.



Figură 2 Grid-ul pentru second house

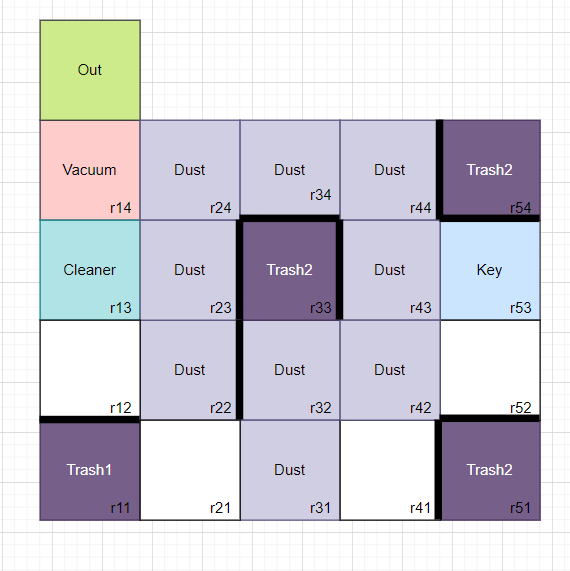
Prima dată este declarat domeniul pe care este bazată problema, apoi obiectele (cleaner, trash1, trash2, trash3, key, vacuum, ieșirea și coordonatele gridului). În partea de init se oferă informații legate de unde se află fiecare obiect important, camerele murdare, cum sunt aranjate coordonatele matricei pe care lucrăm și între ce camere sunt uși. Goal-ul este să se recicleze trash1, trash2 și trash3.

Planul generat respectă pe scurt următorii pași:

* Cleaner-ul ia trash1, îl pune în sac și îl duce afară
* Cleaner-ul ia cheia, dar vede că nu poate să treacă de dust, așa că lasă cheia și merge să ia aspiratorul
* Cu aspiratorul curăță camera de la r31
* Pune trash2 în sac și îl duce afară la reciclat
* Apoi ia cheia și deschide ușa pentru a intra în camera r11
* Pune trash3 în sac și îl duce afară

# A treia problemă - Third House

În a treia problemă sunt date ca și obiective principale cele patru variabile trash1, trash2, trash3 și trash4 și un grid în care se află diferite camere prăfuite, uși închise, un aspirator și o cheie. Scena ar arăta ca în figura2.



Figură 3 Grid-ul pentru third house

Prima dată este declarat domeniul pe care este bazată problema, apoi obiectele (cleaner, trash1, trash2, trash3, trash4, key, vacuum, ieșirea și coordonatele gridului). În partea de init se oferă informații legate de unde se află fiecare obiect important, camerele murdare, cum sunt aranjate coordonatele matricei pe care lucrăm și între ce camere sunt uși. Goal-ul este să se recicleze trash1, trash2, trash3 și trash4 .

Planul generat respectă pe scurt următorii pași:

* Cleaner-ul oservă că nu poate să treacă de dust așa că merge la aspirator
* Aspiră camerele r24, r22, r32, r31, r42
* Ia trash3 și îl duce afară
* Apoi lasă aspiratorul și duce trash4 afară
* Apoi ia cheia și deschide ușa pentru a intra în camera r54
* Deschide și camera r51
* Pune trash1 în sac și îl duce afară
* Pune trash2 în sac și îl duce afară

# Concluzii

# Referințe

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

<https://www.diagrams.net/>

<https://moodle.cs.utcluj.ro/pluginfile.php/136506/mod_resource/content/1/LabIA.pdf>