Експертски системи

(СИ4ЕС)

Реализовање СТРИПС алгоритма планирања

студент
Александар Абу-Самра, 0252/08

професор
Бошко Николић

1. Опис проблема

Овај пројекат је представљен у виду сликовитог проблема света блокова. Наиме, свет блокова се састоји од две врсте блокова - црних и белих - које може да помера роботска рока. Они могу бити спуштени или подигнути, наслагани један на други, или ненаслагани. У случају да се црни блок нађе изнад белог, бели може “попримити” боју и такође постати црн.
Њихова почетна стања могу бити дата следећим предикатским описима:
On (x, y) - блок x је на блоку y
OnTable (x) - блок x је спуштен (на столу)
Clear (x) - нема ништа на блоку x
Holding (x) - рука робота држи блок x
ArmEmpty - рука робота је празна
Black (x) - блок x је црн
White (x) - блок x је бео

Листа могућих акција којима располаже робот:
assimilate (x,y) - блок y поприма боју од блока x
stack (x,y) - ставља блок x на блок y
unstack (x,y) - скида блок x са блока y
pickup (x) - узима блок x са стола
putdown (x) - ставља блок x на сто

Пројекат користи СТРИПС алгоритам планирања при манипулисању овим блоковима. Циљ пројекта је успешно реализовање овог алгоритма уз коришћење програмских језика *C++* или *Java* и боље упознавање студента са алгоритмима за аутоматско планирање.

2. Опис предложеног решења

Овај пројекат је комплетно рађен у програмском језику *Java* уз подршку из *NetBeans* развојног окружења.

Блокови, предикати и акције су представљени засебним класама. Класе блокова у себи садрже информацију о ком блоку се ради. Класе предиката у својим пољима чувају опис предиката и блокове на које се односе. Акције, са друге стране, поред описних и функционалних поља имају и извршну методу која врши све потребне кораке при раду роботске руке са блоковима.

Окружење је свесно са којим блоковима располаже и неће допустити манипулацију непостојећим блоковима.

Класа *BlockWorld* садржи реализацију главне логике и скоро све помоћне методе за управљање системом. Она држи и евиденцију свих потребних података у следећим структурама:
Vector currentState - садржи опис тренутног стања
Stack desiredStack - динамички стек на коме се смештају циљни предикати и акције
Stack originalDesiredStack - циљно стање
LinkedList actionsList - листа предузетих акција
LinkedList blocksList - листа расположивих блокова
,
Над овим структурама се примењује класична логика СТРИПС алгоритма, уз неке додате провере, забране, и оптимизације за конкретан проблем.

Управо то је и највећа мана овог решења. СТРИПС алгоритам, иако сасвим применљив у великом броју проблема аутоматског планирања, уз предикате и акција дате овим пројектом, није увек способан да изнесе валидно решење. Зато је потребно претпоставити неке могуће случајеве, и одступити од строгог придржавања алгоритма, што је поступак подложан грешкама.
У предложеном решењу је отклоњен велики део ових недостатака, што је потврђено успешним проласком кроз све тест примере, али не можемо бити стопроцентно сигурни да ли ће се грешка јавити при некој новој, специфичној ситуацији.

3. Опис корисничког интерфејса

Графички кориснички интерфејс је интуитиван и садржи неке основне алатке помоћу којих се манипулише ситуацијама. При стартовању апликације, кориснику се представља стандардан фрејм апликације, са предефинисаним простором за следеће операције и панеле:

3.1. Праћење тока алгоритма и тренутног стања стекова

Ова секција се налази у горњем левом углу и састоји се из 3 листе - Прва листа представља тренутно стање блокова у систему, друга - стек над којим манипулише алгоритам, а трећа - предузете акције.

3.2. Контролни дугмићи

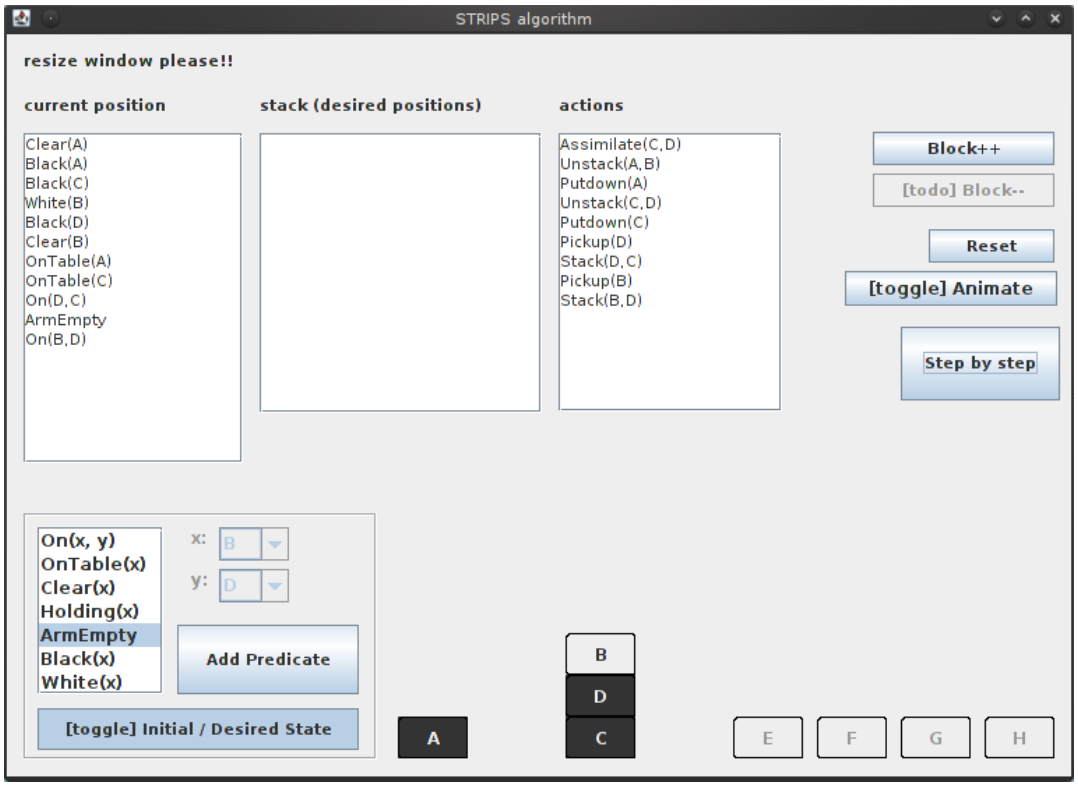
Главна контрола програма се налази овде - у горњем десном углу апликације. Може се додати нови блок у систем, затражити анимирани приказ дешавања у систему, или контрола корак по корак. На крају, корисник може затражити и ресетовање система и повратак у почетно стање.

3.3. Панел за дефинисање проблема

У доњем левом углу се налази панел за описивање почетног и циљног стања блокова у систему. Корисник бира блокове над којима хоће да примени предикат и додаје их у листу. Помоћу дугмета "*[toggle] Initial / Desired State*" корисник се пребацује из мода за описивање почетног стања у мод за описивање циљног стања - и обрнуто.

3.4. Панел за анимацију (графичко праћење извршених акција)

Графички приказ тренутног стања блокова у систему се може пратити у доњем десном углу апликације. При свакој промени вектора *currentState*, може се, у виду једноставних црних и белих влокова, видети и визуелни приказ и тако лакше пратити ток алгоритма.



Скриншот апликације

4. Тест примери

тест број	1	2	3	4	5
почетна позиција	OnTable(A) Clear(A) Black(A) ArmEmpty	OnTable(A) Black(A) On(B,A) White(B) Clear(B) ArmEmpty	OnTable(A) OnTable(B) OnTable(C) Clear(A) Clear(B) Clear(C) White(A) White(B) Black(C) ArmEmpty	OnTable(B) OnTable(D) On(A,B) On(C,D) Clear(A) Clear(C) Black(A) Black(C) White(B) White(D) ArmEmpty	OnTable(E) On(D,E) On(C,D) On(B,C) On(A,B) White(A) White(B) White(C) White(D) Black(E) Clear(A) ArmEmpty
жељена позиција	Clear(A) Holding(A) Black(A)	OnTable(B) Black(B) On(A,B) Clear(A) Black(A) ArmEmpty	OnTable(A) On(C,B) On(B,A) Black(A) Black(C) White(B) Clear(C) ArmEmpty	OnTable(C) OnTable(A) On(B,D) On(D,C) Clear(B) Clear(A) Black(A) Black(D) Black(C) White(B) ArmEmpty	OnTable(A) OnTable(C) OnTable(D) On(B,A) On(E,D) Clear(B) Clear(C) Clear(E) Black(A) Black(B) Black(C) Black(E) ArmEmpty
листа акција	Pickup(A)	Unstack(B,A) Putdown(B) Pickup(A) Stack(A,B) Assimilate(A,B)	Pickup(C) Stack(C,A) Assimilate(C,A) Unstack(C,A) Putdown(C) Pickup(B) Stack(B,A) Pickup(C) Stack(C,B)	Assimilate(C,D) Unstack(A,B) Putdown(A) Unstack(C,D) Putdown(C) Pickup(D) Stack(D,C) Pickup(B) Stack(B,D)	Unstack(A,B) Putdown(A) Unstack(B,C) Putdown(B) Unstack(C,D) Putdown(C) Unstack(D,E) Putdown(D) Pickup(E) Stack(E,C) Assimilate(E,C) Unstack(E,C) Putdown(E) Pickup(C) Stack(C,B) Assimilate(C,B) Unstack(C,B) Putdown(C) Pickup(B) Stack(B,A) Assimilate(B,A) Pickup(E) Stack(E,D)