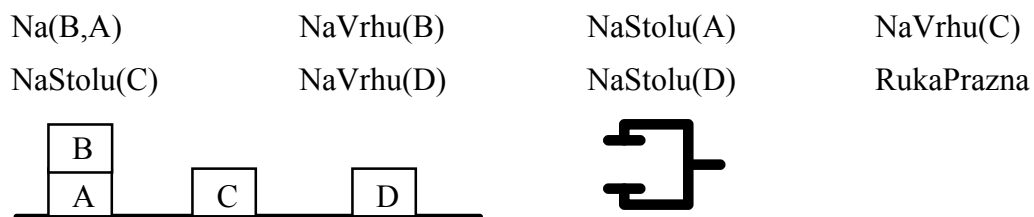


Akcije mehaničke hvataljke (ruke) možemo modelirati sledećim operatorima:

- operator **UZMI_SA_STOLA(x)**:
 PREDUSLOV: $RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)$
 UKLONI: $RukaPrazna; NaVrhu(x); NaStolu(x)$
 DODAJ: $URuci(x)$
- operator **SPUSTI_NA_STO(y)**:
 PREDUSLOV: $URuci(y)$
 UKLONI: $URuci(y)$
 DODAJ: $RukaPrazna; NaVrhu(y); NaStolu(y)$
- operator **SKINI_SA_BLOKA(u,z)**:
 PREDUSLOV: $RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)$
 UKLONI: $RukaPrazna; NaVrhu(u); Na(u,z)$
 DODAJ: $URuci(u); NaVrhu(z)$
- operator **STAVI_NA_BLOK(v,w)**:
 PREDUSLOV: $URuci(v) \wedge NaVrhu(w)$
 UKLONI: $URuci(v); NaVrhu(w)$
 DODAJ: $RukaPrazna; NaVrhu(v); Na(v,w)$

STRIPS algoritam nalaže da se tekuće stanje problema inicijalizuje formalnim opisom početnog stanja i da se na ciljni stek stavi kako složeni cilj (jednak formalnom opisu ciljnog stanja) tako i svaka od komponentata složenog cilja pojedinačno. Posle ove inicijalizacije izgled struktura podataka je sledeći:

TEKUĆE STANJE:



CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

Na(C,A)
 NaVrhu(C)
 NaStolu(A)
 Na(B,D)
 NaVrhu(B)

NaStolu(D)

RukaPrazna

LISTA AKCIJA:

-

Grafički prikaz tekućeg stanja nije deo algoritma već je dat radi lakšeg praćenja promena tekućeg stanja. Prva tri stava sa vrha steka RukaPrazna, NaStolu(D) i NaVrhu(B) su već zadovoljena u tekućem stanju, pa se uklanjaju sa steka. Na vrhu steka ostaje stav Na(B,D) koji trenutno ne važi. **Algoritam nalaže izbor operatora koji u svom DODAJ delu** ima navedeni stav, a to je jedino operator **STAVI_NA_BLOK(v,w)**, pri čemu je izvršena unifikacija $v \Leftrightarrow B$, $w \Leftrightarrow D$. Zatim se na stek stavljaju sam operator, njegov preduslov i svaka komponenta preduslova posebno, posle čega je izgled steka:

CILJNI STEK (raste naniže):

$\text{Na(C,A)} \wedge \text{NaVrhu(C)} \wedge \text{NaStolu(A)} \wedge \text{Na(B,D)} \wedge \text{NaVrhu(B)} \wedge \text{NaStolu(D)} \wedge \text{RukaPrazna}$

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

STAVI_NA_BLOK(B,D)

URuci(B) \wedge NaVrhu(D)

URuci(B)

NaVrhu(D)

Vršni stav je zadovoljen u tekućem stanju dok stav URuci(B) nije zadovoljen. Ponovo, dakle, treba pronaći operator koji ima ovakav stav u svom DODAJ delu. U ovom slučaju postoje dva operatora **UZMI_SA_STOLA** i **SKINI_SA_BLOKA** pa treba izvršiti izbor jednog od njih. STRIPS algoritam ne propisuje kako se vrši izbor, već se koristi poseban algoritam vezan za konkretan problem. U našem slučaju, pošto se blok B u tekućem stanju nalazi na bloku A pogodniji je operator **SKINI_SA_BLOKA(u,z)** uz unifikacije promenljivih $u \Leftrightarrow B$, $z \Leftrightarrow A$, pa se na stek stavlja operator, njegov preduslov i svaki stav preduslova:

CILJNI STEK (raste naniže):

$\text{Na(C,A)} \wedge \text{NaVrhu(C)} \wedge \text{NaStolu(A)} \wedge \text{Na(B,D)} \wedge \text{NaVrhu(B)} \wedge \text{NaStolu(D)} \wedge \text{RukaPrazna}$

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

STAVI_NA_BLOK(B,D)

URuci(B) \wedge NaVrhu(D)

URuci(B)

SKINI_SA_BLOKA(B,A)

RukaPrazna \wedge NaVrhu(B) \wedge Na(B,A)

RukaPrazna

NaVrhu(B)

Na(B,A)

U tekućem stanju svi preduslovi važe pa ih sve skidamo sa steka, na čijem vrhu ostaje operator **SKINI_SA_BLOKA(B,A)**. Prema algoritmu, operator se skida sa steka i upisuje u listu akcija; vrši se modifikacija tekućeg stanja: uklanjaju se stavovi koje operator **SKINI_SA_BLOKA** ima u svojoj UKLONI listi, a dodaju se stavovi iz njegove DODAJ liste. Strukture podataka STRIPS-a posle ovoga izgledaju ovako:

TEKUĆE STANJE

NaStolu(A)

NaVrhu(A)

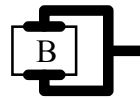
NaVrhu(C)

NaStolu(C)

NaVrhu(D)

NaStolu(D)

URuci(B)



CILJNI STEK (raste naniže):

Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

STAVI_NA_BLOK(B,D)

URuci(B) \wedge NaVrhu(D)

URuci(B)

LISTA AKCIJA:

1. **SKINI_SA_BLOKA(B,A)**

Vršni stav na steku URuci(B) sada važi, kao i složeni preduslov, pa se uklanjaju sa steka na čijem se vrhu sada pojavljuje operator **STAVI_NA_BLOK(B,D)**. Prema algoritmu, operator se skida sa steka i dopisuje u listu akcija a tekuće stanje se ažurira u skladu sa UKLONI i DODAJ listama operatora **STAVI_NA_BLOK**. U tom trenutku imamo sledeći izgled struktura podataka:

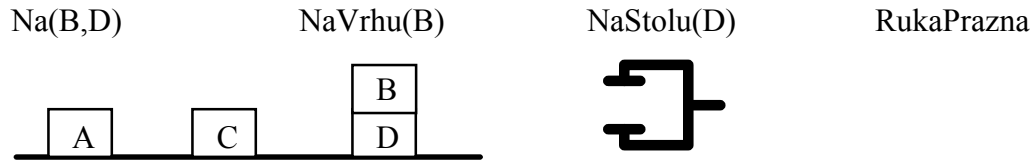
TEKUĆE STANJE

NaStolu(A)

NaVrhu(A)

NaVrhu(C)

NaStolu(C)



CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

LISTA AKCIJA:

1. **SKINI_SA_BLOKA(B,A)**

2. **STAVI_NA_BLOK(B,D)**

Vršna tri stava steka zadovoljena su u tekućem stanju, pa se uklanjaju sa steka. Izgled ciljnog steka posle uklanjanja stavova je:

CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

Na(C,A)

Potrebno je naći operator koji zadovoljava stav $Na(C,A)$ a to je slučaj jedino sa operatorom **STAVI_NA_BLOK(v,w)**, uz unifikacije $v \Leftrightarrow C$, $w \Leftrightarrow A$, posle čega na stek ide prvo operator a zatim i njegovi preduslovi pa je izgled steka:

CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

Na(C,A)

STAVI_NA_BLOK(C,A)

URuci(C) \wedge NaVrhu(A)

URuci(C)

NaVrhu(A)

Stav $NaVrhu(A)$ je zadovoljen, pa se skida sa steka; na vrhu steka ostaje stav $URuci(C)$ koji nije zadovoljen. Operatori koji ga zadovoljavaju su **UZMI_SA_STOLA** i **SKINI_SA_BLOKA**. Koristeći znanje o tekućem stanju i značenju pojedinih operatora može se zaključiti da je operator **UZMI_SA_STOLA(x)** pogodniji. Ovaj operator, nakon unifikacije $x \Leftrightarrow C$, stavlja se na stek kao i njegovi preduslovi nakon čega stek izgleda ovako:

CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

$Na(C,A)$

STAVI_NA_BLOK(C,A)

$URuci(C) \wedge NaVrhu(A)$

$URuci(C)$

$NaVrhu(A)$

UZMI_SA_STOLA(C)

$RukaPrazna \wedge NaVrhu(C) \wedge NaStolu(C)$

$RukaPrazna$

$NaVrhu(C)$

$NaStolu(C)$

Pošto su sve komponente složenog podcilja zadovoljene, uklanjamo ih sa steka kao i sam podcilj te na vrhu ostaje operator **UZMI_SA_STOLA**. Uklanjanjem ovog operatora sa steka, ažuriranjem liste akcija i tekućeg stanja, izgled struktura podataka je sledeći:

TEKUĆE STANJE

$NaStolu(A)$

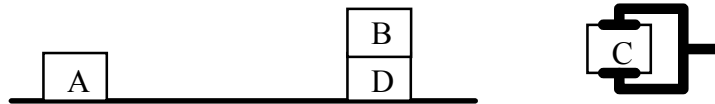
$NaVrhu(A)$

$Na(B,D)$

$NaVrhu(B)$

$NaStolu(D)$

$URuci(C)$



CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$

$Na(C,A)$

STAVI_NA_BLOK(C,A)

$URuci(C) \wedge NaVrhu(A)$

$URuci(C)$

LISTA AKCIJA:

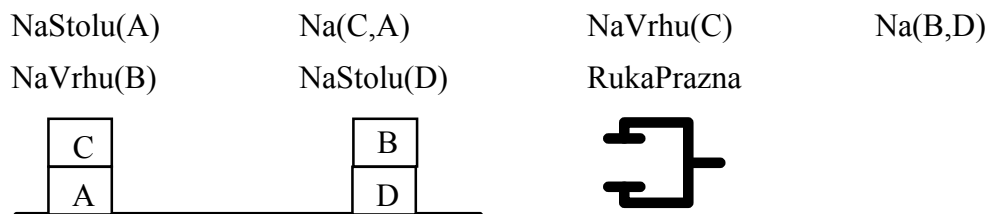
1. **SKINI_SA_BLOKA(B,A)**

2. **STAVI_NA_BLOK(B,D)**

3. **UZMI_SA_STOLA(C)**

Stav $URuci(C)$ je u tekućem stanju zadovoljen pa se uklanja sa steka; kako su obe komponente složenog podcilja koji ostaje na vrhu steka zadovoljene, i taj podcilj se uklanja. Zatim se operator **STAVI_NA_BLOK(C,A)** skida sa steka i upisuje u plan operacija, a tekuće stanje se ažurira preko UKLONI i DODAJ lista ovog operatora. Prema tome, stanje je:

TEKUĆE STANJE



CILJNI STEK (raste naniže):

$Na(C,A) \wedge NaVrhu(C) \wedge NaStolu(A) \wedge Na(B,D) \wedge NaVrhu(B) \wedge NaStolu(D) \wedge RukaPrazna$
 $Na(C,A)$

LISTA AKCIJA:

1. **SKINI_SA_BLOKA(B,A)**
2. **STAVI_NA_BLOK(B,D)**
3. **UZMI_SA_STOLA(C)**
4. **STAVI_NA_BLOK(C,A)**

Sada se vidi da tekuće stanje odgovara ciljnom stanju, pa se sa steka skida kako stav $Na(C,A)$ tako i složeni cilj; stek ostaje prazan što znači da je problem rešen. Izdavanjem liste akcija koje sistem prevode iz početnog u ciljno stanje završava se rad STRIPS algoritma.

Zadatak 1: Svet blokova (problem izbora operatora)

Na slici 1a prikazana je jedna situacija iz sveta cigala. Pronaći koje operatore i kojim redom treba primeniti, da bi se prešlo u situaciju pokazanu na slici 1b, koristeći STRIPS algoritam.



Slika 1

Rešenje

Predstava stanja i operatori mogu se definisati na isti način kao u zadatku **Error! Bookmark not defined.** Početno stanje u ovom slučaju opisano je stavovima:

$NaStolu(A)$ $Na(B,A)$ $NaVrhu(B)$ $RukaPrazna$

dok je ciljno stanje opisano sa:

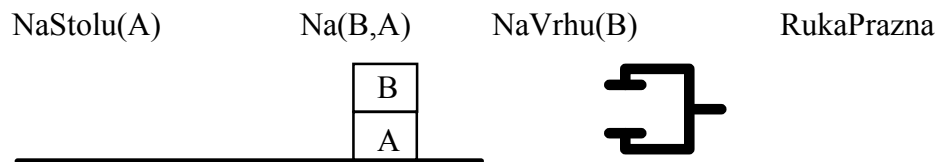
$NaStolu(B)$ $Na(A,B)$ $NaVrhu(B)$ $RukaPrazna$

Nije lako definisati algoritam koji obezbeđuje optimalan redosled izbora operatora u opštem slučaju. Razmotrimo rad STRIPS algoritma ako za izbor operatora usvojimo jednostavno pravilo da se operatori, u situacijama kada postoji mogućnost izbora više od jednog od njih, biraju po redosledu kojim su definisani, naime:

1. **UZMI_SA_STOLA(x)**
2. **SPUSTI_NA_STO(y)**
3. **SKINI_SA_BLOKA(u,z)**
4. **STAVI_NA_BLOK(v,w).**

Inicijalno imamo sledeću situaciju:

TEKUĆE STANJE



CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna
 NaStolu(B)
 Na(A,B)
 NaVrhu(A)
 RukaPrazna

LISTA AKCIJA:

-

Stav RukaPrazna zadovoljen je u početnom stanju pa se ovaj stav skida sa steka. Da bi se zadovoljio stav NaVrhu(A) potrebno je izabrati operator. Mogući operatori, koji imaju stav NaVrhu(A) u svom DODAJ delu, su:

- 1-1 **SPUSTI_NA_STO(A)**
- 1-2 **SKINI_SA_BLOKA(u,A)**
- 1-3 **STAVI_NA_BLOK(A,w)**

pa biramo prvi od njih, **SPUSTI_NA_STO(A)**. Izgled steka posle ažuriranja je:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna
 NaStolu(B)
 Na(A,B)
 NaVrhu(A)
SPUSTI_NA_STO(A)

URuci(A)

Da bismo zadovoljili stav URuci(A) na izboru su sledeći operatori:

2-1 **UZMI_SA_STOLA(A)**

2-2 **SKINI_SA_BLOKA(A,z)**

Biramo operator **UZMI_SA_STOLA(A)** i stavljamo ga na stek zajedno sa preduslovima:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

NaVrhu(A)

SPUSTI_NA_STO(A)

URuci(A)

UZMI_SA_STOLA(A)

RukaPrazna \wedge NaVrhu(A) \wedge NaStolu(A)

RukaPrazna

NaVrhu(A)

NaStolu(A)

Stav NaStolu(A) zadovoljen je u tekućem stanju, pa ga skidamo sa steka. Na vrhu steka ostaje nezadovoljeni stav NaVrhu(A) koji se već pojavljivao kao tekući cilj u tekućem stanju. Algoritam u ovoj situaciji nalaže ponovni izbor operatora **SPUSTI_NA_STO(A)**, a za zadovoljavanje preduslova primene ovog operatora bio bi ponovo izabran operator **UZMI_SA_STOLA(A)** i tako dalje. Ova situacija predstavlja, prema tome, 'ćorsokak' za algoritam izbora operatora.

Algoritam izbora operatora treba dakle da obezbedi vraćanje unatrag (*backtracking*), tako što će obnoviti stanje steka pre izbora operatora **UZMI_SA_STOLA(A)** i izabrati alternativni operator, u ovom slučaju operator **SKINI_SA_BLOKA(A,z)**.

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

NaVrhu(A)

SPUSTI_NA_STO(A)

URuci(A)

SKINI_SA_BLOKA(A,z)

RukaPrazna \wedge Na(A,z) \wedge NaVrhu(A)

RukaPrazna

Na(A,z)

NaVrhu(A)

Treba primetiti da pri prethodnom izboru operatora promenljiva z nije vezana ni za jednu od konstanti A, B, C jer za tim nije postojala potreba. Na vrhu steka opet se pojavio nezadovoljeni stav NaVrhu(A) koji signalizuje pogrešan izbor operatora i nalaže vraćanje na situaciju pre izbora poslednjeg operatora:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

NaVrhu(A)

SPUSTI_NA_STO(A)

URuci(A)

S obzirom da su veæ razmotrene sve mogućnosti za zadovoljavanje stava URuci(A), zaključujemo da je operator **SPUSTI_NA_STO(A)** pogrešno izabran pa se vraćamo na razmatranje situacije pre izbora ovoga operatora:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

NaVrhu(A)

U ovoj situaciji biramo sledeći operator po redosledu primene, a to je **SKINI_SA_BLOKA(u,A)**. Stavljamo ga na stek zajedno sa preduslovima primene:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

NaVrhu(A)

SKINI_SA_BLOKA(u,A)

RukaPrazna \wedge Na(u,A) \wedge NaVrhu(u)

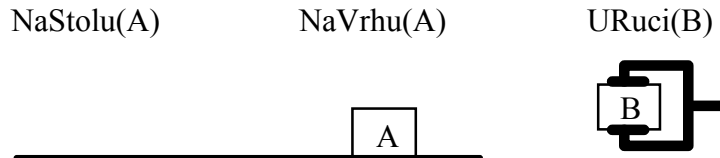
RukaPrazna

Na(u,A)

NaVrhu(u)

Izvršavanje algoritma nastavlja se unifikacijom $u \Leftrightarrow B$ da bi se utvrdilo da se stav NaVrhu(B) nalazi u tekućem stanju. Unifikacijom je određena vrednost i za ostale pojave promenljive u na ciljnom steku tako da je sad na vrhu steka stav Na(B,A). Ovaj stav je zadovoljen, pa se uklanja sa steka kao i stav RukaPrazna. Zadovoljen je i složeni preduslov za primenu operatora **SKINI_SA_BLOKA**(B,A) pa se preduslov i operator uklanjaju sa steka i operator se primenjuje čime se dobija novo tekuće stanje:

TEKUĆE STANJE



CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna
NaStolu(B)
Na(A,B)
NaVrhu(A)

LISTA AKCIJA:

1. **SKINI_SA_BLOKA**(B,A)

Stav na vrhu steka, NaVrhu(A) zadovoljen u ovom stanju pa se uklanja sa steka. Stav Na(A,B) nije zadovoljen u tekućem stanju. Jedini operator koji može da zadovolji ovaj stav je **STAVI_NA_BLOK**(v,w) uz unifikaciju $v \Leftrightarrow A$, $w \Leftrightarrow B$ pa se ovaj operator stavlja na stek zajedno sa preduslovima.

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna
NaStolu(B)
Na(A,B)
STAVI_NA_BLOK(A,B)
URuci(A) \wedge NaVrhu(B)
URuci(A)
NaVrhu(B)

Stav na vrhu steka, NaVrhu(B), nije zadovoljen u tekućem stanju. Mogući operatori za izbor su:

- 3-1 **SPUSTI_NA_STO**(B)
- 3-2 **STAVI_NA_BLOK**(B,w)
- 3-3 **SKINI_SA_BLOKA**(u,B)

Bira se operator **SPUSTI_NA_STO**(y) i ažurira ciljni stek stavljanjem ovoga operatora i njegovih preduslova:

CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

STAVI_NA_BLOK(A,B)

URuci(A) \wedge NaVrhu(B)

URuci(A)

NaVrhu(B)

SPUSTI_NA_STO(B)

URuci(B)

Preduslov URuci(B) je zadovoljen, pa je moguće primeniti operator **SPUSTI_NA_STO**(B) čime se dobija novo tekuće stanje:

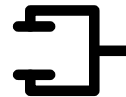
TEKUĆE STANJE

NaStolu(A)
RukaPrazna

NaVrhu(A)

NaStolu(B)

NaVrhu(B)



CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

STAVI_NA_BLOK(A,B)

URuci(A) \wedge NaVrhu(B)

URuci(A)

NaVrhu(B)

LISTA AKCIJA:

1. **SKINI_SA_BLOKA**(B,A)

2. **SPUSTI_NA_STO**(B)

Stav NaVrhu(B) zadovoljen je u ovom stanju, pa se uklanja sa vrha steka. Za zadovoljavanje stava URuci(A) na raspolaganju su operatori:

4-1 **UZMI_SA_STOLA**(A)

4-2 SKINI_SA_BLOKA(A,z)

Bira se prvi od ovih operatora i stavlja na stek zajedno sa preduslovima. Izgled steka je:

CILJNI STEK (raste naniže):

$\text{NaStolu}(B) \wedge \text{Na}(A,B) \wedge \text{NaVrhu}(A) \wedge \text{RukaPrazna}$

$\text{NaStolu}(B)$

$\text{Na}(A,B)$

STAVI_NA_BLOK(A,B)

$\text{URuci}(A) \wedge \text{NaVrhu}(B)$

$\text{URuci}(A)$

UZMI_SA_STOLA(A)

$\text{RukaPrazna} \wedge \text{NaVrhu}(A) \wedge \text{NaStolu}(A)$

RukaPrazna

$\text{NaVrhu}(A)$

$\text{NaStolu}(A)$

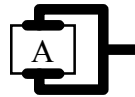
Svi preduslovi za primenu operatora **UZMI_SA_STOLA(A)** zadovoljeni su u tekućem stanju, pa se zajedno sa operatorom uklanjaju sa steka. Primenom ovoga operatora dobija se novo stanje:

TEKUĆE STANJE

$\text{NaStolu}(B)$

$\text{NaVrhu}(B)$

$\text{URuci}(A)$



CILJNI STEK (raste naniže):

$\text{NaStolu}(B) \wedge \text{Na}(A,B) \wedge \text{NaVrhu}(A) \wedge \text{RukaPrazna}$

$\text{NaStolu}(B)$

$\text{Na}(A,B)$

STAVI_NA_BLOK(A,B)

$\text{URuci}(A) \wedge \text{NaVrhu}(B)$

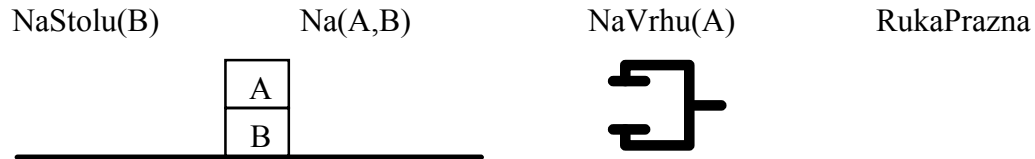
$\text{URuci}(A)$

LISTA AKCIJA:

1. **SKINI_SA_BLOKA(B,A)**
2. **SPUSTI_NA_STO(B)**
3. **UZMI_SA_STOLA(A)**

Preduslovi za primenu operatora **STAVI_NA_BLOK(A,B)** su u potpunosti zadovoljeni pa se redom skidaju sa steka. Ovaj operator takodje se uklanja sa steka i primenjuje na tekuće stanje čime se dobija novo tekuće stanje.

TEKUĆE STANJE



CILJNI STEK (raste naniže):

NaStolu(B) \wedge Na(A,B) \wedge NaVrhu(A) \wedge RukaPrazna

NaStolu(B)

Na(A,B)

LISTA AKCIJA:

1. **SKINI_SA_BLOKA(B,A)**
2. **SPUSTI_NA_STO(B)**
3. **UZMI_SA_STOLA(A)**
4. **STAVI_NA_BLOK(A,B)**

Svi stavovi na steku su zadovoljeni pa se redom skidaju sa steka dok se stek ne isprazni čime se okončava algoritam. Rešenje problema dato je listom akcija.

Čitaocu ostaje da proveri da li se može definisati bolji redosled operatora pri izboru koji bi smanjio broj vraćanja unatrag.