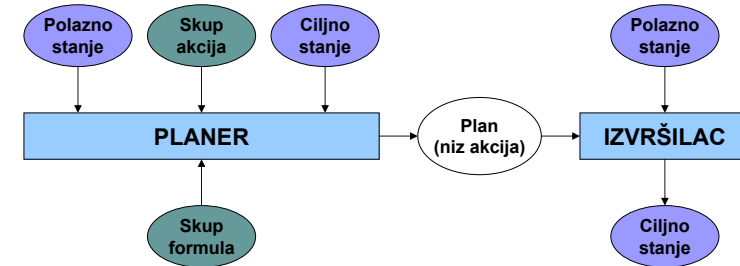




Planiranje

Aleksandar Milosavljević
Vladan Mihajlović

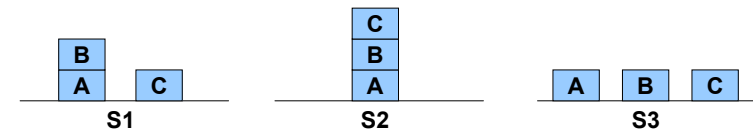
Planiranje



Green-ova metoda

Svet blokova - opis stanja

- Primeri stanja u svetu blokova



- Svako stanje može da se opiše korišćenjem 3 predikata:
 - Clear(*x*)** - nijedan blok ne stoji na bloku *x*
 - Table(*x*)** - blok *x* se nalazi na tabli
 - On(*x*, *y*)** - blok *x* se nalazi na bloku *y*



Svet blokova - opis stanja

- Četvrti predikat T se koristi da se opis stanja poveže sa oznakom stanja:
 - **T(*stanje_bloka*, *oznaka_stanja*)**
- Opis stanja **S1**:
 - T(Clear(B), S1)
 - T(Clear(C), S1)
 - T(Table(A), S1)
 - T(Table(C), S1)
 - T(On(B, A), S1)



Svet blokova - opis stanja

- Opis stanja **S2**:
 - T(Clear(C), S2)
 - T(Table(A), S2)
 - T(On(C, B), S2)
 - T(On(B, A), S2)
- Opis stanja **S3**:
 - T(Clear(A), S3)
 - T(Clear(B), S3)
 - T(Clear(C), S3)
 - T(Table(A), S3)
 - T(Table(B), S3)
 - T(Table(C), S3)



Svet blokova - skup akcija

- Akcije prevode svet blokova iz jednog stanja u drugo.
- Postoje 3 moguće akcije (operacije):
 - **U(x, y)** - *unstack*
 - skida blok **x** sa bloka **y** i stavlja ga na tablu
 - **S(x, y)** - *stack*
 - stavlja blok **x**, koji se nalazi na tabli, na blok **y**
 - **M(x, y, z)** - *move*
 - pomera blok **x**, koji se nalazi na bloku **y**, na blok **z**



Definicije operatora

- Definicija operatora **U(x, y)**:

$$T(\text{On}(x, y), s) \wedge T(\text{Clear}(x), s) \Rightarrow T(\text{Table}(x), \text{do}(U(x, y), s)) \wedge T(\text{Clear}(y), \text{do}(U(x, y), s))$$
- Definicija operatora **S(x, y)**:

$$T(\text{Table}(x), s) \wedge T(\text{Clear}(x), s) \wedge T(\text{Clear}(y), s) \Rightarrow T(\text{On}(x, y), \text{do}(S(x, y), s))$$
- Definicija operatora **M(x, y, z)**:

$$T(\text{Clear}(x), s) \wedge T(\text{On}(x, y), s) \wedge T(\text{Clear}(z), s) \Rightarrow T(\text{On}(x, z), \text{do}(M(x, y, z), s)) \wedge T(\text{Clear}(y), \text{do}(M(x, y, z), s))$$



Funkcija *do*

- Funkcija **do** se koristi za definiciju novog stanja posle primene nekog od operatora:
 - $\text{do}(\text{operator}, \text{stanje}) \rightarrow \text{novo_stanje}$
- Primeri:
 - $\text{do}(\text{S}(\text{C}, \text{B}), \text{S1}) \rightarrow \text{S2}$
 - $\text{do}(\text{U}(\text{B}, \text{A}), \text{do}(\text{U}(\text{C}, \text{B}), \text{S2})) \rightarrow \text{S3}$



Frame aksiome

- *Frame* aksiome se koriste da definišu prelaze iz tekućeg stanja u novo stanje blokova koji nisu pod uticajem operatora.
- Za svaki od operatora U, S i M definiše se po jedna *frame* aksioma za stanja blokova **Clear(x)**, **Table(x)**, **On(x, y)**.



Frame aksiome

- *Frame* aksiome operatora **U(x, y)**:
 - $T(\text{Clear}(u), s) \Rightarrow T(\text{Clear}(u), \text{do}(\text{U}(x, y), s))$
 - $T(\text{Table}(u), s) \Rightarrow T(\text{Table}(u), \text{do}(\text{U}(x, y), s))$
 - $T(\text{On}(u, w), s) \wedge u \neq x \Rightarrow T(\text{On}(u, w), \text{do}(\text{U}(x, y), s))$



Frame aksiome

- *Frame* aksiome operatora **S(x, y)**:
 - $T(\text{Clear}(u), s) \wedge u \neq y \Rightarrow T(\text{Clear}(u), \text{do}(\text{S}(x, y), s))$
 - $T(\text{Table}(u), s) \wedge u \neq x \Rightarrow T(\text{Table}(u), \text{do}(\text{S}(x, y), s))$
 - $T(\text{On}(u, w), s) \Rightarrow T(\text{On}(u, w), \text{do}(\text{S}(x, y), s))$



Frame aksiome

- **Frame aksiome operatora $M(x, y, z)$:**

- $T(\text{Clear}(u), s) \wedge u \neq z \Rightarrow T(\text{Clear}(u), \text{do}(M(x, y, z), s))$
- $T(\text{Table}(u), s) \Rightarrow T(\text{Table}(u), \text{do}(M(x, y, z), s))$
- $T(\text{On}(u, w), s) \wedge u \neq x \Rightarrow T(\text{On}(u, w), \text{do}(M(x, y, z), s))$



Green-ova metoda

- Planiranje se zasniva na predikatskoj logici:
 - Sve akcije moraju biti iz skupa $\{U, S, M\}$
 - Mora postojati dokaz da skup akcija prevodi početno stanje u ciljno stanje.
 - Izvođenje plana se zasniva na rezoluciji.
- Teorema koja se dokazuje rezolucijom po Green-ovoj metodi je:
 - **Goal**(do(a, S1)) \Rightarrow **Ans**(a)
 - Predikat **Goal** definiše ciljno stanje, a predikat **Ans** definiše skup akcija potreban da početno stanje prevede u ciljno.
 - **Goal**(t) \Leftrightarrow opis ciljnog stanja t



Zadatak 1.

- Izvesti plan za prevođenje blokova iz stanja S1 u stanje S2:



- **Rešenje:**

- Analizom problema se uočava da akcija U(A, B) prevodi blokove iz stanja S1 u S2.



Zadatak 1. (rešenje)

- Definicija početog stanja S1:
 - $T(\text{Clear}(A), S1) \wedge T(\text{Table}(C), S1) \wedge T(\text{On}(A, B), S1) \wedge T(\text{On}(B, C), S1)$
- Definicija cilja:
 - $\text{Goal}(t) \Leftrightarrow T(\text{Table}(A), t)$
 - $T(\text{Table}(A), \text{do}(a, S1)) \Rightarrow \text{Ans}(a)$
- Prevođenjem u klauzulni oblik dobijamo...



Zadatak 1. (rešenje)

- Početno stanje:
 - K1: $\{T(\text{Clear}(A), S1)\}$
 - K2: $\{T(\text{Table}(C), S1)\}$
 - K3: $\{T(\text{On}(A, B), S1)\}$
 - K4: $\{T(\text{On}(B, C), S1)\}$
- Operator $U(x, y)$:
 - K5: $\{T(\text{Table}(x5), \text{do}(U(x5, y5), s5)), \neg T(\text{On}(x5, y5), s5), \neg T(\text{Clear}(x5), s5)\}$
 - K6: $\{T(\text{Clear}(y6), \text{do}(U(x6, y6), s6)), \neg T(\text{On}(x6, y6), s6), \neg T(\text{Clear}(x6), s6)\}$
- Definicija cilja (kada je dovoljna jedna akcija):
 - K7: $\{\neg T(\text{Table}(A), \text{do}(a, S1)), \text{Ans}(a)\}$



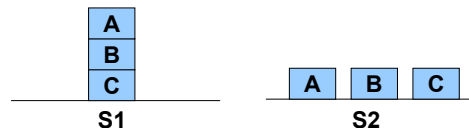
Zadatak 1. (rešenje)

- Poklapanjem klauzula K7 i K5, sa listom smena:
 - $(x5 / A; a / U(A, y5); s5 / S1)$
- dobijamo:
 - K8: $\{\text{Ans}(U(A, y5)), \neg T(\text{On}(A, y5), S1), \neg T(\text{Clear}(A), S1)\}$
- Poklapanjem klauzula K8 i K3, sa listom smena:
 - $(y5 / B)$
- dobijamo
 - K9: $\{\text{Ans}(U(A, B)), \neg T(\text{Clear}(A), S1)\}$
- Poklapanjem klauzula K9 i K1 dobijamo:
 - K10: $\{\text{Ans}(U(A, B))\}$



Zadatak 2.

- Izvesti plan za prevođenje blokova iz stanja S1 u stanje S2:



- Rešenje:
 - Analizom problema se uočava da niz akcija $U(A, B)$ i $U(B, C)$ prevodi blokove iz stanja S1 u S2.



Zadatak 2. (rešenje)

- Definicija početog stanja S1:
 - $T(\text{Clear}(A), S1) \wedge T(\text{Table}(C), S1) \wedge T(\text{On}(A, B), S1) \wedge T(\text{On}(B, C), S1)$
- Definicija cilja:
 - $\text{Goal}(t) \Leftrightarrow T(\text{Table}(B), t)$
 - $T(\text{Table}(B), \text{do}(a2, \text{do}(a1, S1))) \Rightarrow \text{Ans}(a1, a2)$
- Prevođenjem u klauzulni oblik dobijamo...



Zadatak 2. (rešenje)

- Početno stanje:
 - K1: {T(Clear(A), S1)}
 - K2: {T(Table(C), S1)}
 - K3: {T(On(A, B), S1)}
 - K4: {T(On(B, C), S1)}
- Operator U(x, y):
 - K5: {T(Table(x5), do(U(x5, y5), s)),
¬T(On(x5, y5), s5), ¬T(Clear(x5), s5)}
 - K6: {T(Clear(y6), do(U(x6, y6), s6)),
¬T(On(x6, y6), s6), ¬T(Clear(x6), s6)}



Zadatak 2. (rešenje)

- Frame aksiome za operator U:
 - K7: {¬T(Clear(u7), s7), T(Clear(u7), do(U(x7, y7), s7))}
 - K8: {¬T(Table(u8), s8), T(Table(u8), do(U(x8, y8), s8))}
 - K9: {¬T(On(u9, w9), s9), ¬u9≠x9,
T(On(u9, w9), do(U(x9, y9), s9))}
- Definicija cilja (kada su potrebne dve akcije):
 - K10: {¬T(Table(B), do(a2, do(a1, S1))),
Ans(a1, a2)}



Zadatak 2. (rešenje)

- Poklapanjem klauzula K10 i K5, sa listom smena:
 - (x5 / B; a2 / U(B, y5); s5 / do(a1, S1))
- dobijamo:
 - K11: {Ans(a1, U(B, y5)),
¬T(On(B, y5), do(a1, S1)),
¬T(Clear(B), do(a1, S1))}



Zadatak 2. (rešenje)

- Poklapanjem klauzula K11 i K6, sa listom smena:
 - (y6 / B; a1 / U(x6, B); s6 / S1)
- dobijamo:
 - K12: {Ans(U(x6, B), U(B, y5)),
¬T(On(B, y5), do(U(x6, B), S1)),
¬T(On(x6, B), S1), ¬T(Clear(x6), S1)}



Zadatak 2. (rešenje)

- Poklapanjem klauzula K12 i K1, sa listom smena:
 - $(x6 / A)$
- dobijamo:
 - K13: $\{Ans(U(A, B), U(B, y5)), \neg T(On(B, y5), do(U(A, B), S1)), \neg T(On(A, B), S1)\}$



Zadatak 2. (rešenje)

- Poklapanjem klauzula K13 i K3 dobijamo:
 - K14: $\{Ans(U(A, B), U(B, y5)), \neg T(On(B, y5), do(U(A, B), S1))\}$
- Poklapanjem klauzule K14 sa frame aksiomom K9, sa listom smena:
 - $(u9 / B; x9 / A, y9 / B, s9 / S1, w9 / y5)$
- dobijamo:
 - K15: $\{Ans(U(A, B), U(B, y5)), \neg T(On(B, y5), S1), \neg B \neq A\}$



Zadatak 2. (rešenje)

- Kako su A i B različiti blokovi možemo da dodamo podrazumevanu klauzulu:
 - $\{B \neq A\}$
- Poklapanjem klauzule K15 sa ovom podrazumevanom klauzulom dobijamo:
 - K16: $\{Ans(U(A, B), U(B, y5)), \neg T(On(B, y5), S1)\}$



Zadatak 2. (rešenje)

- Konačno, poklapanjem klauzula K16 i K4, sa listom smena:
 - $(y5 / C)$
- dobijamo odgovor:
 - K17: $\{Ans(U(A, B), U(B, C))\}$
- Redosled akcija koje treba sprovesti da bi se svet blokova preveo iz stanja S1 u stanje S2 je:
 - $U(A, B)$, pa onda $U(B, C)$.





STRIPS algoritam

STRIPS algoritam

- STRIPS algoritam služi za rešavanje problema strategijom planiranja.
- Problem se predstavlja u formalnoj logici.
- Cilj STRIPS-a je da se nađe sekvenca operatora koja sistem prevodi iz početnog stanja u ciljno stanje.



Strukture podataka

- STRIPS algoritma koristi sledeće strukture podataka:
 - **Tekuće stanje** problema opisano u predikatskoj logici.
 - Inicijalno je to opis početnog stanja.
 - **Ciljni stek** koji sadrži stavove koji odgovaraju trenutnom (pod)cilju.
 - Inicijalno je to opis ciljnog stanje.
 - **Lista akcija** koja na kraju sadrži sekvencu operacija koja predstavlja plan
 - Inicijalno je lista prazna.



Operatori promene stanja

- Operatori se definišu zadavanjem tri liste stavova:
 - **PREDUSLOV** - lista stavova koja moraju biti ispunjeni u tekućem stanju da bi operator mogao biti primenjen.
 - **UKLONI** - lista stavova koji se uklanjaju iz tekućeg stanja u trenutku primene operatora.
 - **DODAJ** - lista stavova koji se dodaju tekućem stanju nakon primene operatora.



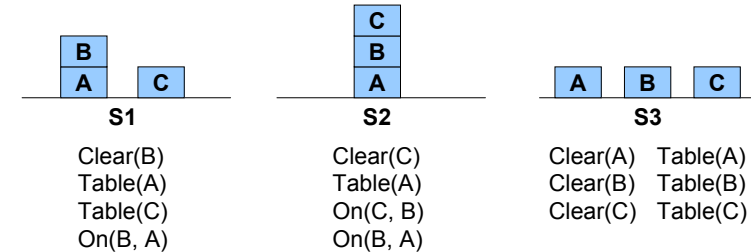
Opis algoritma

- Definiše se početno stanje.
- Definiše se cilj i postavi na stek.
- Dok stek nije prazan uzima se stav sa vrha steka:
 - Ako je stav (pod)cilj proveriti da li je zadovoljen:
 - Ako je zadovoljen uklanja se sa steka.
 - U suprotnom se bira operator koji ga zadovoljava i stavlja na stek. Stavljaju se i svi njegovi PREDUSLOVI.
 - Ako je stav operator:
 - Skida se sa steka i dodaje se listi akcija.
 - Uklanjaju se stavovi prema listi UKLONI iz tekućeg stanja.
 - Dodaju se stavovi prema listi DODAJ u tekuće stanje.
- Izdati listu akcija.



Opis stanja

- Za opis stanja i cilja koriste se ranije definisani predikati:
 - **Clear(x)** - nijedan blok ne stoji na bloku **x**
 - **Table(x)** - blok **x** se nalazi na tabli
 - **On(x, y)** - blok **x** se nalazi na bloku **y**



Operatori

- **Koriste se ranije definisani operatori u svetu blokova**
 - **U(x, y)** - *unstack*
 - skida blok **x** sa bloka **y** i stavlja ga na tablu
 - **S(x, y)** - *stack*
 - stavlja blok **x**, koji se nalazi na tabli, na blok **y**
 - **M(x, y, z)** - *move*
 - pomera blok **x**, koji se nalazi na bloku **y**, na blok **z**



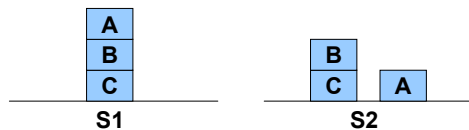
Definicije operatora

- **U(x, y)**
 - PREDUSLOV: **Clear(x), On(x, y)**
 - UKLONI: **On(x, y)**
 - DODAJ: **Table(x), Clear(y)**
- **S(x, y)**
 - PREDUSLOV: **Table(x), Clear(x), Clear(y)**
 - UKLONI: **Table(x), Clear(y)**
 - DODAJ: **On(x, y)**
- **M(x, y, z)**
 - PREDUSLOV: **Clear(x), On(x, y), Clear(z)**
 - UKLONI: **On(x, y), Clear(z)**
 - DODAJ: **On(x, z), Clear(y)**



Zadatak 1.

- Koristeći STRIPS algoritam naći sekvencu operatora koji prevode svet blokova iz stanja S1 u stanje S2:



Zadatak 1. (rešenje)

- Inicijalno stanje STRIPS struktura je:
 - Tekuće stanje:
 - Clear(A), Table(C), On(A, B), On(B, C)
 - Ciljni stek:
 - Table(A)
 - Lista akcija je inicijalno prazna.
- Uzimamo stav sa vrha steka -> **Table(A)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto ga nema u tekućem stanju biramo operator koji ga ima u DODAJ delu -> **U(x, y)**
 - Izvrši se unifikacija (pod)cilja i DODAJ dela operatora **U(x, y)**, a zatim (ukoliko je potrebno) i PREDUSLOVA operatora sa tekućim stanjem:
 - Dobija se lista smena: **(x / A, y / B)**



Zadatak 1. (rešenje)

- Novo stanje ciljnog steka sadrži operator i njegove preduslove (**prvo se dodaje operator, pa onda preduslovi**):
 - Clear(A)
 - On(A, B)
 - U(A, B)
 - Table(A)
- Uzimamo stav sa vrha steka -> **Clear(A)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto se (pod)cilj nalazi u tekućem stanju uklanja se sa steka.
- Uzimamo stav sa vrha steka -> **On(A, B)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto se (pod)cilj nalazi u tekućem stanju uklanja se sa steka.



Zadatak 1. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), On(A, B), On(B, C)
- Ciljni stek:
 - U(A, B)
 - Table(A)
- Uzimamo stav sa vrha steka -> **U(A, B)**
 - Kako je reč o operatoru, skida se sa steka i dodaje se **listi akcija**.
 - Stavovi sadržani u UKLONI listi operatora U(A, B) se uklanjaju iz tekućeg stanja.
 - Stavovi sadržani u DODAJ listi operatora U(A, B) se dodaju u tekuće stanje.



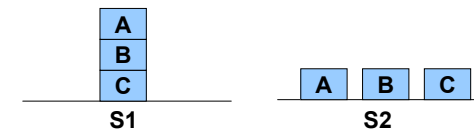
Zadatak 1. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), On(B, C), Table(A), Clear(B)
- Ciljni stek:
 - Table(A)
- Uzimamo stav sa vrha steka -> **Table(A)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto ga ima u tekućem stanju, uklanja se iz steka.
- Stek je nakon ovog koraka prazan.
 - Izdaje se lista akcija: **U(A, B)**



Zadatak 2.

- Koristeći STRIPS algoritam naći sekvencu operatora koji prevode svet blokova iz stanja S1 u stanje S2:



Zadatak 2. (rešenje)

- Inicijalno stanje STRIPS struktura je:
 - Tekuće stanje:
 - Clear(A), Table(C), On(A, B), On(B, C)
 - Ciljni stek:
 - Table(B)
 - Lista akcija je inicijalno prazna.
- Uzimamo stav sa vrha steka -> **Table(B)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto ga nema u tekućem stanju biramo operator koji ga ima u DODAJ delu -> **U(x, y)**
 - Izvrši se unifikacija (pod)cilja i DODAJ dela operatora **U(x, y)**, a zatim (ukoliko je potrebno) i PREDUSLOVA operatora sa tekućim stanjem:
 - Dobija se lista smena: **(x / B, y / C)**



Zadatak 2. (rešenje)

- Novo stanje ciljnog steka sadrži operator i njegove preduslove (**prvo se dodaje operator, pa onda preduslovi**):
 - Clear(B)
 - On(B, C)
 - U(B, C)
 - Table(B)
- Uzimamo stav sa vrha steka -> **Clear(B)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto ga nema u tekućem stanju biramo operator koji ga ima u DODAJ delu -> **U(x, y)** ili **M(x, y, z)**
 - Koristeći heuristiku bira se operator **U(x, y)** jer je moguća unifikacija sa (pod)ciljem **(y / B)** i preduslova sa tekućim stanjem **(x / A)**.



Zadatak 2. (rešenje)

- Novo stanje ciljnog steka sadrži operator i njegove preduslove (**prvo se dodaje operator, pa onda preduslovi**):
 - Clear(A)
 - On(A, B)
 - U(A, B)
 - Clear(B)
 - On(B, C)
 - U(B, C)
 - Table(B)
- Uzimamo stav sa vrha steka -> **Clear(A)**
 - Pošto se (pod)cilj nalazi u tekućem stanju uklanja se sa steka.
- Uzimamo stav sa vrha steka -> **On(A, B)**
 - Pošto se (pod)cilj nalazi u tekućem stanju uklanja se sa steka.



Zadatak 2. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), On(A, B), On(B, C)
- Ciljni stek:
 - U(A, B)
 - Clear(B)
 - On(B, C)
 - U(B, C)
 - Table(B)
- Uzimamo stav sa vrha steka -> **U(A, B)**
 - Kako je reč o operatoru, skida se sa steka i dodaje se **listi akcija**.
 - Stavovi sadržani u UKLONI listi operatora U(A, B) se uklanjaju iz tekućeg stanja.
 - Stavovi sadržani u DODAJ listi operatora U(A, B) se dodaju u tekuće stanje.



Zadatak 2. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), On(B, C), Table(A), Clear(B)
- Ciljni stek:
 - Clear(B)
 - On(B, C)
 - U(B, C)
 - Table(B)
- Lista akcija: U(A, B)
- Uzimamo stav sa vrha steka -> **Clear(B)**
 - Pošto ga ima u tekućem stanju, uklanja se iz steka.
- Uzimamo stav sa vrha steka -> **On(B, C)**
 - Pošto ga ima u tekućem stanju, uklanja se iz steka.



Zadatak 2. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), On(B, C), Table(A), Clear(B)
- Ciljni stek:
 - U(B, C)
 - Table(B)
- Lista akcija: U(A, B)
- Uzimamo stav sa vrha steka -> **U(B, C)**
 - Kako je reč o operatoru, skida se sa steka i dodaje se **listi akcija**.
 - Stavovi sadržani u UKLONI listi operatora U(B, C) se uklanjaju iz tekućeg stanja.
 - Stavovi sadržani u DODAJ listi operatora U(B, C) se dodaju u tekuće stanje.



Zadatak 2. (rešenje)

- Tekuće stanje:
 - Clear(A), Table(C), Table(A), Clear(B), Table(B), Clear(C)
- Ciljni stek:
 - Table(B)
- Lista akcija: U(A, B), U(B, C)
- Uzimamo stav sa vrha steka -> **Table(B)**
 - Kako je reč o (pod)cilju proveravamo da li je zadovoljen.
 - Pošto ga ima u tekućem stanju, uklanja se iz steka.
- Stek je nakon ovog koraka prazan.
 - Izdaje se lista akcija: **U(A, B), U(B, C)**

