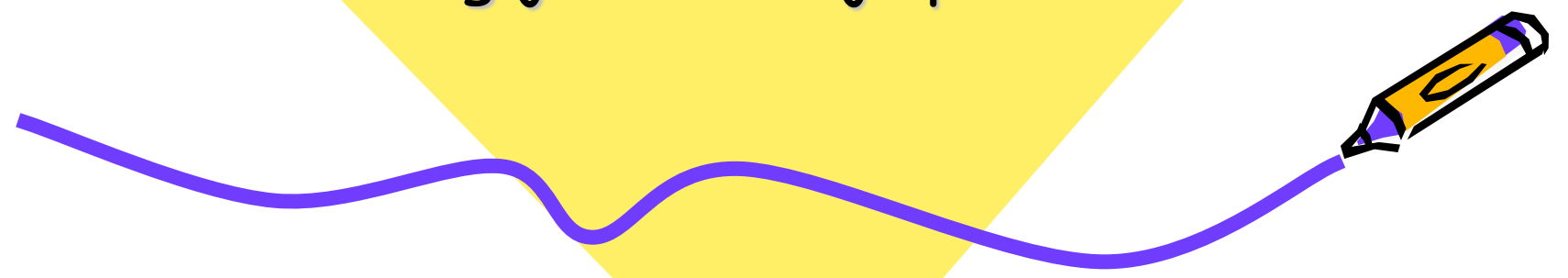




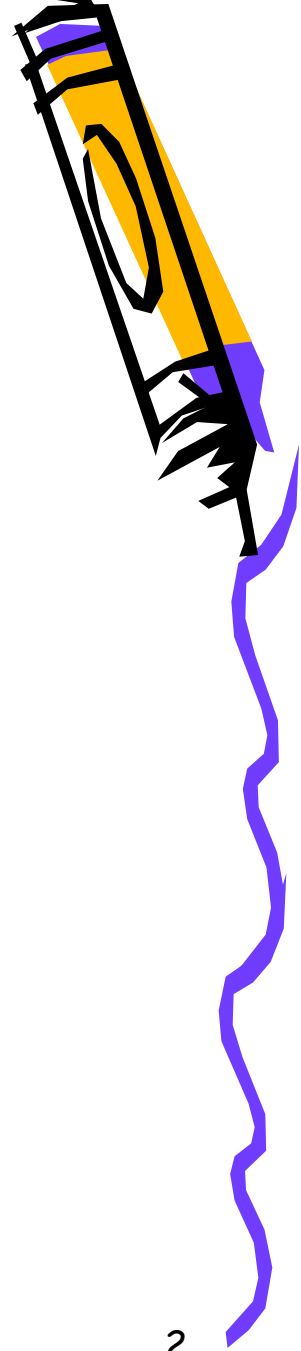
Ekspertski Sistemi Vežbe

Strategije rešavanja problema

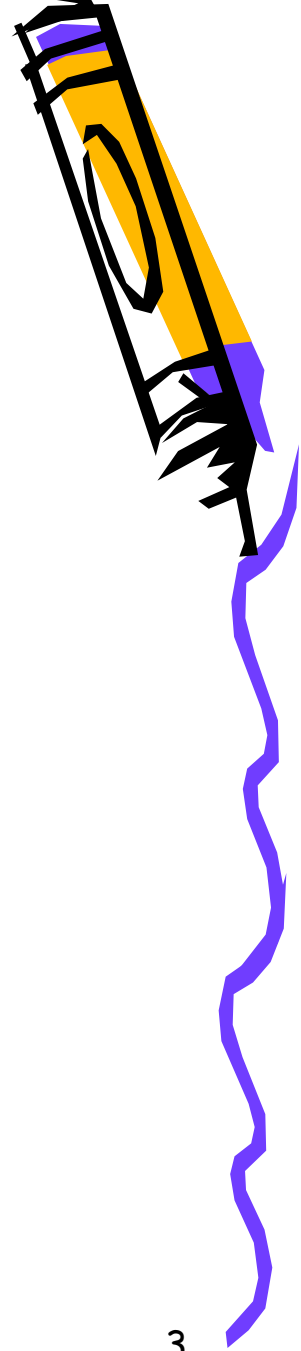


Program vežbi

- Algoritmi pretraživanja
- Teorija igara
- Formalna logika
- Produkcioni sistemi
- **Strategije rešavanja problema**
- Uvod u mašinsko učenje
- Rad u neizvesnom okruženju

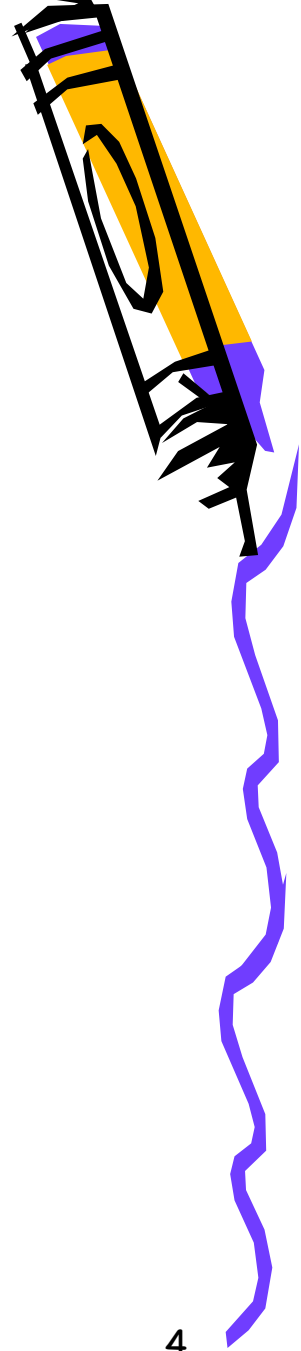


- Planiranje
- Metod zadovoljenja ograničenja
- Metod sukcesivnih aproksimacija (predavanja)



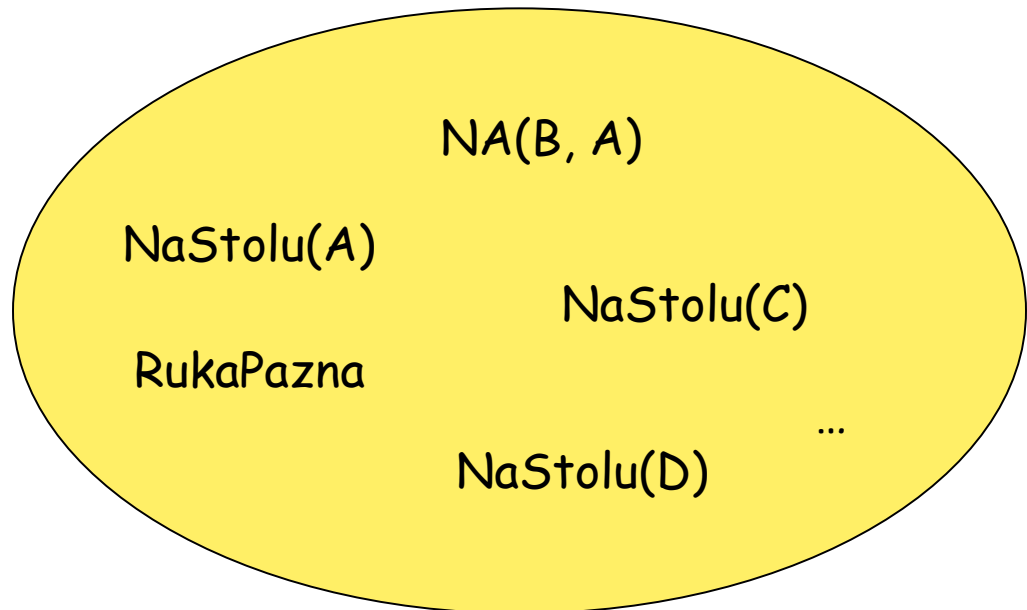
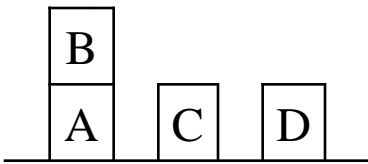
Planiranje

- Proces odlučivanja o akcijama pre njihovog izvršavanja
- Sistemi planiranja se zasnivaju na simuliranju akcija na bazi modela domena



STRIPS

- Služi za rešavanje problema strategijom planiranja
- STRIPS koristi predikatsku logiku za predstavljanje tekućeg stanja sistema (skup izraza koji zajedno kazuju kakva je trenutna situacija)



STRIPS

- Skup operatora koji se koristi za modifikaciju tekućeg stanja

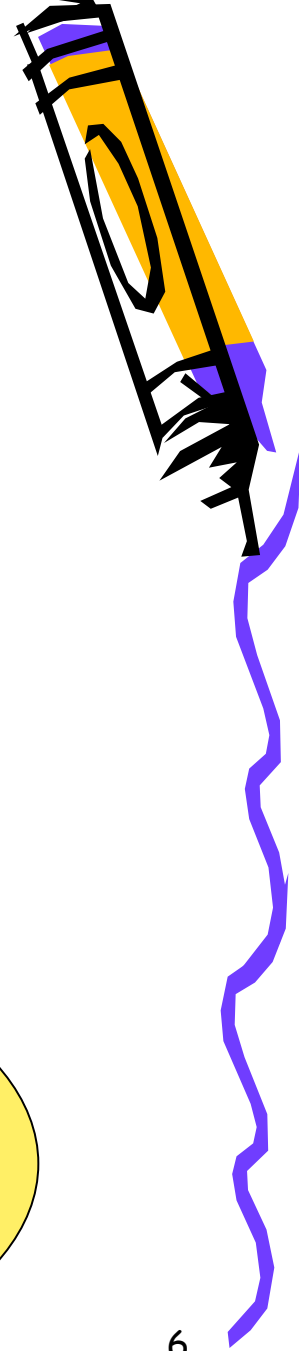
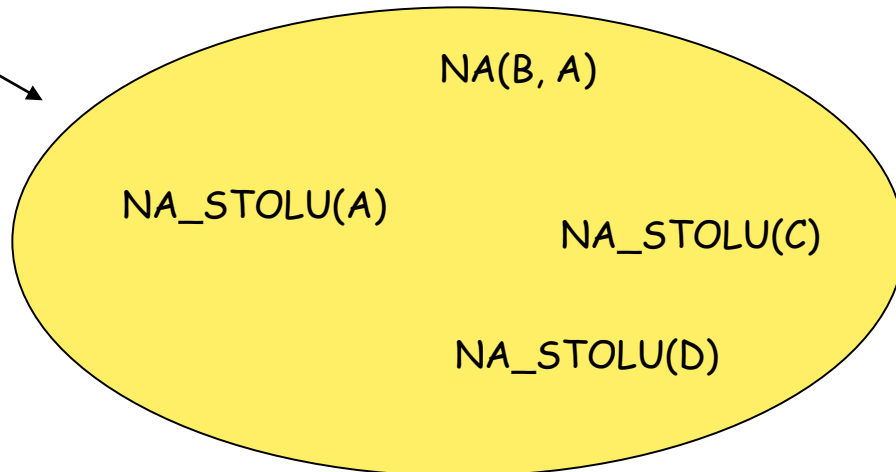
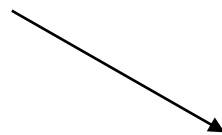
UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna; NaVrhu(x); NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

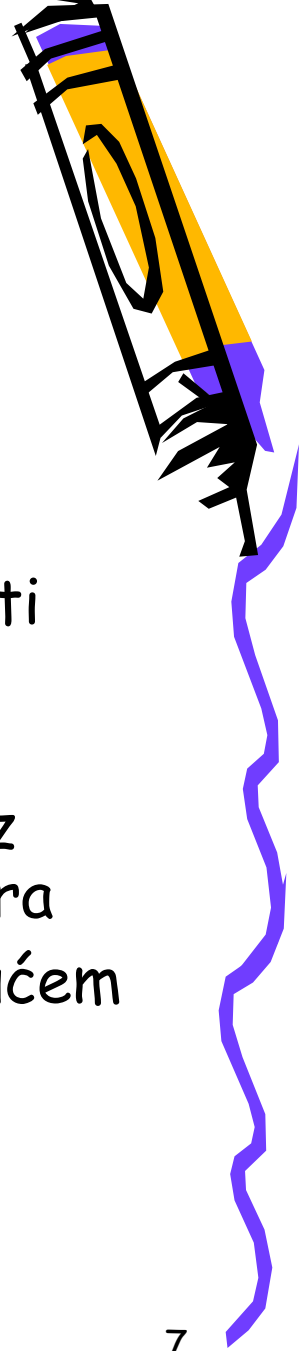
DODAJ: URuci(x)

...



STRIPS

- Operatori promene stanja (tri liste stavova):
 - PREDUSLOV je lista stavova koji moraju biti ispunjeni u tekućem stanju da bi operator mogao biti primenjen
 - UKLONI je lista stavova koji se uklanjaju iz tekućeg stanja u trenutku primene operatora
 - DODAJ je lista stavova koji se dodaju tekućem stanju u trenutku primene operatora

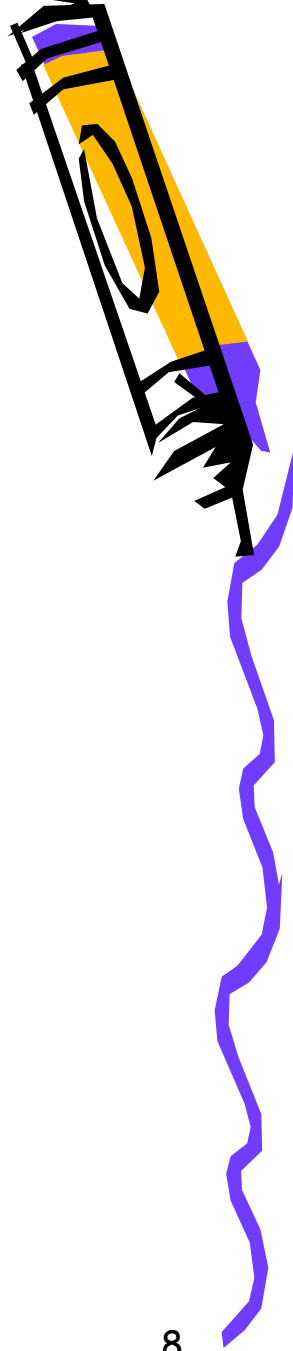


STRIPS

- Cilj je pronaći sekvencu operatora koja sistem prevodi iz početnog u ciljno stanje

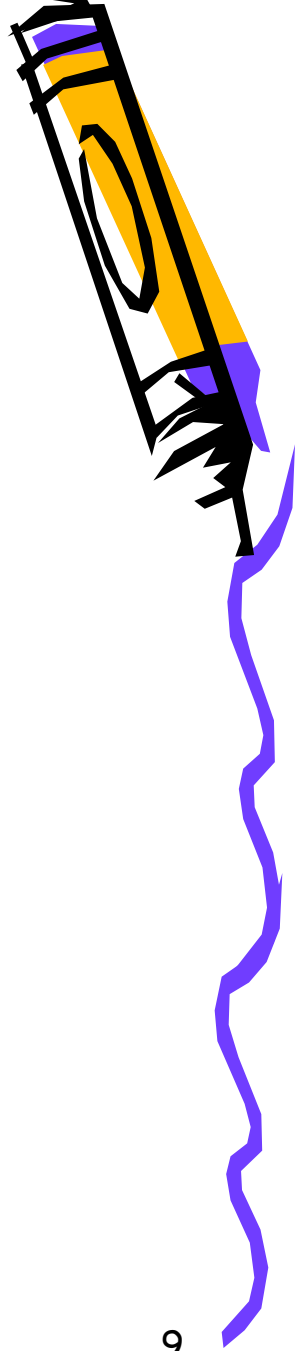
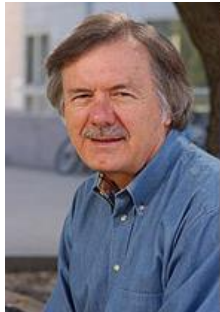


1. SKINI_SA_BLOKA(B,A)
2. STAVI_NA_BLOK(B,D)
3. UZMI_SA_STOLA(C)
4. STAVI_NA_BLOK(C,A)



STRIPS

- **S**Tanford **R**esearch **I**nstitute
Problem **S**olver
- 1971
- *Richard Fikes i Nils Nilsson*



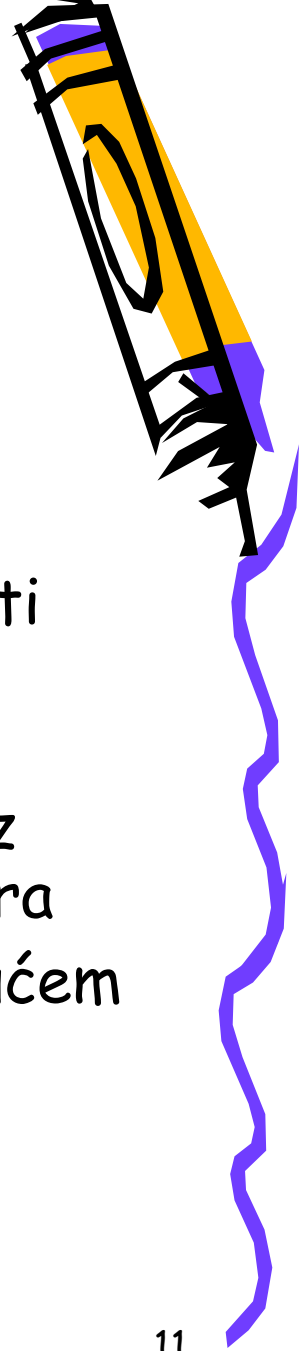
STRIPS

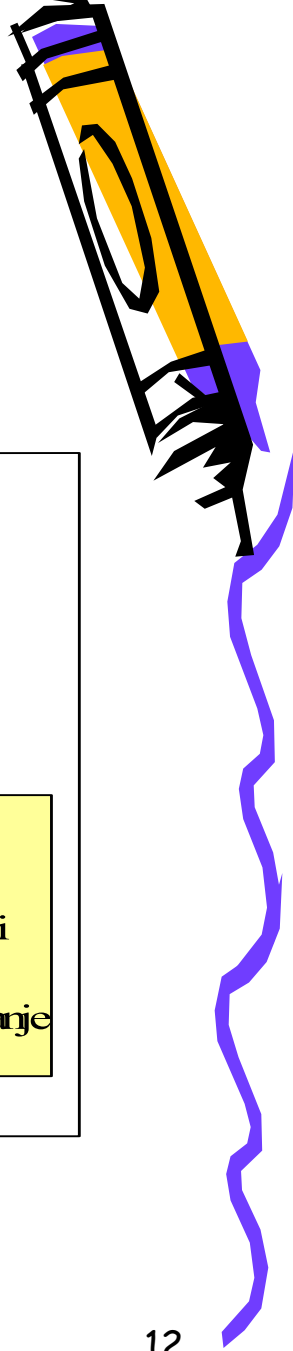
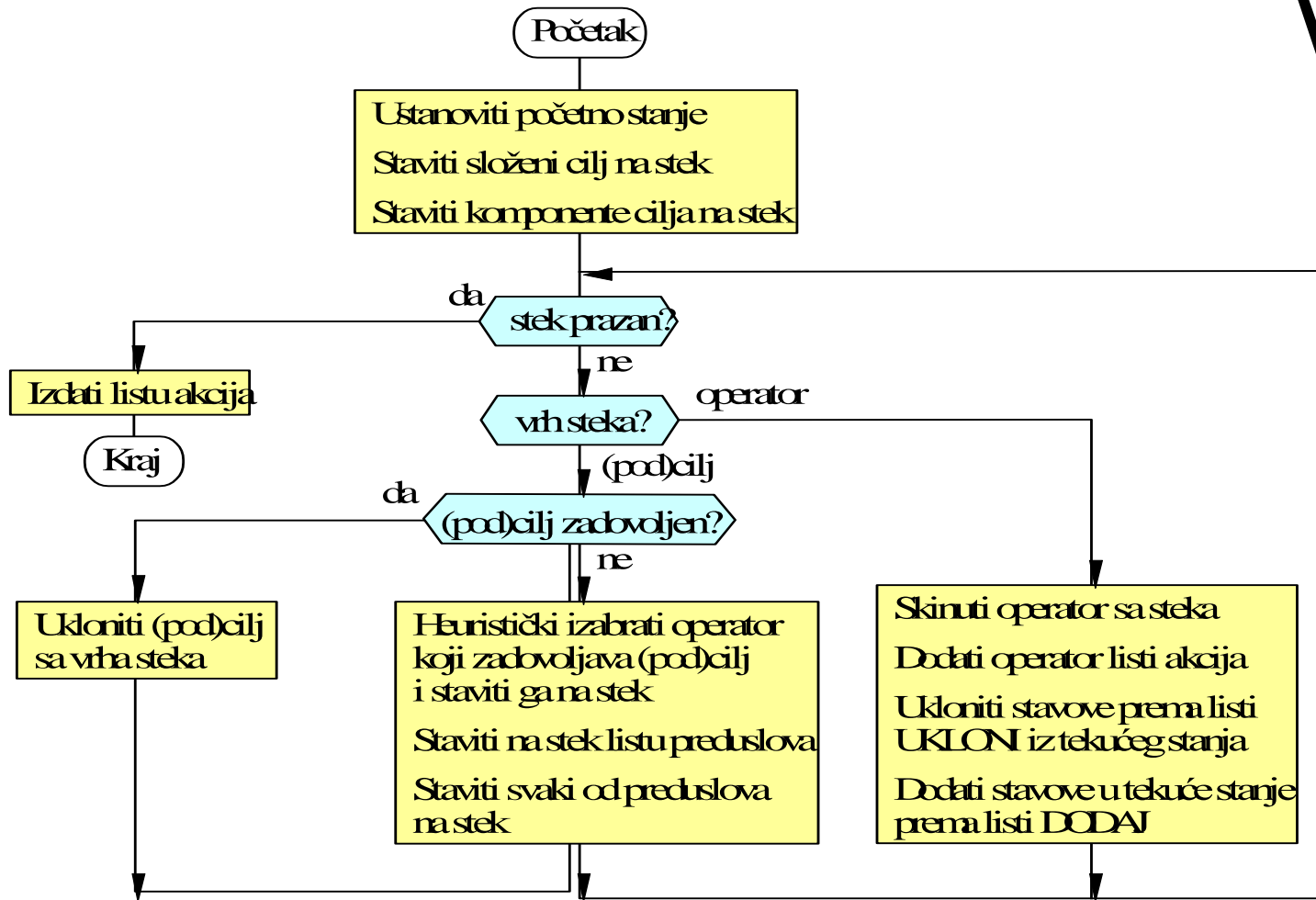
- Koristi sledeće strukture podataka:
 - Tekuće stanje problema (opisano predikatskom logikom). Inicijalno je startno stanje
 - Ciljni stek koji sadrži stavove koji odgovaraju trenutnom (pod)cilju u toku rada algoritma i operatore izabrane za primenu za koje uslovi još nisu zadovoljeni



STRIPS

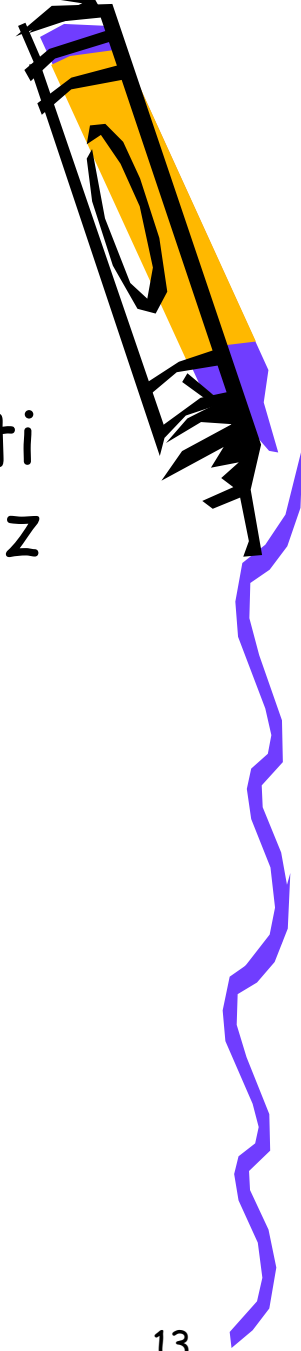
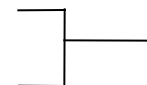
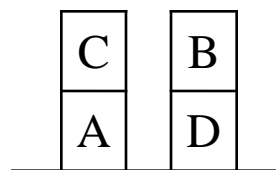
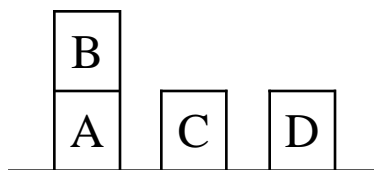
- Operatori promene stanja (tri liste stavova):
 - PREDUSLOV je lista stavova koji moraju biti ispunjeni u tekućem stanju da bi operator mogao biti primenjen
 - UKLONI je lista stavova koji se uklanjaju iz tekućeg stanja u trenutku primene operatora
 - DODAJ je lista stavova koji se dodaju tekućem stanju u trenutku primene operatora





Zadatak 1: Svet blokova

Primenom STRIPS mehanizma generisati niz operatora kojima bi se niz blokova uz pomoć mehaničke hvataljke preveo iz početnog u završno stanje



Uvedeni predikati za opis stanja

- $Na(B, A)$ označava da se blok B nalazi na bloku A
- $NaStolu(C)$ označava da se blok C nalazi neposredno na stolu
- $NaVrhu(B)$ označava da se iznad bloka B ne nalazi nijedan drugi blok
- $RukaPrazna$ označava da nema blokova u hvataljci
- $URuci(A)$ označava da se u hvataljci nalazi blok A



Početno stanje

Na(B,A)

NaStolu(A)

NaStolu(C)

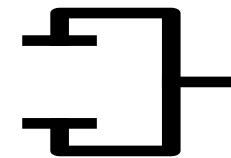
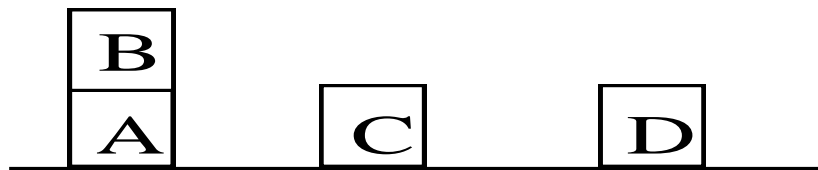
NaStolu(D)

NaVrhu(B)

NaVrhu(C)

NaVrhu(D)

RukaPrazna



Ciljno stanje

Na(C,A)

NaStolu(A)

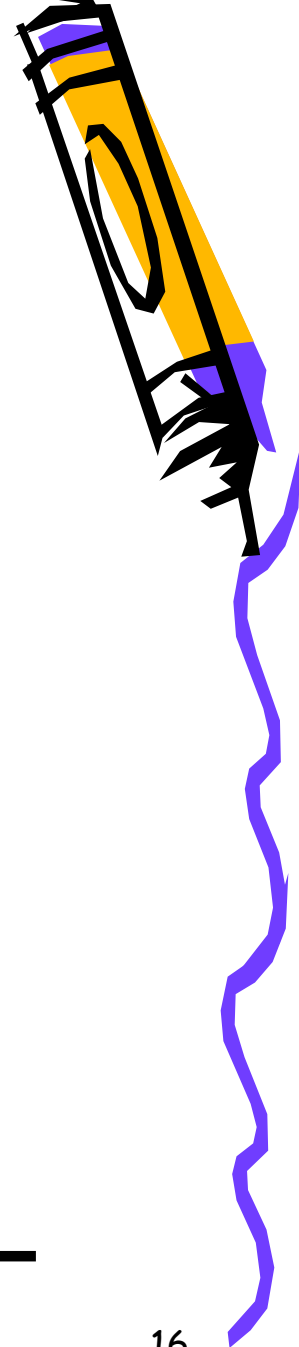
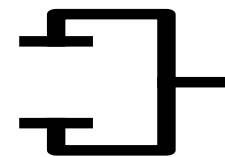
NaVrhu(B)

RukaPrazna

NaVrhu(C)

Na(B,D)

NaStolu(D)



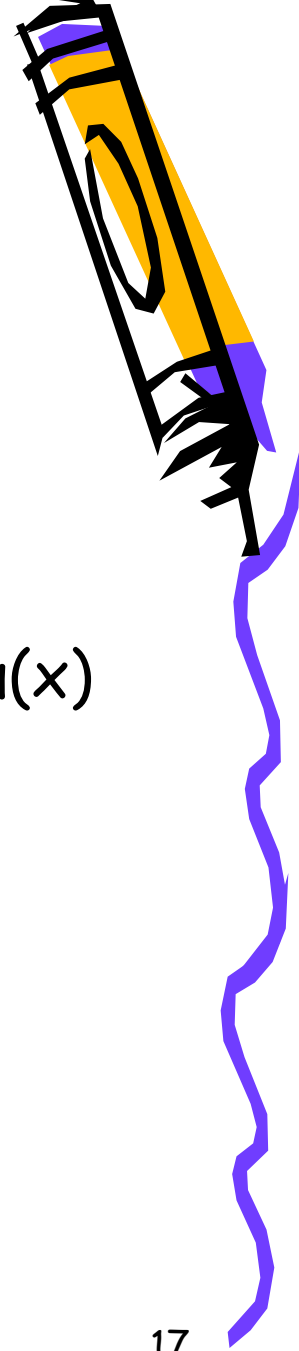
Uvedeni operatori (akcije hvataljke)

- operator **UZMI_SA_STOLA(x)**:

PREDUSLOV: $RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)$

UKLONI: $RukaPrazna; NaVrhu(x); NaStolu(x)$

DODAJ: $URuci(x)$



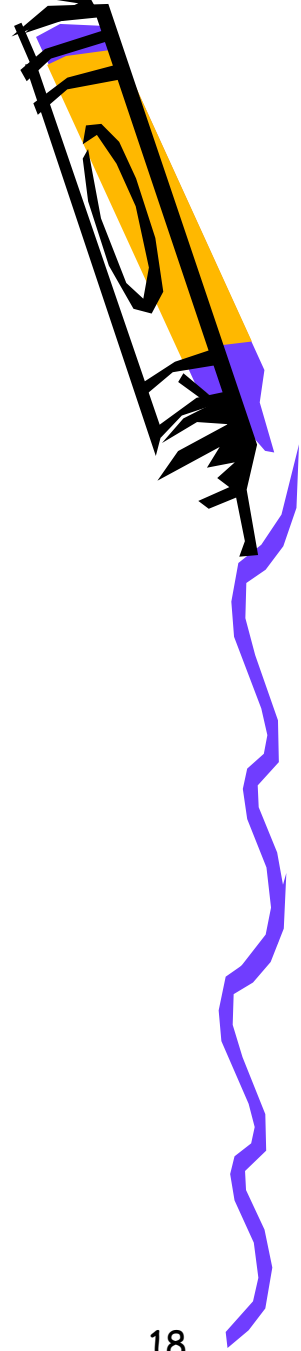
Uvedeni operatori (akcije hvataljke)

- operator **SPUSTI_NA_STO(y)**:

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)



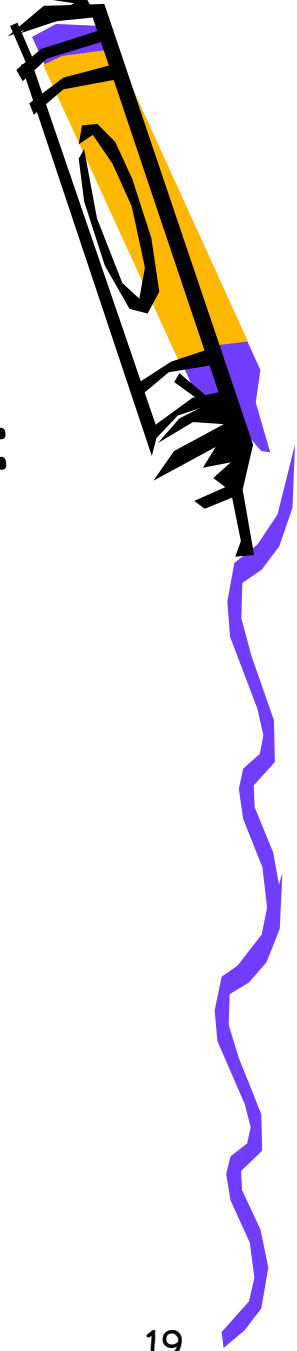
Uvedeni operatori (akcije hvataljke)

- operator **SKINI_SA_BLOKA**(u, z):

PREDUSLOV: $RukaPrazna \wedge NaVrhu(u) \wedge Na(u, z)$

UKLONI: $RukaPrazna; NaVrhu(u); Na(u, z)$

DODAJ: $URuci(u); NaVrhu(z)$



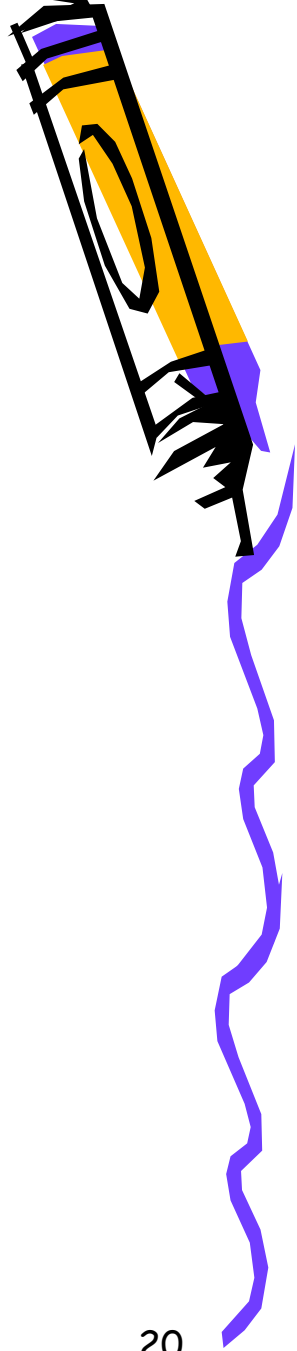
Uvedeni operatori (akcije hvataljke)

- operator **STAVI_NA_BLOK**(v, w):

PREDUSLOV: URuci(v) NaVrhu(w)

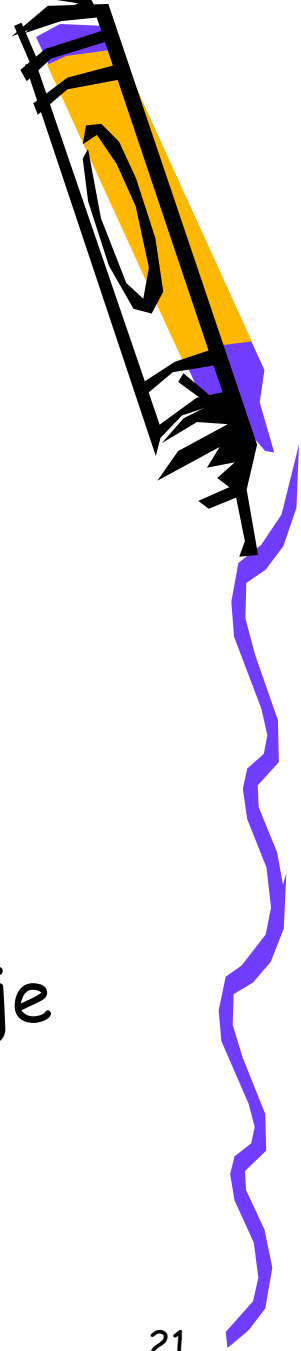
UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v, w)



Inicijalizacija algoritma

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



Tekuće stanje:

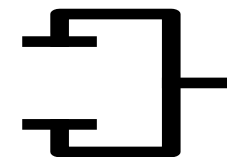
Na(B,A) NaVrhu(B)
NaStolu(A) NaVrhu(C)
NaStolu(C) NaVrhu(D)
NaStolu(D) RukaPrazna

Ciljni stek:

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:

-



Tekuće stanje:

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

Ciljni stek:

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)

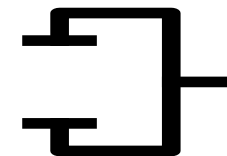
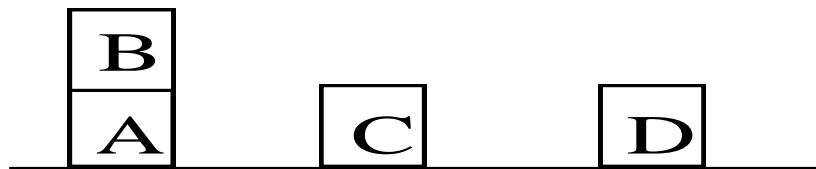
NaVrhu(B)

NaStolu(D)

RukaPrazna

Lista akcija:

-



Tekuće stanje:

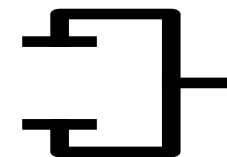
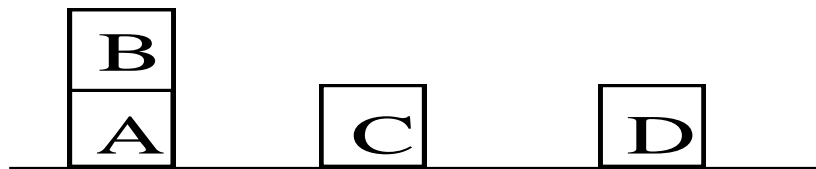
Na(B,A) NaVrhu(B)
NaStolu(A) NaVrhu(C)
NaStolu(C) NaVrhu(D)
NaStolu(D) RukaPrazna

Ciljni stek:

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:

-



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

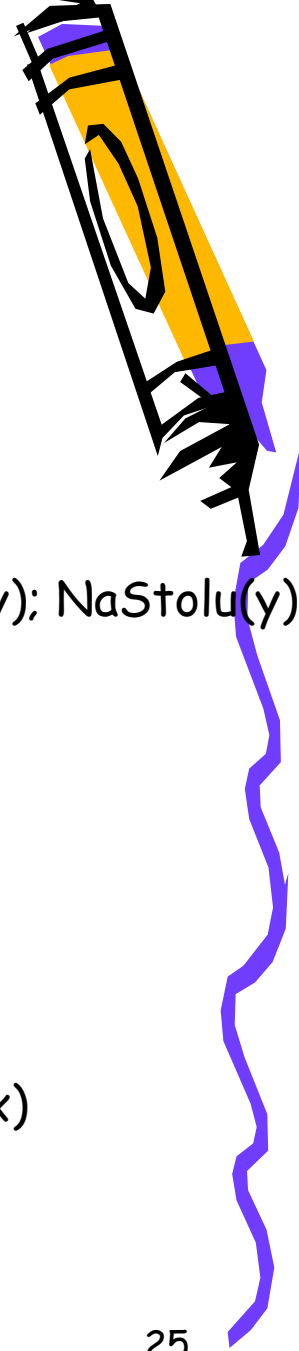
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

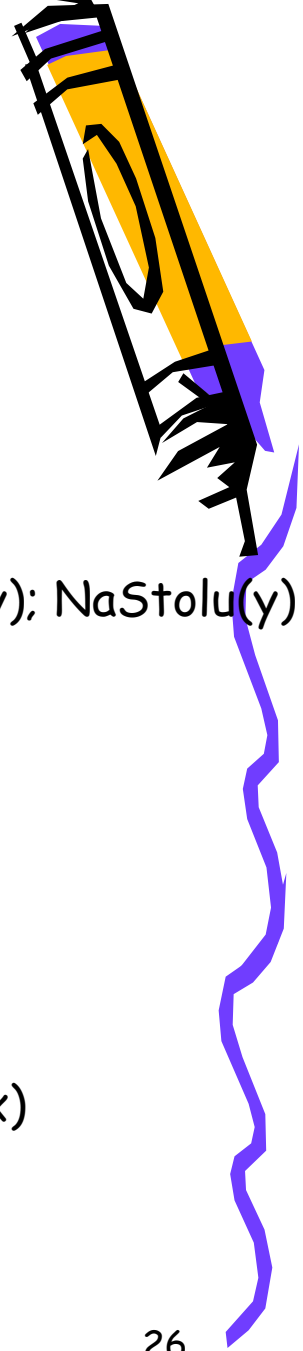
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Na stek dodajemo operator, preduslov i svaku komponentu preduslova

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

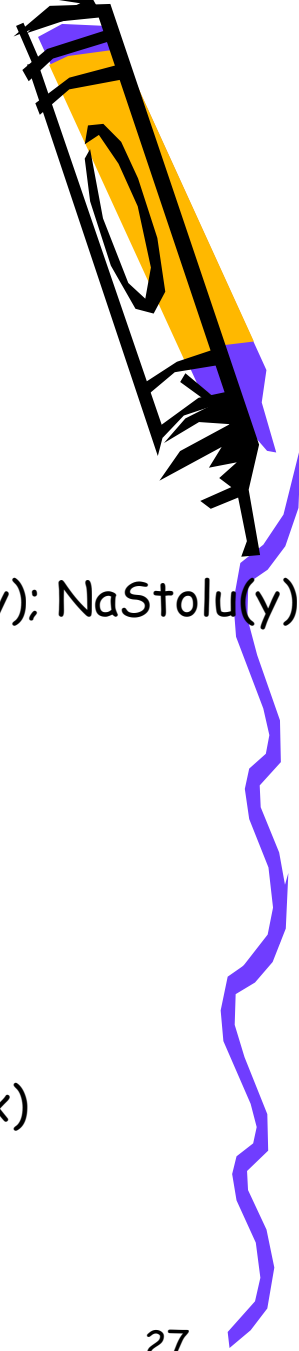
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Tekuće stanje:

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

Ciljni stek:

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)

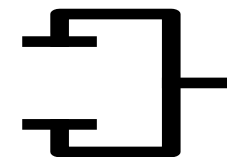
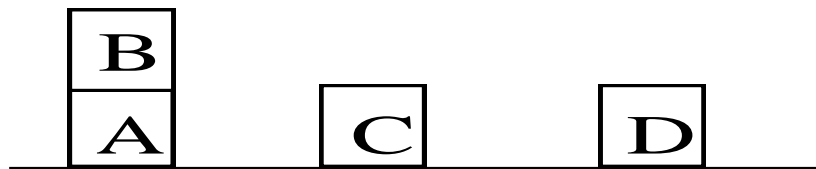
URuci(B) \wedge NaVrhu(D)

URuci(B)

NaVrhu(D)

Lista akcija:

-



Tekuće stanje:

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

Ciljni stek:

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)

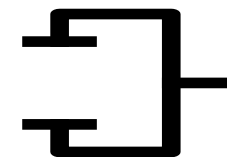
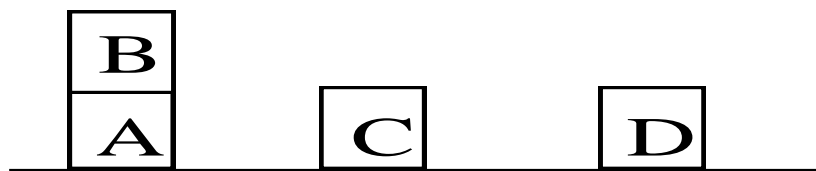
URuci(B) \wedge NaVrhu(D)

URuci(B)

NaVrhu(D)

Lista akcija:

-



Tekuće stanje:

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

Ciljni stek:

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)

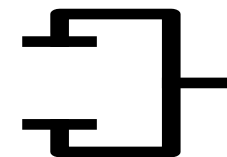
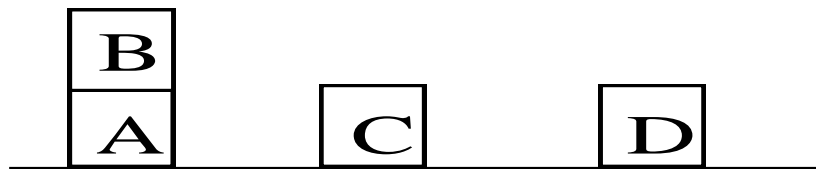
STAVI_NA_BLOK(B,D)

URuci(B) \wedge NaVrhu(D)

URuci(B)

Lista akcija:

-



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

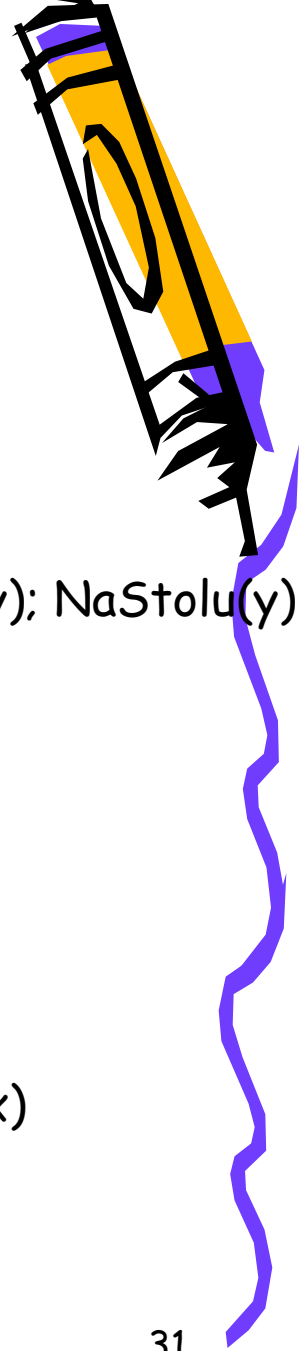
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

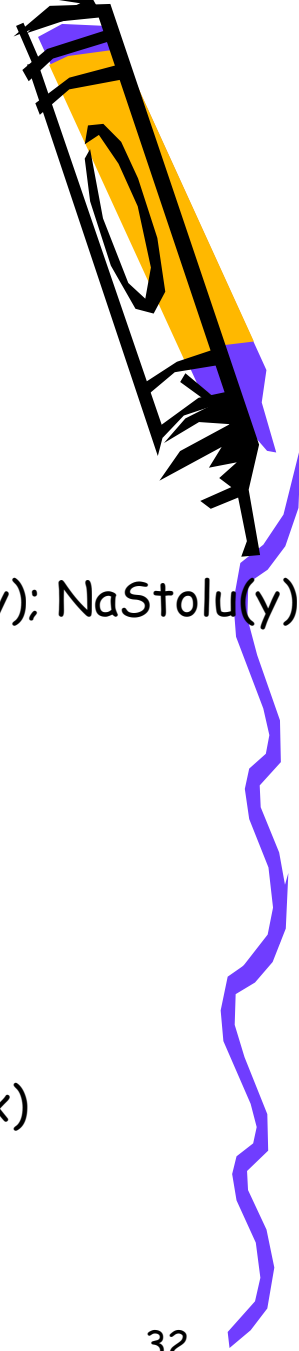
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Tekuće stanje:

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

Ciljni stek:

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)

SKINI_SA_BLOKA(B,A)

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

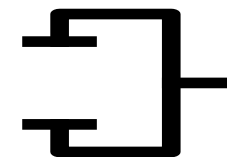
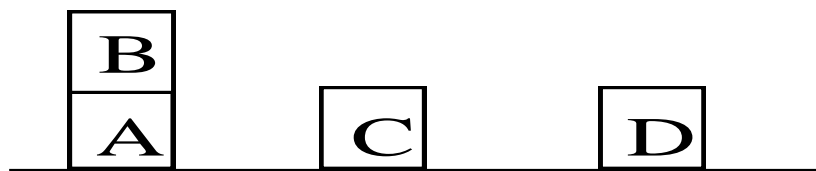
RukaPrazna

NaVrhu(B)

Na(B,A)

Lista akcija:

-



Tekuće stanje:

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

Ciljni stek:

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)

SKINI_SA_BLOKA(B,A)

RukaPrazna NaVrhu(B) Na(B,A)

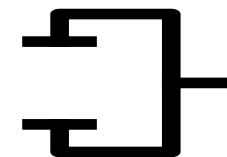
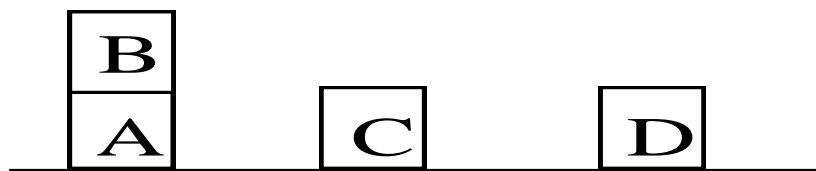
RukaPrazna

NaVrhu(B)

Na(B,A)

Lista akcija:

-



Tekuće stanje:

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

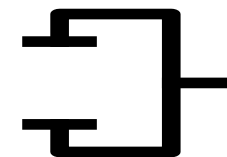
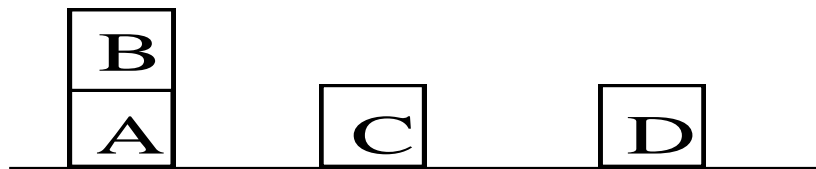
Ciljni stek:

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:

SKINI_SA_BLOKA(B,A)

=> modifikacija sveta



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaStolu(C) NaVrhu(C)

NaStolu(D) NaVrhu(D)

URuci(B)

Ciljni stek:

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

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

STAVI_NA_BLOK(B,D)

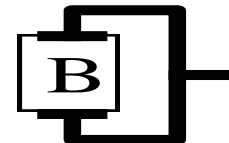
URuci(B) \wedge NaVrhu(D)

URuci(B)

Lista akcija:

SKINI_SA_BLOKA(B,A)

=> modifikacija sveta



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaStolu(C) NaVrhu(C)

NaStolu(D) NaVrhu(D)

URuci(B)

Ciljni stek:

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

Na(C,A)

NaVrhu(C)

NaStolu(A)

Na(B,D)

STAVI_NA_BLOK(B,D)

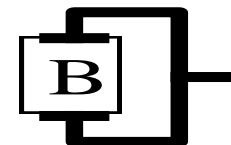
URuci(B) \wedge NaVrhu(D)

URuci(B)

Lista akcija:

SKINI_SA_BLOKA(B,A)

=> modifikacija sveta



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaStolu(C) NaVrhu(C)

NaStolu(D) NaVrhu(D)

URuci(B)

Ciljni stek:

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

Na(C,A)

NaVrhu(C)

NaStolu(A)

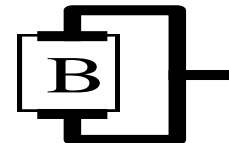
Na(B,D)

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

=> modifikacija sveta



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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)

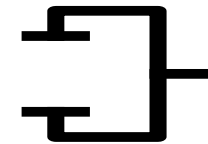
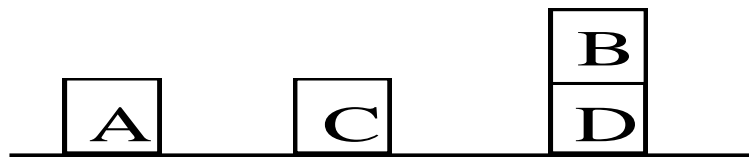
NaStolu(A)

Na(B,D)

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

Na(C,A)

NaVrhu(C)

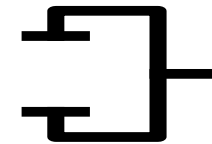
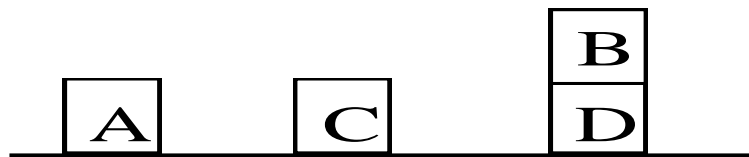
NaStolu(A)

Na(B,D)

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

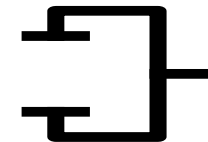
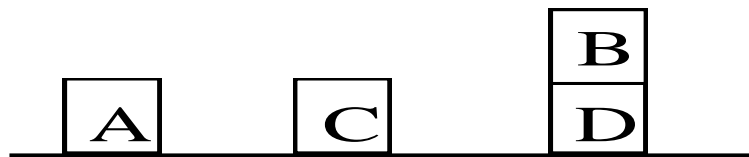
$\text{Na}(C,A) \wedge \text{NaVrhu}(C) \wedge \text{NaStolu}(A) \wedge \text{Na}(B,D) \wedge \text{NaVrhu}(B) \wedge \text{NaStolu}(D) \wedge \text{RukaPrazna}$

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

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

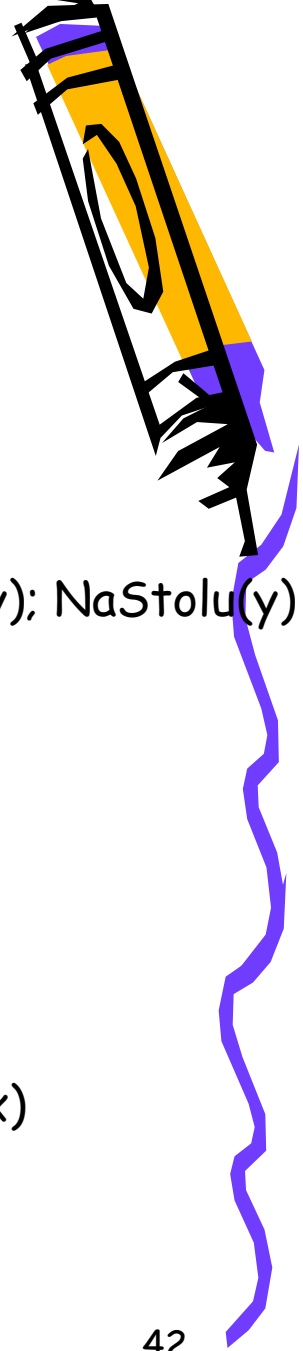
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

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

STAVI_NA_BLOK(C,A)

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

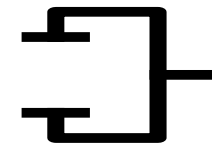
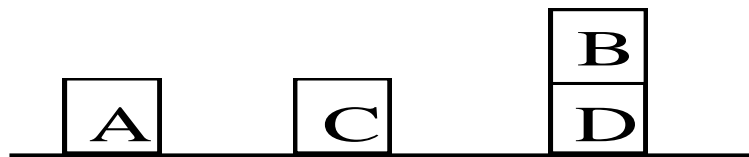
$\text{URuci}(C)$

$\text{NaVrhu}(A)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

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

STAVI_NA_BLOK(C,A)

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

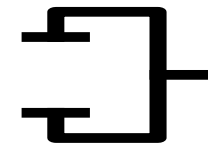
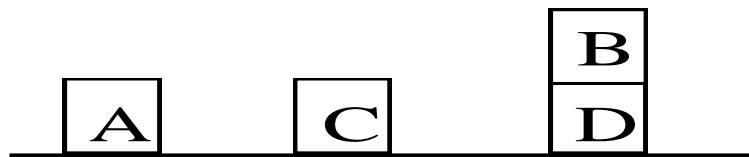
$\text{URuci}(C)$

$\text{NaVrhu}(A)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

$Na(C,A)$

STAVI_NA_BLOK(C,A)

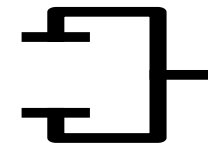
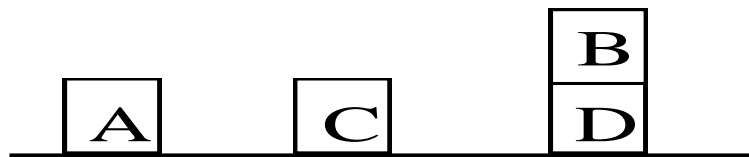
$URuci(C) \wedge NaVrhu(A)$

$URuci(C)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Izbor operatora

STAVI_NA_BLOK(v,w):

PREDUSLOV: URuci(v) NaVrhu(w)

UKLONI: URuci(v); NaVrhu(w)

DODAJ: RukaPrazna; NaVrhu(v); Na(v,w)

SPUSTI_NA_STO(y):

PREDUSLOV: URuci(y)

UKLONI: URuci(y)

DODAJ: RukaPrazna; NaVrhu(y); NaStolu(y)

SKINI_SA_BLOKA(u,z):

PREDUSLOV: RukaPrazna \wedge NaVrhu(u) \wedge Na(u,z)

UKLONI: RukaPrazna; NaVrhu(u); Na(u,z)

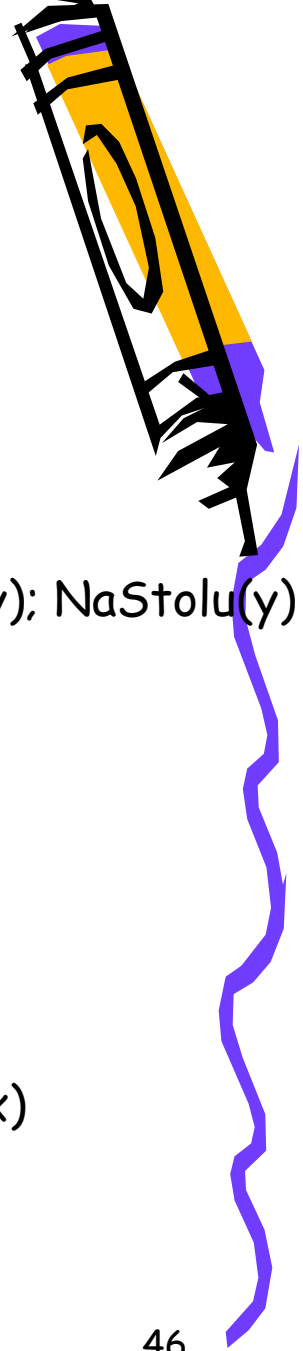
DODAJ: URuci(u); NaVrhu(z)

UZMI_SA_STOLA(x):

PREDUSLOV: RukaPrazna \wedge NaVrhu(x) \wedge NaStolu(x)

UKLONI: RukaPrazna; NaVrhu(x); NaStolu(x)

DODAJ: URuci(x)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

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

STAVI_NA_BLOK(C,A)

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

$\text{URuci}(C)$

UZMI_SA_STOLA(C)

$\text{RukaPrazna NaVrhu}(C) \text{ NaStolu}(C)$

RukaPrazna

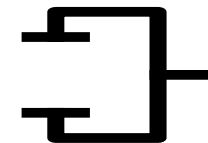
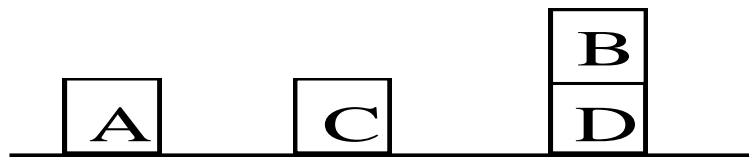
$\text{NaVrhu}(C)$

$\text{NaStolu}(C)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

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

STAVI_NA_BLOK(C,A)

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

$\text{URuci}(C)$

UZMI_SA_STOLA(C)

RukaPrazna NaVrhu(C) NaStolu(C)

RukaPrazna

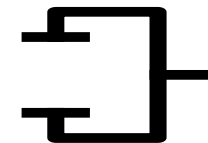
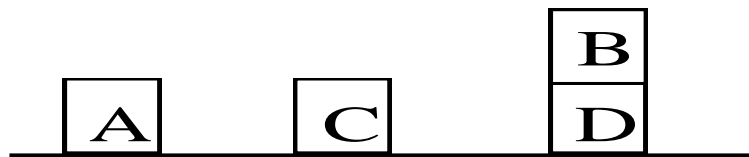
NaVrhu(C)

NaStolu(C)

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

NaVrhu(C) NaStolu(C)

Na(B, D) NaVrhu(B)

NaStolu(D) RukaPrazna

Ciljni stek:

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

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

STAVI_NA_BLOK(C,A)

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

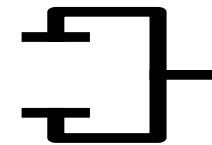
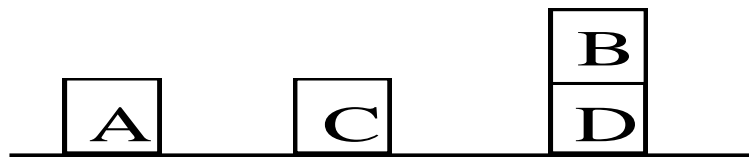
$\text{URuci}(C)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

UZMI_SA_STOLA(C)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

Na(B, D) NaVrhu(B)

NaStolu(D) URuci(C)

Ciljni stek:

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

STAVI_NA_BLOK(C,A)

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

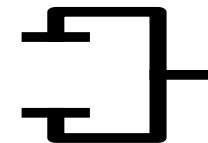
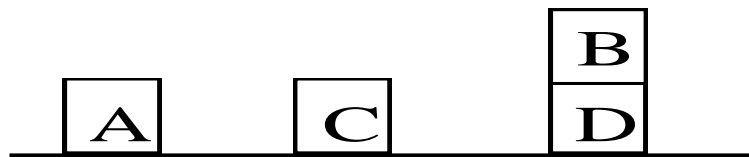
$\text{URuci}(C)$

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

UZMI_SA_STOLA(C)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

Na(B, D) NaVrhu(B)

NaStolu(D) URuci(C)

Ciljni stek:

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

STAVI_NA_BLOK(C,A)

URuci(C) \wedge NaVrhu(A)

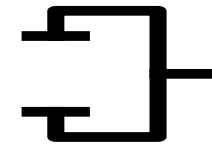
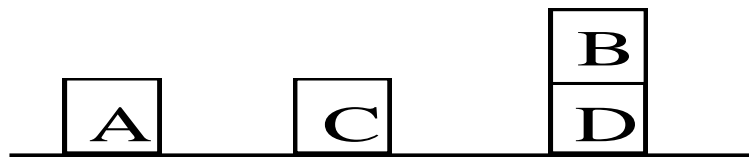
URuci(C)

Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

UZMI_SA_STOLA(C)



Tekuće stanje:

NaStolu(A) NaVrhu(A)

Na(B, D) NaVrhu(B)

NaStolu(D) URuci(C)

Ciljni stek:

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

Na(C,A)

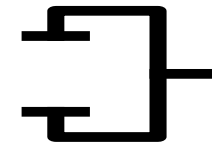
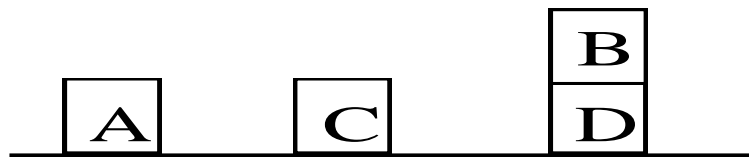
Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

UZMI_SA_STOLA(C)

STAVI_NA_BLOK(C,A)



Tekuće stanje:

NaStolu(A) Na(C, A)

NaVrhu(C) Na(B, D)

NaVrhu(B) NaStolu(D)

RukaPrazna

Ciljni stek:

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

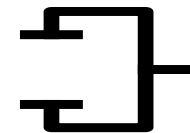
Lista akcija:

SKINI_SA_BLOKA(B,A)

STAVI_NA_BLOK(B,D)

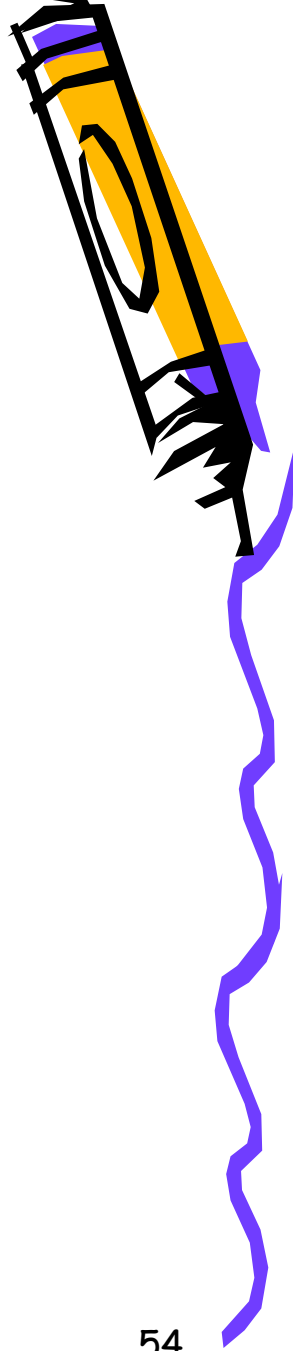
UZMI_SA_STOLA(C)

STAVI_NA_BLOK(C,A)



Zadatak 2: Svet blokova (problem izbora operatora)

Dato je jedno stanje iz sveta blokova.
Pronaći koje operatore i kojim redom
treba primeniti, da bi se prešlo u ciljnu
poziciju, koristeći STRIPS algoritam



Isti predikati

- Početno stanje

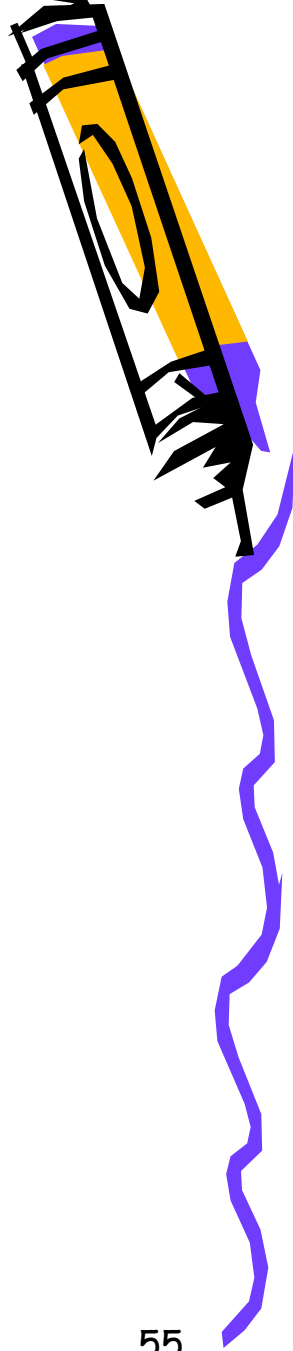
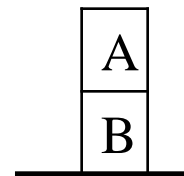
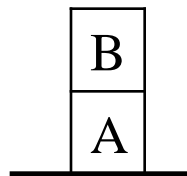
NaStolu(A) Na(B,A)

NaVrhu(B) RukaPrazna

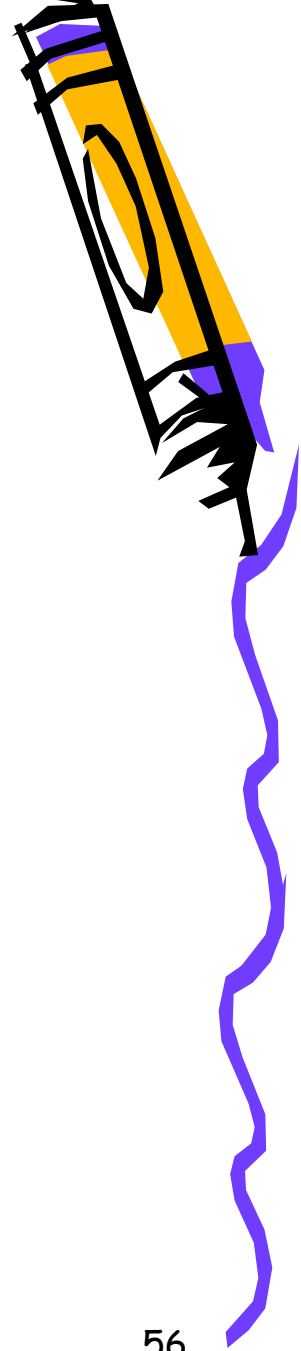
- Ciljno stanje

NaStolu(B) Na(A,B)

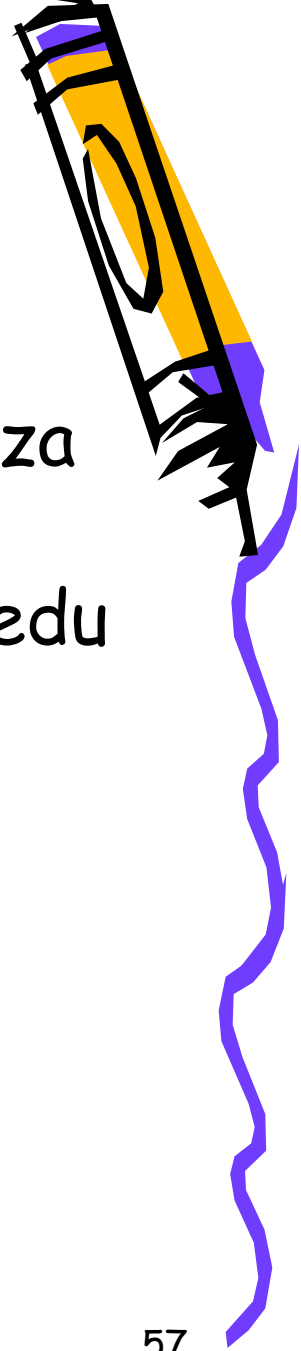
NaVrhu(A) RukaPrazna



- Nije lako definisati algoritam koji obezbedjuje optimalan redosled izbora operatora u opštem slučaju



- Razmotrimo rad STRIPS algoritma ako za izbor operatora usvojimo jednostavno pravilo da se operatori biraju po redosledu koji su definisani:
 - UZMI_SA_STOLA(x)
 - SPUSTI_NA_STO(y)
 - SKINI_SA_BLOKA(u,z)
 - STAVI_NA_BLOK(v,w)



Tekuće stanje:

NaStolu(A) NA(B,A)

NaVrhu(B) RukaPrazna

Ciljni stek:

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

NaStolu(B)

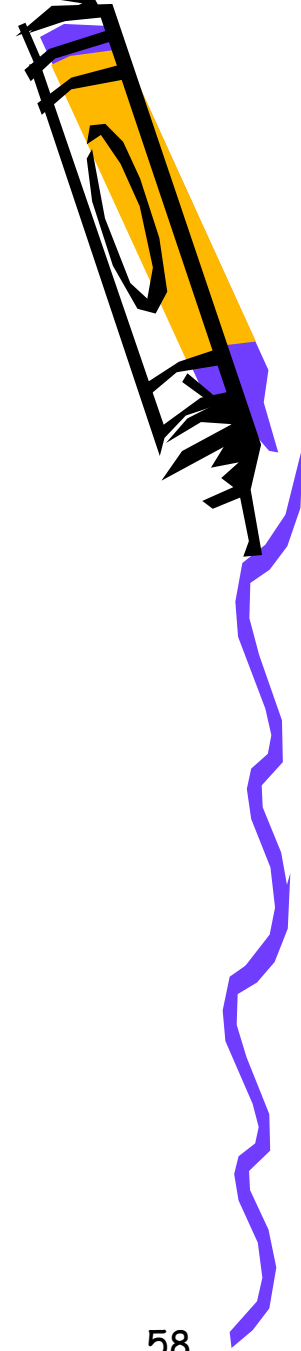
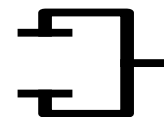
NA(A,B)

NaVrhu(A)

RukaPrazna

Lista akcija:

-



Tekuće stanje:

NaStolu(A) NA(B,A)

NaVrhu(B) RukaPrazna

Ciljni stek:

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

NaStolu(B)

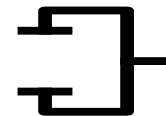
NA(A,B)

NaVrhu(A)

RukaPrazna

Lista akcija:

-



Tekuće stanje:

NaStolu(A) NA(B,A)

NaVrhu(B) RukaPrazna

Ciljni stek:

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

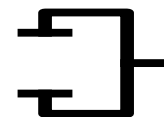
NaStolu(B)

NA(A,B)

NaVrhu(A)

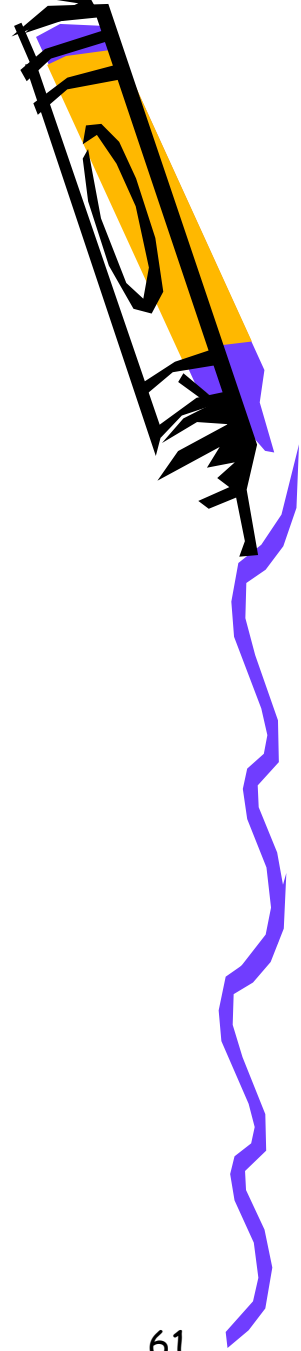
Lista akcija:

-



Mogući operatori

- **SPUSTI_NA_STO(A)**
- SKINI_SA_BLOKA(x,A)
- STAVI_NA_BLOK(A,w)



Tekuće stanje:

NaStolu(A) NA(B,A)

NaVrhu(B) RukaPrazna

Ciljni stek:

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

NaStolu(B)

NA(A,B)

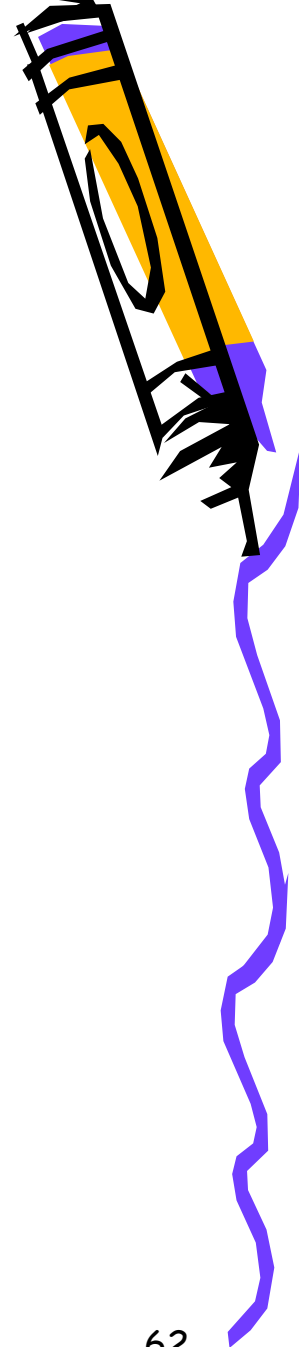
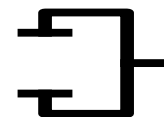
NaVrhu(A)

SPUSTI_NA_STO(A)

URuci(A)

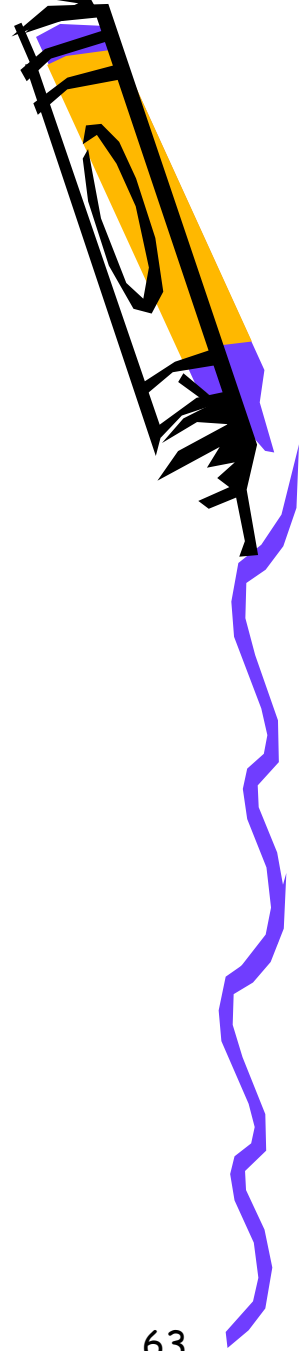
Lista akcija:

-



Mogući operatori

- UZMI_SA_STOLA(A)
- SKINI_SA_BLOKA(A,z)



Tekuće stanje:

$\text{NaStolu}(A) \wedge \text{NA}(B,A)$

$\text{NaVrhu}(B) \wedge \text{RukaPrazna}$

Ciljni stek:

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

$\text{NaStolu}(B)$

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

$\text{NaVrhu}(A)$

SPUSTI_NA_STO(A)

$\text{URuci}(A)$

UZMI_SA_STOLA(A)

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

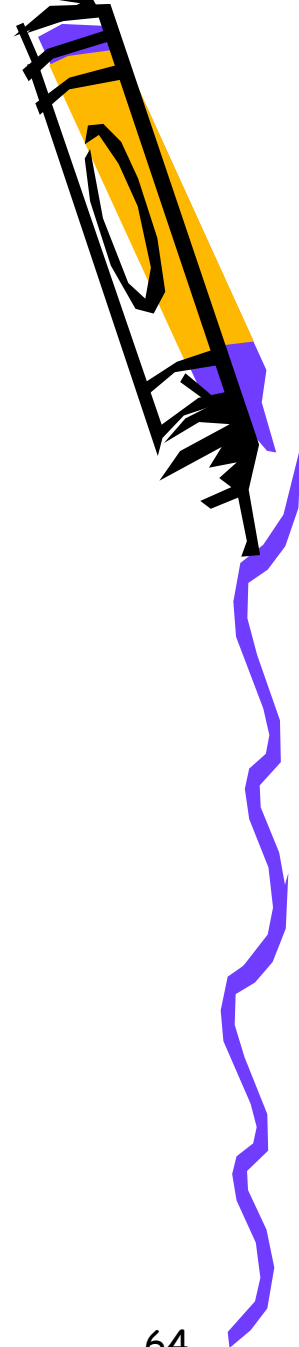
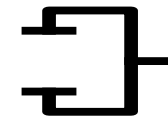
RukaPrazna

$\text{NaVrhu}(A)$

$\text{NaStolu}(A)$

Lista akcija:

-



Tekuće stanje:

$\text{NaStolu}(A) \wedge \text{NA}(B,A)$

$\text{NaVrhu}(B) \wedge \text{RukaPrazna}$

Ciljni stek:

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

$\text{NaStolu}(B)$

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

$\text{NaVrhu}(A)$

SPUSTI_NA_STO(A)

$\text{URuci}(A)$

UZMI_SA_STOLA(A)

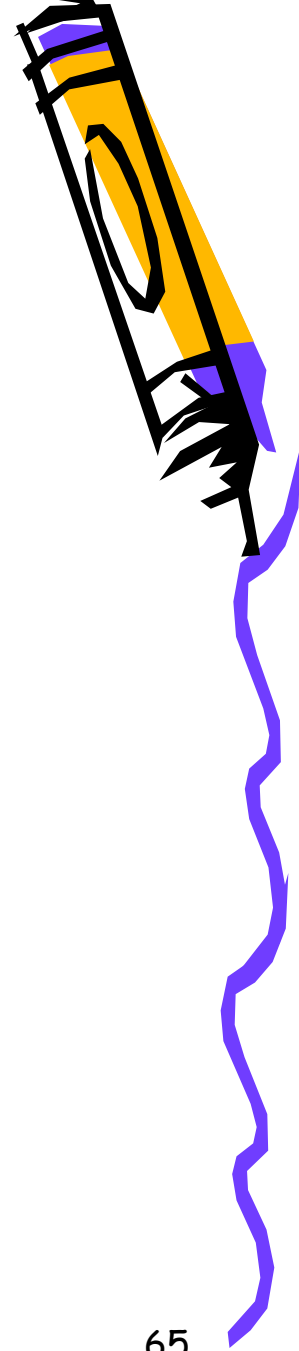
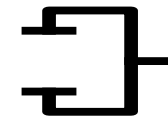
$\text{RukaPrazna} \wedge \text{NaVrhu}(A) \wedge \text{NaStolu}(A)$

RukaPrazna

$\text{NaVrhu}(A)$

Lista akcija:

-



Tekuće stanje:

NaStolu(A) NA(B,A)

NaVrhu(B) RukaPrazna

Ciljni stek:

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)~~

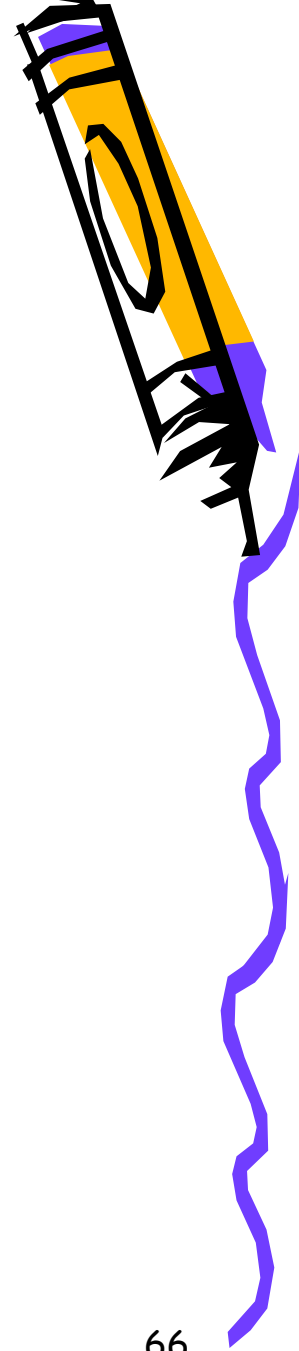
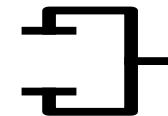
~~RukaPrazna~~

~~NaVrhu(A)~~

Lista akcija:

-

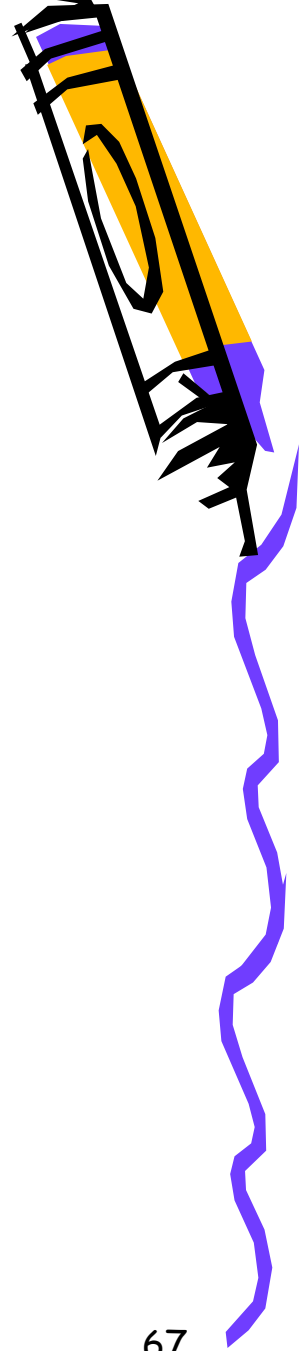
Vraćanje unatrag



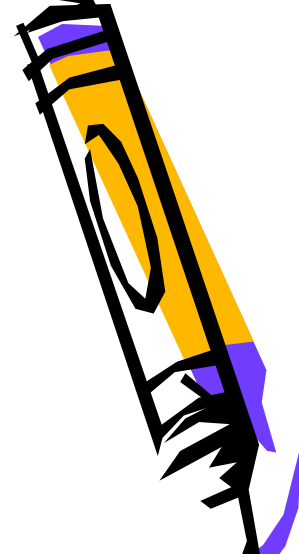
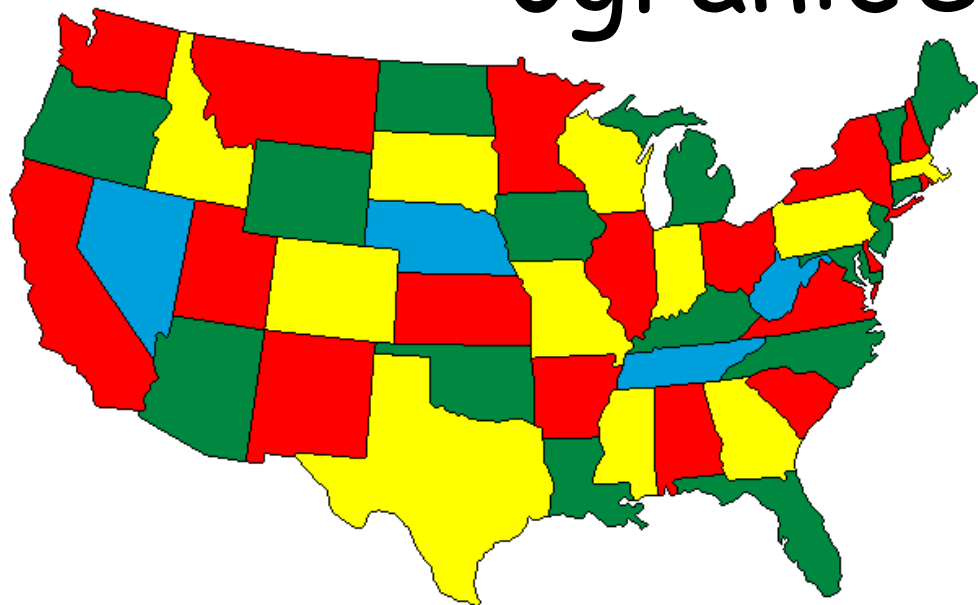
Mogući operatori

- UZMI_SA_STOLA(A)
- SKINI_SA_BLOKA(A,z)

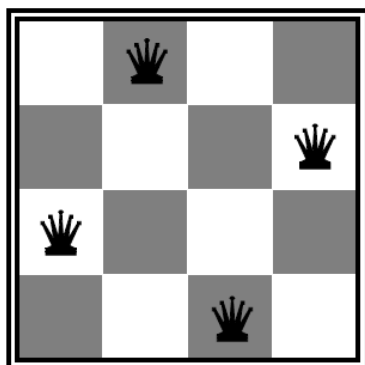
....



Metod zadovoljenja ograničenja



8			4	6			7
						4	
	1					6	5
5		9		3		7	8
				7			
	4	8		2		1	3
	5	2					9
		1					
3			9	2			5



Constraint satisfaction problems (CSPs)

- Standardna pretraga:

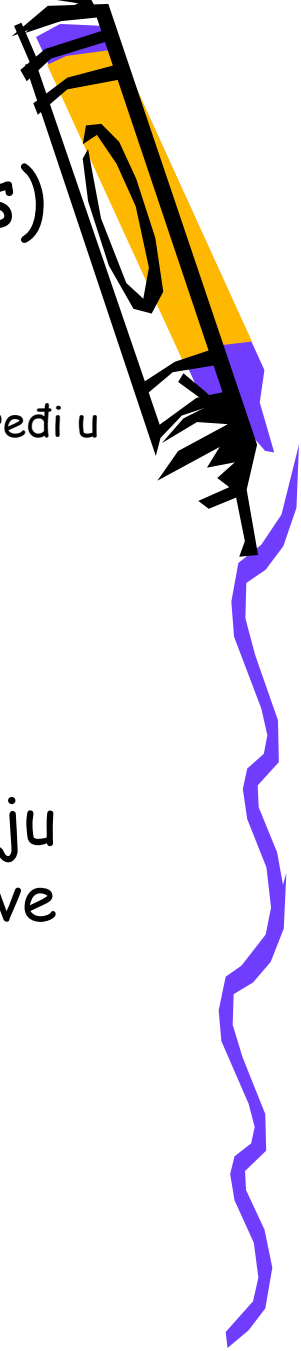
Stanje je „crna kutija“ - bilo koja struktura koja podržava funkciju „pređi u sledeće stanje“, heurističku funkciju i testiranje cilja

Definicija:

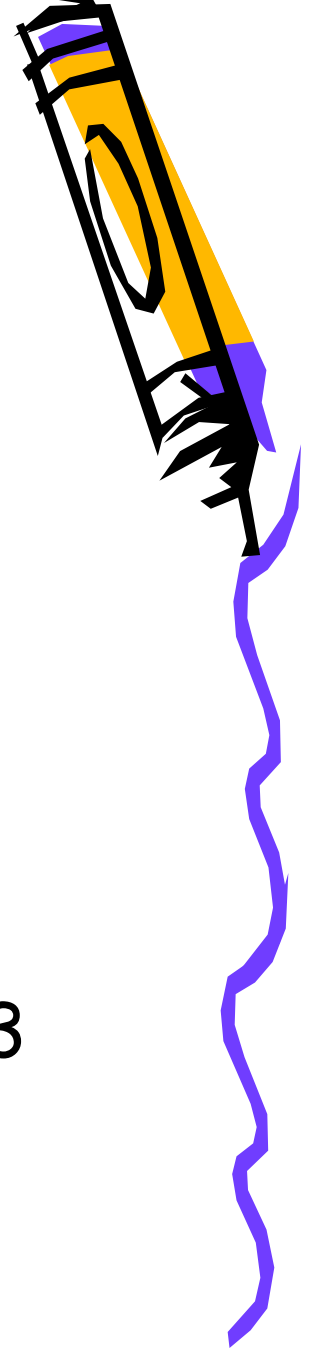
Stanje je definisano **promenljivama** X_i sa **vrednostima** iz **domena** D_i

Test cilja je skup **ograničenja** koje specificiraju dozvoljene kombinacije vrednosti za podskupove promenljivih

Rešenje je **potpuna** i **konzistentna** dodela



Vrste ograničenja



Unarna ograničenja uključuju samo jednu promenljivu

npr. $SA \neq green$

Binarna ograničenja uključuju parove promenljivih

npr. $SA \neq WA$

Ograničenja **višeg reda** uključuju više od 3 promenljive

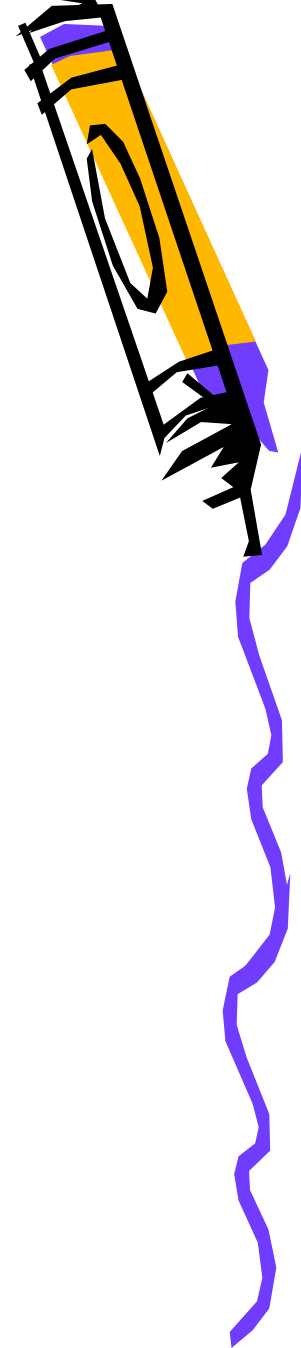
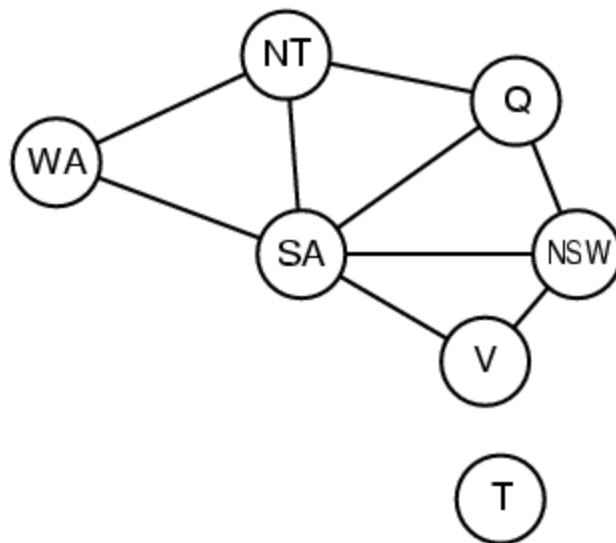
npr. kriptoaritmetički problem



Graf ograničenja

Binarni CSP: svako ograničenje povezuje najviše dve promenljive

Graf ograničenja: čvorovi su promenljive, grane (lukovi) su ograničenja



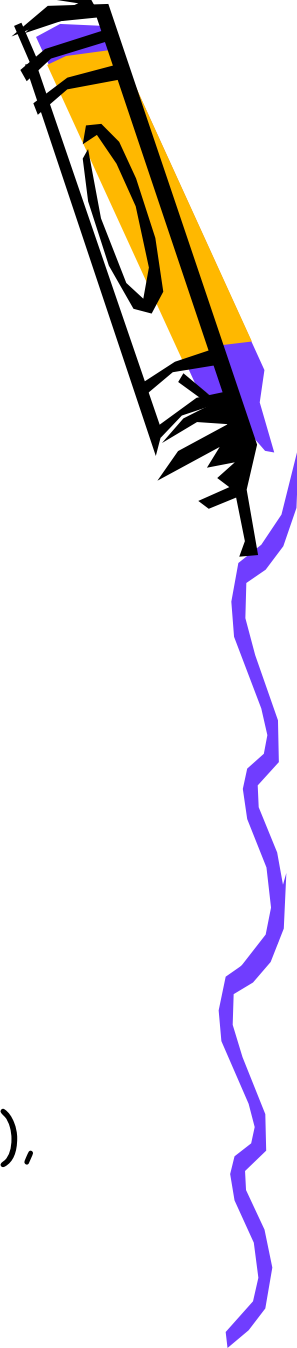
Primer: bojenje mape



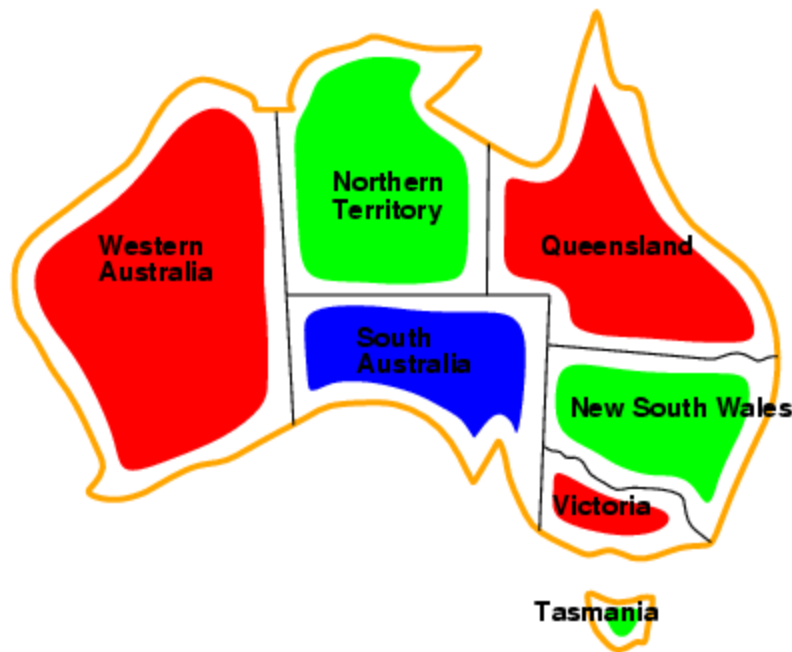
Promenljive: WA, NT, Q, NSW, V, SA, T

Domen: {red, green, blue}

Ograničenja: susedni regioni moraju imati različite boje
npr. $WA \neq NT$, ili $(WA, NT) \notin \{(red, green), (red, blue), (green, red), (green, blue), (blue, red), (blue, green)\}$



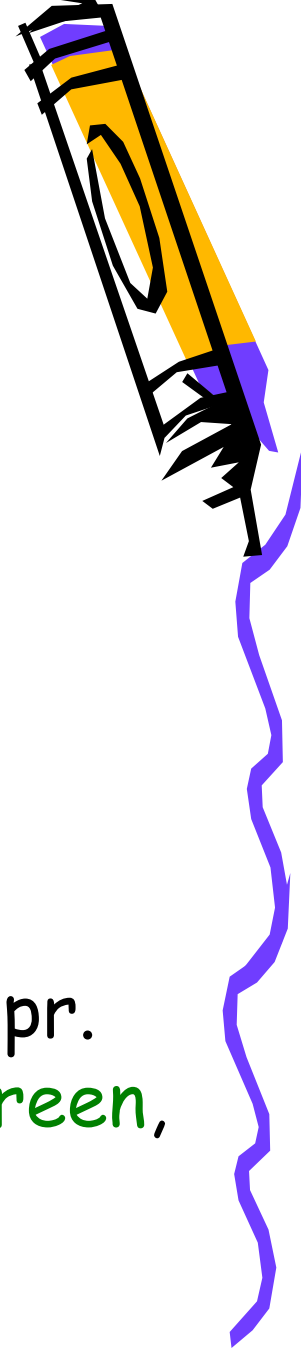
Primer: bojenje mape



Rešenja su *potpune* i *konzistentne* dodele, npr.

WA = red, NT = green, Q = red, NSW = green,

V = red, SA = blue, T = green



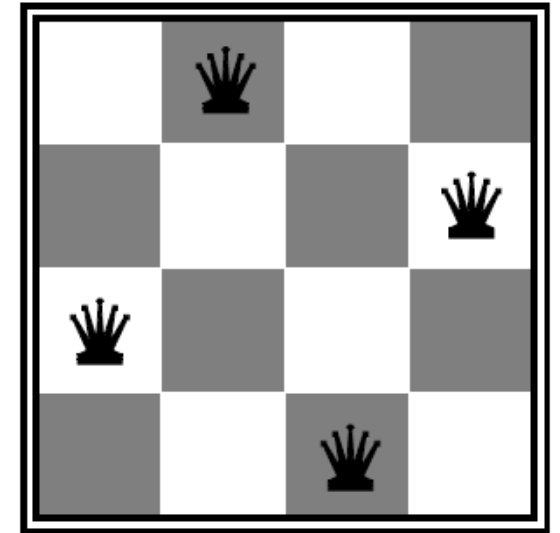
N-Queens

Promenljive: Q_i

Domen: $\{1, \dots, N\}$

Ograničenja:

$\forall i, j$ ne_napadaju_se (Q_i, Q_j)



Primer: Cryptarithmic

Promenljive: T, W, O, F, U, R

X_1, X_2

Domen: $\{0, 1, 2, \dots, 9\}$

Ograničenja:

Alldiff(T, W, O, F, U, R)

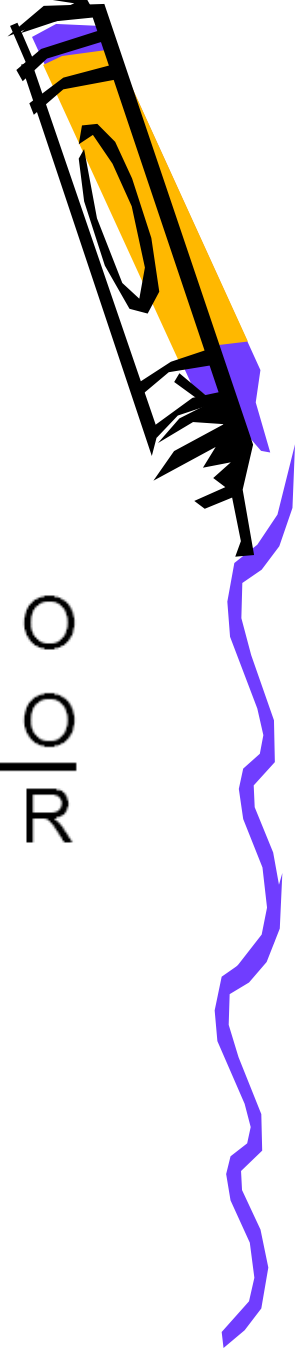
$$O + O = R + 10 * X_1$$

$$W + W + X_1 = U + 10 * X_2$$

$$T + T + X_2 = O + 10 * F$$

$$T \neq 0, F \neq 0$$

$$\begin{array}{r} \text{T W O} \\ + \text{T W O} \\ \hline \text{F O U R} \end{array}$$



Primer: Sudoku

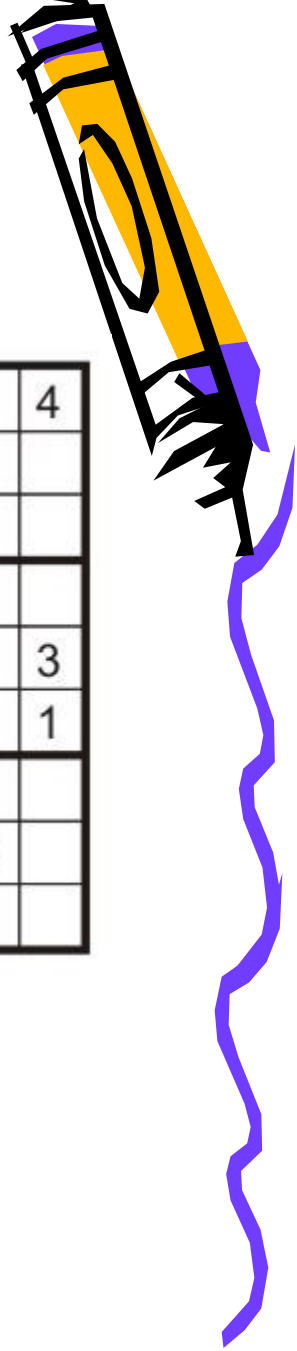
Promenljive: X_{ij}

Domen: $\{1, 2, \dots, 9\}$

Ograničenja:

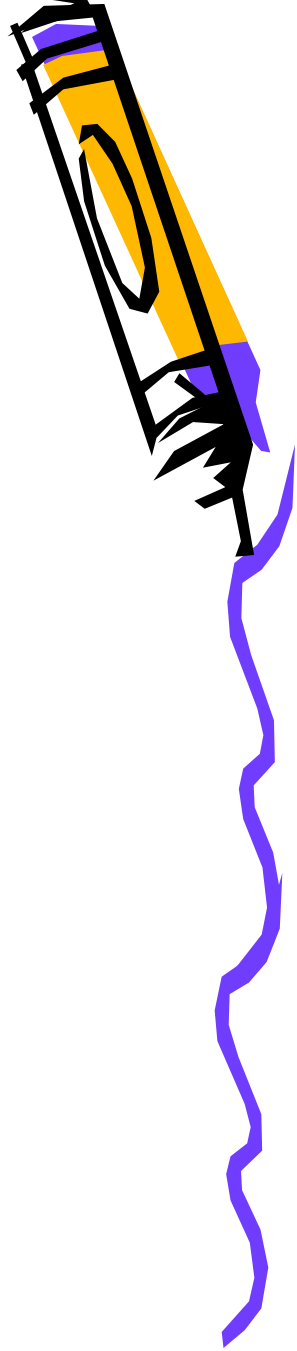
Alldiff(X_{ij} u istoj jedinici)

				8			4
	8	4		1	6		
			5			1	
1	3	8			9		
6	8		X_{ij}		4		3
	2			9	5		1
	7			2			
			7	8		2	6
2			3				



Realni CSPs

- Podela predmeta
npr. ko predaje koji predmet
- Raspored časova
- Raspored saobraćaja
- Planiranje raspored prostorija
- ...



Formulacija standardne pretrage

Stanja:

Vrednosti koje su dodeljene do sada

Inicijalno stanje:

Prazna dodela { }

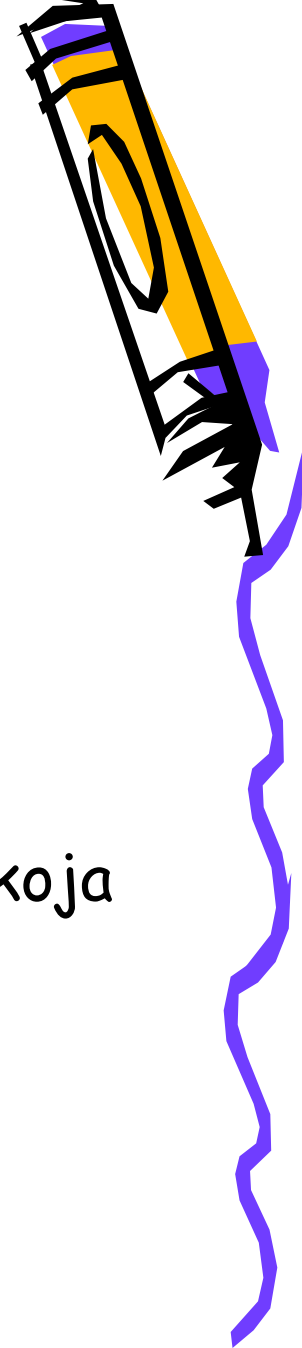
Sledbenik funkcija:

Izabrati bilo koju promenljivu kojoj nije dodeljena vrednost i dodeliti joj vrednost koja ne krši ni jedno ograničenje.

Neuspeh ako nije moguće izvršiti dodelu.

Test cilja:

Dodela je kompletna i zadovoljava sva ograničenja.



Backtracking search

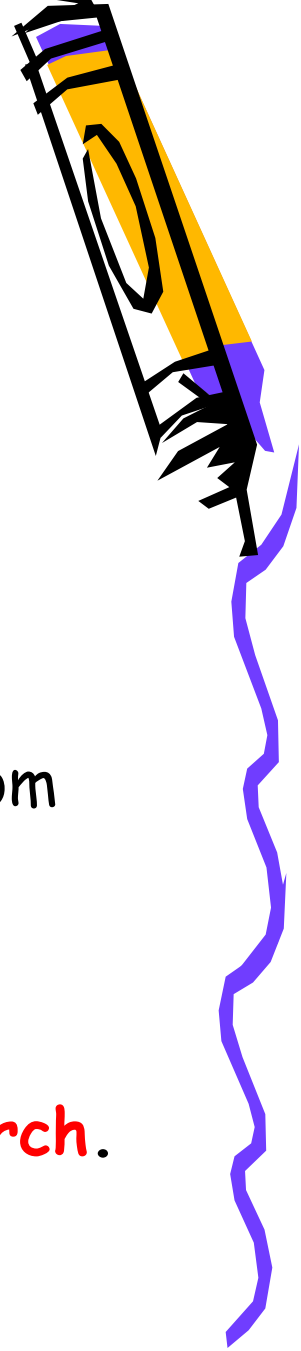
U CSP-u, dodela vrednosti promenljivama je **komutativna**

npr. $[WA = \text{red then } NT = \text{green}]$ je isto kao i $[NT = \text{green then } WA = \text{red}]$

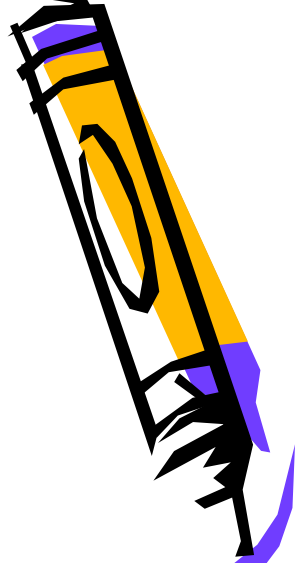
Razmatramo dodele jednoj promenljivoj po svakom nivou (fiksiramo redosled dodela).

Ukupno ima m^n listova.

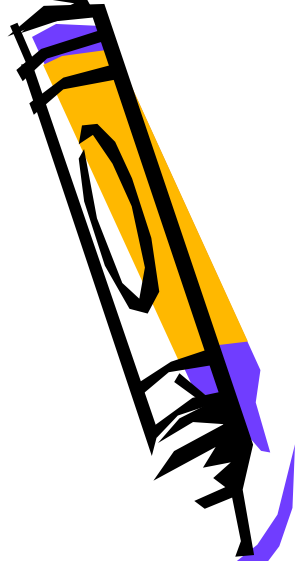
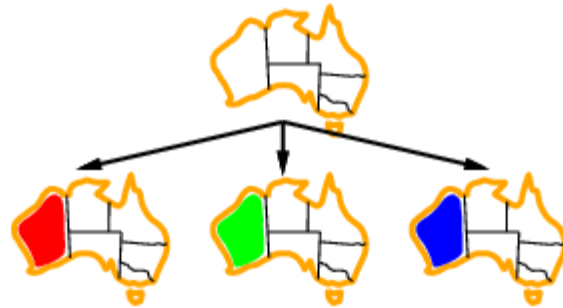
Depth-first search za CSP sa dodelom vrednosti jednoj promenljivoj se zove **backtracking search**.



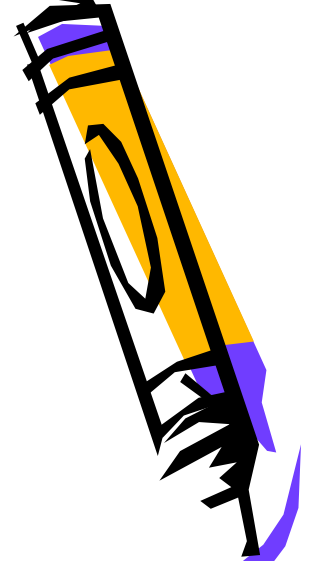
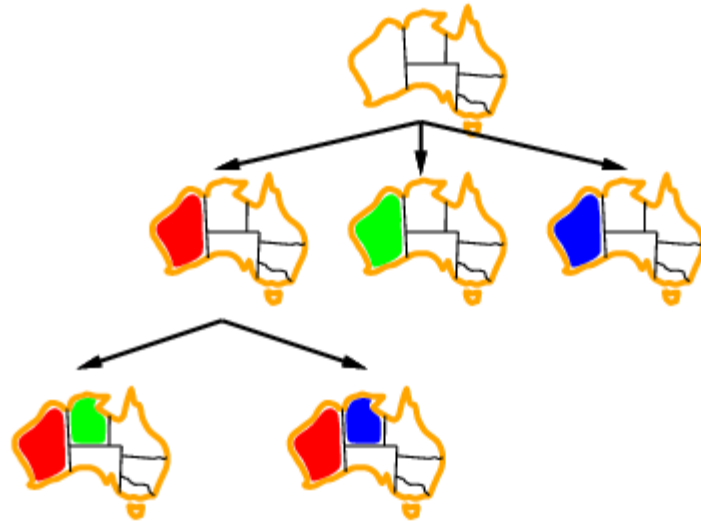
Primer



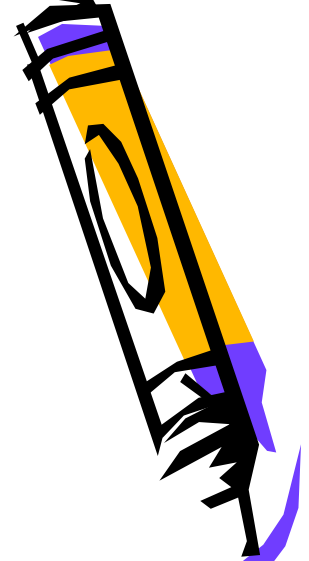
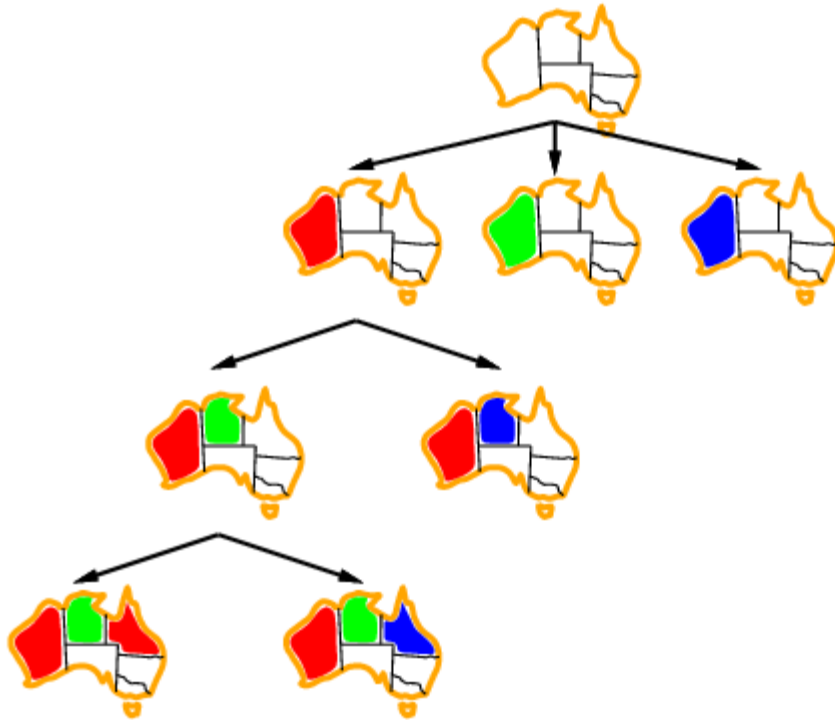
Primer



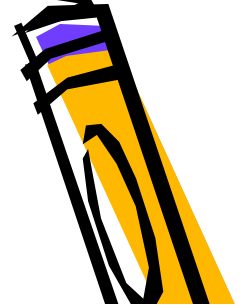
Primer



Primer



Backtracking search algoritam



```
function RECURSIVE-BACKTRACKING(assignment, csp)
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp)
    if value is consistent with assignment given CONSTRAINTS[csp]
      add {var = value} to assignment
      result ← RECURSIVE-BACKTRACKING(assignment, csp)
      if result ≠ failure then return result
      remove {var = value} from assignment
  return failure
```

Poboljšanje efikasnosti:

- Kojoj sledećoj promenljivoj dodeliti vrednost?
- U kom redosledu treba probati njene vrednosti?
- Možemo li otkriti neizbežan neuspeh unapred?

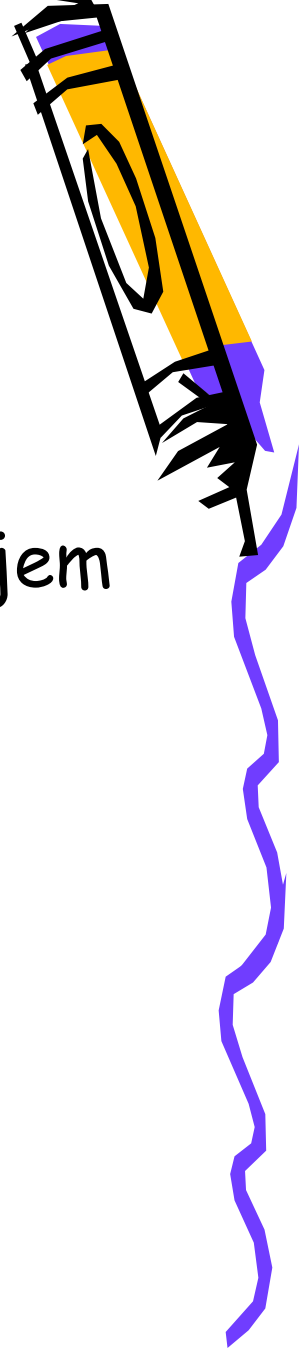
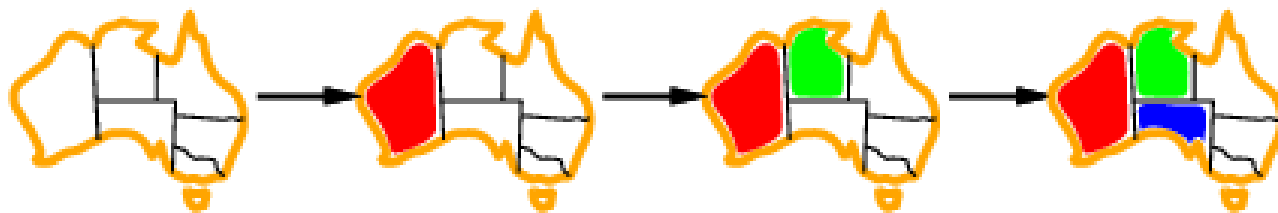


Kojoj sledećoj promenljivoj dodeliti vrednost?

Najograničenijoj promenljivoj:

Izabrati promenljivu sa najmanjim brojem legalnih vrednosti

tzv. **minimum remaining values (MRV)** heuristika

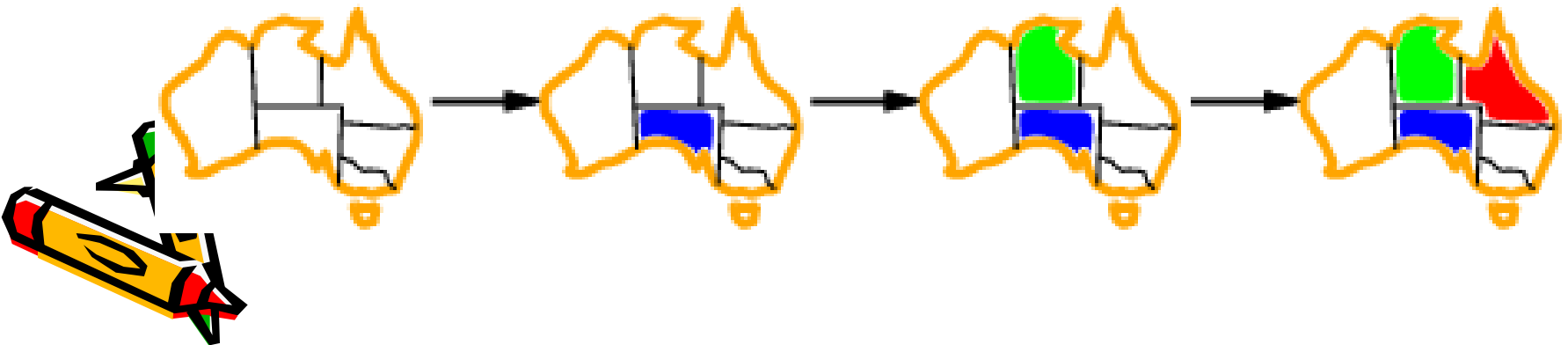


Kojoj sledećoj promenljivoj dodeliti vrednost?

Promenljivoj koja najviše ograničava (degree heuristic):

Izabrati promenljivu koja nameće najviše ograničenja ostalim promenljivama.

Rešava problem izjednačenih vrednosti MRV heuristike.



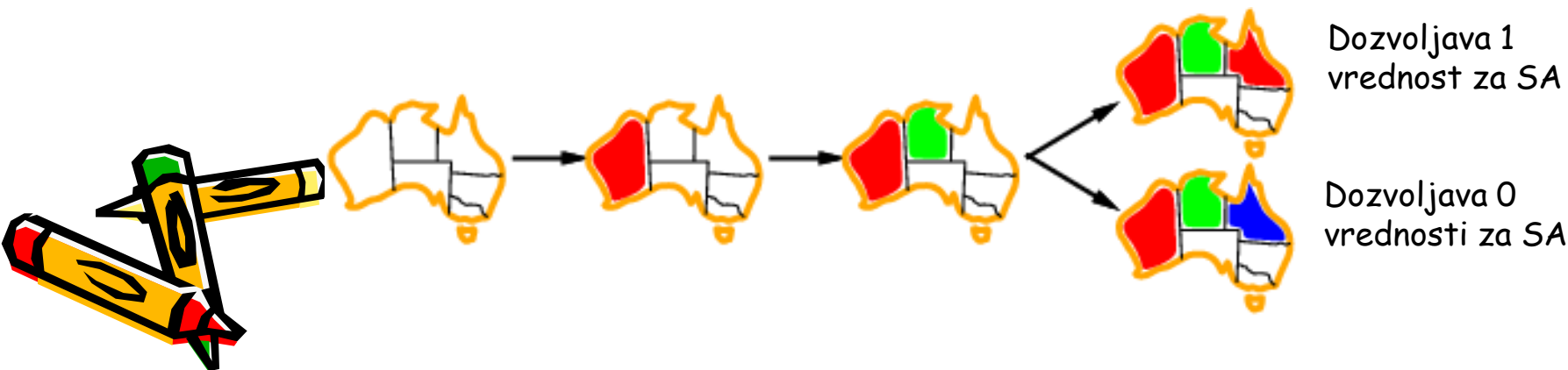
Najmanje ograničavajuća vrednost?

Ako je izabrana promenljiva, u kom redosledu treba probati njene vrednosti?

Izabrati vrednost koja najmanje ograničava:

ona koja odbacuje najmanje vrednosti preostalih promenljivih

Koju vrednost treba izabrati za Q?

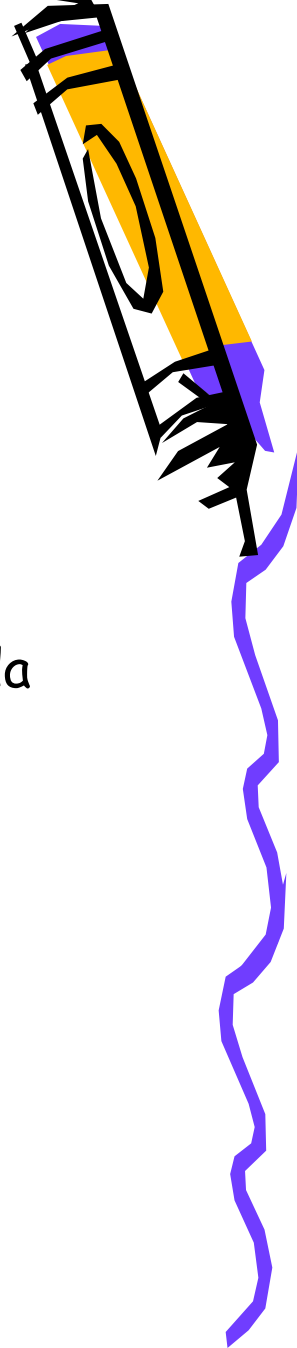


Forward checking

Ideja:

Pratiti preostale dozvoljene vrednosti za promenljive kojima ništa nije dodeljeno.

Prekinuti pretragu kada nijedna promenljiva ne može da dobije vrednost.

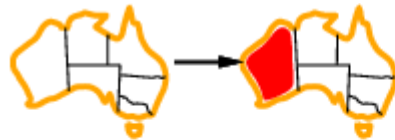


Forward checking

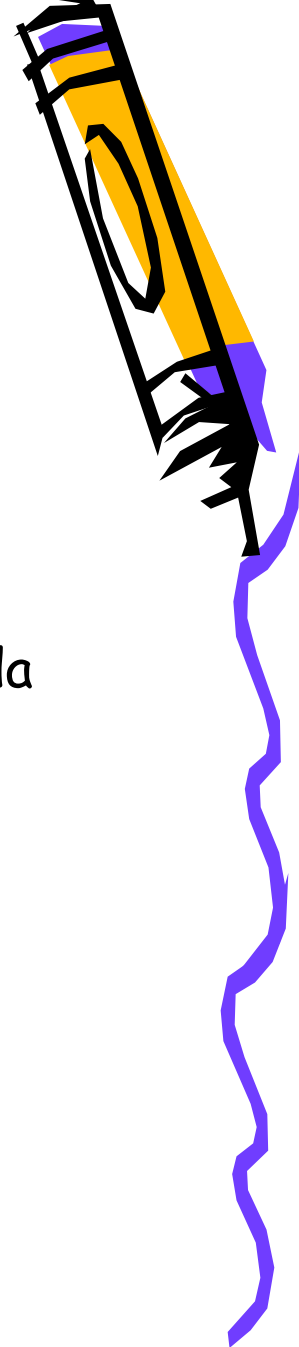
Ideja:

Pratiti preostale dozvoljene vrednosti za promenljive kojima ništa nije dodeljeno.

Prekinuti pretragu kada nijedna promenljiva ne može da dobije vrednost.



WA	NT	Q	NSW	V	SA	T
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■



Forward checking

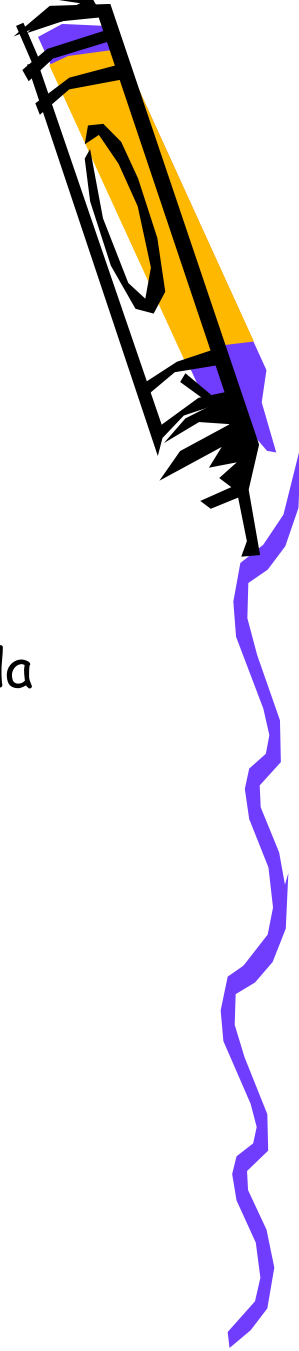
Ideja:

Pratiti preostale dozvoljene vrednosti za promenljive kojima ništa nije dodeljeno.

Prekinuti pretragu kada nijedna promenljiva ne može da dobije vrednost.



WA	NT	Q	NSW	V	SA	T
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■



Forward checking

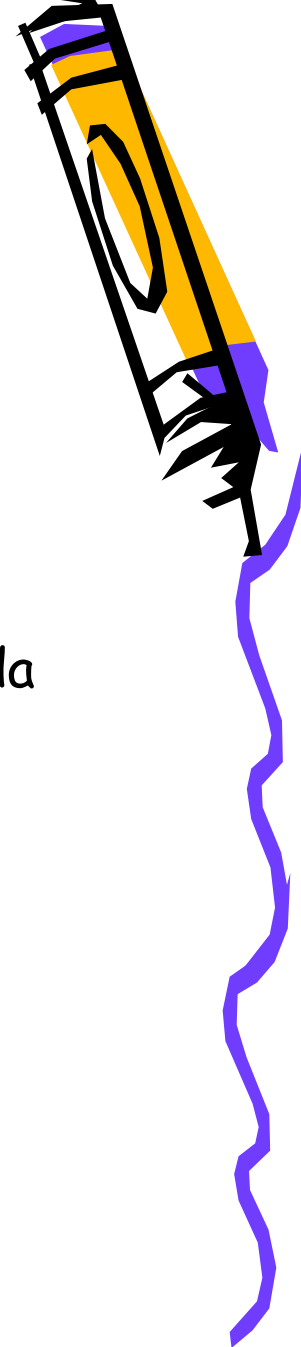
Ideja:

Pratiti preostale dozvoljene vrednosti za promenljive kojima ništa nije dodeljeno.

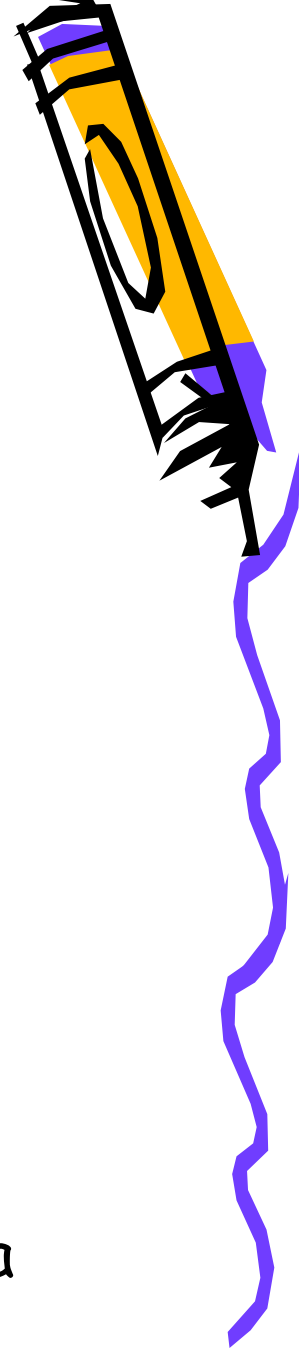
Prekinuti pretragu kada nijedna promenljiva ne može da dobije vrednost.



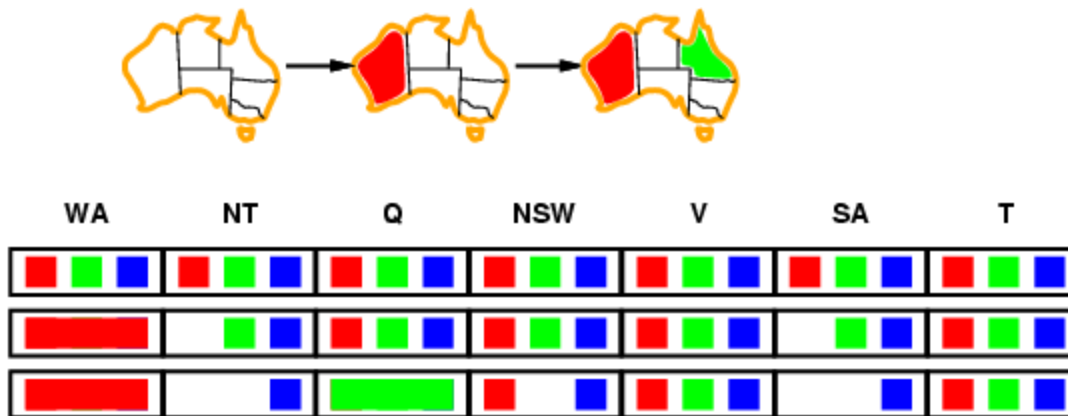
WA	NT	Q	NSW	V	SA	T
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■
■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■	■ ■ ■



Propagacija ograničenja



Forward checking propagira informacije od dodeljenih ka nedodeljenim promenljivama, ali ne omogućava rano otkrivanje svih grešaka:



NT i SA ne mogu biti plave!



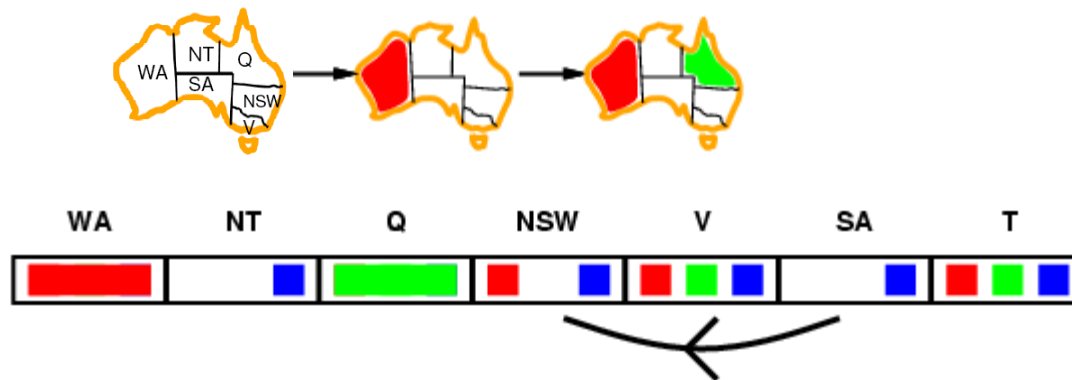
Propagiranje ograničenja u više navrata nameće lokalno ograničenje.

Konzistencija luka (arc consistency)

Najjednostavnija forma propagacije ograničenja omogućava da je svaki par promenljivih **konzistentan**:

$X \rightarrow Y$ je konzistentan akko za **svaku** vrednost X postoji **neka** dozvoljena vrednost za Y

Prilikom provere $X \rightarrow Y$, odbaciti sve vrednosti X za koje ne postoji dozvoljena vrednost za Y

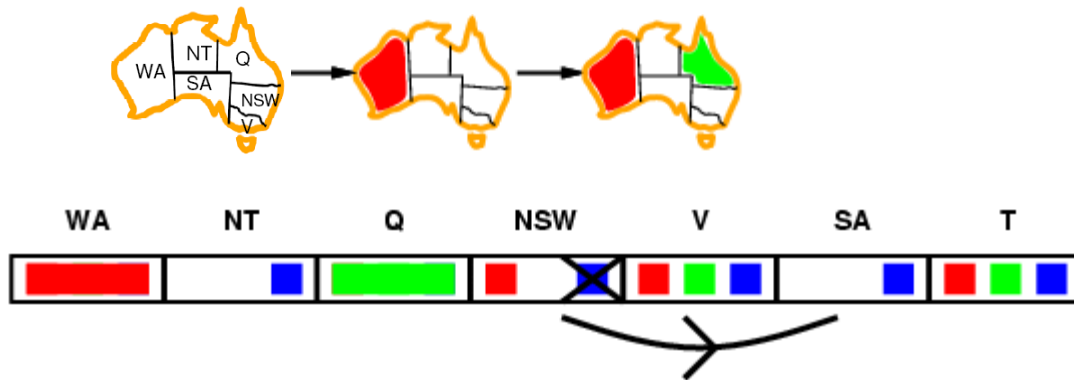


Konzistencija luka (arc consistency)

Najjednostavnija forma propagacije ograničenja omogućava da je svaki par promenljivih **konzistentan**:

$X \rightarrow Y$ je konzistentan akko za **svaku** vrednost X postoji **neka** dozvoljena vrednost za Y

Prilikom provere $X \rightarrow Y$, odbaciti sve vrednosti X za koje ne postoji dozvoljena vrednost za Y



Ako X izgubi vrednost, sve parove $Z \rightarrow X$ treba preispitati

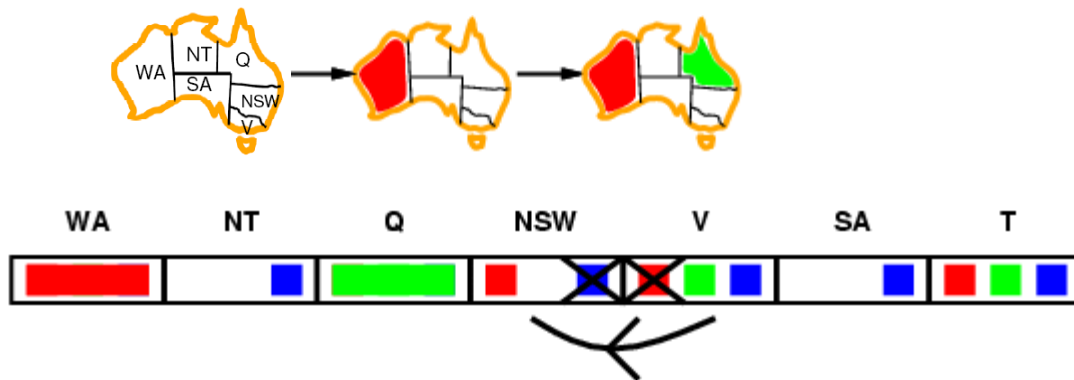


Konzistencija luka (arc consistency)

Najjednostavnija forma propagacije ograničenja omogućava da je svaki par promenljivih **konzistentan**:

$X \rightarrow Y$ je konzistentan akko za **svaku** vrednost X postoji **neka** dozvoljena vrednost za Y .

Prilikom provere $X \rightarrow Y$, odbaciti sve vrednosti X za koje ne postoji dozvoljena vrednost za Y .



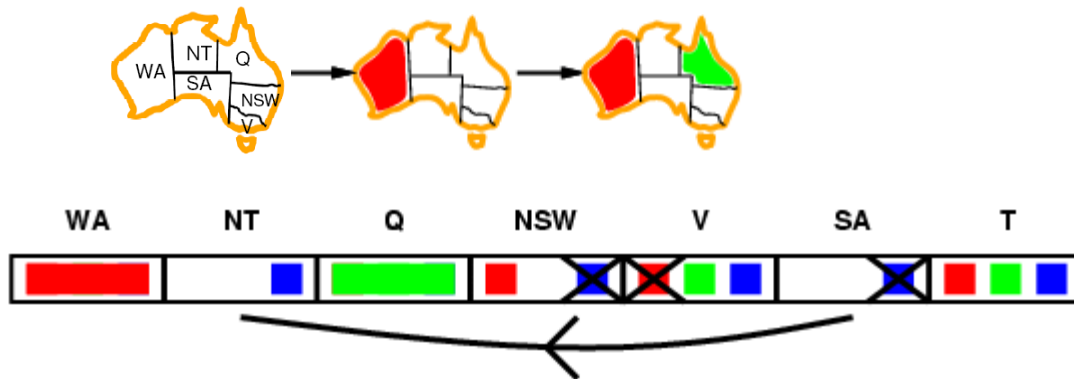
Ako X izgubi vrednost, sve parove $Z \rightarrow X$ treba preispitati

Konzistencija luka (arc consistency)

Najjednostavnija forma propagacije ograničenja omogućava da je svaki par promenljivih **konzistentan**:

$X \rightarrow Y$ je konzistentan akko za **svaku** vrednost X postoji **neka** dozvoljena vrednost za Y .

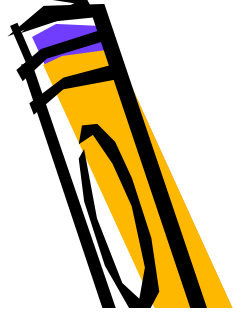
Prilikom provere $X \rightarrow Y$, odbaciti sve vrednosti X za koje ne postoji dozvoljena vrednost za Y .



Ako X izgubi vrednost, sve parove $Z \rightarrow X$ treba preispitati.

Brže se otkrije greška. Može se pokrenuti nakon svake dodele.

Arc consistency algorithm AC-3



function **AC-3**(*csp*) returns the CSP, possibly with reduced domains

inputs: *csp*, a binary CSP with variables $\{X_1, X_2, \dots, X_n\}$

local variables: *queue*, a queue of arcs, initially all the arcs in *csp*

while *queue* is not empty

$(X_i, X_j) \leftarrow \text{REMOVE-FIRST}(\textit{queue})$

 if **REMOVE-INCONSISTENT-VALUES**(X_i, X_j) then

 for each X_k in **NEIGHBORS**[X_i] do

 add (X_k, X_i) to *queue*

function **REMOVE-INCONSISTENT-VALUES**(X_i, X_j) returns true iff succeeds

removed \leftarrow false

 for each x in **DOMAIN**[X_i]

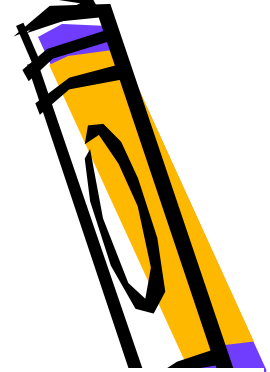
 if no value y in **DOMAIN**[X_j] allows (x, y) to satisfy the constraint $X_i \leftrightarrow X_j$

 then delete x from **DOMAIN**[X_i]; *removed* \leftarrow true



 return *removed*



Backtracking search sa zaključivanjem



```
function RECURSIVE-BACKTRACKING(assignment, csp)
  if assignment is complete then return assignment
  var ← SELECT-UNASSIGNED-VARIABLE(VARIABLES[csp], assignment, csp)
  for each value in ORDER-DOMAIN-VALUES(var, assignment, csp)
    if value is consistent with assignment given CONSTRAINTS[csp]
      add {var = value} to assignment
      result ← RECURSIVE-BACKTRACKING(assignment, csp)
      if result ≠ failure then return result
      remove {var = value} from assignment
  return failure
```



forward checking i/ili
arc consistency ide ovde

Lokalna pretraga za CSP

Hill-climbing, recimo, zahteva da je dodela kompletna tj. da su svim promenljivama dodeljene vrednosti

Da bi se primenio CSP:

Dozvoliti stanja sa neazdovoljenim ograničenjima

Pokušati popravku stanja preraspodelom dodeljenih vrednosti

Izbor promenljive:

Nasumičan izbor bilo koje promenljive koja ima konflikt

Izbor vrednosti upotrebom **min-broj-konflikta** heuristike:

Izabrati vrednost koja krši najmanji broj ograničenja

Npr. hill-climb sa $h(n)$ = ukupan broj prekršenih ograničenja



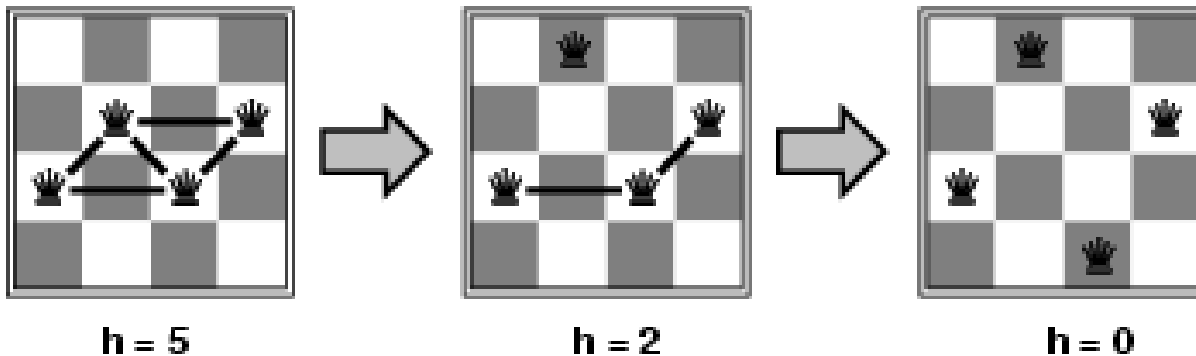
Primer: 4-Queens

Stanje: 4 kraljice u 4 kolone ($4^4 = 256$ stanja)

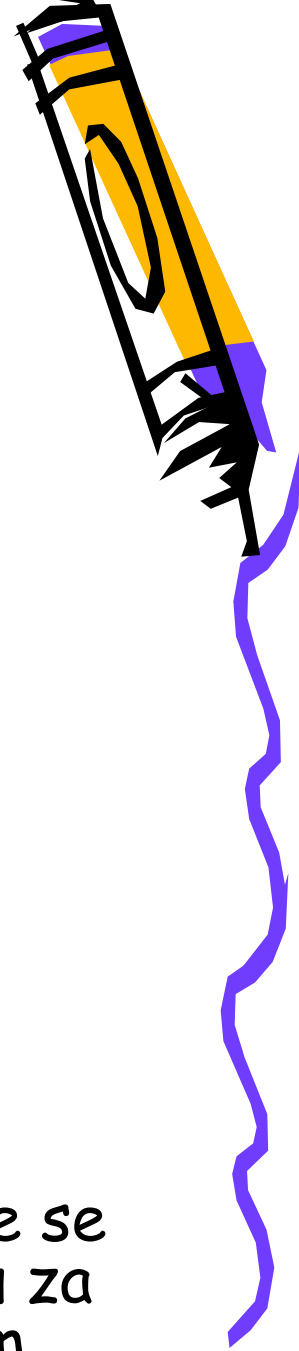
Akcije: pomeri kraljicu u koloni

Test cilja: nema napada

Evaluacija: $h(n) =$ broj napada



Sa nasumičnim početnim stanjem, može se rešiti n -queens u skoro konstantnom vremenu za bilo koju vrednost n sa velikom verovatnoćom (npr. $n = 10,000,000$)



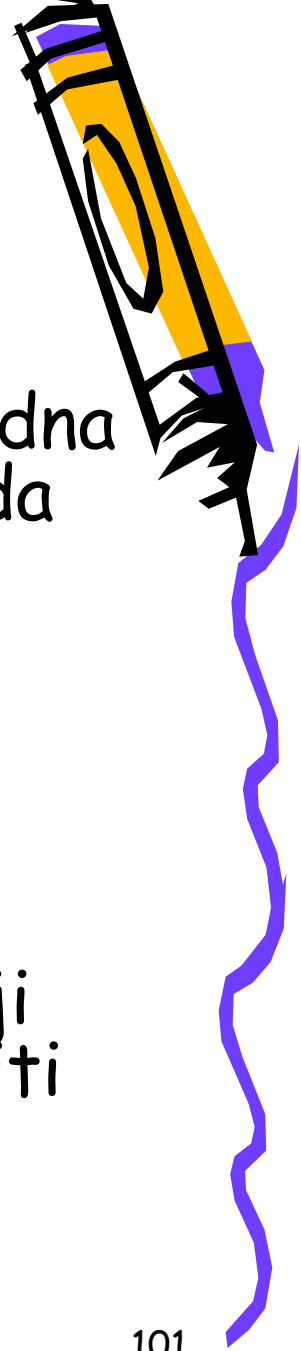
Zadatak 3:

Kriptoaritmetički problem

- U sledećoj operaciji sabiranja dva dekadna broja cifre su zamenjene slovima tako da različitim slovima odgovaraju različite cifre.

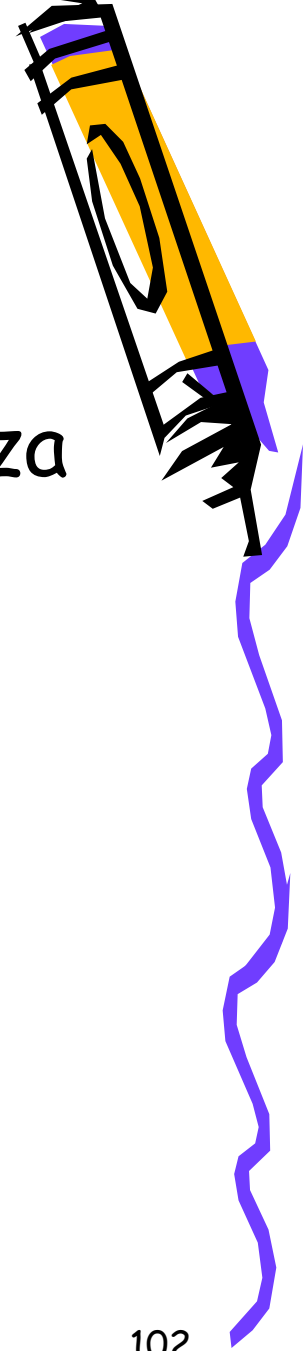
$$\begin{array}{r} \text{SEND} \\ +\text{MORE} \\ \hline \text{MONEY} \end{array}$$

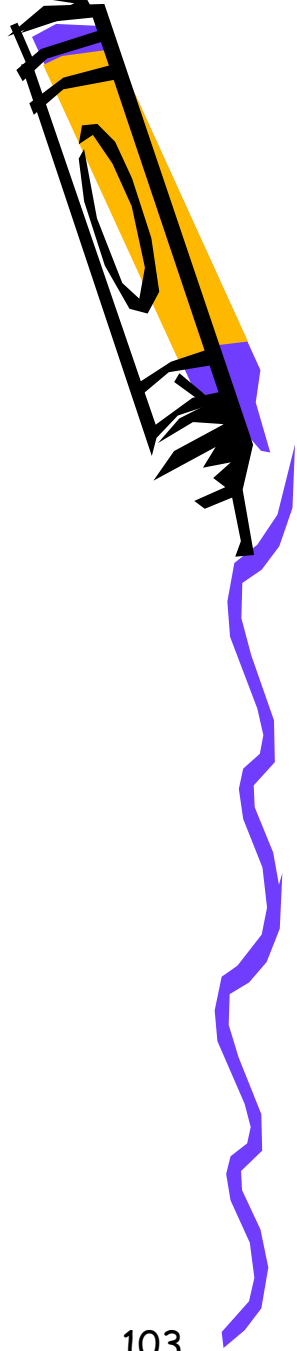
- Metodom proste relaksacije pronaći koji broj predstavlja svaka od cifara. Usvojiti da je $E = 5$



Metoda relaksacije

- Napraviti liste mogućih vrednosti za sve slobodne promenljive (one čije vrednosti želimo da odredimo)





M {1, 2, 3, 4, 5, 6, 7, 8, 9}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

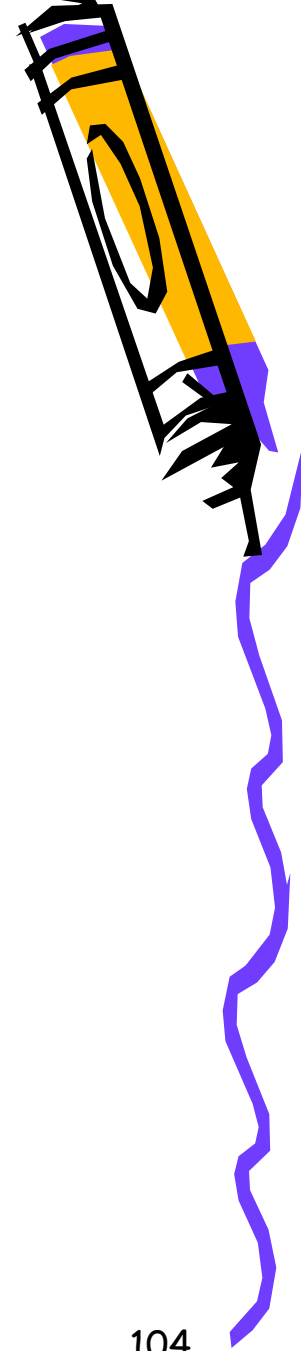
C100 {0, 1}

C10 {0, 1}

C1 {0, 1}



- Obeležiti aktivne promenljive



Aktivne: M S O C100 N C10 R C1 D Y

Neaktivne: E

M {1, 2, 3, 4, 5, 6, 7, 8, 9}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

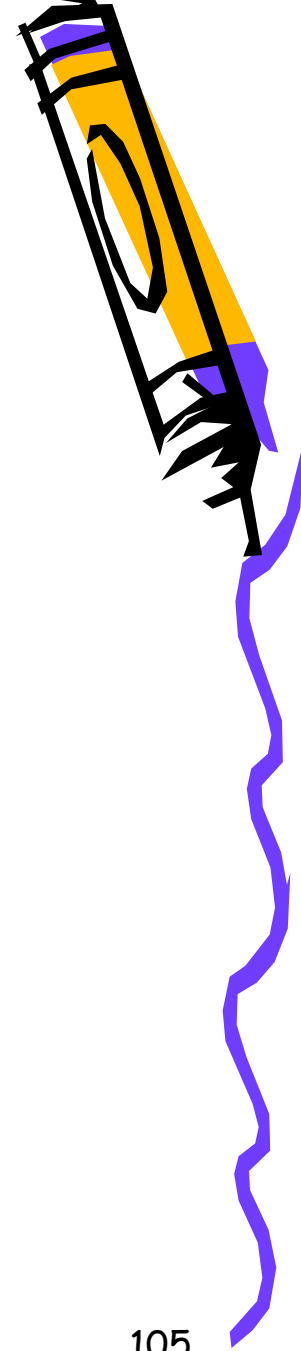
D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

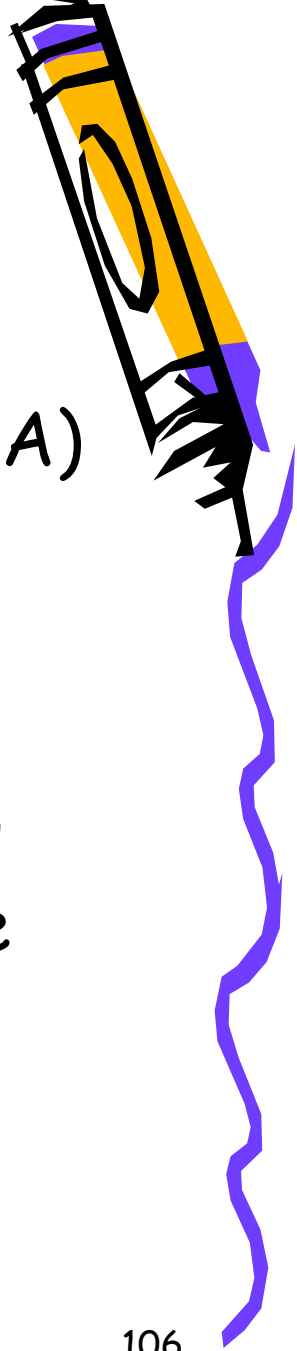
C100 {0, 1}

C10 {0, 1}

C1 {0, 1}



- Izabrati neku od aktivnih promenljivih (A)
- Može se pri izboru koristiti neka heuristika
- Usvojićemo heuristiku da od aktivnih promenljivih za razmatranje biramo onu koja predstavlja cifru ili prenos najveće težine



Aktivne: M S O C100 N C10 R C1 D Y

Neaktivne: E

M {1, 2, 3, 4, 5, 6, 7, 8, 9}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

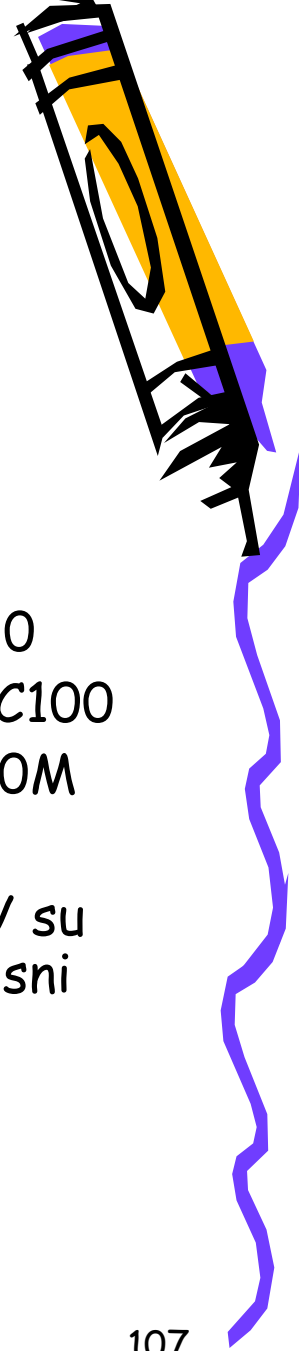
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

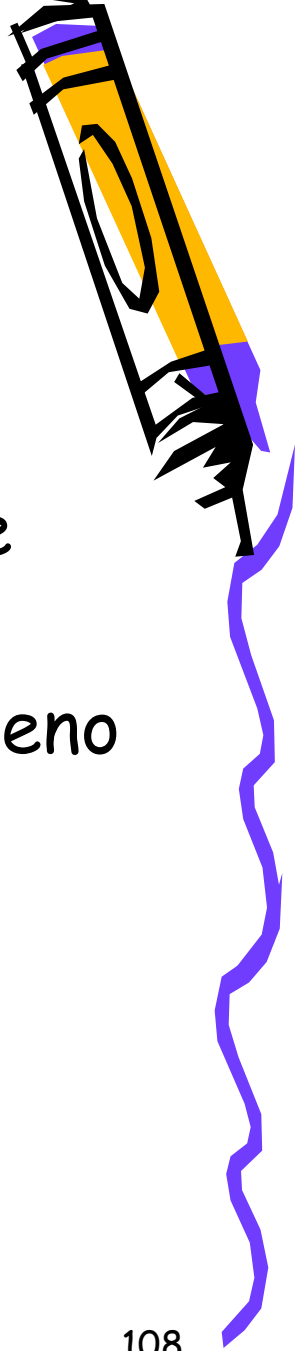
$$O3: S + O + C10 = N + 10C100$$

$$O4: S + M + C100 = O + 10M$$

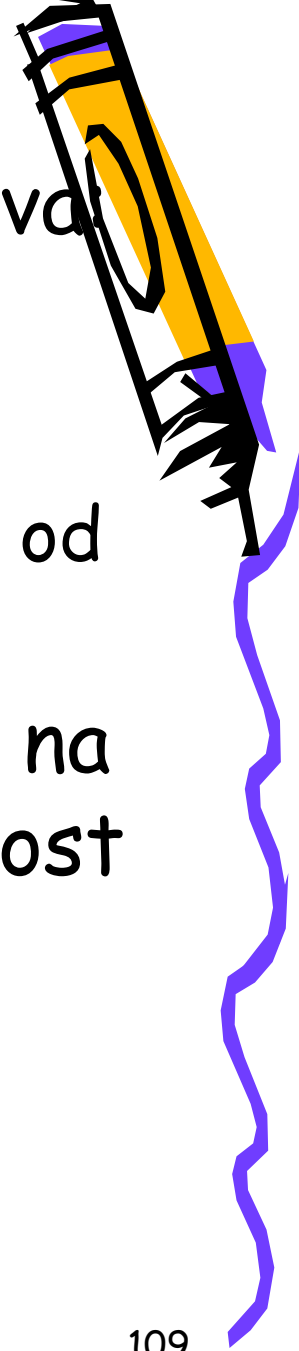
O5: M, S, O, N, 5, R, D, Y su
svi medjusobno nezavisni



- Za svaku moguću vrednost V za A razmatrati svako od ograničenja u kome figuriše A
- Proveriti da li je ograničenje C zadovoljeno ako A ima vrednost V
- Ako se C ne može zadovoljiti, izbaciti V
- Obeležiti promenljivu A kao neaktivnu



- Obeležiti kao aktivne promenljive koje zadovoljavaju sledeća tri uslova:
 - Pomenute su u ograničenjima sa A
 - Prethodno su bile neaktivne
 - U skupu mogućih vrednosti imaju više od jedne vrednosti
- Ako je skup vrednosti za A sveden na jednu vrednosti, zameniti tu vrednost u svim ograničenjima



Aktivne: M S O C100 N C10 R C1 D Y

Neaktivne: E

M > 1 ??

M {1, 2, 3, 4, 5, 6, 7, 8, 9}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

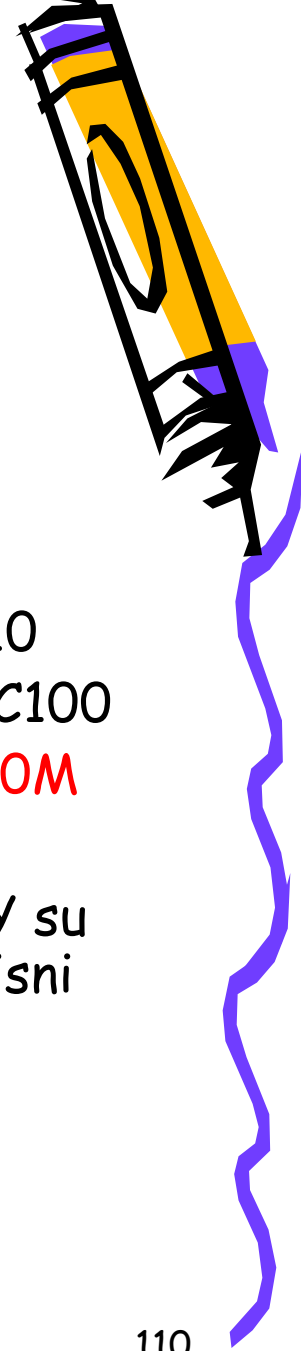
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + O + C10 = N + 10C100$$

$$O4: S + M + C100 = O + 10M$$

O5: M, S, O, N, 5, R, D, Y su
svi medjusobno nezavisni



Aktivne: S O C100 N C10 R C1 D Y

Neaktivne: E M

M {1}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

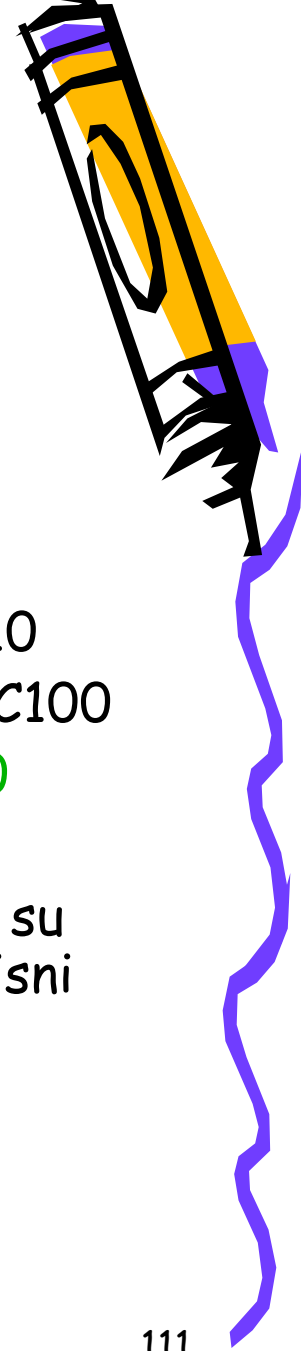
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + O + C10 = N + 10C100$$

$$O4: S + 1 + C100 = O + 10$$

O5: 1, S, O, N, 5, R, D, Y su
svi medjusobno nezavisni



Aktivne: S O C100 N C10 R C1 D Y

Neaktivne: E M

M {1}

S {1, 2, 3, 4, 5, 6, 7, 8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Za koje S je O4 zadovoljen ??

Ograničenja

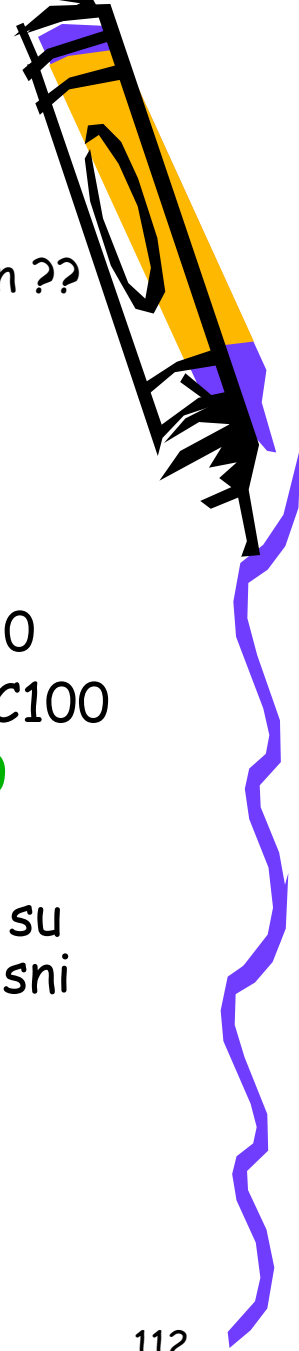
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + O + C10 = N + 10C100$$

$$O4: S + 1 + C100 = O + 10$$

O5: 1, S, O, N, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: O C100 N C10 R C1 D Y

Neaktivne: E M S

M {1}

S {8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

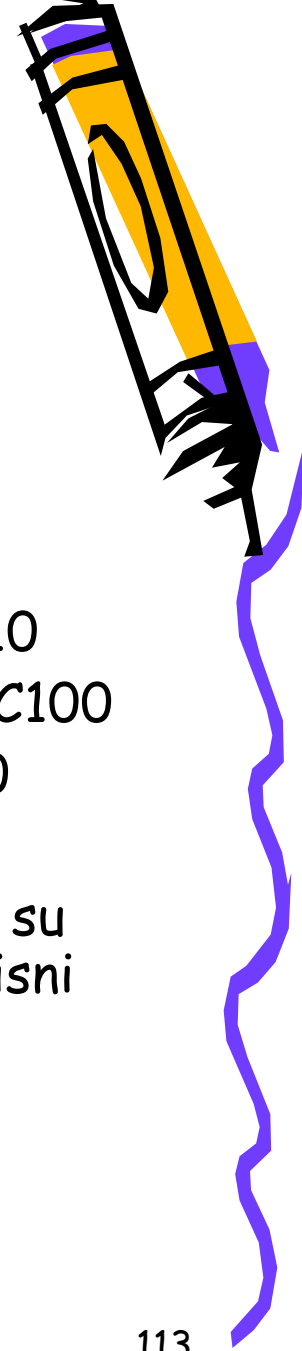
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + O + C10 = N + 10C100$$

$$O4: S + 1 + C100 = O + 10$$

O5: 1, S, O, N, 5, R, D, Y su
svi medjusobno nezavisni



Aktivne: **O** C100 N C10 R C1 D Y

Neaktivne: E M S

M {1}

S {8, 9}

O {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

O: 1 ili 0 ??

Ograničenja

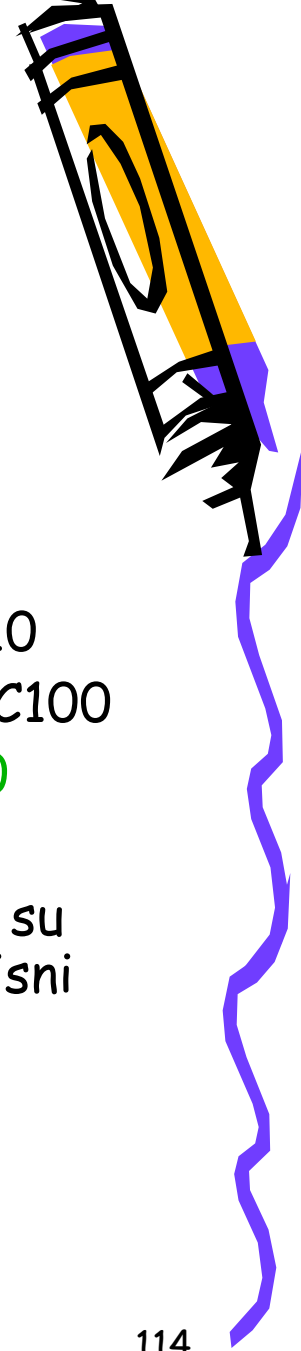
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + O + C10 = N + 10C100$$

$$O4: S + 1 + C100 = O + 10$$

O5: 1, S, O, N, 5, R, D, Y su
svi medjusobno nezavisni



Aktivne: S C100 N C10 R C1 D Y

Neaktivne: E M O

M {1}

S {8, 9}

O {0}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + C10 = N + 10C100$$

$$O4: S + 1 + C100 = 10$$

O5: 1, S, O, N, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: C100 N C10 R C1 D Y

Neaktivne: E M O S

M {1}

S {8, 9}

O {0}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0, 1}

C10 {0, 1}

C1 {0, 1}

Ograničenja

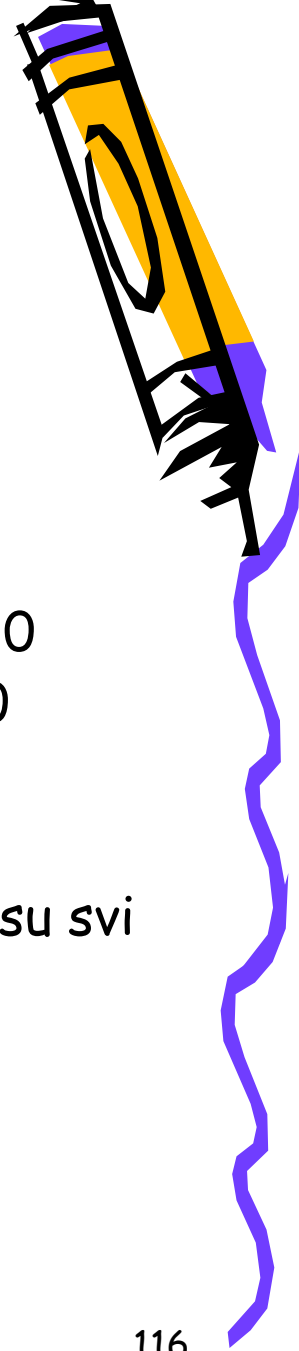
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + C10 = N + 10C100$$

$$O4: S + 1 + C100 = 10$$

O5: 1, S, O, N, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: S N C10 R C1 D Y

Neaktivne: E M O C100

M {1}

S {8, 9}

O {0}

E {5}

N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {0, 1}

C1 {0, 1}

Ograničenja

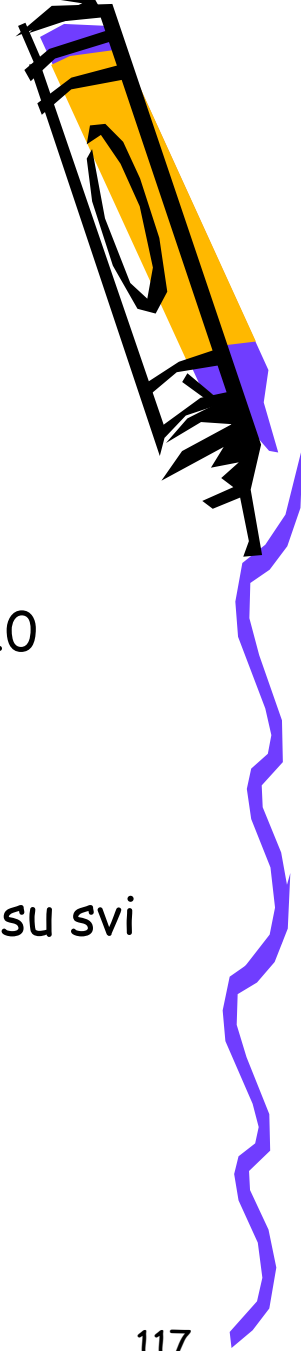
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + C10 = N$$

$$O4: S + 1 = 10$$

O5: 1, S, O, N, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: N C10 R C1 D Y
Neaktivne: E M O C100 S

M {1}
S {9}
O {0}
E {5}
N {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
C100 {0}
C10 {0, 1}
C1 {0, 1}

Ograničenja

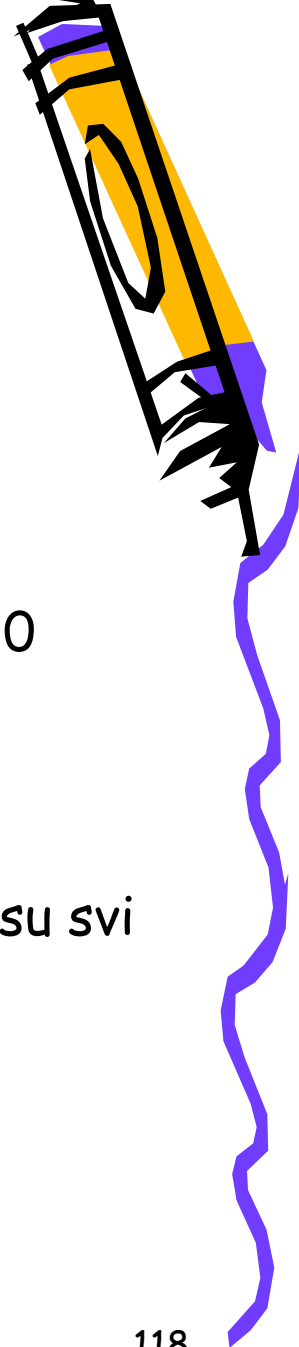
$$O1: D + 5 = Y + 10C1$$

$$O2: N + R + C1 = 5 + 10C10$$

$$O3: 5 + C10 = N$$

$$O4: 10 = 10$$

O5: 1, S, O, N, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: C10 R C1 D Y

Neaktivne: E M O C100 S N

M {1}

S {9}

O {0}

E {5}

N {6}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {0, 1}

C1 {0, 1}

Ograničenja

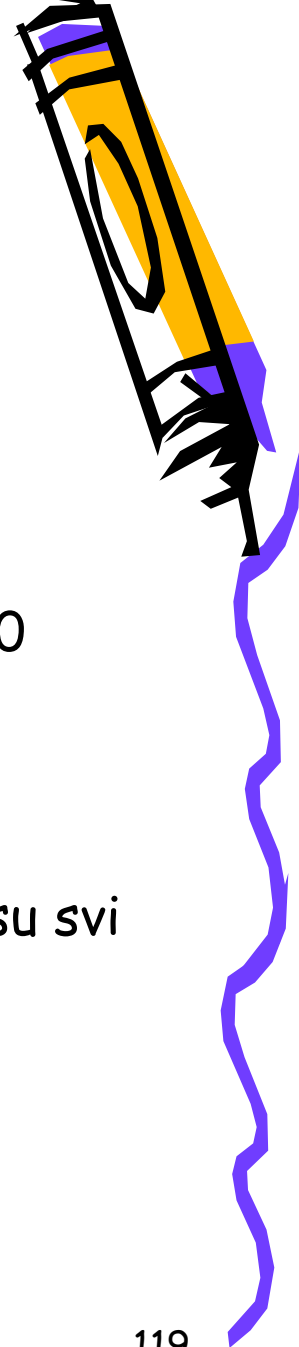
$$O1: D + 5 = Y + 10C1$$

$$O2: 6 + R + C1 = 5 + 10C10$$

$$O3: 5 + C10 = 6$$

$$O4: 10 = 10$$

O5: 1, 9, 0, 6, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: R C1 D Y

Neaktivne: E M O C100 S N C10

M {1}

S {9}

O {0}

E {5}

N {6}

R {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {1}

C1 {0, 1}

Ograničenja

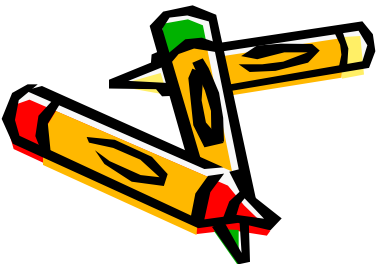
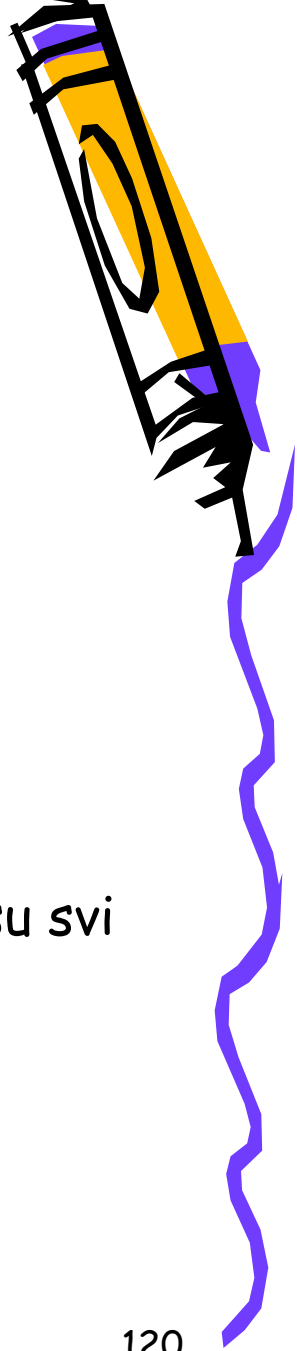
$$O1: D + 5 = Y + 10C1$$

$$O2: 6 + R + C1 = 15$$

$$O3: 6 = 6$$

$$O4: 10 = 10$$

O5: 1, 9, 0, 6, 5, R, D, Y su svi medjusobno nezavisni



Aktivne: C1 D Y

Neaktivne: E M O C100 S N C10 R

M {1}

S {9}

O {0}

E {5}

N {6}

R {8}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {1}

C1 {0, 1}

Ograničenja

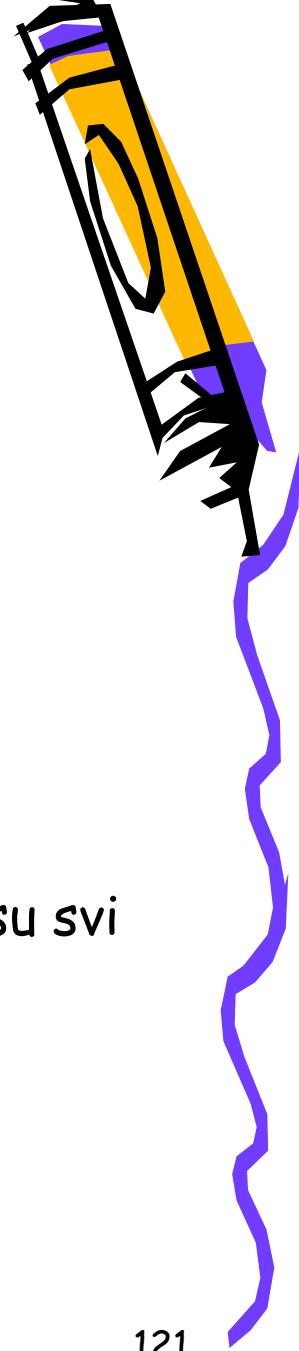
$$O1: D + 5 = Y + 10C1$$

$$O2: 14 + C1 = 15$$

$$O3: 6 = 6$$

$$O4: 10 = 10$$

O5: 1, 9, 0, 6, 5, 8, D, Y su svi medjusobno nezavisni



Aktivne: D Y

Neaktivne: E M O C100 S N C10 R C1

M {1}

S {9}

O {0}

E {5}

N {6}

R {8}

D {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {1}

C1 {1}

Ograničenja

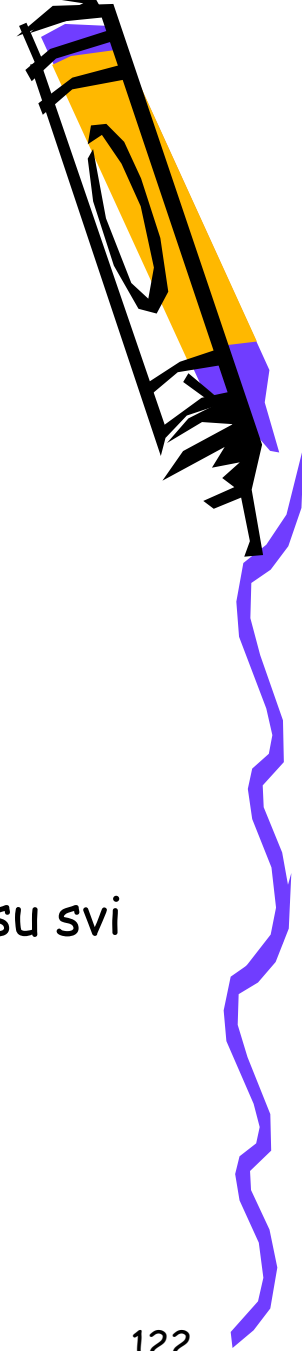
$$O1: D + 5 = Y + 10$$

$$O2: 15 = 15$$

$$O3: 6 = 6$$

$$O4: 10 = 10$$

O5: 1, 9, 0, 6, 5, 8, D, Y su svi medjusobno nezavisni



Aktivne: Y

Neaktivne: E M O C100 S N C10 R C1 D

M {1}

S {9}

O {0}

E {5}

N {6}

R {8}

D {7}

Y {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

C100 {0}

C10 {1}

C1 {1}

Ograničenja

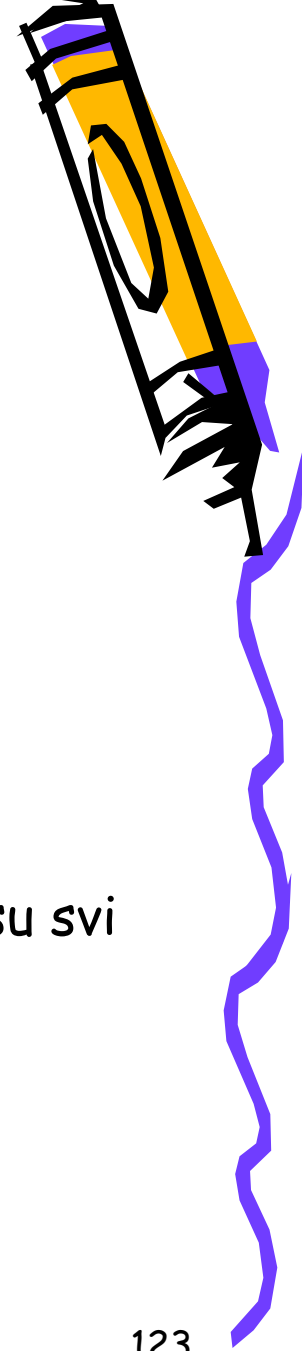
$$O1: 12 = Y + 10$$

$$O2: 15 = 15$$

$$O3: 6 = 6$$

$$O4: 10 = 10$$

O5: 1, 9, 0, 6, 5, 8, 7, Y su svi medjusobno nezavisni



Aktivne:

Neaktivne: E M O C100 S N C10 R C1 D Y

M {1}

S {9}

O {0}

E {5}

N {6}

R {8}

D {7}

Y {2}

C100 {0}

C10 {1}

C1 {1}

Ograničenja

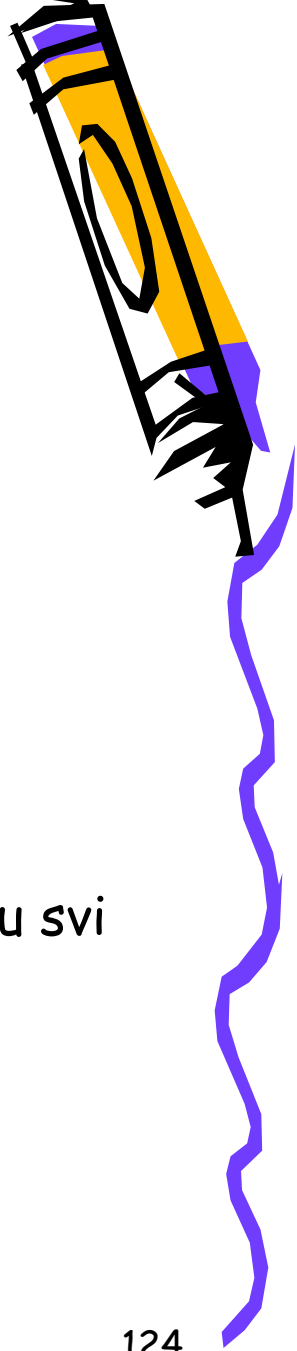
O1: 12 = 12

O2: 15 = 15

O3: 6 = 6

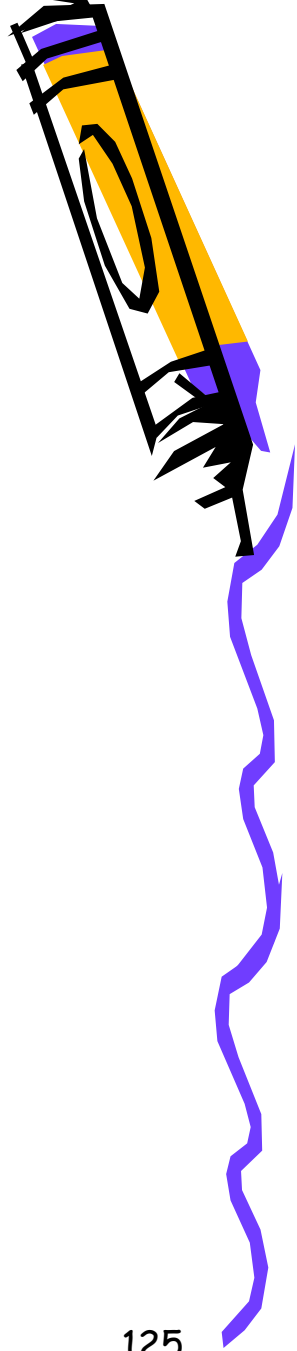
O4: 10 = 10

O5: 1, 9, 0, 6, 5, 8, 7, 2 su svi medjusobno nezavisni



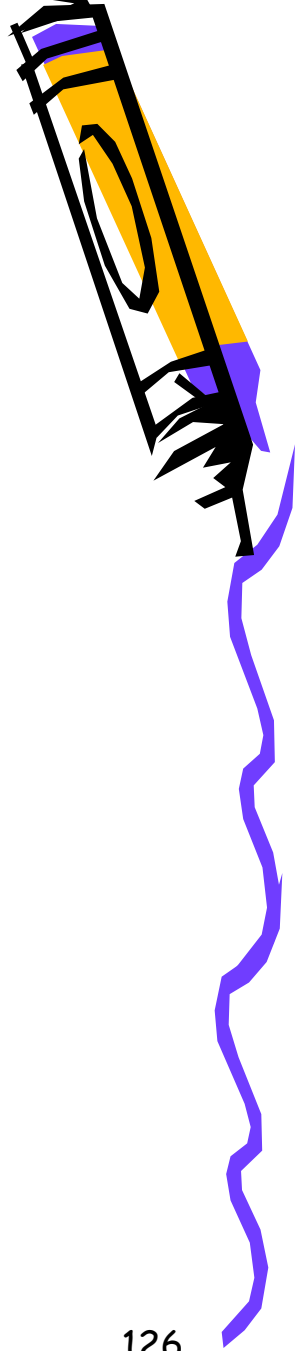
SEND
+MORE

MONEY



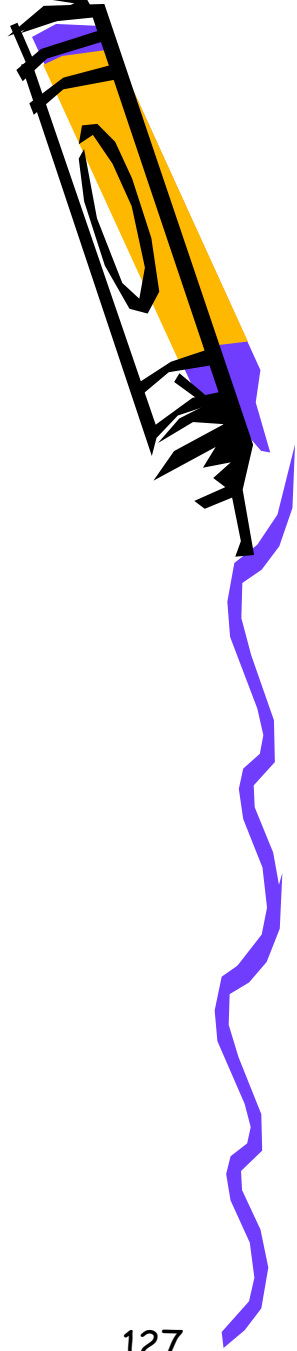
S5ND
+MOR5

MON5Y



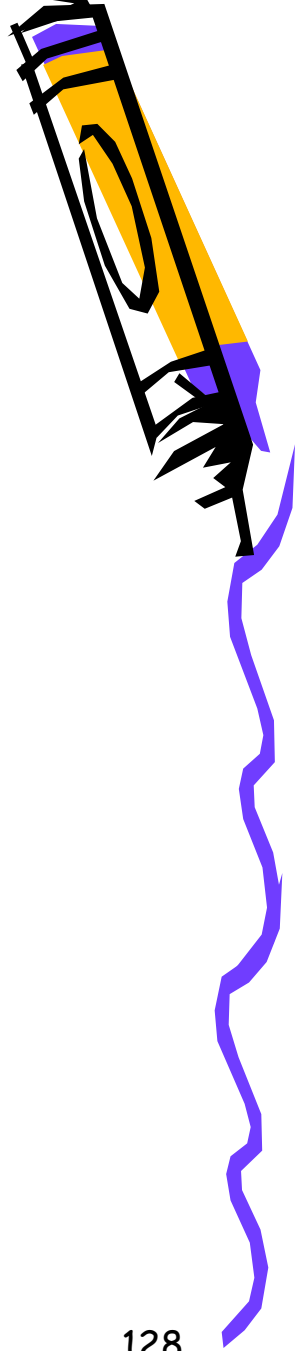
S5ND
+10R5

10N5Y

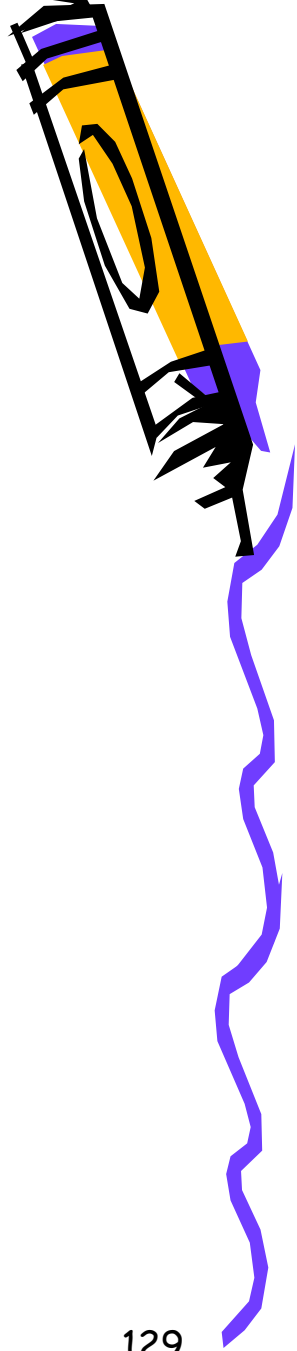


S5ND
+10R5

10N5Y



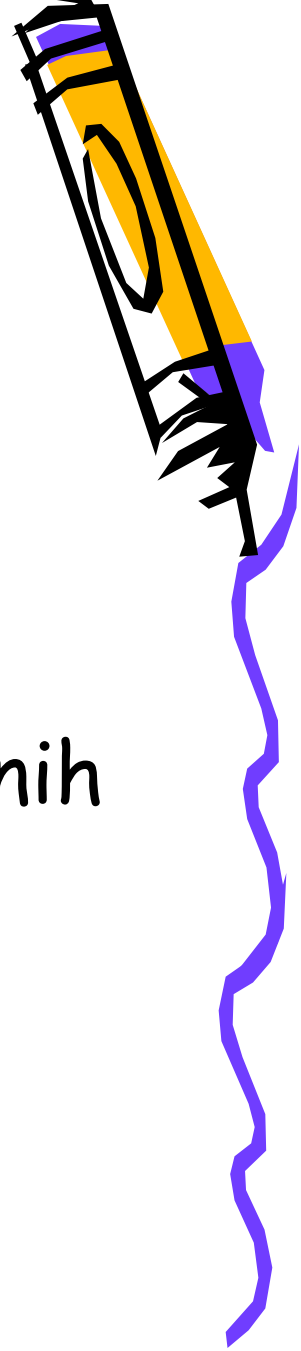
$$\begin{array}{r} 9567 \\ +1085 \\ \hline 10652 \end{array}$$



Zadatak 4: Raspored

Nikola, Dragan, Milan i Vanja imaju časove u isto vreme. Potrebno je dodeliti im učionice.

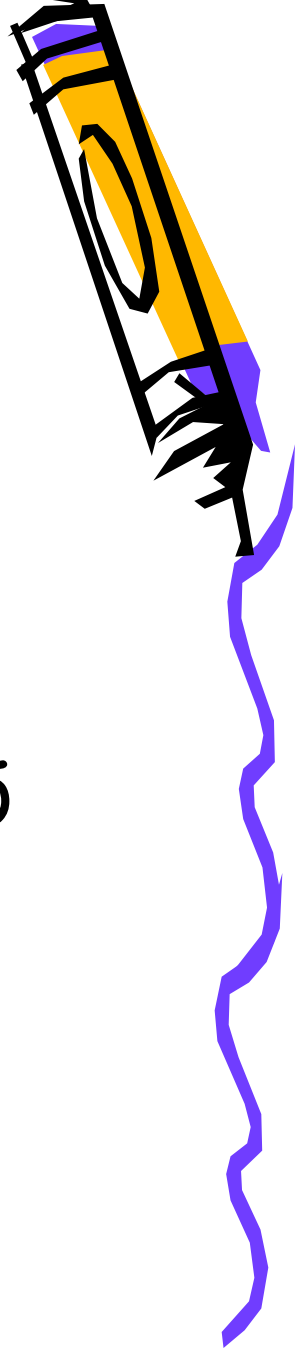
Svaki nastavnik ima svoj skup omiljenih učionica, što zbog njihove opremljenosti, što zbog lokacije.



Raspored

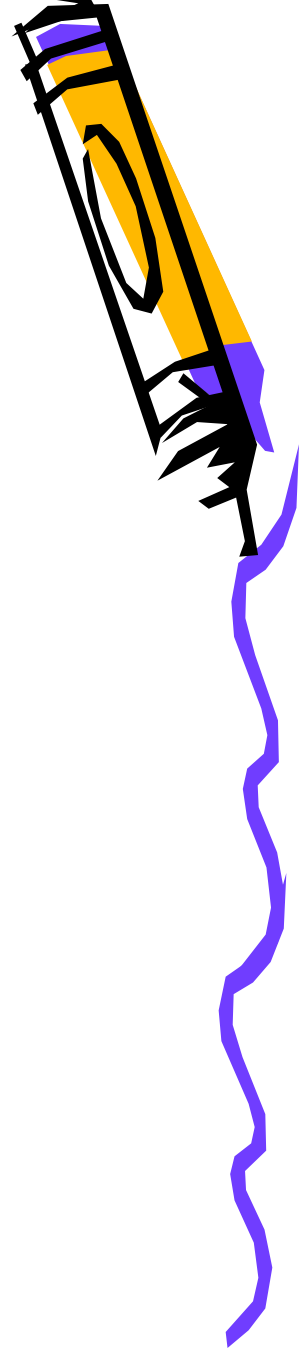
Želje su sledeće:

- Nikola - 61, 57, 59, 309, Paviljon lab25
- Dragan - 61, 57, 59, Paviljon lab25
- Milan - Paviljon lab25 i 61
- Vanja - Paviljon lab25



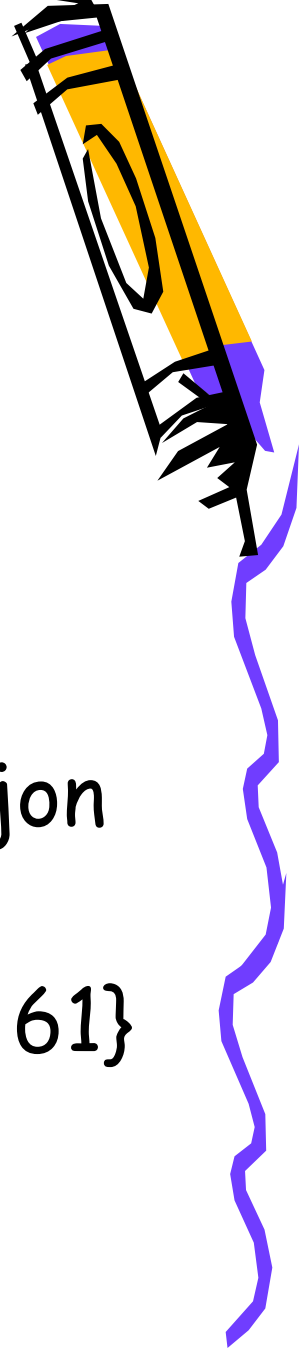
Raspored - rešenje

- Graf je potpuno povezan.
- Ograničenja su binarna.



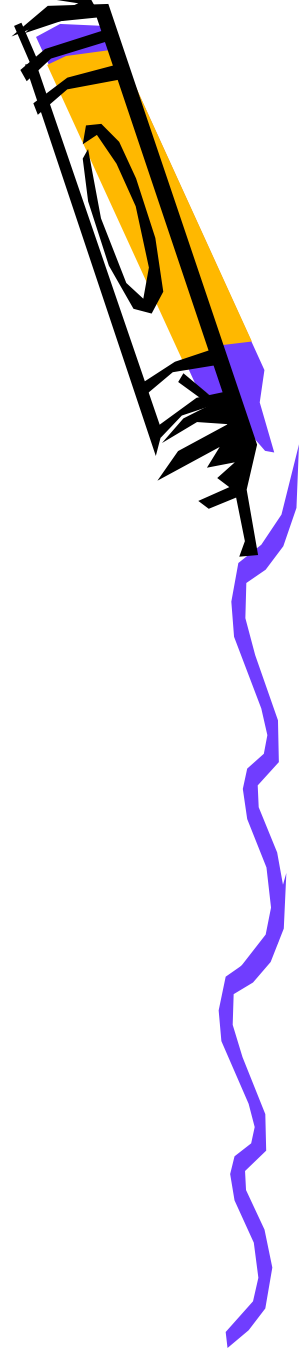
Raspored - rešenje

- Dodela = {}
- Domen(Vanja) = {61, 57, 59, 309, Paviljon lab25}
- Domen(Nikola) = {61, 57, 59, Paviljon lab25}
- Domen(Dragan) = {Paviljon lab25 i 61}
- Domen(Milan) = {Paviljon lab25}



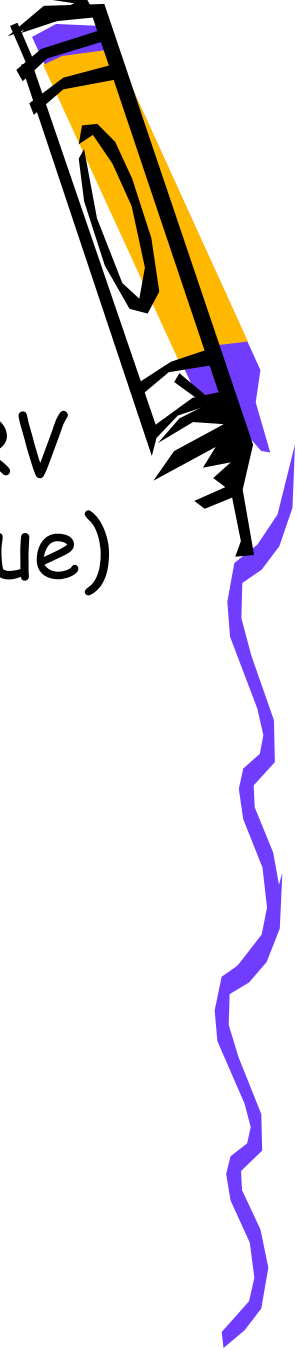
Raspored - rešenje

- Koristićemo:
- backtrack algoritam
- heuristiku
- forward checking i
- arc consistency



Raspored - rešenje

- Biramo promenljivu upotrebom MRV heuristike (minimum remaining value)
- Izbor je Milan
 - samo jedna vrednost, Paviljon lab25



Raspored - rešenje

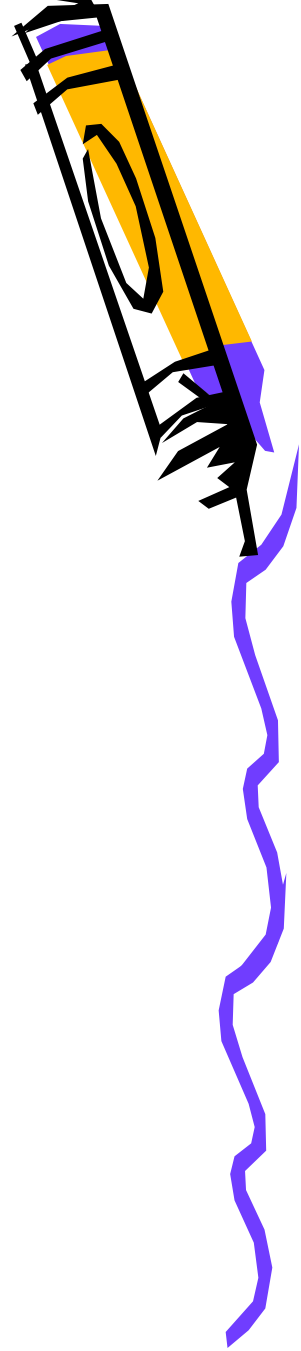
- Zaključivanje (inference) 1:
Forward checking

{{**Milan = Paviljon lab25**}.

Domen(Vanja) = (61, 57, 59, 309)

Domen(Nikola) = (61, 57, 59)

Domen(Dragan) = (61)



Raspored - rešenje

Zaključivanje (inference) 2: Arc consistency

Red:

Nikola->Dragan

Dragan->Nikola

Nikola->Milan

Milan->Nikola

Nikola->Vanja

Vanja->Nikola

Dragan->Milan

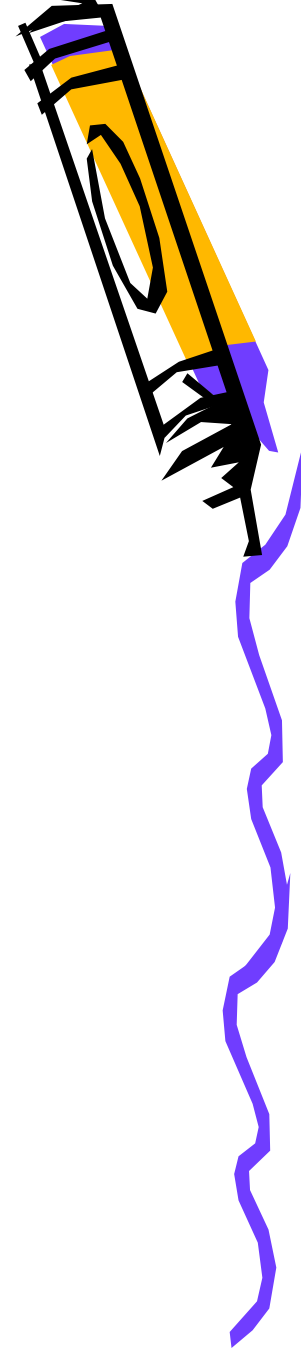
Milan->Dragan

Dragan->Vanja

Vanja->Dragan

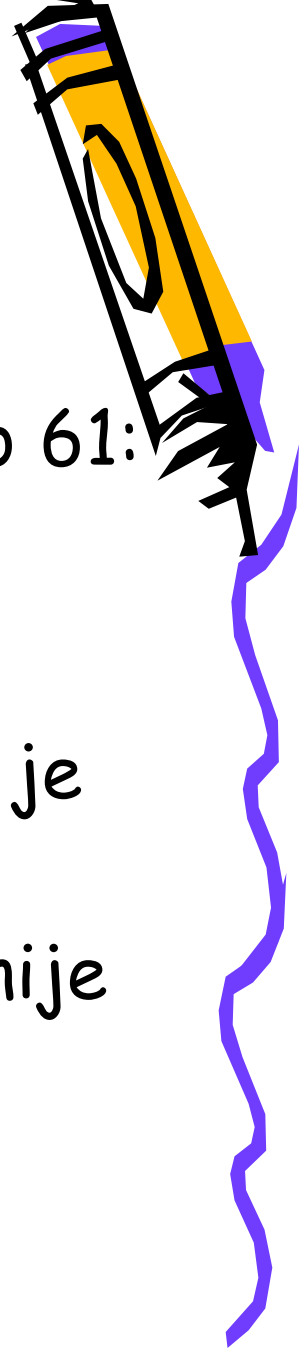
Milan->Vanja

Vanja->Milan



Raspored - rešenje

- Da bi lukovi bili konzistentni, uklanjamo 61: Vanja-→Dragan i Nikola-→Dragan bili konzistentni.
- Ako Vanja dobije učionicu **61**, ne može je dobiti i Dragan
- Kako je to jedini izbor za Dragana, to nije dozvoljena vrednost za Vanju
- Analogno i za Nikolu



Raspored - rešenje

- U red dodajemo:

Dragan->Vanja, Milan->Vanja i Nikola->Vanja

Dragan->Nikola, Milan->Nikola i Vanja->Nikola

- Razlog

- njihov domen je promenjen



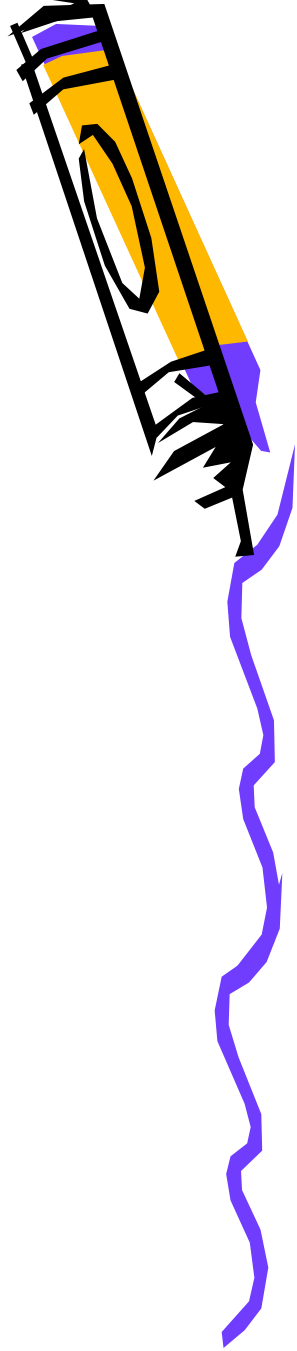
Raspored - rešenje

- Sve promenljive su arc consistent
- Gotov je korak zaključivanja
- Nastavljamo Backtracking
({Milan = Paviljon lab25}.

$$\text{Domen}(\text{Vanja}) = (\text{61}, 57, 59, 309)$$

$$\text{Domen}(\text{Nikola}) = (\text{61}, 57, 59)$$

$$\text{Domen}(\text{Dragan}) = (61)$$



Raspored - rešenje

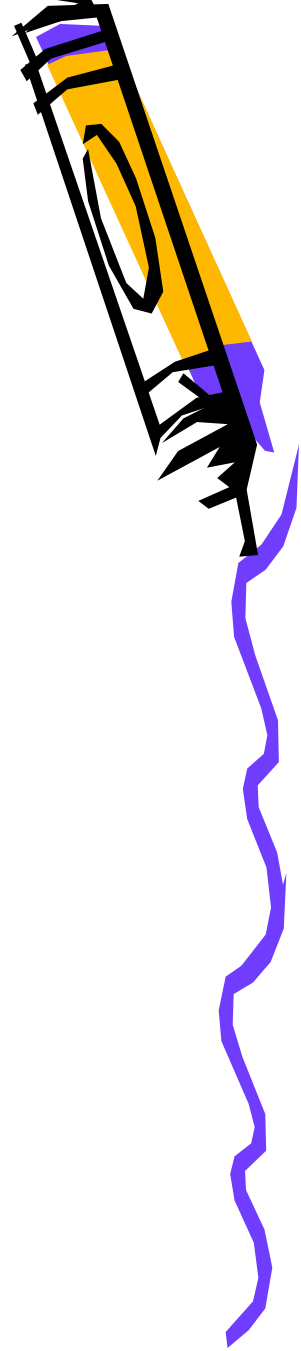
- Zaključivanje (inference) 1:
Forward checking

{Milan = Paviljon lab25}.

Domen(Vanja) = (61, 57, 59, 309)

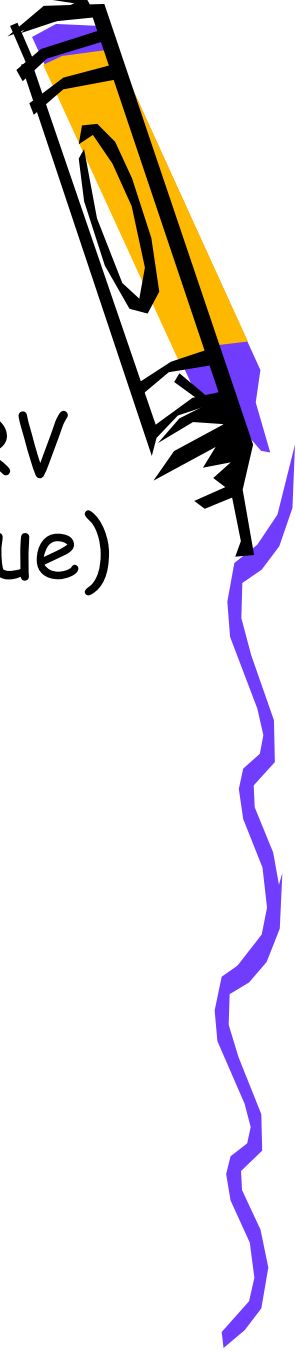
Domen(Nikola) = (61, 57, 59)

Domen(Dragan) = (61)



Raspored - rešenje

- Biramo promenljivu upotrebom MRV heuristike (minimum remaining value)
- Izbor je Dragan
 - samo jedna vrednost, 61



Raspored - rešenje

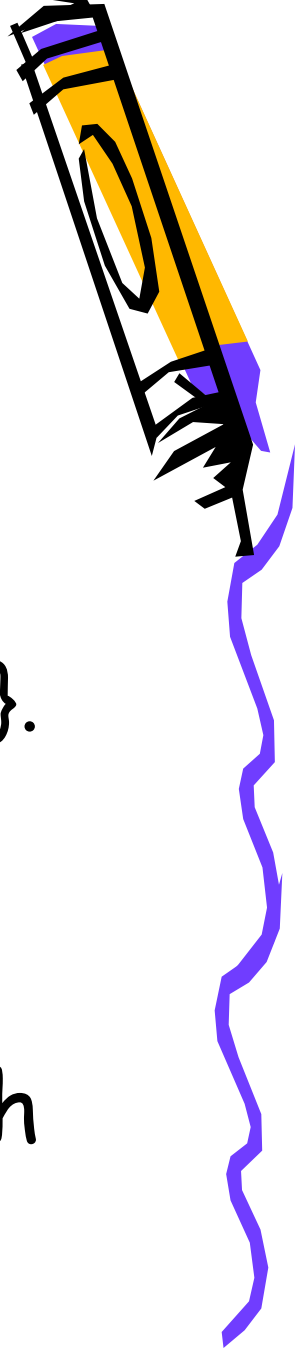
- Zaključivanje (inference) 1:
Forward checking

{Milan = Paviljon lab25, Dragan = 61}.

Domen(Vanja) = (57, 59, 309)

Domen(Nikola) = (57, 59)

- Uklanjamo 61 iz domena ostalih
promenljivih



Raspored - rešenje

Zaključivanje (inference) 2: Arc consistency

Red:

Nikola->Dragan

Dragan->Nikola

Nikola->Milan

Milan->Nikola

Nikola->Vanja

Vanja->Nikola

Dragan->Milan

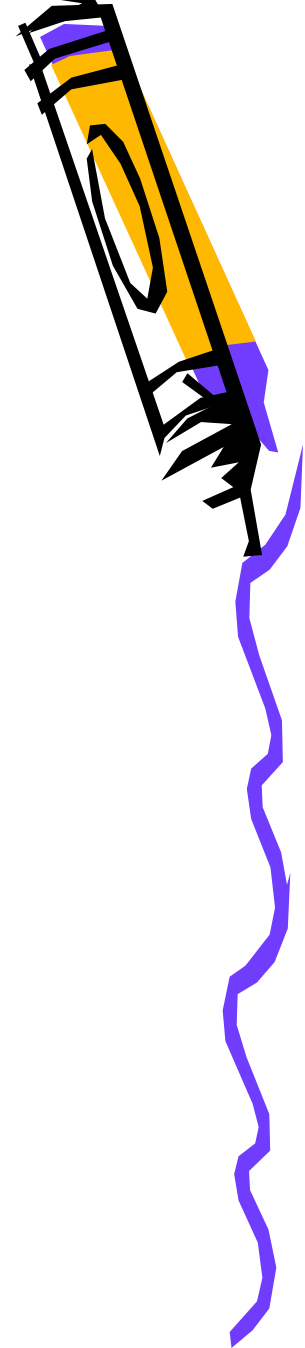
Milan->Dragan

Dragan->Vanja

Vanja->Dragan

Milan->Vanja

Vanja->Milan

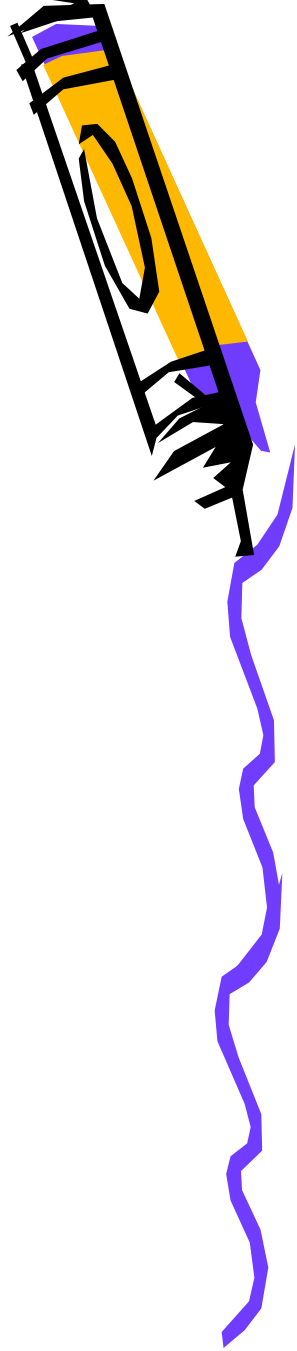


Raspored - rešenje

- Sve promenljive su arc consistent
- Gotov je korak zaključivanja
- Nastavljamo Backtracking
({Milan = Paviljon lab25, Dragan = 61}.

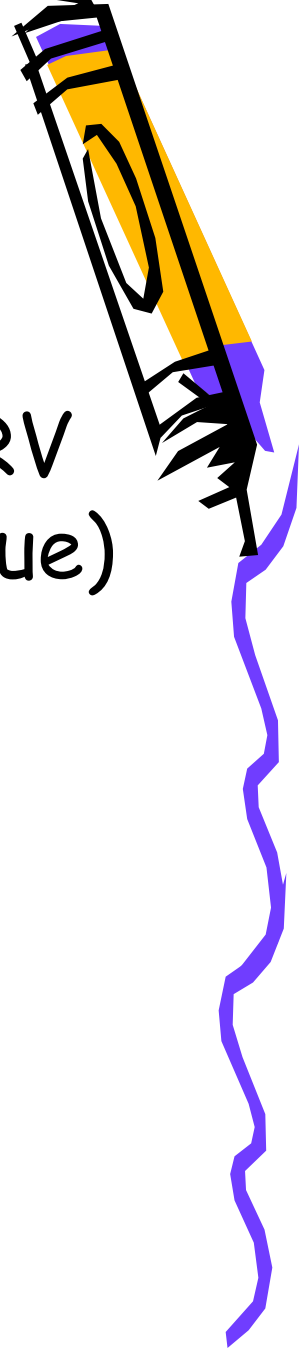
$$\text{Domen}(\text{Vanja}) = (57, 59, 309)$$

$$\text{Domen}(\text{Nikola}) = (57, 59)$$



Raspored - rešenje

- Biramo promenljivu upotrebom MRV heuristike (minimum remaining value)
- Izbor je Nikola
 - dve vrednosti, 57 i 59
- Nerešen rezultat
 - nasumice biramo **57**



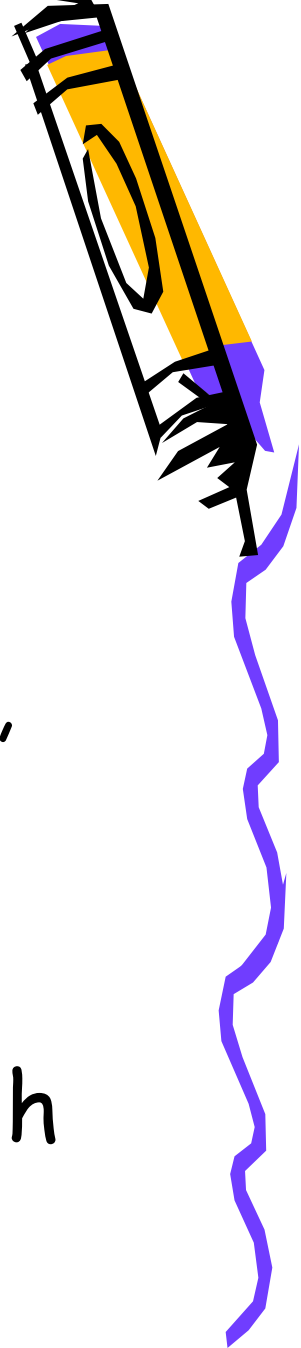
Raspored - rešenje

- Zaključivanje (inference) 1:
Forward checking

{Milan = Paviljon lab25, Dragan = 61,
Nikola = 57}.

Domen(Vanja) = (~~57~~, 59, 309)

- Uklanjammo **57** iz domena ostalih
promenljivih



Raspored - rešenje

Zaključivanje (inference) 2: Arc consistency

Red:

Nikola->Dragan

Dragan->Nikola

Nikola->Milan

Milan->Nikola

Nikola->Vanja

Vanja->Nikola

Dragan->Milan

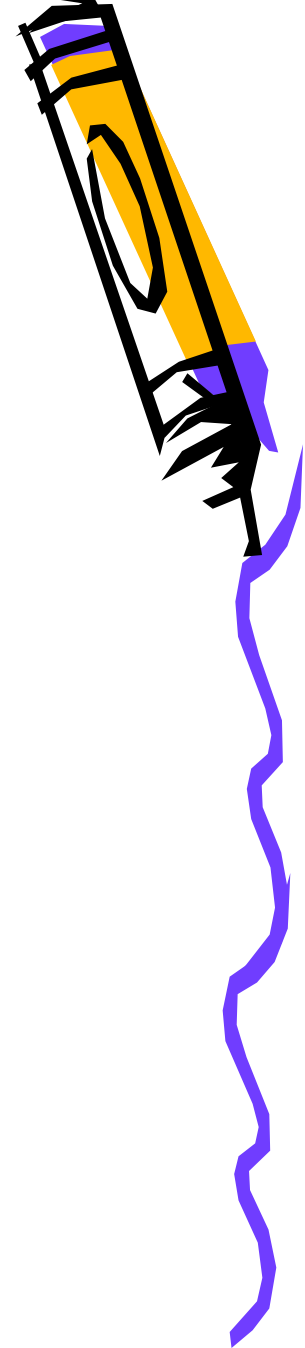
Milan->Dragan

Dragan->Vanja

Vanja->Dragan

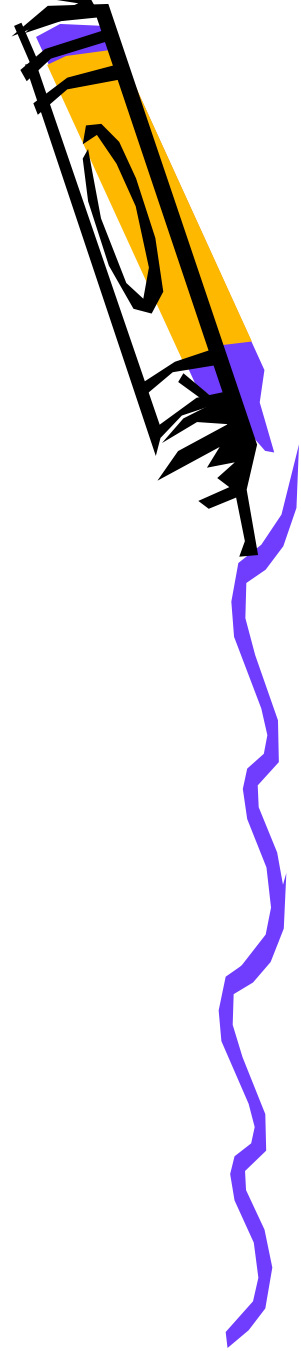
Milan->Vanja

Vanja->Milan



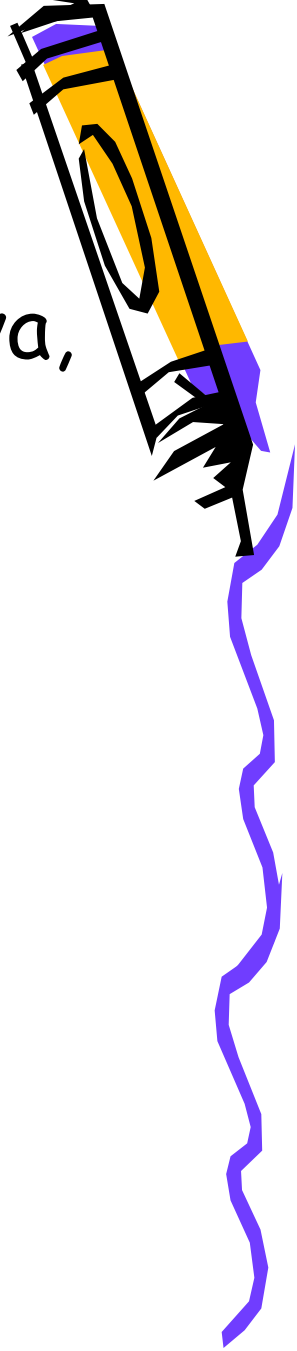
Raspored - rešenje

- Sve promenljive su arc consistent
 - Gotov je korak zaključivanja
 - Nastavljamo Backtracking
- ({Milan = Paviljon lab25, Dragan = 61,
Nikola = 57}).
- Domen(Vanja) = (59, 309)



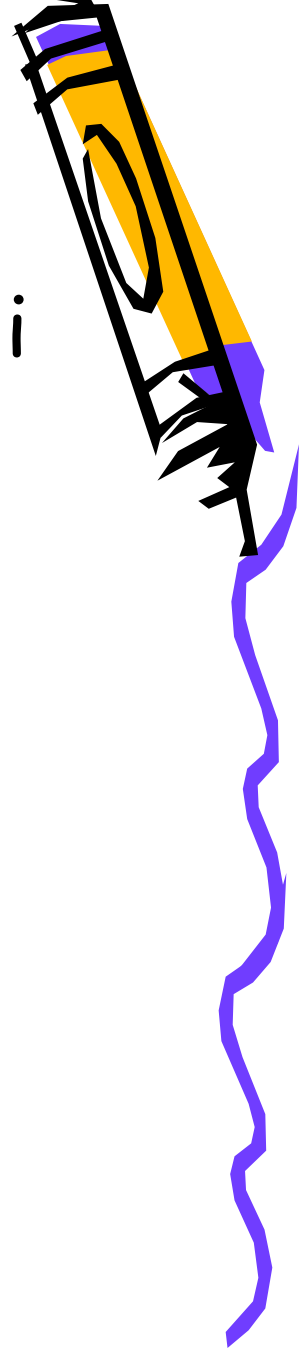
Raspored - rešenje

- Kako je ostala samo jedna promenljiva, možemo izabrati bilo koju vrednost
- Rešenje je:
 - Milan = Paviljon lab25
 - Dragan = 61
 - Nikola = 57
 - Vanja = 59 ili 309



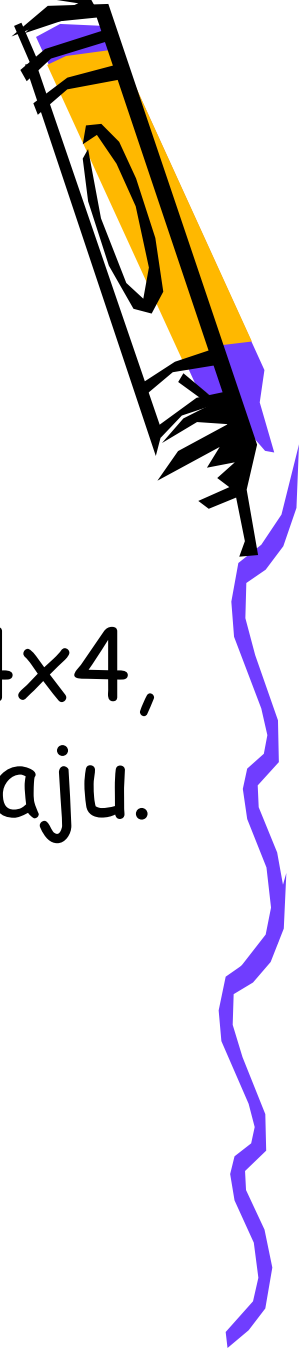
Raspored - rešenje

- Da smo za promenljivu Nikola izabrali vrednost 59, rešenje bi bilo:
 - Milan = Paviljon lab25
 - Dragan = 61
 - Nikola = 59
 - Vanja = 57 ili 309



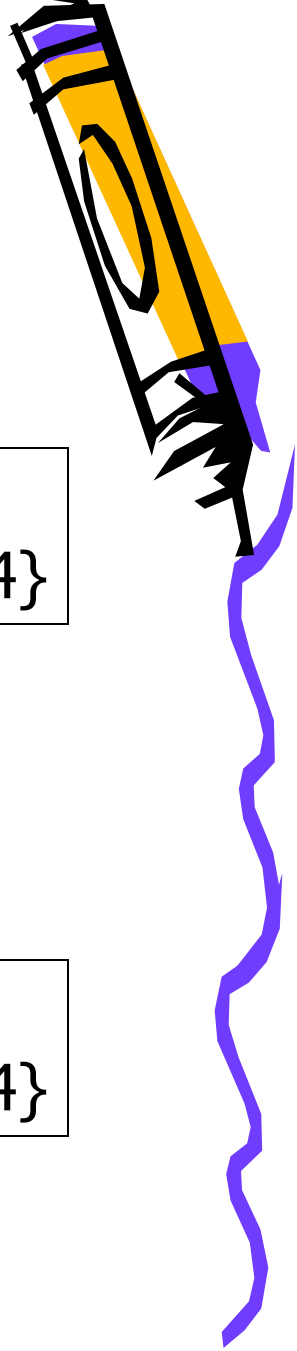
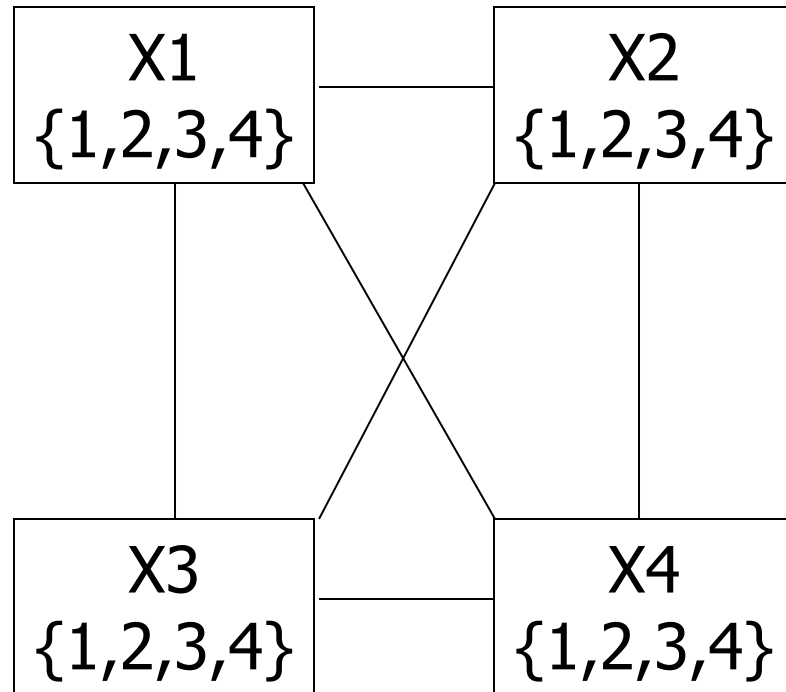
Zadatak 5: 4-Queens problem

Rasporediti 4 kraljice na tabli 4×4 ,
tako da se međusobno ne napadaju.










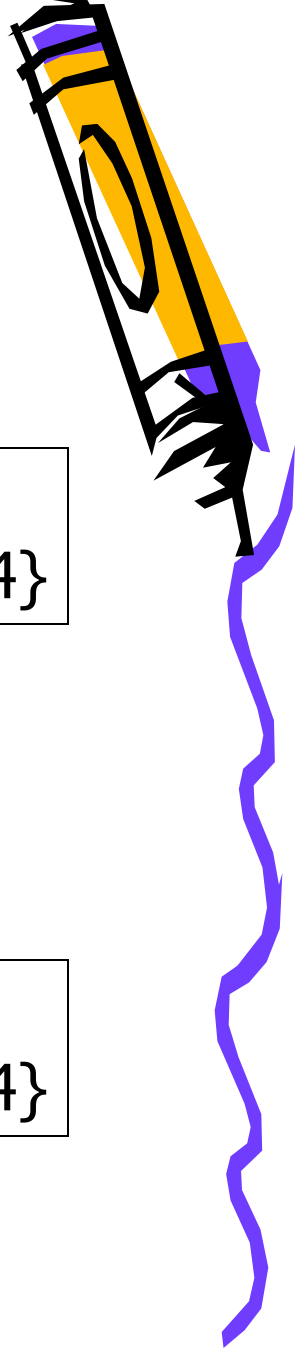
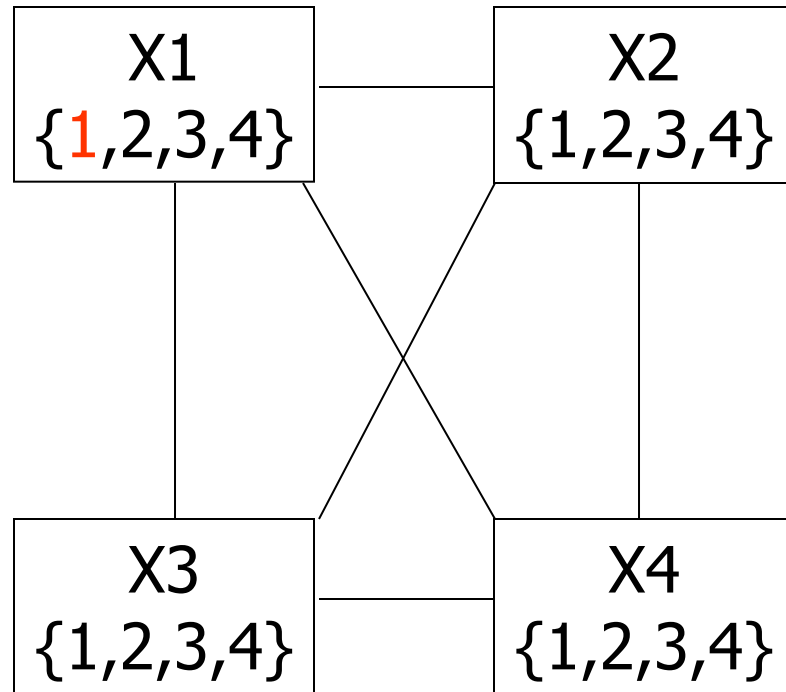
Rešenje: 4-Queens problem

	1	2	3	4
1				
2				
3				
4				










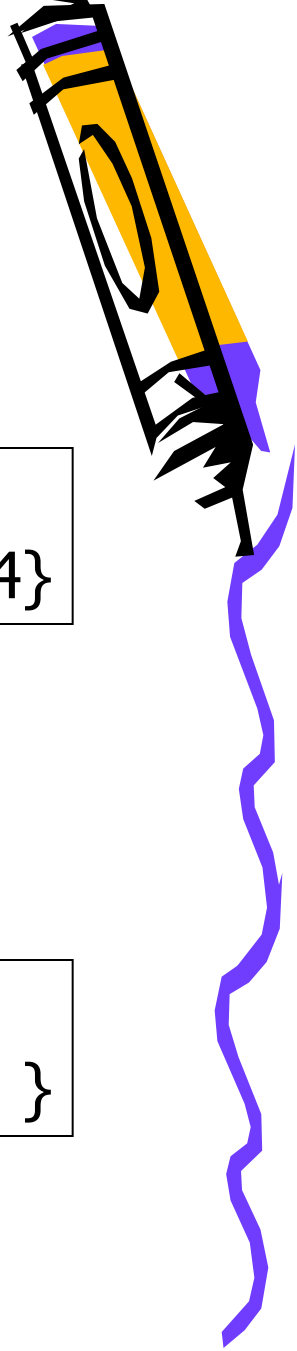
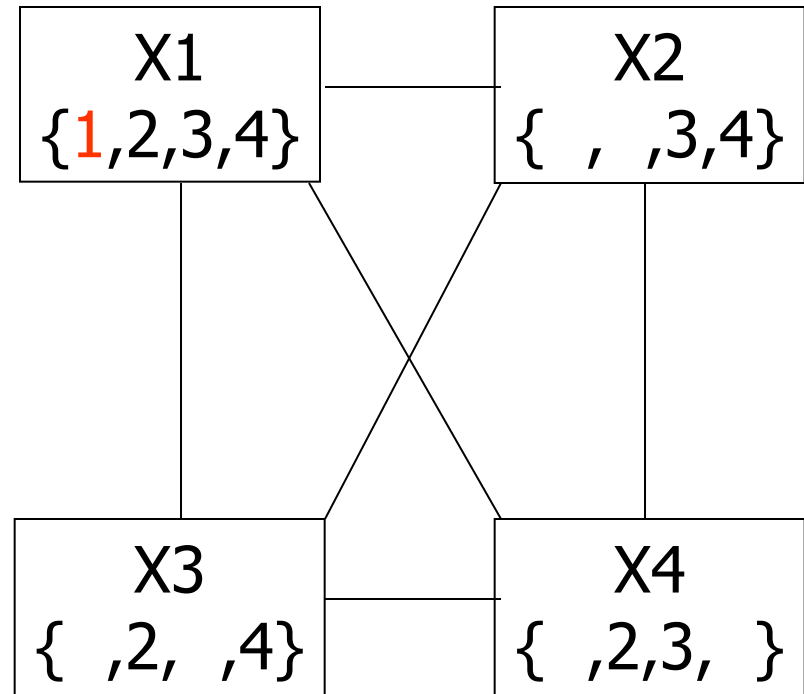
Rešenje: 4-Queens problem

	1	2	3	4
1				
2				
3				
4				



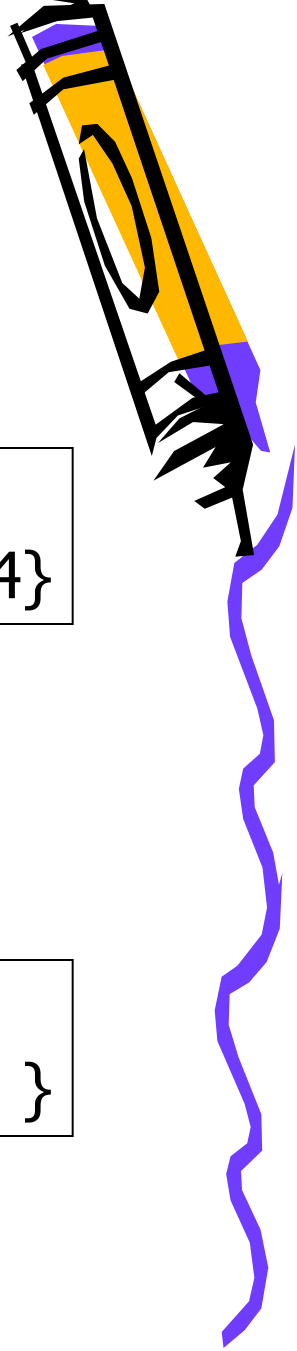
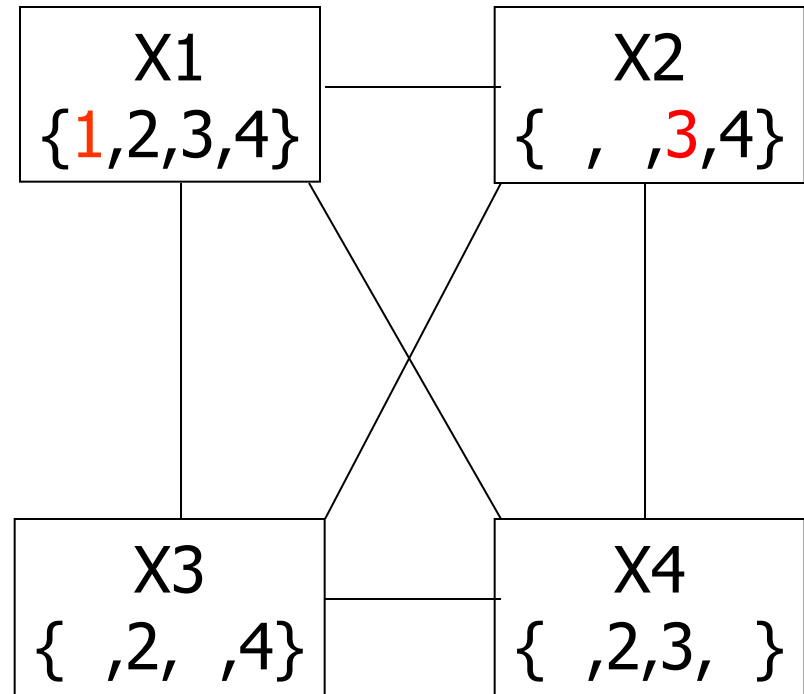
Rešenje: 4-Queens problem

	1	2	3	4
1				
2				
3				
4				



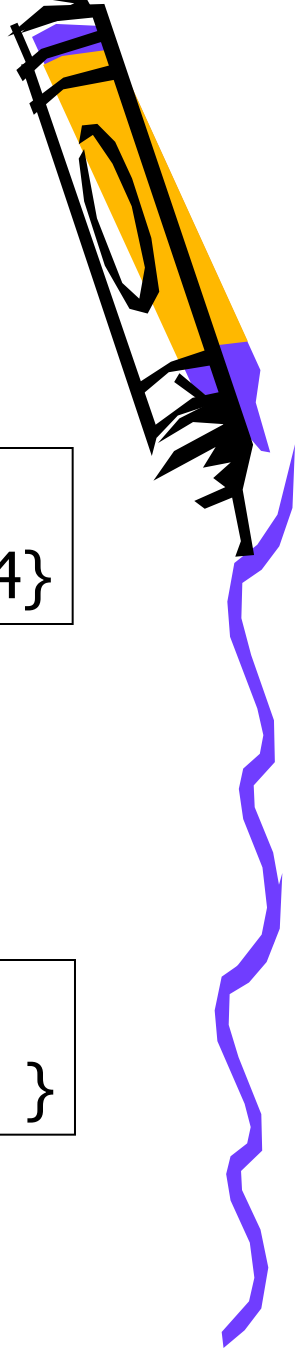
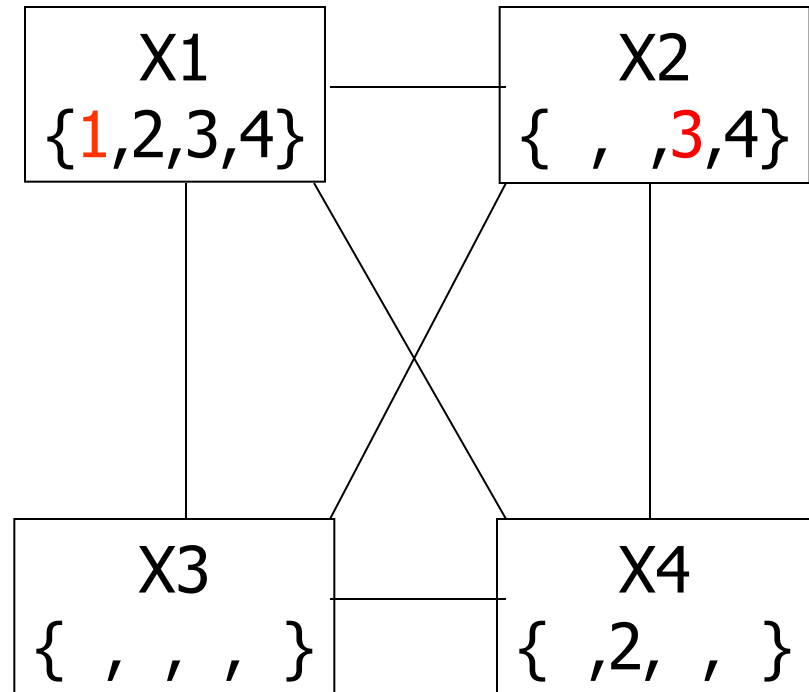
Rešenje: 4-Queens problem

	1	2	3	4
1	★	●	●	●
2	■	●	●	□
3	□	★	●	●
4	■	□	●	●



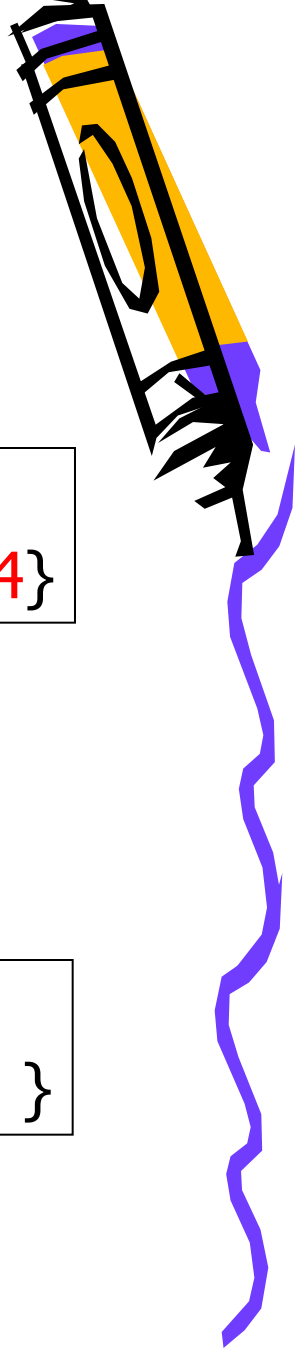
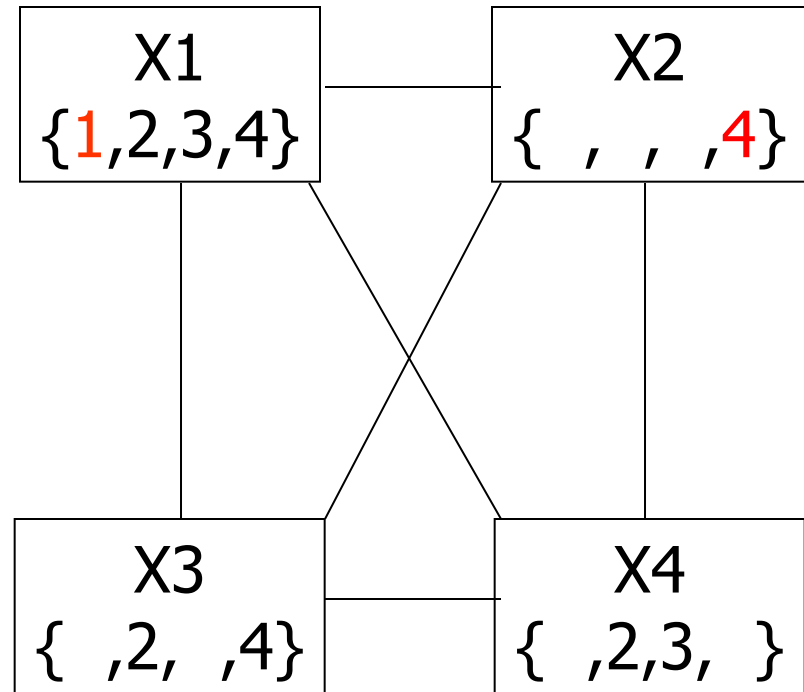
Rešenje: 4-Queens problem

	1	2	3	4
1	★	●	●	●
2	■	●	●	□
3	□	★	●	●
4	■	□	●	●














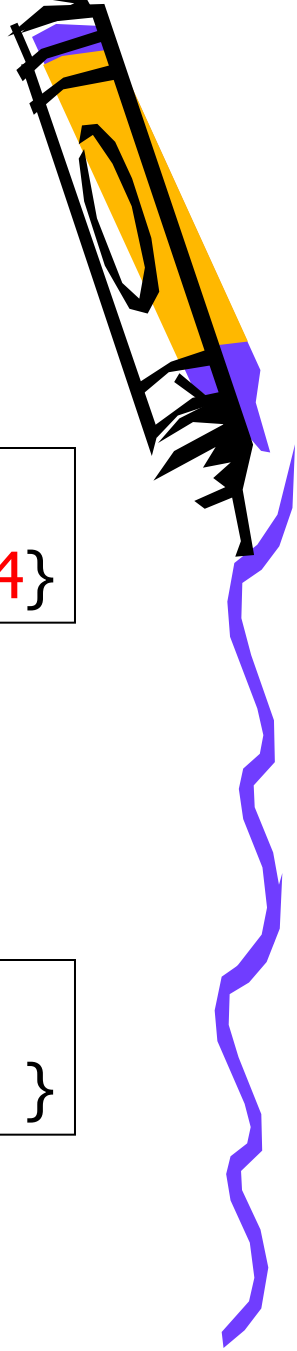
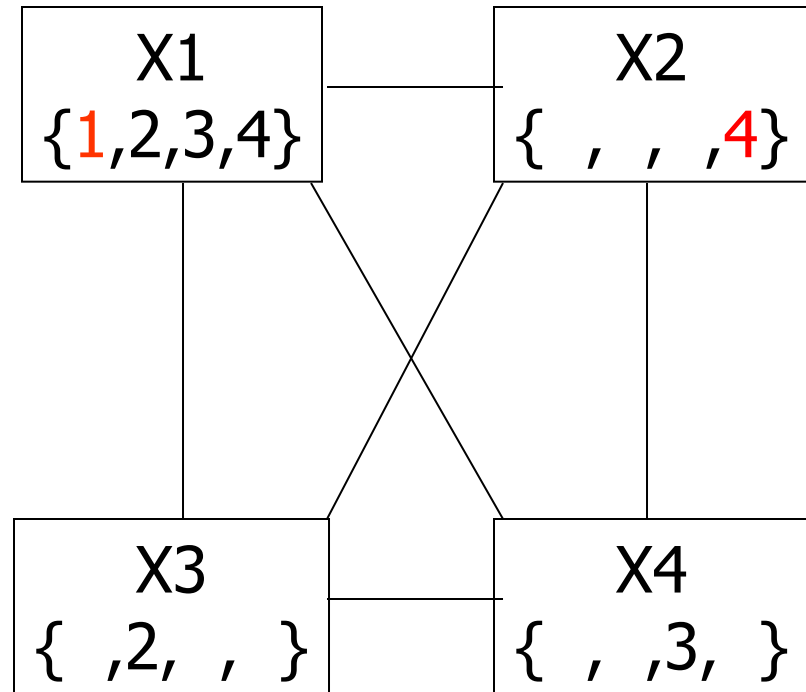
Rešenje: 4-Queens problem

	1	2	3	4
1	★	●	●	●
2	■	●	■	●
3	■	●	●	■
4	■	★	●	●



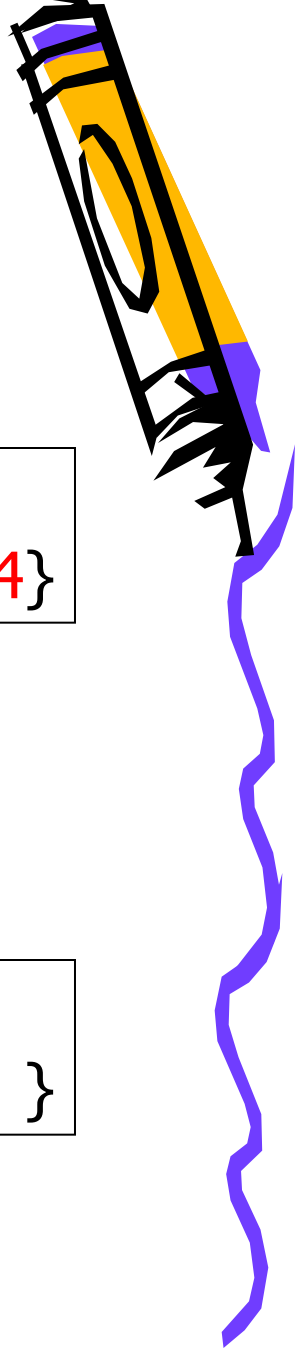
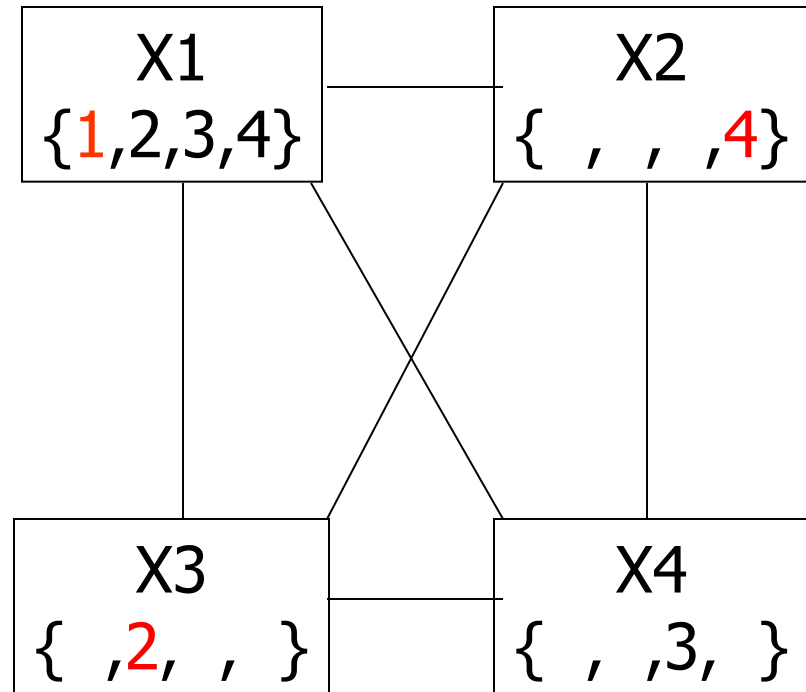
Rešenje: 4-Queens problem

	1	2	3	4
1				
2				
3				
4				



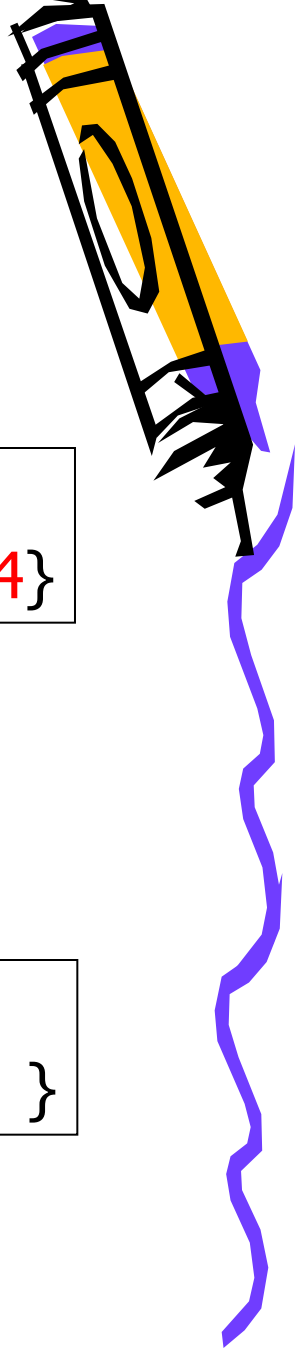
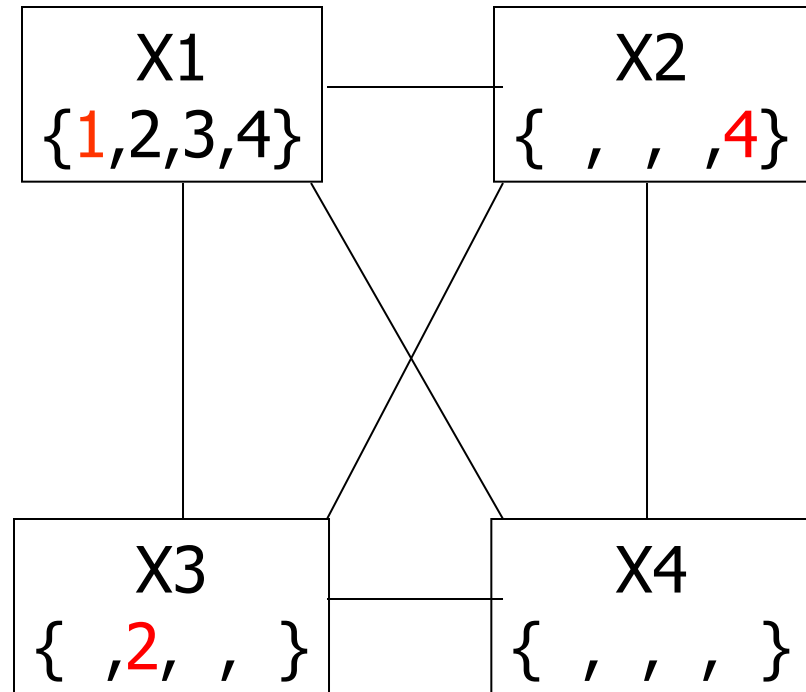
Rešenje: 4-Queens problem

	1	2	3	4
1	★	●	●	●
2		●	★	●
3		●	●	●
4		★	●	●



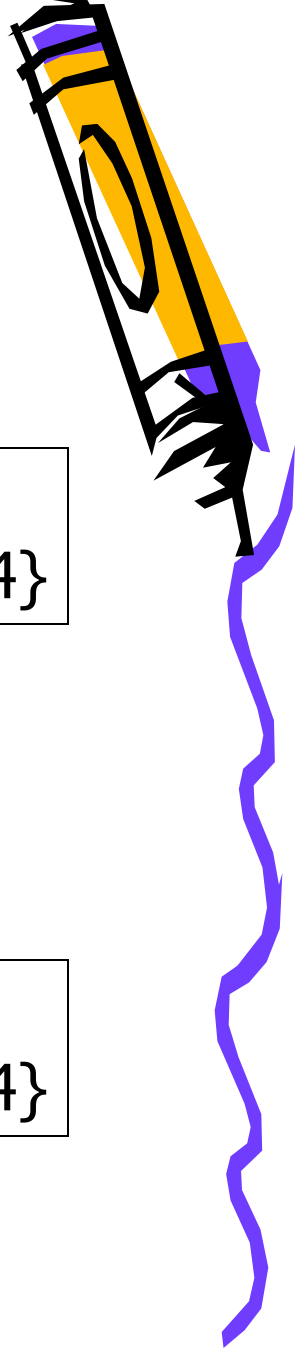
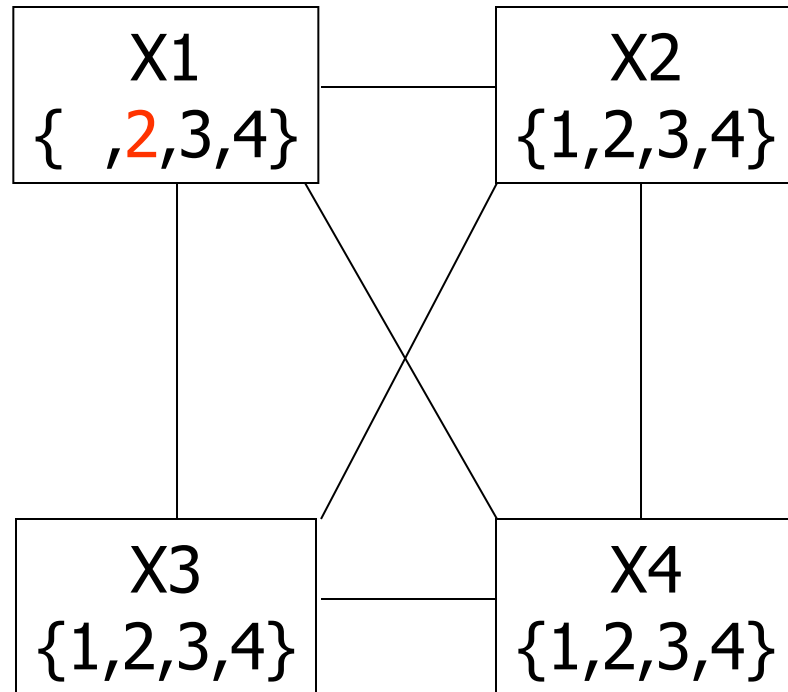
Rešenje: 4-Queens problem

	1	2	3	4
1	★	●	●	●
2		●	★	●
3		●	●	●
4		★	●	●



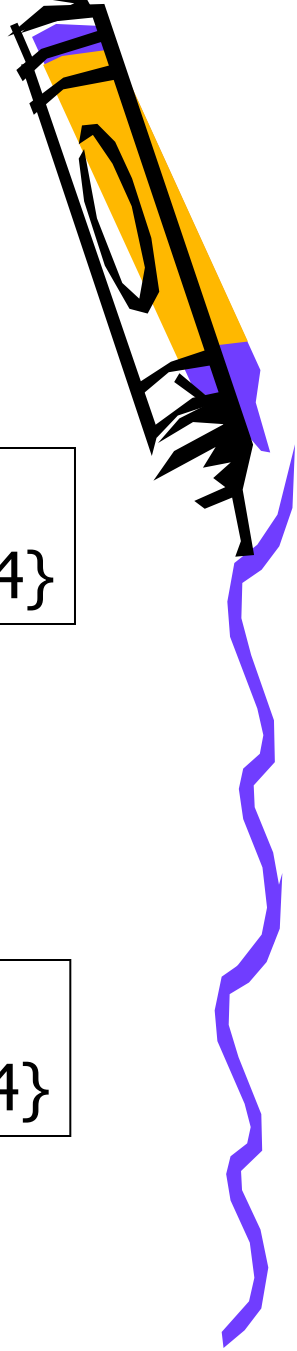
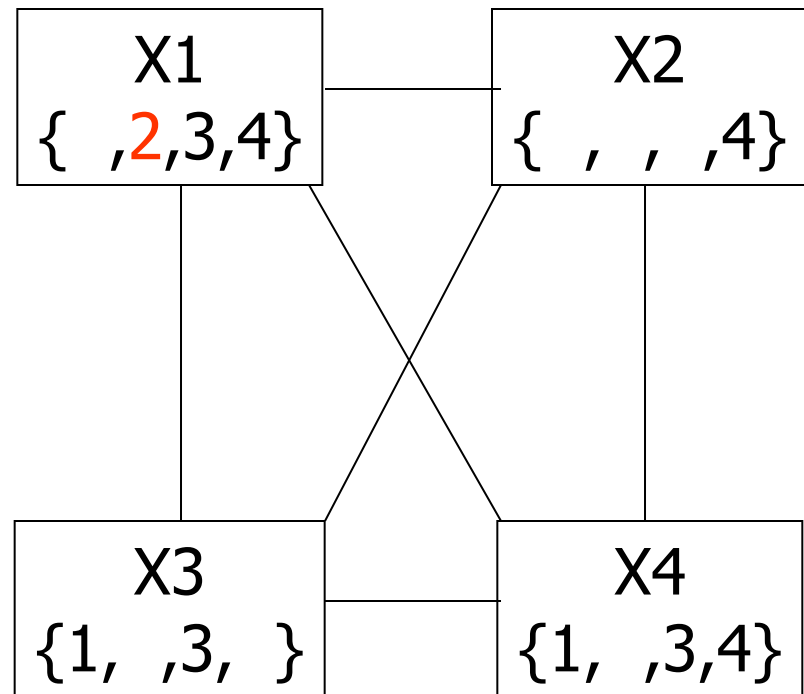
Rešenje: 4-Queens problem

	1	2	3	4
1		●		
2	★	●	●	●
3		●		
4			●	



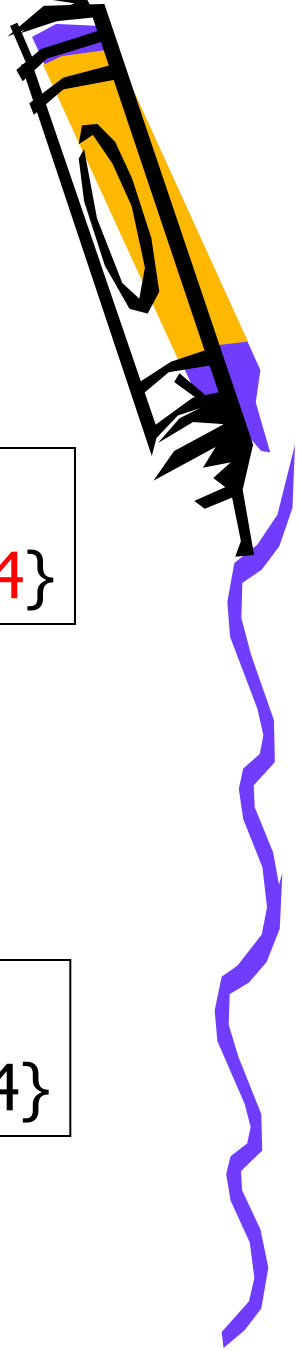
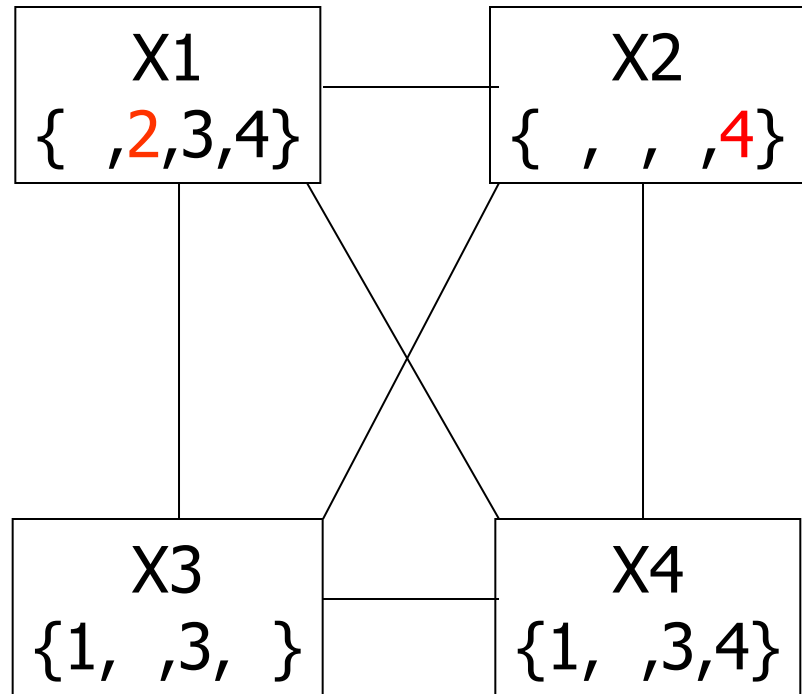
Rešenje: 4-Queens problem

	1	2	3	4
1		●		
2	★	●	●	●
3		●		
4			●	



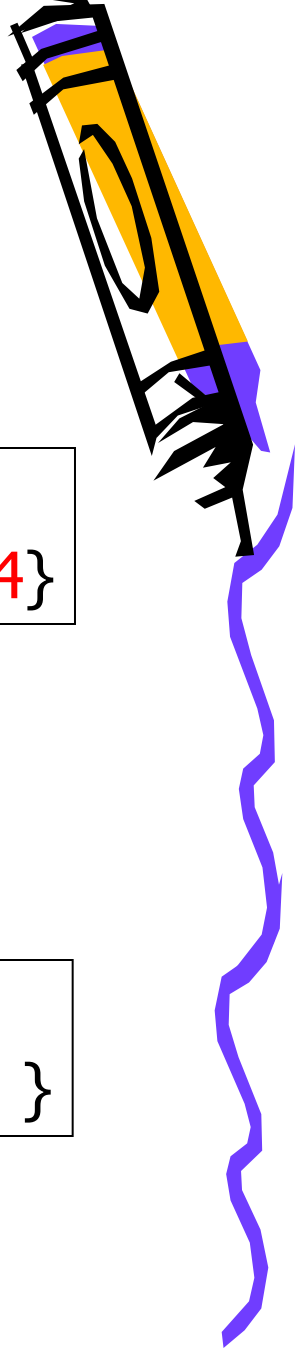
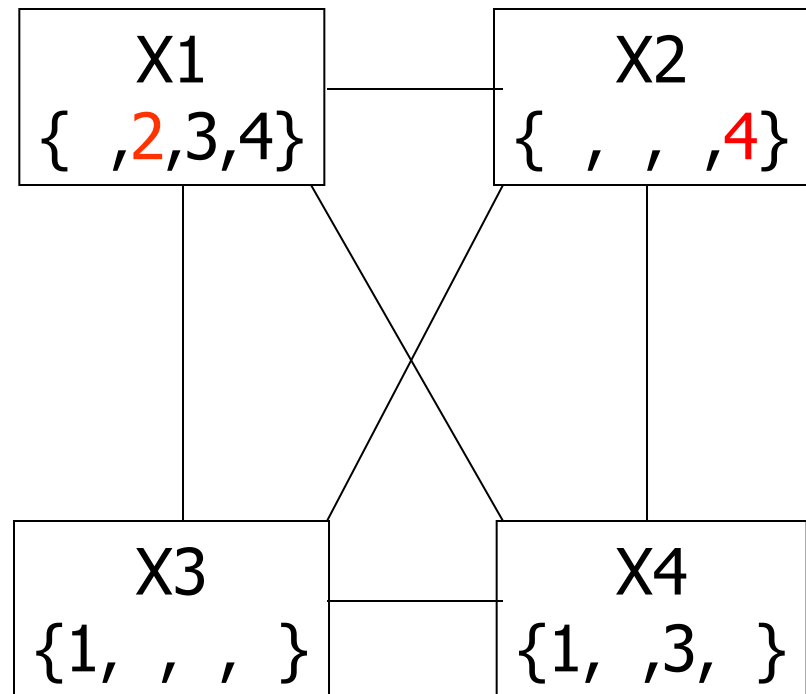
Rešenje: 4-Queens problem

	1	2	3	4
1		●		
2	★	●	●	●
3		●	●	
4		★	●	●



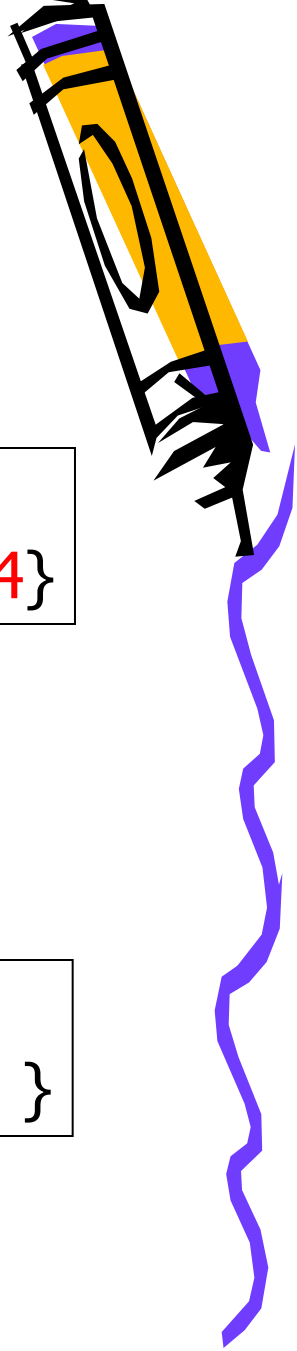
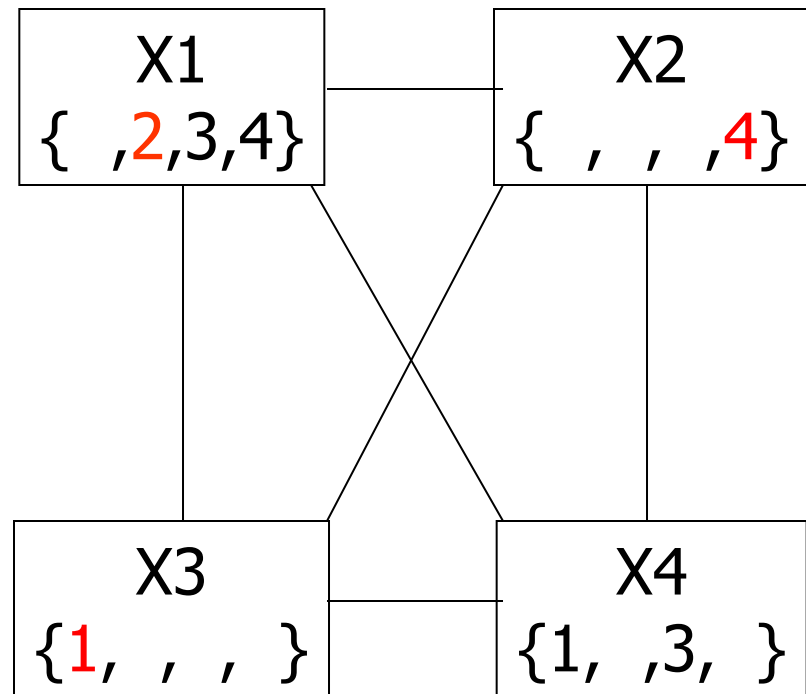
Rešenje: 4-Queens problem

	1	2	3	4
1		●		
2	★	●	●	●
3		●	●	
4		★	●	●



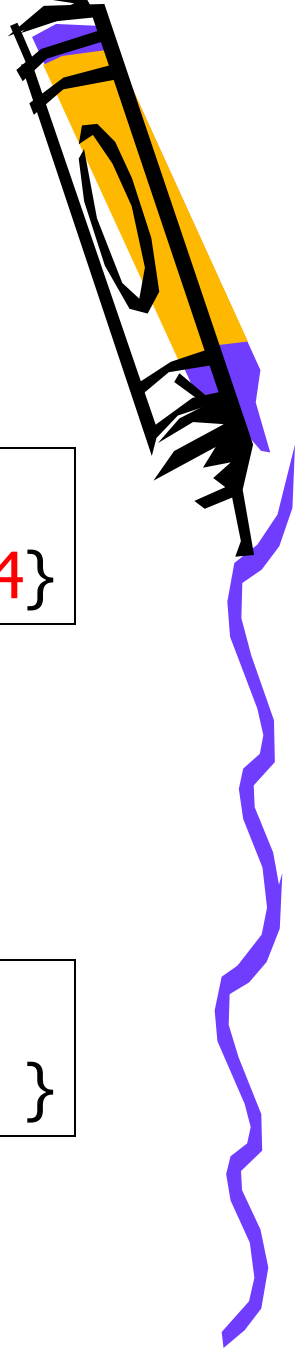
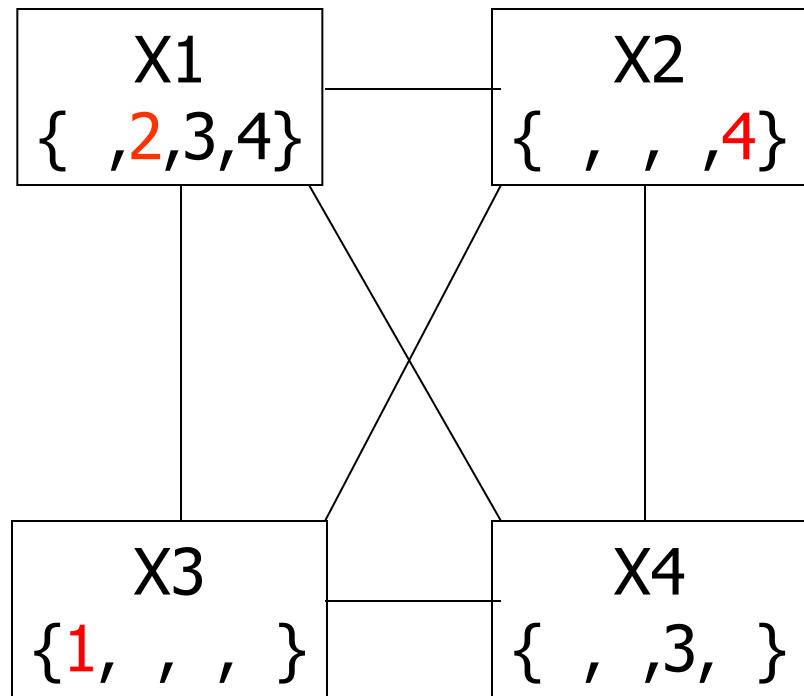
Rešenje: 4-Queens problem

	1	2	3	4
1		●	★	●
2	★	●	●	●
3		●	●	
4		★	●	●



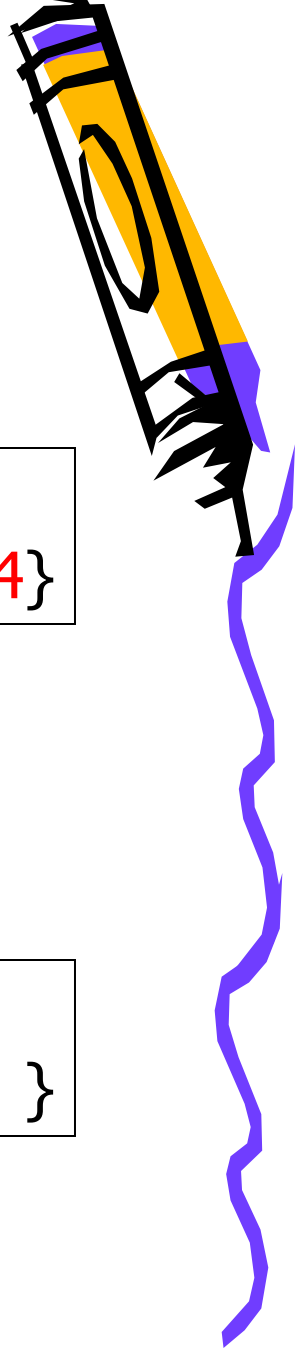
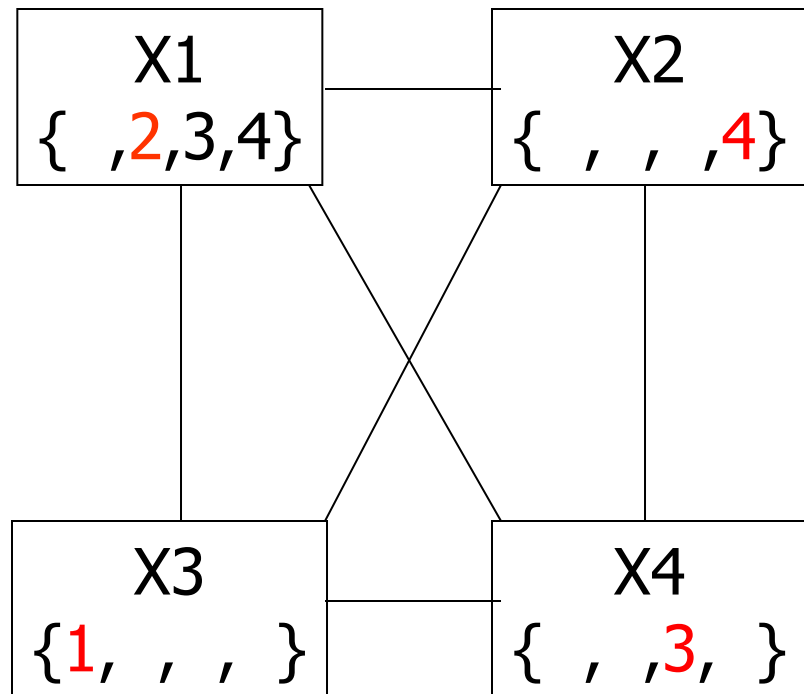
Rešenje: 4-Queens problem

	1	2	3	4
1		●	★	●
2	★	●	●	●
3		●	●	
4		★	●	●



Rešenje: 4-Queens problem

	1	2	3	4
1		●	★	●
2	★	●	●	●
3		●	●	★
4		★	●	●



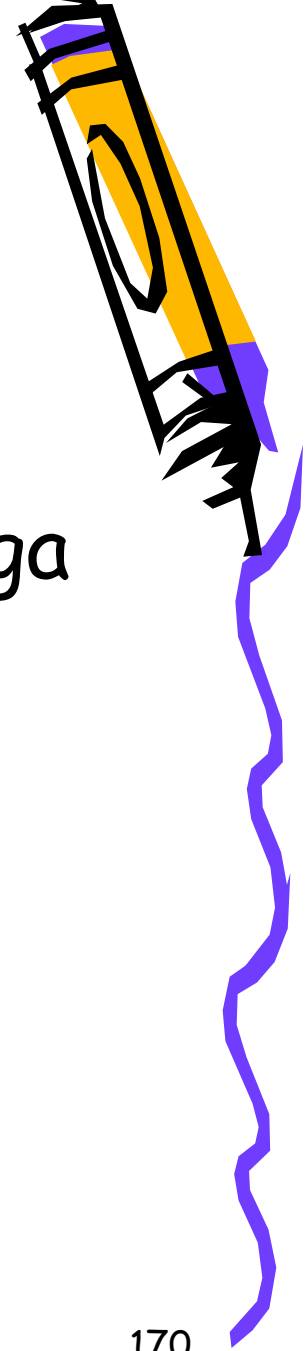
Zadatak 6: Problem tri muzičara

- Petar, Jovan i Pavle su muzičari. Jedan od njih svira saksofon, drugi gitaru, a treći doboše. Jedan od njih se plaši broja 13, drugi mačaka, a treći se boji visine.
- Poznato je:
 - Petar i gitarista su paraglajdisti
 - Pavle i svirač saksofona vole mačke
 - Dobošar živi u stanu broj 13 na trinaestom spratu

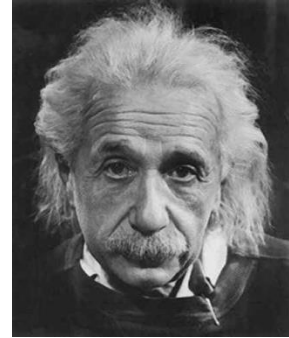


Odrediti

- Za Petra, Jovana i Pavla koji instrument svira svaki od njih i čega se koji od njih boji.



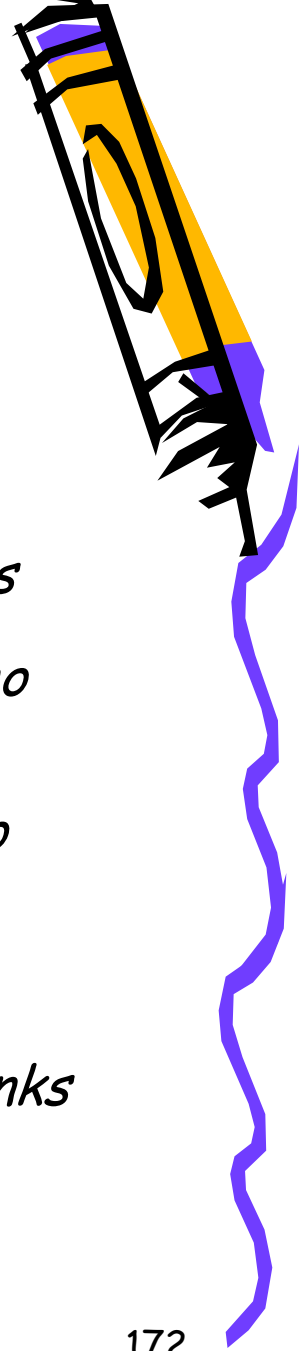
- *This brainteaser, reportedly written by Einstein is difficult and Einstein said that 98% of the people in the world could not figure it out. Which percentage are you in?*
- *There are five houses in a row in different colors. In each house lives a person with a different nationality. The five owners drink a different drink, smoke a different brand of cigar and keep a different pet, one of which is a Walleye Pike.*



The question is- who owns the fish?

Hints:

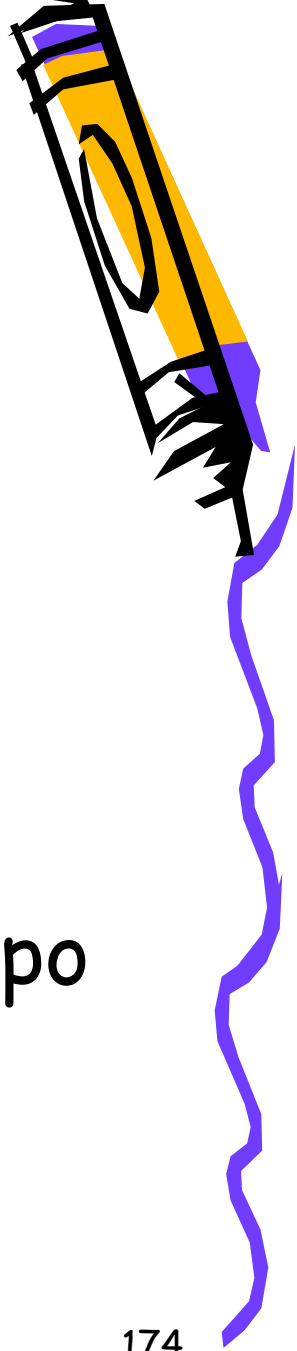
- 1. The Brit lives in the red house.*
- 2. The Swede keeps dogs as pets.*
- 3. The Dane drinks tea.*
- 4. The green house is on the left of the white house.*
- 5. The green house owner drinks coffee.*
- 6. The person who smokes Pall Malls keeps birds.*
- 7. The owner of the yellow house smokes Dunhills.*
- 8. The man living in the house right in the center drinks milk.*
- 9. The man who smokes Blends lives next to the one who keeps cats.*
- 10. The Norwegian lives in the first house.*
- 11. The man who keeps horses lives next to the one who smokes Dunhills.*
- 12. The owner who smokes Bluemasters drinks beer.*
- 13. The German smokes Princes.*
- 14. The Norwegian lives next to the blue house.*
- 15. The man who smokes Blends has a neighbor who drinks water.*



Analiza problema

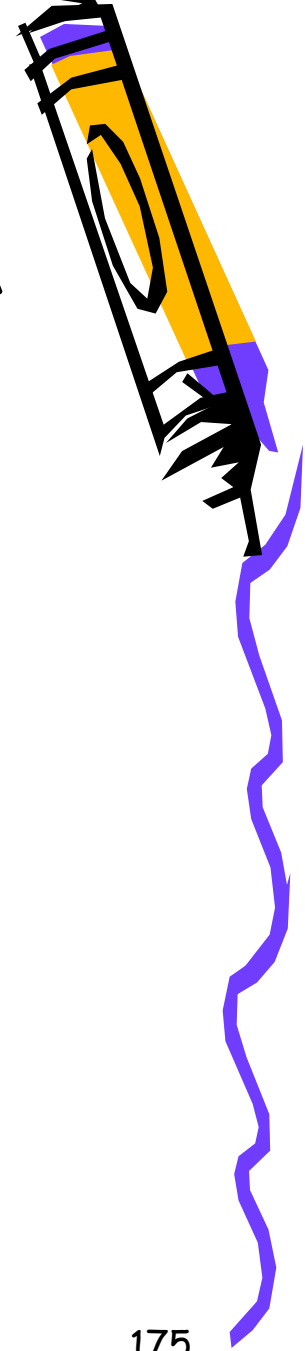


- Svaka od osoba opisana je sa svoje tri osobine:
 - Ime
 - Zanimanje
 - Strah
- Svaka osobina može imati jednu od konkretne tri vrednosti
- Definišemo tri relacije identiteta, po jednu za svaki par osobina



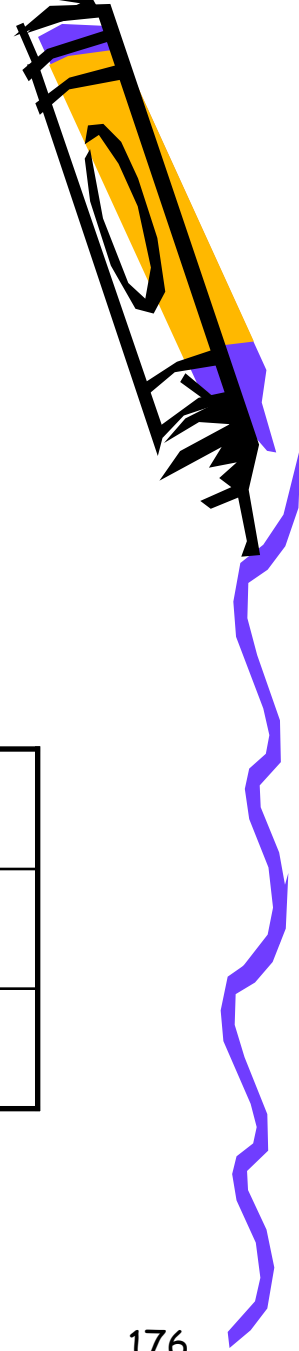
- Neka NOT(Petar, gitarista) znači da osoba sa imenom Petar nije gitarista. Činjenice date u postavci iskazuju se na sledeći način:

1. NOT(Petar, gitarista)
2. NOT(Petar, plaši se visine)
3. NOT(gitarista, plaši se visine)
4. NOT(Pavle, plaši se mačaka)
5. NOT(Pavle, saksofonista)
6. NOT(saksofonista, plaši se mačaka)
7. NOT(dobošar, plaši se broja 13)
8. NOT(dobošar, plaši se visine)



- Relacije možemo predstaviti tabelarno. Svaki red i kolona su označeni konkretnim vrednostima neke osobine
- NOT ulaz u tabeli označava da je relacija identiteta isključiva
- YES označava da relacija važi

	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

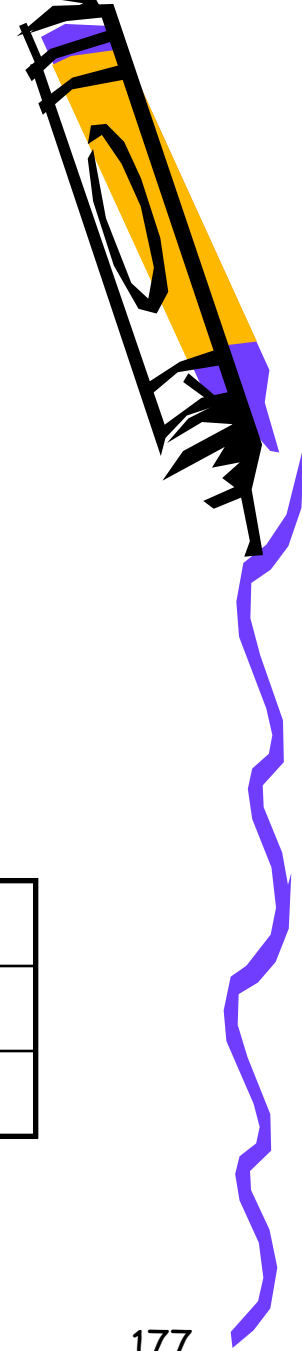


plaši se broja 13 plaši se mačaka plaši se visine

Petar		NOT,2
Pavle	NOT,4	
Jovan		

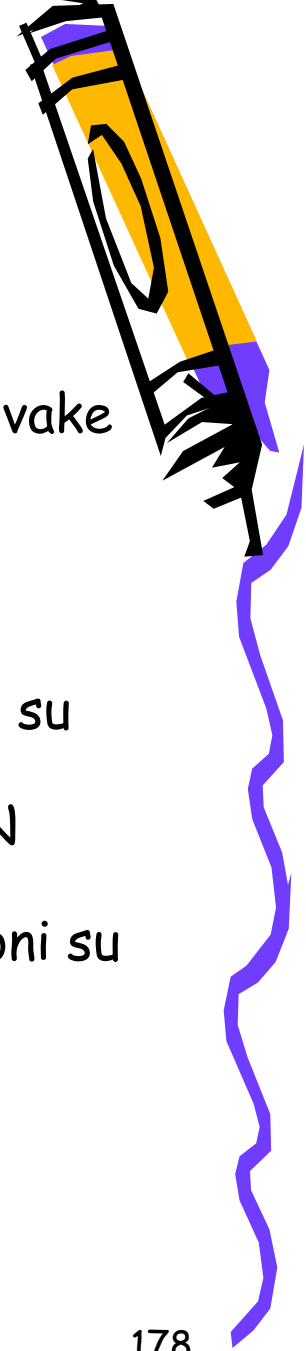
plaši se broja 13 plaši se mačaka plaši se visine

gitarista		NOT,3
saksofonista	NOT,6	
dobošar	NOT,7	NOT,8

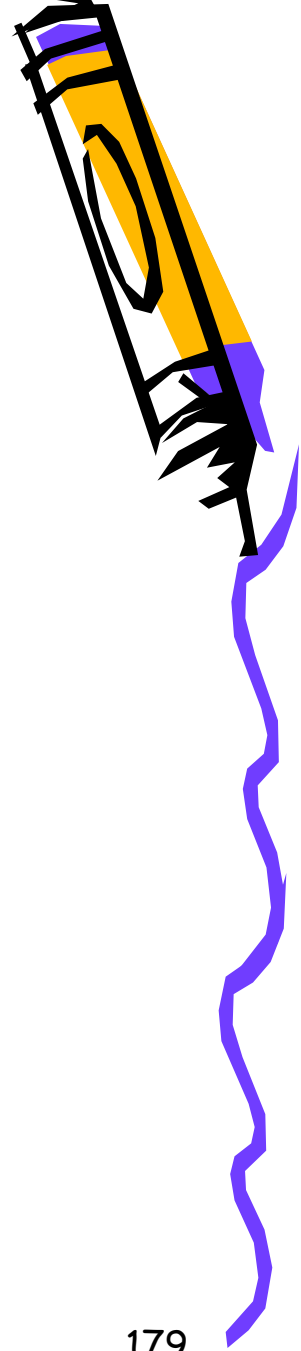


Kako popuniti tabele ?

- Svaka osoba poseduje svoju karakterističnu vrednost svake od osobina!
1. IF svi ulazi u jednoj vrsti su NOT izuzev jednog THEN preostali je YES
 2. IF jedan ulaz u vrsti je YES THEN svi ostali u toj vrsti su NOT
 3. IF svi ulazi u jednoj koloni su NOT izuzev jednog THEN preostali je YES
 4. IF jedan ulaz u koloni je YES THEN svi ostali u toj koloni su NOT
 5. IF važi YES(x,y) i NOT(y,z) THEN može se zaključiti NOT(x,z)



- Pravila popunjavanja izraza su ograničenja koja su svojstvena problemu



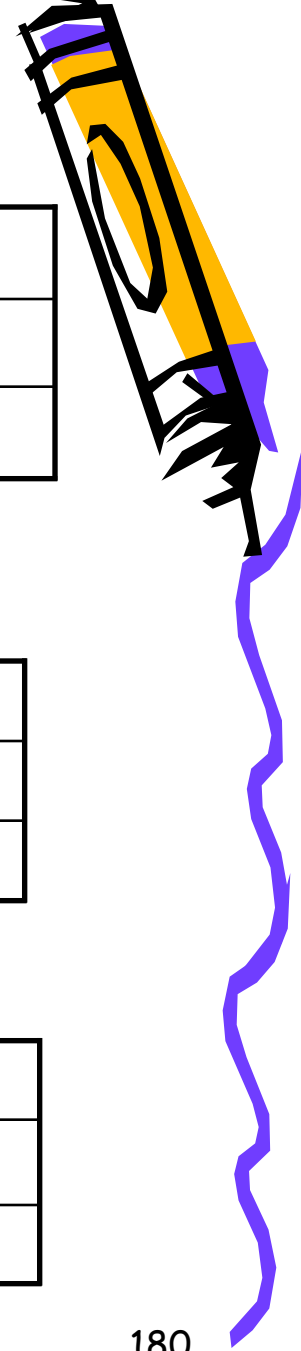
	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista			NOT,3
saksofonista		NOT,6	.
dobošar	NOT,7		NOT,8



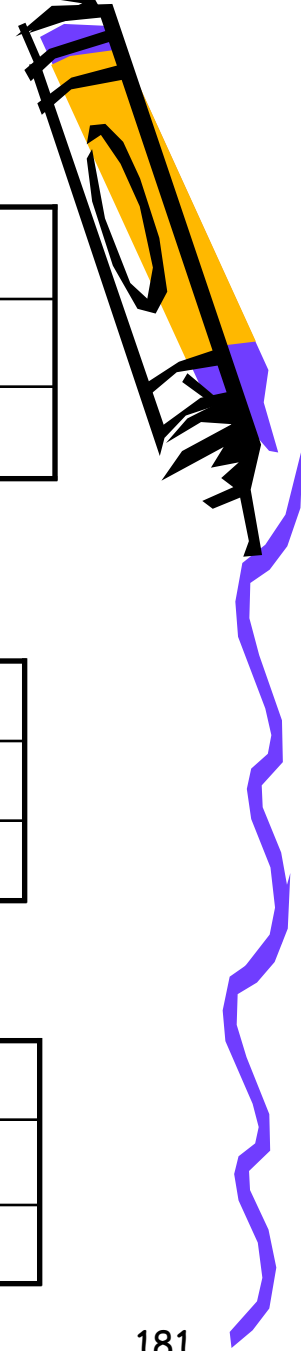
	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

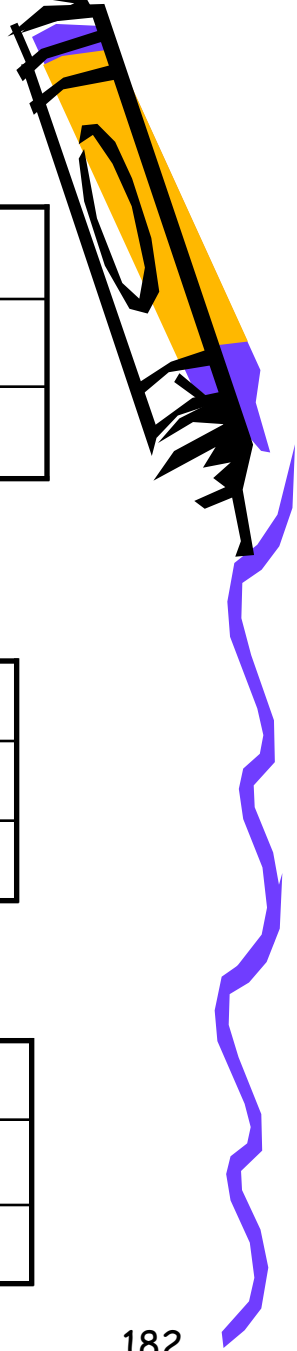
plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista			NOT,3
saksofonista	.	NOT,6	YES
dobošar	NOT,7		NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

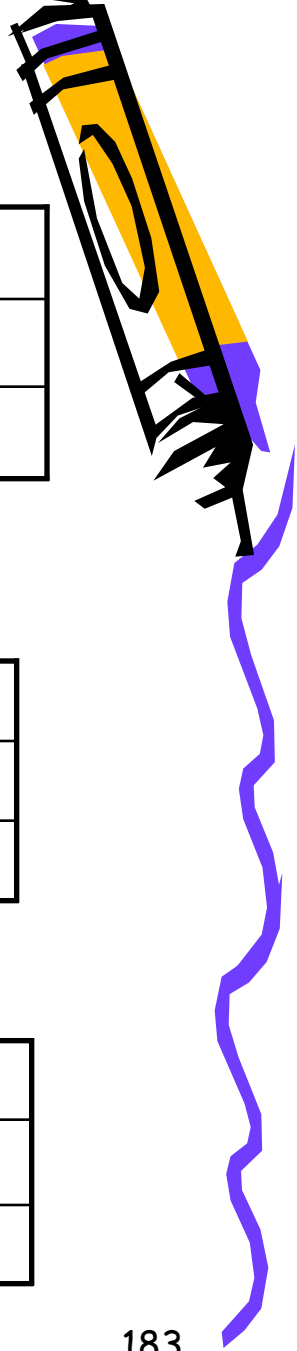
plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	.		NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7		NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

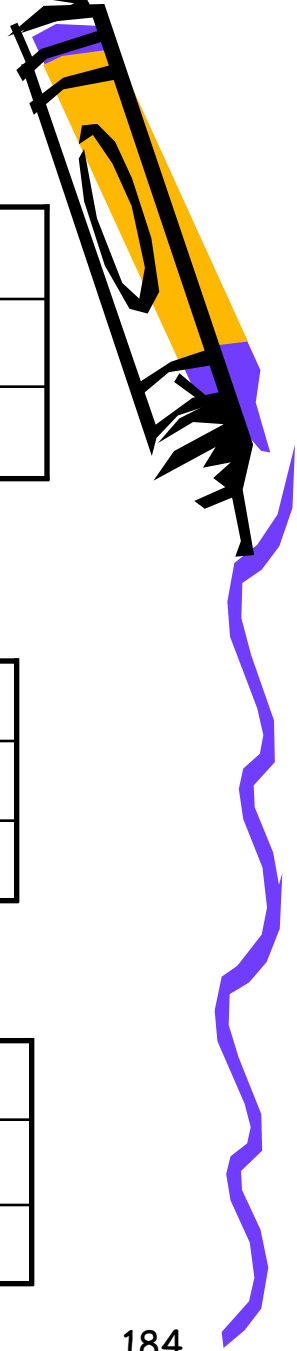
plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	YES		NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	.	NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	YES	.	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8



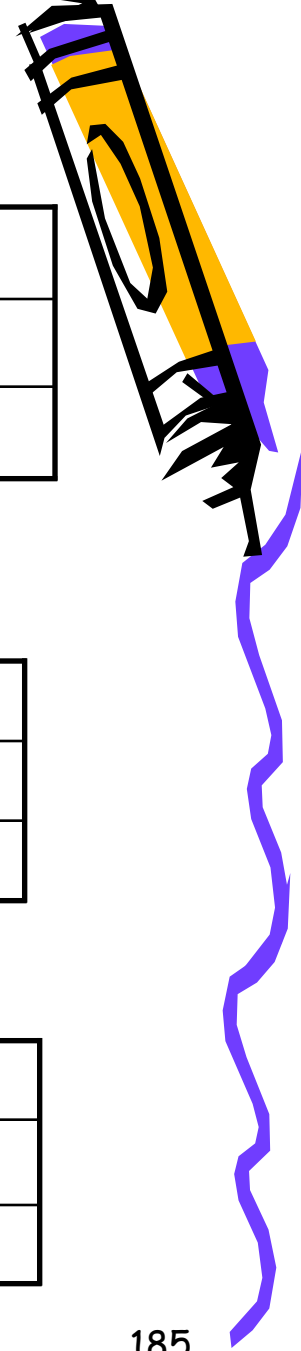
	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle		NOT,5	.
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

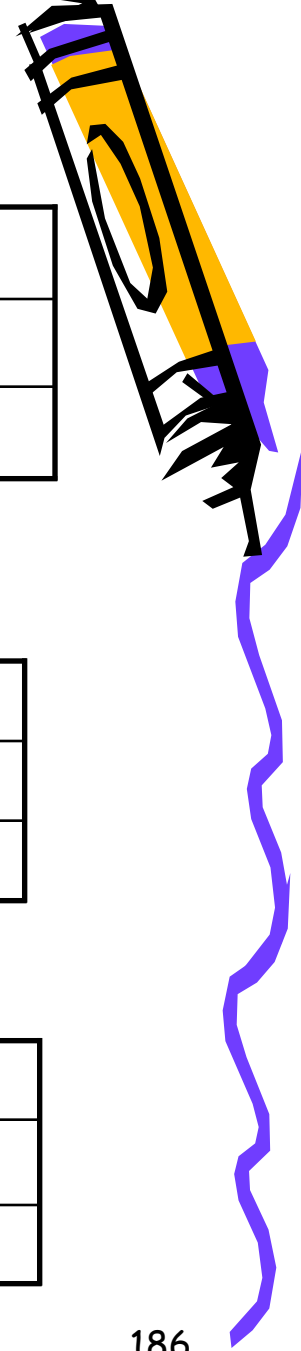
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8

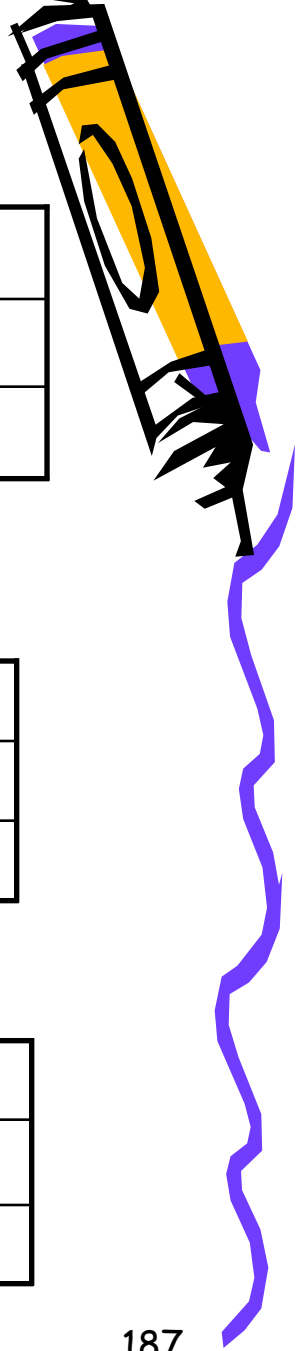


	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle	.	NOT,5	NOT
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
Petar			NOT,2
Pavle		NOT,4	
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1		
Pavle	YES	NOT,5	NOT
Jovan	.		

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8

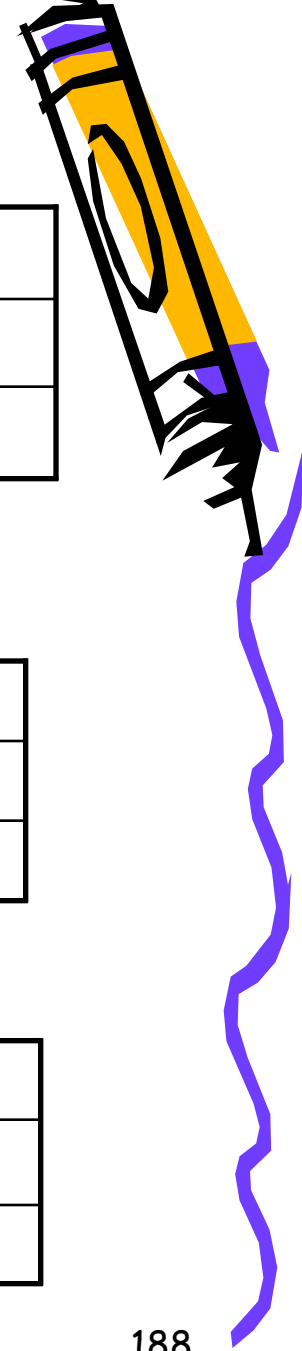


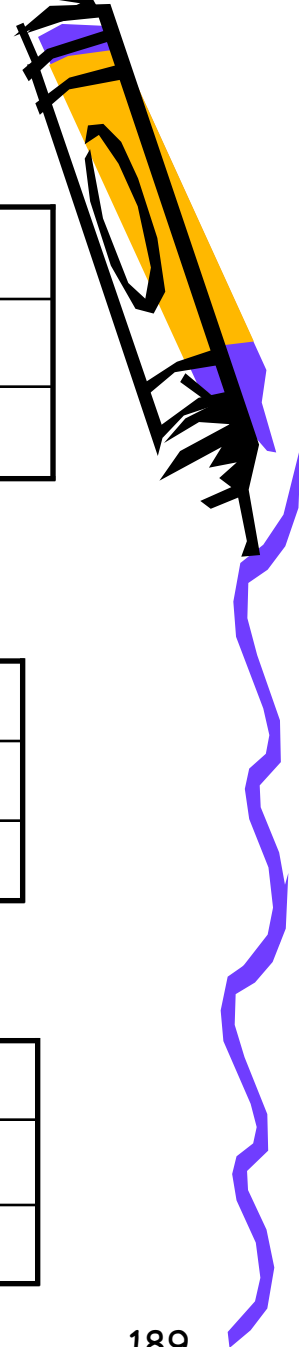
	gitarista	saksofonista	dobošar
Petar	NOT,1	.	
Pavle	YES	NOT,5	NOT
Jovan	NOT		

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1	NOT	
Pavle	YES	NOT,5	NOT
Jovan	NOT	.	

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8

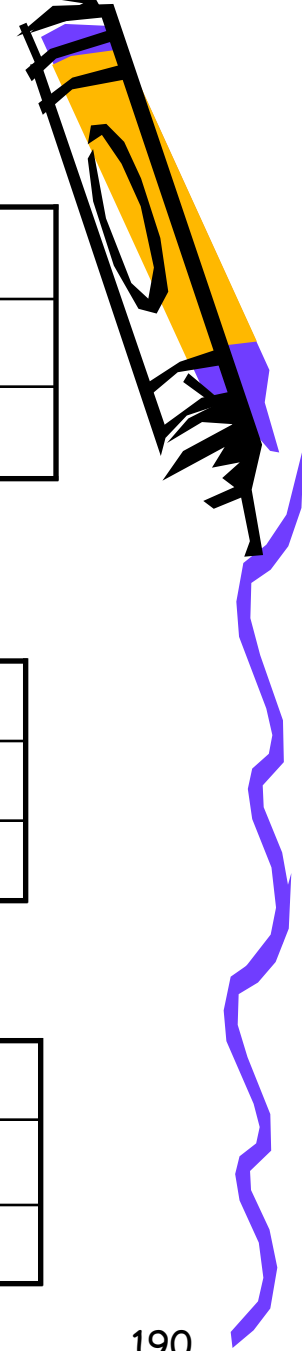


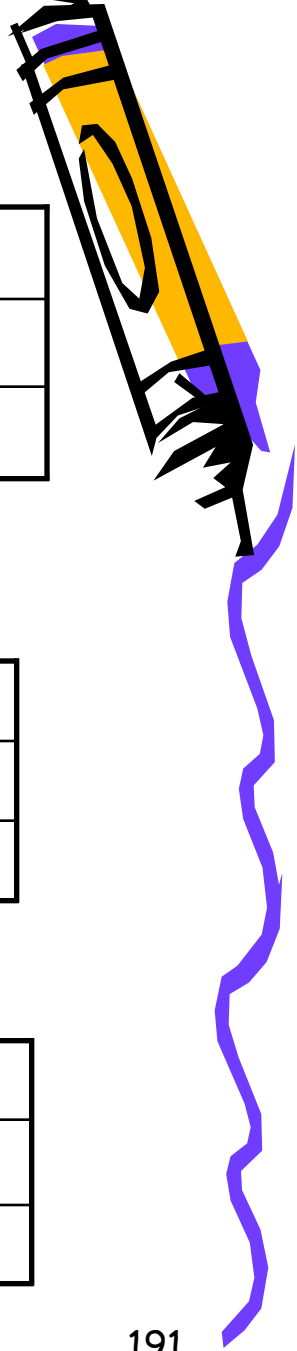
	gitarista	saksofonista	dobošar
Petar	NOT,1	NOT	.
Pavle	YES	NOT,5	NOT
Jovan	NOT	YES	

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1	NOT	YES
Pavle	YES	NOT,5	NOT
Jovan	NOT	YES	.

plaši se broja 13 plaši se mačaka plaši se visine

Petar			NOT,2
Pavle		NOT,4	
Jovan			

plaši se broja 13 plaši se mačaka plaši se visine

gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8

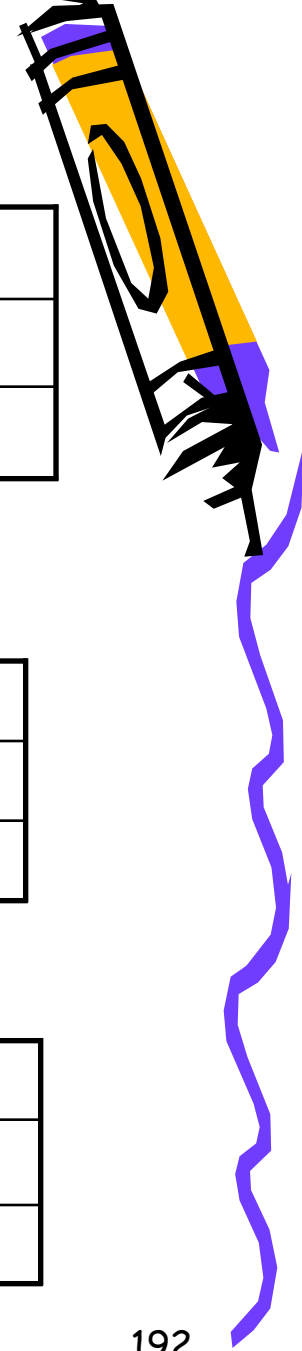


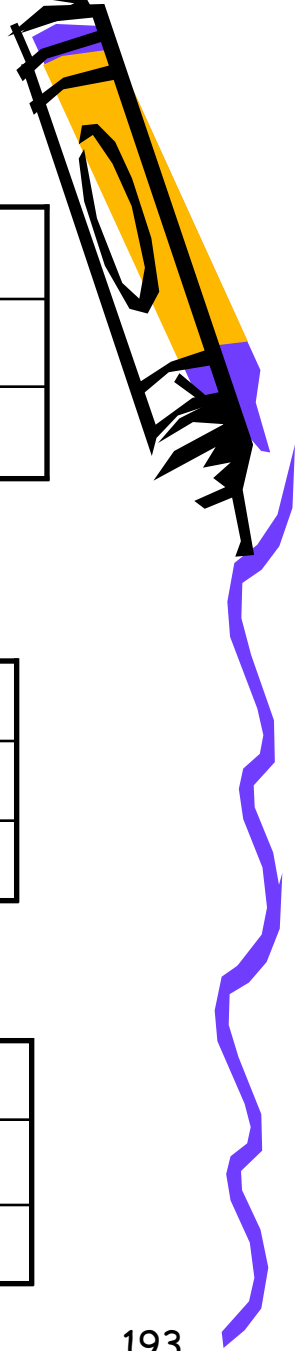
	gitarista	saksofonista	dobošar
Petar	NOT,1	NOT	YES
Pavle	YES	NOT,5	NOT
Jovan	NOT	YES	NOT

plaši se broja 13 plaši se mačaka plaši se visine

Petar	.		NOT,2
Pavle		NOT,4	
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8





	gitarista	saksofonista	dobošar
Petar	NOT,1	NOT	YES
Pavle	YES	NOT,5	NOT
Jovan	NOT	YES	NOT

plaši se broja 13 plaši se mačaka plaši se visine

Petar	NOT		NOT,2
Pavle		NOT,4	
Jovan			

	plaši se broja 13	plaši se mačaka	plaši se visine
gitarista	YES	NOT	NOT,3
saksofonista	NOT	NOT,6	YES
dobošar	NOT,7	YES	NOT,8

