

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Irena Oršolić

Samostalni studentski projekt
iz predmeta "Uvod u teoriju računarstva"

Zadatak broj 2034.

Zagreb, lipanj 2011.

Samostalni studentski projekt iz predmeta “Uvod u teoriju računarstva”

Student: Irena Oršolić

Matični broj studenta: 0036461463

Zadatak broj 2034.:

Jezik L nad abecedom $\{a, b, c, d\}$ čine prazan niz i svi mogući nizovi u kojima je broj znakova a i b te c i d jednak, odnosno $n_a = n_b$ i $n_c = n_d$. DKA $M(m)$ prihvaća niz iz jezika L ako za svaki prefiks niza vrijedi da je $|n_a - n_b| \leq m$ i $|n_c - n_d| \leq m$. Definirati (nije potrebno programski ostvariti) minimalne DKA $M(1)$, DKA $M(2)$ i DKA $M(3)$. Odrediti broj stanja minimalnog DKA $M(m)$ u ovisnosti o parametru m .

Uvod

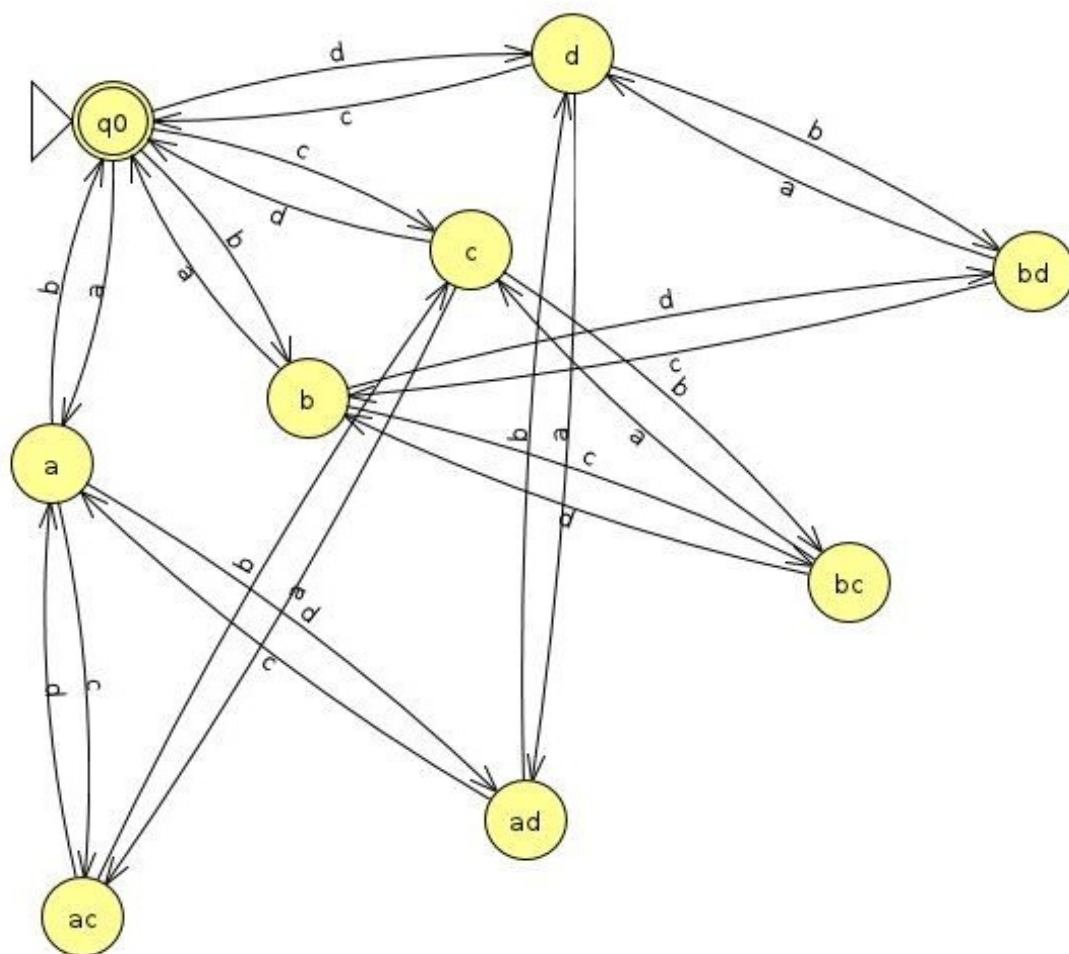
Zadatak ovoga projekta bio je konstruirati DKA(m) (gdje je $m=1,2$ i 3) koji prihvaćaju nizove koji zadovoljavaju sljedeća dva uvjeta:

1. broj znakova a i b te c i d je jednak
2. za svaki prefiks niza vrijedi da je $|n_a - n_b| \leq m$ i $|n_c - n_d| \leq m$

U ostvarenju ovoga projekta koristila sam programski alat JFLAP dostupan na <http://www.cs.duke.edu/csed/jflap/>

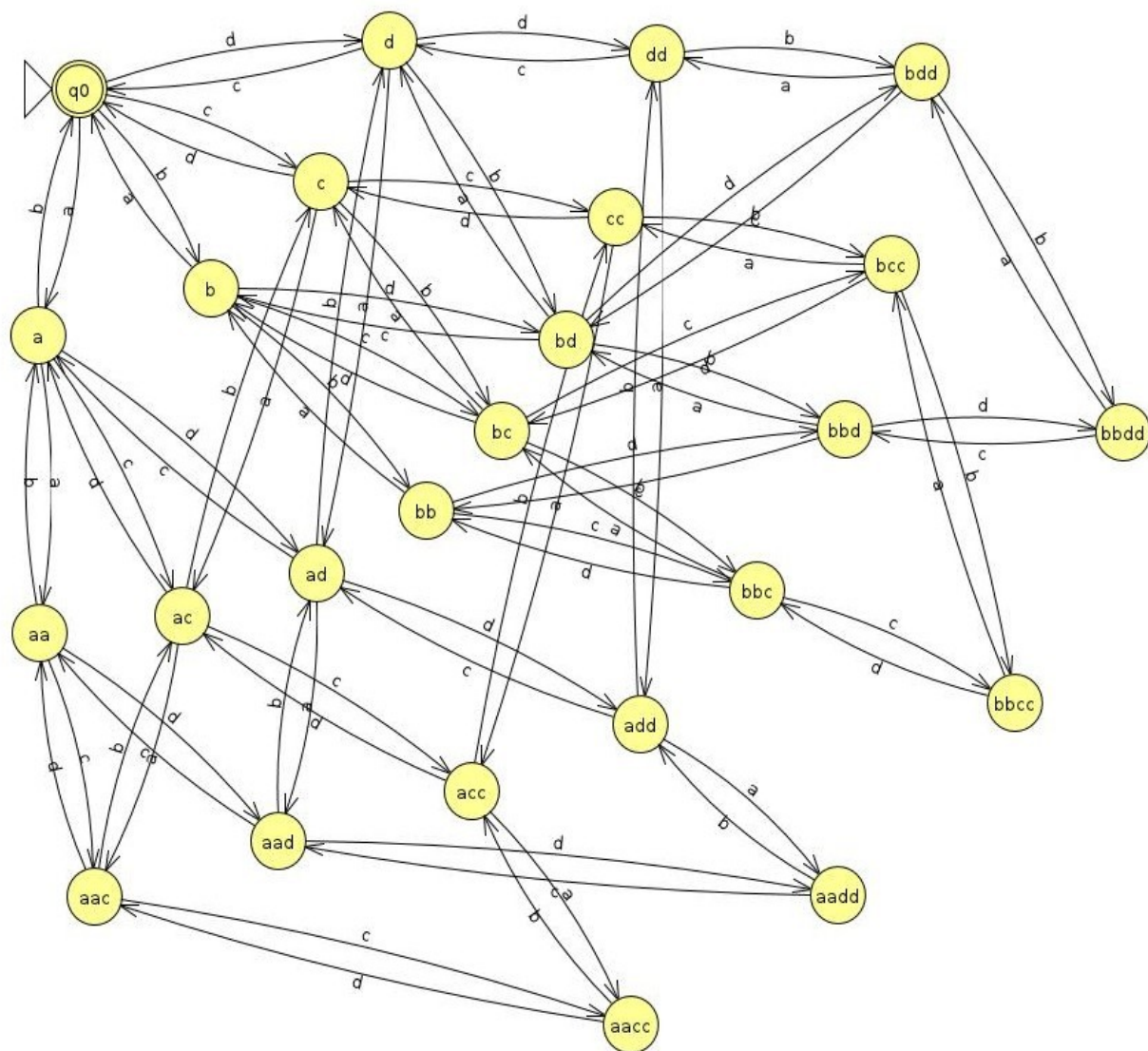
Ostvarenje

Krenut ću od DKA(1). Iz početnog stanja q_0 , čitanjem znaka na ulazu, automat može prijeći u 4 moguća stanja, ovisno o tome koji znak pročita (a , b , c ili d). Primjera radi, automat je pročitao znak " a ". Postoje opet 4 mogućnosti za znak koji može slijedeći pročitati. Ukoliko pročita " a " odlazi u stanje iz kojeg nikako ne može doći u prihvatljivo stanje, koji god znak slijedeći pročitao. Neka automat čita niz " $aabb$ ". Niz zadovoljava uvjet $n_a = n_b$, ali ne i drugi uvjet, jer je u za prefiks " aa " $|n_a - n_b| = |2 - 0| > 1$. Dakle, ponovno čitanje " a " dovesti će automat u stanje u kojem ostaje bez obzira koje znakove dalje čitao. Na shemama je ovo stanje ignorirano zbog preglednosti modela. Ukoliko automat nakon " a " pročita " b " vraća se u q_0 , jer se niz " ab " prihvaća, ali i zbog toga što se čitanjem " b " nakon " a " razlika između broja znakova a i b (i c i d) vraća u 0, a upravo to je ono što karakterizira inicijalno stanje ovog automata. Ako je nakon " a " pročitao znak " c ", automat prelazi u stanje označeno sa " ac ". Iz njega može, primjerice, čitanjem znaka " b " prijeći u stanje kakvo bi imao da je pročitao samo znak " c ". Kad bi se nakon " a " pročitao znak " d ", situacija bi bila identična, samo se radi o drugim znakovima. Ovom logikom može se izgraditi čitav automat. Slijedi grafički prikaz izrađen u JFLAP-u.

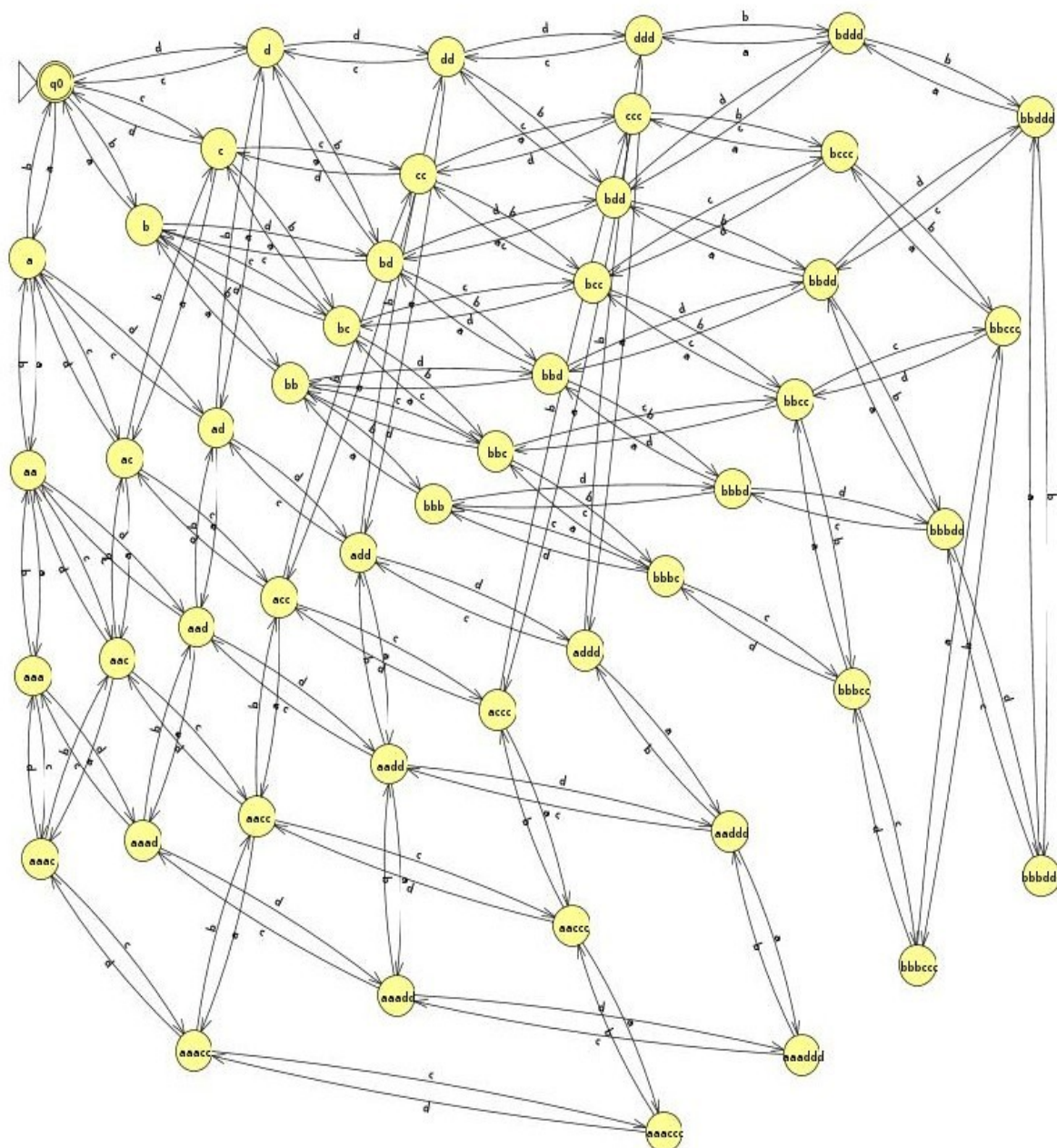


minimalni DKA(1)

Na isti način ostvarila sam i DKA(2) te DKA(3).



minimalni DKA(2)



minimalni DKA(3)

Minimizaciju DKA-ova sam također napravila pomoću JFLAP-a, ali se pokazalo da sam već napravila minimalne automate pa sam ostavila svoje sheme kako bih na njima mogla lakše pokazati kako sam uočila pravilnost u broju stanja u ovisnosti o parametru m . Automate sam radila u razinama koje predstavljaju broj učitanih znakova. Za DKA(1) broj razina je 2, DKA(2) ima 4 razine, a DKA(3) 6. Uočila sam da je broj stanja DKA(1) $1+4+4$, kod DKA(2) $1+4+8+8+4$, a kod DKA(3) $1+4+8+12+12+8+4$. To sam poopćila kao $1+4*2(1+2+...+m)$, tj. $1+(m*(m+1))/2$ gdje je 1 početno stanje, a m parametar zadan u zadatku.

Zaključak

U ovom seminarskom radu ostvarila sam DKA koji prihvaća nizove iz zadanog jezika koji ispunjavaju zadani uvjet. Na samome početku, svoj zadatak samo počela ostvarivati uz pomoć olovke i papira što se već za DKA(2) pokazalo izrazito kompleksnim. U potrazi za alatom uz pomoću kojega bih si olakšala posao, a na preporuku kolega, downloadala sam JFLAP. Uz pomoć njega sam uočila pravilnosti i istovjetnost stanja, što je na papiru bilo gotovo nemoguće.

Uz ovu dokumentaciju o izradi SSP-a priložene su sheme u .gif i u .jff formatu.