

Interpretacija programa

30. lipnja 2017.

Ovo je "open book" kolokvij. Dozvoljeno je korištenje bilo kakvih materijala — bilješke s vježbi/predavanja, vlastiti USBovi s riješenim zadacima, Python help, tutoriali, postovi na online forumima,... — **nastalih prije** kolokvija (npr. dozvoljeno je na *StackOverflowu* naći rješenje nekog zadatka, ali nije dozvoljeno tamo postaviti pitanje kako se rješava neki zadatak). Također, nije dozvoljena komunikacija (razgovor, *chat*, razmjena papira, USBova, ili bilo kakvih materijala) **među** studentima.

Na <http://ted.math.hr:54327> nalazi se zadnja verzija biblioteke `pj.py`, te predložak datoteke `pj_digisklop.py`. Rješenje zadatka pišite **isključivo** u lokalnu kopiju datoteke `pj_digisklop.py`, te na kraju tu datoteku *uploadajte* na <http://ted.math.hr:54327>. Odgovore na pitanja koja se nalaze u zadatku pišite u komentare unutar koda.

Datoteka `pj_digisklop.py`, uključivo sa svim testovima koji su inicijalno u njoj, treba se moći izvršiti u Pythonu bez grešaka. Ako imate neki kod koji po Vašem mišljenju pokazuje ideju rješenja, ali iz nekog razloga ne radi, napišite ga u komentar. Korisno je u datoteku uključiti još neke testove pored onih obaveznih.

Maksimalno vrijeme rješavanja je 180 minuta. U kolokviju je moguće osvojiti 35 bodova plus 5 bonus bodova (bonus bodovi ne ulaze u minimum za prolaz).

Digitalni sklopovi opisuju se logičkim izrazima. Njihovi ulazi opisuju se slovima. Tri su elementarna logička sklopa (*vrata*): NOT (ima točno jedan ulaz), te AND i OR (imaju dva ili više ulaza).

Sami logički izrazi, pored slova koja označavaju ulaze, imaju još neke simbole koji kažu kako se ulazi slažu u sklopove. $a+b+c+d$ označava OR vrata s 4 ulaza, dok abc označava AND vrata s 3 ulaza. Apostrof označava negaciju: a' su NOT vrata s ulazom a . Oble zagrade $()$ označavaju zasebni sklop čiji izlaz direktno sudjeluje u većem sklopu, dok uglate zagrade $[]$ označavaju zasebni sklop čiji izlaz je negiran (drugim riječima, $[sklop]$ je pokratak za $(sklop)'$). Primjerice, $xy'+x'y$ označava XOR, dok $[xy]$ označava NAND sklop s ulazima x i y .

Logički sklopovi često se realiziraju pomoću samo jedne vrste tzv. *univerzalnih* NAND vrata (s jednim ili više ulaza). Elementarna vrata je lako realizirati: NOT je jednostavno NAND s jednim ulazom, AND je kompozicija od NOT i NAND, dok se OR može dobiti de Morganovim pravilom. Na tako dobivenim NAND-sklopovima često se provode brojne optimizacije: najjednostavnija je eliminacija dvostruke negacije, gdje se $[[x]]$ zamjenjuje sa x .

[5b] Napišite leksički analizator za jezik DS digitalnih sklopova. Zanimarite praznine u ulaznom stringu. [5b] Potrebne tipove tokena odredite sami.

Napišite [5b] beskontekstnu gramatiku i [10b] sintaksni analizator za jezik DS. Apstraktna sintaksna stabla neka odgovaraju elementarnim vratima: `And(ulazi)`, `Or(ulazi)`, te `Not(ulaz)`. [-5b: Smijete simulirati npr. AND s 5 ulaza pomoću 4 AND vrata s po 2 ulaza.]

[10b] Napišite semantički analizator u obliku NAND-realizatora za jezik DS. NAND vrata reprezentirajte običnim listama čiji elementi su slova (stringovi duljine 1) ili iste takve liste. Primjerice, $['x', 'y', ['x']]$ je sklop od dva NAND vrata: jedna imaju 1 ulaz, x , dok druga imaju 3 ulaza: x , y i izlaz iz prvih (koji je negacija od x).

[bonus 5b] Napišite optimizator NAND-sklopova koje proizvodi realizator iz prošlog odlomka, koji pronalazi i uklanja sve dvostruke negacije iz sklopa. Primjerice, $[[[[['x'], [['y']], [['x']]]]]]$ treba pretvoriti u gornji primjer.