

AUTOMATSKO REZONOVANJE – KOLOKVIJUM – APRIL 2016

1. (25%) Dokazati (u *Isabelle*-u) da je formula:

$$(((p \Rightarrow q) \vee (\neg r \wedge s)) \wedge (s \Rightarrow ((p \wedge \neg q) \vee r)) \Rightarrow (p \Rightarrow q)) \Rightarrow ((\neg r \wedge s) \Rightarrow (p \Rightarrow q))$$

teorema prirodne dedukcije za klasičnu iskaznu logiku.

2. (50%) U programskom jeziku *C/C++*:

- a) omogućiti predstavljanje CNF iskaznih formula (u obliku skupa skupova literala, ili nekako drugačije). Omogućiti prikaz na standardnom izlazu.
- b) implementirati funkciju *resolve*(c_1, c_2, p) koja primenjuje pravilo rezolucije nad klauzama c_1 i c_2 po literalu p (pretpostavka je da c_1 sadrži p , a c_2 sadrži $\neg p$). Ovim pravilom se dobija nova klauza r koja sadrži sve literalne iz obe polazne klauze, ali bez literala p iz c_1 i literala $\neg p$ iz c_2 (tj. $r = (c_1 \setminus p) \cup (c_2 \setminus \neg p)$). Funkcija treba da eliminiše duplikate iz dobijene klauze. Ukoliko je novodobijena klauza tautologija (tj. sadrži i neko iskazno slovo q i njegovu negaciju), tada funkcija vraća *false*, a u suprotnom vraća *true*.
- c) implementirati funkciju *eliminate*(f, p) koja uklanja iskazno slovo p iz CNF formule f tako što primenjuje pravilo rezolucije (koristeći funkciju *resolve* iz dela pod b) nad svakim parom klauza (c_1, c_2) takvim da $p \in c_1$ i $\neg p \in c_2$. Klauze koje se dobiju na ovaj način se dodaju u CNF formulu, a originalne klauze (dakle, sve one koje sadrže bilo p bilo $\neg p$) se brišu. Ostale klauze iz originalne formule (one koje ne sadrže ni p ni $\neg p$) ostaju u CNF formuli nepromenjene. Ako se rezolucijom dobije tautologija, tada se ta klauza ne dodaje u CNF formulu. Ako se rezolucijom dobije prazna klauza, funkcija vraća *false*, a u suprotnom vraća *true*.
- d) implementirati funkciju *checkSat*(f) koja za datu CNF formulu f eliminiše jednu po jednu promenljivu, koristeći funkciju *eliminate* iz dela pod c). Ako bilo koji poziv funkcije *eliminate* vrati *false*, tada i *checkSat* vraća *false* (formula je nezadovoljiva jer je izvedena prazna klauza). Ako se proces završi bez izvođenja prazne klauze, funkcija vraća *true* (formula je zadovoljiva).
- e) Napisati program koji kreira CNF formulu:

$$[[p, \neg q, r], [\neg p, \neg q], [q, p, \neg r], [q, \neg p, \neg r]]$$

a zatim za nju proverava da li je zadovoljiva, pozivom funkcije *checkSat* iz dela pod d).

3. (25%) Dat je pomerački trobitni registar (p, q, r) i *carry bit* c . Obavlja se sledeća operacija: najpre se sadržaj registra obrće u ogledalu, zatim se dobijeni sadržaj pomera u desno kroz *carry* (što znači da vrednost bita c dolazi na najviši bit u registru (umesto p), a najniži bit iz registra (bit r) se potiskuje u c), nakon čega se sadržaj registra ponovo okrene u ogledalu. Dokazati da se nakon 4 primene ove složene operacije registar i *carry bit* vraćaju u početno stanje. Zadatak rešiti svođenjem na SAT. Dobijenu iskaznu formulu kodirati u *DIMACS* formatu, a zatim pokrenuti *minisat* rešavač.