

Tabele odlučivanja kroz primere

Seminarski rad u okviru kursa
Verifikacija softvera
Matematički fakultet

Andela Damnjanović, 59/2019
mi19059@alas.matf.bg.ac.rs

31.12.2023.

Sadržaj

1	Uvod	2
2	Tabele odlučivanja	2
2.1	Kreiranje tabela odlučivanja	2
2.2	Dobre i loše strane tabela odlučivanja	4
3	Primeri	4
3.1	Logovanje	4
3.2	Kredit	5
3.3	Katalozi	7
3.4	Otpremanje slika	10
3.5	Čekovi	12
4	Zaključak	15
	Literatura	15

1 Uvod

Pre nego što se softver pusti u upotrebu, potrebno ga je verifikovati. Najčešći način verifikacije softvera jeste testiranjem. Dobro odrađen proces testiranja doprinosi podizanju nivoa pouzdanosti softvera, dok istovremeno smanjuje prostor za grešku. Stoga je veoma važno biti upoznat sa načinima, vrstama i tehnikama testiranja [7].

Jedna od značajnijih tehnika testiranja softvera je tehnika testiranja crne kutije. Ona podrazumeva generisanje test primera bez razmatranja interne strukture koda. Dakle, odgovarajući test primeri se generišu isključivo na osnovu specifikacije. Kao što se iz navedenog opisa može pretpostaviti, testiranje crne kutije predstavlja vid testiranja softvera iz ugla korisnika [7].

Za potpunu verifikaciju softvera potrebno je proveriti sve moguće kombinacije ulaza. Međutim, čak i za naizgled jednostavne programe, ispitivanje svih mogućih slučajeva može se pokazati netrivialno. Stoga cilj testiranja crne kutije nije da generiše sve moguće test primere, već da se pronade neki podskup skupa testova koji će dobro reprezentovati skup svih testova [5]. Tabele odlučivanja, kojima je ovaj rad i posvećen, upravo predstavljaju jedan od načina za testiranje crne kutije. Stoga će sekcija 2 biti posvećena detaljnijem opisu tabela odlučivanja, načinu njihovog kreiranja i razmatranju dobrih i loših strana ovih tabela, nakon čega će u sekciji 3 biti demonstrirana njihova primena na odgovarajućim primerima.

2 Tabele odlučivanja

Kao što je i rečeno u uvodnom delu ovog rada, testiranje crne kutije podrazumeva validaciju sistema bez ikakvih predznanja o unutrašnjem dizajnu pomenutog sistema. Neke jednostavnije tehnike testiranja crne kutije, poput klase ekvivalencije i graničnih vrednosti, mogu da obrađuju samo jedan ulazni uslov u svakom trenutku. Međutim, postavlja se pitanje šta raditi kada se na ulazu nalazi više različitih uslova i radnji. Upravo zbog ovakvih slučajeva, razvijena je tehnika tabela odlučivanja [5, 6].

Kao što i samo ime kaže, ova tehnika predstavlja tabelarni prikaz ulaznih uslova i rezultujućih radnji. Iz nje se jasno mogu uočiti uzroci i posledice, te se stoga naziva još i tabela uzroka i posledica [5, 6]. Informacije koje se nalaze u tabeli odlučivanja zatim se koriste za generisanje test primera kako bi se proverilo da li sistem radi kako bi trebalo.

2.1 Kreiranje tabela odlučivanja

Sada, nakon upoznavanja sa pojmom i funkcijom tabela odlučivanja, nameće se pitanje kako se date tabele konstruišu. Pre kreiranja same tabele, najpre je potrebno prepoznati funkciju ili podsistem koji treba da reaguje na kombinaciju ulaznih uslova. Sam sistem ne bi trebalo da sadrži veliki broj ulaza, jer će onda posleđično i broj njihovih kombinacija postati prevelik, te ga neće biti moguće ispratiti. No, kako su realni sistemi daleko od idealnih teorijskih modela, često se dešava da je broj mogućih ulaza velik. Kada se to desi, preporučena strategija podrazumeva podelu svih uslova na podskupove. Nakon kreiranja odgovarajućih podskupova, oni se testiraju jedan po jedan. Kada se uoče aspekti

koje je potrebno ukombinovati, oni se unose u tabelu odlučivanja [6].

Sam proces pripreme može se podeliti na četiri osnovna koraka. Prvi korak podrazumeva prepoznavanje svih mogućih ulaznih vrednosti od kojih na kraju zavisi rezultat, dok je drugi korak posvećen prepoznavanju odgovarajućih akcija koje se mogu dogoditi u sistemu. Nakon što su svi uslovi i rezultujuće akcije poznati, u trećem koraku vrši se generisanje svih mogućih kombinacija uslova, dok se u finalnom, četvrtom koraku, na osnovu svake pojedinačne kombinacije utvrđuje odgovarajuća akcija [5].

Nakon što su svi preduslovi zadovoljeni, može se pristupiti pravljenju pomenute tabele. Izgled šablona za kreiranje tabele odlučivanja može se naći na slici 2.1. Kao što se sa date slike može videti, gornji redovi rezervisani su za informacije o ulaznim uslovima dok donji redovi tabele sadrže podatke o rezultujućim akcijama. Takođe se može uočiti da se spiskovi svih mogućih uslova i akcija (dobijenih iz prva dva koraka) nalaze skroz levo u tabeli, dok je ostatak tabele rezervisan za nabranje kombinacija i rezultujućih akcija (dobijenih u poslednja dva koraka).

Spisak uslova	Ulazne vrednosti
Spisak akcija	Rezultujuće vrednosti

Slika 1: Izgled šablona za tabele odlučivanja

Sada, nakon upoznavanja sa izgledom i načinima kreiranja tabele, potrebno je opisati notaciju koja se koristi. Što se tiče ulaznih uslova, ukoliko su binarni, oni mogu biti ispunjeni ili neispunjeni. Za označavanje je uslov ispunjen, u zavisnosti od preferencija, mogu se koristiti sledeće oznake: **Y**, **T** ili **1**. Sa druge strane, ako uslov nije ispunjen, to se označava sa: **N**, **F** ili **0**. Ukoliko za krajnji ishod nije bitno da li je ulazni uslov ispunjen ili nije mogu se koristiti oznake - ili **N/A** [6]. Treba pomenuti da, ukoliko ulazni uslov nije binarni nego recimo numerički, onda će se u tabeli naći brojevi. Slično važi i za kategoričke vrednosti, tj. ako je ulazni uslov kategorički u tabeli će se naći odgovarajuća oznaka kategorije.

Analogna pravila važe za akcije. Ukoliko akcija treba da se desi, pored već pomenutih oznaka **Y**, **T** i **1** može se koristiti i **X**, dok u slučaju da akcija ne treba da se izvrši pored postojećih oznaka **N**, **F** i **0** takođe može samo ostaviti prazno polje [6].

Za potrebe ovog rada, biće korišćena sledeća notacija: za ispunjene uslove i akcije koje treba da se izvrše biće korišćena oznaka **Y**, za neispunjene uslove **N**, za uslove od kojih rezultat ne zavisi -, a za akcije koje se ne dešavaju, biće ostavljena prazna polja u tabeli.

2.2 Dobre i loše strane tabela odlučivanja

U uvodnom poglavlju 1 je pomenuto da su tabele odlučivanja korisne kada treba ispitati više ulaznih uslova istovremeno. No, to nije jedina pozitivna strana korišćenja tabela odlučivanja. Pored navedenog, tabele odlučivanja odlikuju i jednostavnost (veoma ih je lako napraviti i koristiti, ali i naučiti, što je takođe jako bitno), jasan način obeležavanja, kao i lako identifikovanje uzročno-posledičnih veza. Zbog svoje preciznosti i konciznosti veoma se često sreću i u dokumentaciji. Takođe, tabele odlučivanja olakšavaju komunikaciju između ljudi (pogotovo između testera i programera), olakšavaju vizualizaciju ponašanja sistema i bolje koriste podrutine sistema [4].

Međutim, i pored svih pozitivnih strana, ni tabele odlučivanja nisu svemoćne. Najveći problem jeste njihov eksponencijalni rast (po broju ulaznih uslova) što znači da tabele vrlo brzo postaju ogromne. Samim tim, testiranje svih kombinacija često može biti nepraktično ili nemoguće [4]. U ovakvim situacijama potrebno je odrediti podskup kombinacija koji će biti testiran, što nije lako uzimajući u obzir da upravo od izabranog podskupa zavisi kvalitet testiranja [6, 5]. Može se i desiti da je potrebno i više od jedne tabele odlučivanja za bi se logika mogla dokumentovati [4].

Još jedan problem na koji treba obratiti pažnju, a koji je demonstriran u primerima 3.3 i 3.5 jeste problem redundantnosti kombinacija. Iako su navedeni primeri mali po obimu, pa se ne čini da je redundantnost veliki propust, ovo jeste veoma ozbiljan nedostatak koji će u velikim sistemima napraviti velike probleme. Kao rešenje nameće se sledeći pristup: nije potrebno testirati svaku kombinaciju, već svaku varijaciju [6]. Treba napomenuti da danas postoje i softverski programi koji mogu da otkriju da li se u tabeli odlučivanja nalaze redundantnosti, kao i da otkriju da li postoje slučajevi koji tabelom nisu pokriveni [4].

Na kraju, čak iako su gorenavedeni problemi rešeni, i dalje treba obratiti pažnju na granične slučajeve. Ukoliko su ulazni podaci numerički, treba obratiti pažnju na sve slučajeve koji se nalaze na granici između dva slučaja i utvrditi unapred kom slučaju će biti pripojen dati granični slučaj. Za rešavanje ovog problema, mogu se, uz tabele odlučivanja, koristiti i druge tehnike za testiranje crne kutije kao što su već pomenuto particionisanje i klase ekvivalencije [6].

3 Primeri

Ova sekcija će biti posvećena demonstraciji korišćenja tabela odlučivanja kroz primere. Svaki primer će najpre biti opisan, a zatim će se, kroz korake opisane u sekciji 2.1, dobiti sve informacije potrebne za kreiranje tabela odlučivanja. Najzad, biće prikazane rezultujuće tabele. Primeri koji će biti obrađeni su: primer sa ekranom za logovanje, primer sa odobravanjem kredita i čekova, primer sa otpremanjem slika, te primer sa katalogima.

3.1 Logovanje

Najpre će biti razmotren sledeći primer: potrebno je napraviti tabelu odlučivanja koja će raditi sa ekranima za prijavljivanje. Ukoliko korisnik unese ispravno korisničko ime i lozinku prikazuje mu se profil, a ukoliko neki od podataka ne bude unesen ili ne bude validan ispisuje se poruka o grešci [3].

Korak 1. Najpre je potrebno odrediti sve moguće ulazne vrednosti. Na osnovu samog teksta zadatka može se zaključiti da su ulazni uslovi sledeći:

- Korisničko ime je ili korektno uneto ili nije i
- Lozinka je ili korektno uneta ili nije.

Korak 2. Zatim je potrebno identifikovati koje se akcije mogu desiti. Ovaj podatak se takođe može pročitati iz teksta zadatka. Moguće akcije su dakle prikazivanje korisničkog naloga ili ispis greške.

Korak 3. Nakon što je napravljen spisak svih mogućih ulaza, potrebno je generisati i sve kombinacije ulaznih uslova koje se mogu pojaviti na ulazu. Pošto primer nije komplikovan, moguće je napraviti spisak svih mogućih kombinacija. Kombinovanjem svih mogućih parova ulaza dobijamo sledeće:

- **kombiancija 1:** i korisničko ime i lozinka su korektno navedeni
- **kombiancija 2:** ni korisničko ime ni lozinka nisu korektno navedeni
- **kombiancija 3:** korisničko ime je korektno navedeno, dok lozinka nije i
- **kombiancija 4:** lozinka je dobro navedena, dok korisničko ime nije.

Korak 4. Sada je samo potrebno odrediti akcije koje odgovaraju svakoj od kombinacija. Za to će biti potrebni rezultati dobijeni u prethodnim koracima. Da bi se odredilo koju akciju treba pridružiti kojoj kombinaciji, potrebno je ponovo se vratiti na tekst zadatka. Iz samog teksta možemo zaključiti sledeće:

- **kombiancija 1:** -> prikazuje se korisnički nalog
- **kombiancija 2:** -> ispisuje se greška
- **kombiancija 3:** -> ispisuje se greška
- **kombiancija 4:** -> ispisuje se greška.

Sada je još potrebno oraganizovati dobijene podatke u tabelu. Svakoj koloni odgovara po jedan par ulaznih vrednosti, dok redovima u tabeli odgovaraju nazivi ulaznih uslova i njihove vrednosti ili nazivi akcija i informacija da li se data akcija dešava ili ne. Konačno rešenje je prikazano u tabeli 1. Ono što je posebno zanimljivo u vezi tabela odlučivanja, a može se direktno videti u navedenoj tabeli (kao i u svim ostalim tabelama koje slede), jeste činjenica da su akcije međusobno isključive, tj. u svakoj koloni tačno je jedna akcija obeležena kao ona koja će se desiti.

3.2 Kredit

Nakon savladavanja osnova u primeru 3.1, sada će biti prikazan malo kompleksniji primer. Neka je potrebno razviti bankarsku aplikaciju koja će korisniku nakon unetih odgovarajućih ulaznih podataka prikazati da li je stekao uslov za odobravanje kredita ili ne, kao i da li je potrebno priložiti još dokumenata da bi ta odluka bila doneta. Informacije koje korisnik treba da unese su da li je zaposlen ili ne, kao i kolika su mu mesečna primanja [6]. Takođe, dat je sledeći spisak uslova na osnovu kojih se određuje rezultujuća akcija:

Tabela 1: Tabela odlučivanja na primeru ekrana za logovanje

	U1	U2	U3	U4
Uslovi				
Korektno korisničko ime	Y	Y	N	N
Korektna lozinka	Y	N	Y	N
Akcije				
Prikaz korisničkog naloga	Y			
Informacija o greški		Y	Y	Y

- Ukoliko je klijent zaposlen i mesečna primanja su veća ili jednaka 75.000 dolara, kredit se odobrava,
- Ukoliko je klijent zaposlen i mesečna primanja su između 25.000 i 75.000, onda je potrebno priložiti dodatne dokumente,
- Ukoliko je klijent zaposlen i ima manja primanja od 25.000, onda će zahtev biti odbijen,
- Ukoliko klijent nije zaposlen i ako su mu mesečna primanja veća ili jednaka 75.000, onda je potrebno priložiti dodatne dokumente i
- Ukoliko klijent nije zaposlen i ako su mu mesečna primanja manja od 75.000, zahtev za kredit biće odbijen.

Korak 1. Kao i u primeru 3.1 prvo je potrebno odrediti sve moguće ulazne vrednosti. Ponovo, na osnovu samog teksta zadatka može se zaključiti da su ulazni uslovi sledeći:

- Klijent je zaposlen ili nezaposlen
- Mesečna primanja klijenta su < 25.000 ili nisu
- Mesečna primanja klijenta su $25.000-75.000$ ili nisu i
- Mesečna primanja klijenta su > 75.000 ili nisu.

Korak 2. Zatim je potrebno identifikovati skup akcija koje se mogu desiti. Moguće akcije su odobravanje prava na kredit, odbijanje kredita i zahtev za dostavu dodatne dokumentacije.

Korak 3. Sada je potrebno generisati sve kombinacije ulaznih uslova. Pošto ni ovaj primer nije komplikovan, i sada je moguće napraviti spisak svih mogućih kombinacija. Kombinovanjem svih mogućih parova ulaza dobijamo sledeće:

- **kombiancija 1:** Klijent je zaposlen, sa platom manjom od 25.000
- **kombiancija 2:** Klijent je zaposlen, sa platom iz opsega 25.000-75.000
- **kombiancija 3:** Klijent je zaposlen, sa platom ≥ 75.000
- **kombiancija 4:** Klijent je nezaposlen, sa platom manjom od 25.000
- **kombiancija 5:** Klijent je nezaposlen, sa platom iz opsega 25.000-75.000 i

- **kombiancija 6:** Klijent je nezaposlen, sa platom ≥ 75.000

Korak 4. Ponovo, potrebno je odrediti akcije koje odgovaraju svakoj od kombinacija. Za to će biti potrebni rezultati dobijeni u prethodnim koracima. Da bi se odredilo koju akciju treba pridružiti kojoj kombinaciji, potrebno je ponovo se vratiti na tekst zadatka. Ponovnim čitanjem teksta izvode sledeći zaključci:

- **kombiancija 1:** -> Zahtev odbijen
- **kombiancija 2:** -> Potrebna dodatna dokumentacija
- **kombiancija 3:** -> Zahtev prihvaćen
- **kombiancija 4:** -> Zahtev odbijen
- **kombiancija 5:** -> Zahtev odbijen
- **kombiancija 6:** -> Potrebna dodatna dokumentacija.

Nakon uspešnog izvršenja prethodnih koraka, ostalo je još samo napraviti odgovarajuću tabelu, te je konačno rešenje je prikazano u tabeli 2. Kao što se iz priložene tabele može videti, bez obzira na ulaznu kombinaciju, uvek se dešava samo jedna akcija. Takođe treba napomenuti da bi ovde moglo da dođe do problema sa graničnim vrednostima, te je veoma bitno znati u koji slučaj treba smestiti granični slučaj jer od toga zavisi krajnja akcija koja se dešava.

Tabela 2: Tabela odlučivanja na primeru zahteva za dobijanje kredita

	U1	U2	U3	U4	U5	U6
Uslovi						
Plaćen?	Y	Y	Y	N	N	N
<25.000	Y	N	N	Y	N	N
25-75.000	N	Y	N	N	Y	N
>75.000	N	N	Y	N	N	Y
Akcije						
Odobreno	N	N	Y	N	N	N
Odbijeno	Y	N	N	Y	Y	N
Dodatak	N	Y	N	N	N	Y

3.3 Katalozi

Danas svi najveći proizvođači imaju razvijene sisteme preporuka. Cilj ovih sistema jeste da prate preferencije klijenata i da im, na osnovu njih, preporuče slične artikle. Neka je zadatak razviti sistem koji će slično funkcionisati. Tačnije, na osnovu informacije iz kog je kataloga klijent u prošlosti naručivao artikle, sistem treba da zaključi koji od dostupnih kataloga treba da pošalje klijentu kako bi ga zainteresovao za još jednu kupovinu. Artikli se mogu naručiti iz jesenjeg, Božićnog ili specijalnog kataloga. Sistem treba da utvrdi da li je klijentu potrebno poslati ovogodišnji Božićni ili specijalni katalog ili oba [2]. Za odabir kataloga koje treba poslati, koriste se sledeća pravila:

- Ako je klijent naručivao iz sva tri kataloga, šalju se i Božićni i specijalni katalog
- Ako je klijent naručivao iz jesenjeg i Božićnog kataloga, šalje se samo Božićni katalog
- Ako je klijent naručivao iz jesenjeg i specijalnog kataloga, šalje se samo specijalni katalog
- Ako je klijent naručivao samo iz jesenjeg kataloga, šalje se samo Božićni katalog
- Ako je klijent naručivao samo iz Božićnog kataloga, šalje se samo Božićni katalog
- Ako je klijent naručivao samo iz specijalnog kataloga, šalje se samo specijalni katalog
- Ako je klijent naručivao iz specijalnog i Božićnog kataloga, šalju se oba kataloga
- Ako klijent nije naručivao ni iz jednog od pomenutih kataloga, šalje se samo Božićni katalog

Korak 1. Ponovo je prvi korak određivanje svih mogućih ulaznih vrednosti. Na osnovu samog teksta zadatka može se zaključiti da su ulazni uslovi sledeći:

- Klijent jeste ili nije naručio artikal iz Božićnog kataloga
- Klijent jeste ili nije naručio artikal iz specijalnog kataloga
- Klijent jeste ili nije naručio artikal iz jesenjeg kataloga

Korak 2. Nakon pronalaženja mogućih ulaznih vrednosti, potrebno je identifikovati skup akcija koje se mogu desiti. Ovaj podatak se takođe može pročitati iz teksta zadatka. Moguće akcije su: slanje Božićnog kataloga, slanje specijalnog kataloga ili slanje oba kataloga.

Korak 3. Nakon što je napravljen spisak svih mogućih ulaza, potrebno je generisati i sve kombinacije ulaznih uslova. Iako kompleksniji od dosadašnjih, ni ovaj primer nije previše komplikovan, pa je moguće napraviti spisak svih mogućih kombinacija. Uparivanjem svih mogućih ulaza dobijamo sledeće:

- **kombiancija 1:** Klijent je naručivao artikle iz svakog od kataloga
- **kombiancija 2:** Klijent je naručivao iz Božićnog i jesenjeg kataloga
- **kombiancija 3:** Klijent je naručivao iz specijalnog i jesenjeg kataloga
- **kombiancija 4:** Klijent je naručivao samo iz jesenjeg kataloga
- **kombiancija 5:** Klijent je naručivao iz Božićnog i specijalnog kataloga
- **kombiancija 6:** Klijent je naručivao samo iz Božićnog kataloga
- **kombiancija 7:** Klijent je naručivao samo iz specijalnog kataloga
- **kombiancija 8:** Klijent nije naručivao ni iz jednog od ponuđenih kataloga.

Korak 4. Na kraju, potrebno je odrediti akcije koje odgovaraju svakoj od kombinacija. Za to će, kao i do sad, biti potrebni rezultati dobijeni u prethodnim koracima. Da bi se odredilo koju akciju treba pridružiti kojoj kombinaciji, potrebno je ponovo se vratiti na tekst zadatka. Tada se izvode sledeći zaključci:

- **kombiancija 1:** -> Poslati oba kataloga
- **kombiancija 2:** -> Poslati Božićni katalog
- **kombiancija 3:** -> Poslati specijalni katalog
- **kombiancija 4:** -> Poslati Božićni katalog
- **kombiancija 5:** -> Poslati oba kataloga
- **kombiancija 6:** -> Poslati Božićni katalog
- **kombiancija 7:** -> Poslati specijalni katalog
- **kombiancija 8:** -> Poslati Božićni katalog

Rešenje se može naći u tabeli 3. Međutim, ako se dobijena tabela pažljivije prouči, može se doći do zaključka da se u njoj nalaze i **redundantnosti**. Na primer, ako se pogledaju ulazi U1 i U5 može se zaključiti da ukoliko je klijent naručivao i iz Božićnog i iz specijalnog kataloga, onda nije bitno da li je ili ne naručivao artikle iz jesenjeg kataloga, rezultat je isti. Stoga se ova dva slučaja mogu stopiti u jedan. Dalje, ako se uporede ulazi U2 i U6 može se zaključiti da će mu, ukoliko je klijent naručivao artikle iz Božićnog kataloga ali ne i iz specijalnog, biti poslat Božićni katalog bez obzira da li je naručivao iz jesenjeg ili ne, tako da se i ova dva uslova mogu stopiti u jedan. Na analogan način, mogu se spojiti parovi ulaza: U3 i U7 ¹ i U4 i U8 ².

Tabela 3: Tabela odlučivanja na primeru slanja kataloga

	U1	U2	U3	U4	U5	U6	U7	U8
Uslovi								
Jesenji	Y	Y	Y	Y	N	N	N	N
Božićni	Y	Y	N	N	Y	Y	N	N
Specijalni	Y	N	Y	N	Y	N	Y	N
Akcije								
Pošalji Božićni	N	Y	N	Y	N	Y	N	Y
Pošalji specijalni	N	N	Y	N	N	N	Y	N
Pošalji oba	Y	N	N	N	Y	N	N	N

Dakle, tabela se može uprostiti. Za postizanje željenih rezultata, potrebno je stopiti uslove U1 i U5 u novi uslov koji će biti nazvan U1', uslove U2 i U6 u nov uslov U2', uslove U3 i U7 u nov uslov U3' i, na kraju, uslove U4 i U8 u nov uslov U4'. Rezultat ovakvog uprošćavanja dat je u tabeli 4.

¹jer ukoliko je naručeno iz specijalnog kataloga, ali ne i Božićnog izbor ostaje isti bez obzira da li je klijent naručivao iz jesenjeg kataloga ili ne

²ponovo, ukoliko klijent nije naručivao ni iz Božićnog ni iz specijalnog kataloga rezultat je isti bez obzira da li je naručivao iz jesenjeg kataloga ili ne

Tabela 4: Pojednostavljena tabela odlučivanja na primeru slanja kataloga

	U1'	U2'	U3'	U4'
Uslovi				
Jesenji	-	-	-	-
Božićni	Y	Y	N	N
Specijalni	Y	N	Y	N
Akcije				
Pošalji Božićni	N	Y	N	Y
Pošalji specijalni	N	N	Y	N
Pošalji oba	Y	N	N	N

Međutim, posao još nije završen jer ako se i tabela 4 pažljivije pogleda, može se uočiti da i ona sadrži redundantnosti. Naime, ako se pogledaju uslovi U2' i U4' može se videti da krajnji rezultat zapravo ne zavisi od toga da li je klijent naručivao iz Božićnog kataloga ili nije, pa se i ova dva uslova mogu spojiti u jedan — U2". Najzad, konačno rešenje ovog problema dato je u tabeli 5.

Tabela 5: Konačna tabela odlučivanja na primeru slanja kataloga

	U1'	U2"	U3'
Uslovi			
Jesenji	-	-	-
Božićni	Y	-	N
Specijalni	Y	N	Y
Akcije			
Pošalji Božićni	N	Y	N
Pošalji specijalni	N	N	Y
Pošalji oba	Y	N	N

3.4 Otpremanje slika

Neka je sada potrebno napraviti softver čija će uloga biti da prima slike koje klijent želi da otpremi. Međutim, da bi slika mogla biti otpremljena potrebno je da bude u .jpg formatu, da bude manja od 32kB i da joj rezolucija bude 137*177. Ukoliko neki od uslova nije ispunjen, ispisuje se poruka koja sadrži informaciju o tome koji uslovi nisu ispunjeni [3].

Korak 1. Kao i do sad, najpre se određuje spisak mogućih ulaznih vrednosti. Mogući ulazi su:

- Klijent jeste ili nije prosledio sliku u .jpg formatu
- Klijent jeste ili nije prosledio sliku manju od 32kB
- Klijent jeste ili nije prosledio sliku dimanzija 137*177

Korak 2. Dalje, potrebno je identifikovati skup akcija koje se mogu desiti. Moguće akcije su:

- Slika je otpremljena
- Slika nije otpremljena
- Ispisuje se poruka o grešci zbog neodgovarajućeg formata
- Ispisuje se poruka o grešci zbog neodgovarajuće veličine
- Ispisuje se poruka o grešci zbog neodgovarajućih dimenzija
- Ispisuje se poruka o grešci zbog neodgovarajućeg formata i neodgovarajuće veličine
- Ispisuje se poruka o grešci zbog neodgovarajućeg formata i neodgovarajućih dimenzija
- Ispisuje se poruka o grešci zbog neodgovarajuće veličine i neodgovarajućih dimenzija

Korak 3. U sledećem koraku potrebno je generisati i sve kombinacije ulaznih uslova. I u ovom slučaju je moguće napraviti spisak svih mogućih kombinacija:

- **kombiancija 1:** Klijent je poslao sliku koja zadovoljava sve uslove
- **kombiancija 2:** Klijent je poslao sliku koja nije odgovarajućih dimenzija
- **kombiancija 3:** Klijent je poslao sliku koja nije odgovarajuće veličine
- **kombiancija 4:** Klijent je poslao sliku koja nije odgovarajućih dimenzija ni odgovarajuće veličine
- **kombiancija 5:** Klijent je poslao sliku koja nije odgovarajućeg formata
- **kombiancija 6:** Klijent je poslao sliku koja nije odgovarajućih dimenzija ni odgovarajućeg formata
- **kombiancija 7:** Klijent je poslao sliku koja nije odgovarajuće veličine ni odgovarajućeg formata
- **kombiancija 8:** Klijent je poslao sliku koja ne ispunjava nijedan od zahtevanih uslova.

Korak 4. Na kraju, svakoj kombinaciji je potrebno pridružiti odgovarajuću akciju:

- **kombiancija 1:** -> Slika je uspešno otpremljena
- **kombiancija 2:** -> Poruka o neodgovarajućim dimenzijama slike
- **kombiancija 3:** -> Poruka o neodgovarajućoj veličini
- **kombiancija 4:** -> Poruka o neodgovarajućoj veličini i o neodgovarajućim dimenzijama
- **kombiancija 5:** -> Poruka o neodgovarajućem formatu
- **kombiancija 6:** -> Poruka o neodgovarajućem formatu i neodgovarajućim dimenzijama

- **kombiancija 7:** -> Poruka o neodgovarajućem formatu i neodgovarajućoj veličini
- **kombiancija 8:** -> Poruka o neodgovarajućem formatu, veličini i dimanzijama

Ostalo je samo još smestiti podatke u tabelu. Međutim, sličnost ovog primera sa primerom 3.3 navodi na razmišljanje da li i u ovom slučaju postoje redundantnosti. Odgovor na ovo pitanje je odričan. Naime, dovoljno je pogledati u spisak akcija pridruženim kombinacijama i zaključiti da se svaka akcija javlja u tačno jednom slučaju, te je samim tim nemoguće da tabela sadrži bilo kakve redundantnosti. Stoga, tabela 6 zaista predstavlja konačno rešenje. Treba napomenuti da oznaka E u tabeli označava da je došlo do greške, a sve što sledi iza slova jeste spisak neispunjenih uslova.

Tabela 6: Tabela odlučivanja na primeru otpremanja slika								
	U1	U2	U3	U4	U5	U6	U7	U8
Uslovi								
.jpg format	Y	Y	Y	Y	N	N	N	N
<32kB	Y	Y	N	N	Y	Y	N	N
137*177	Y	N	Y	N	Y	N	Y	N
Akcije								
Otpremanje slike	Y	N	N	N	N	N	N	N
E: dimenzije	N	Y	N	N	N	N	N	N
E: format	N	N	N	N	Y	N	N	N
E: veličina	N	N	Y	N	N	N	N	N
E: veličina i format	N	N	N	N	N	N	Y	N
E: dimenzije i veličina	N	N	N	Y	N	N	N	N
E: dimenzije i format	N	N	N	N	N	Y	N	N
E: dimenzije, veličina i format	N	N	N	N	N	N	N	Y

3.5 Čekovi

Kao što se može videti, do sada su svi ulazni uslovi za tabele odlučivanja mogli nesmetano da stanu na jednu stranu lista formata A4. Naravno, u realnim sistemima izuzetno retko se dešava da je broj ulaznih stanja ovako mali. Zato će biti razmotren nešto obimniji primer. Potrebno je napraviti sistem koji će da prihvata ili odbija čekove [1]. Ulazni uslove čine sledeće informacije:

- vrednost čeka (manje ili više od 100 dolara)
- da li klijent poseduje vozačku dozvolu ili ne
- da li klijent ima garantnu karticu za ček
- da li je ček dvopartijski
- da li klijent označen kao "preferirani kupac".

Ovih pet binarnih uslova proizvešće na kraju $2^5=32$ moguće ulazne kombinacije. To je čak četiri puta više kombinacija nego što je do sada bilo razmatrano. Naravno, računaru neće biti problem da reši ni ovoliki broj slučajeva, ali, kako broj mogućih kombinacija eksponencijalno raste, vrlo brzo će ogroman broj kombinacija postati problem čak i za računar. Stoga je potrebno pronaći način da se, kao u primeru 3.3, broj ulaznih kombinacija što je moguće više smanji. No, pre preduzimanja bilo čega, potrebno je poznavati uslove pri kojima je ček prihvaćen ili odbijen. Za prihvatanje čeka neophodno je da je ispunjen neki od sledećih uslova:

- klijent ima garantnu karticu i vozačku dozvolu
- ček je na manje od 100 dolara i klijent pokazuje vozačku dozvolu ili garantnu karticu
- ček je na manje od 100 dolara i klijent je "preferirani kupac"
- klijent je "preferirani kupac" i ček je dvopartijski. U svim ostalim

slučajevima ček je odbijen.

Korak 1. Ponovo je najpre potrebno odrediti moguće ulazne vrednosti:

- Ček jeste ili nije na manje od 100 dolara
- Klijent poseduje ili ne poseduje vozačku dozvolu
- Klijent ima ili nema garantnu karticu
- Ček jeste ili nije dvopartijski
- Klijent jeste ili nije označen kao "preferirani kupac"

Korak 2. Zatim, potrebno je identifikovati skup akcija koje se mogu desiti. Moguće akcije su:

- Ček je prihvaćen
- Ček je odbijen

Korak 3. U ovom trenutku dolazi do problema — postoji preveliki broj kombinacija. Naravno, moguće je sesti i ispisati sve moguće kombinacije, no to bi bilo veoma nepraktično. Stoga je potrebno, kao što je i napomenuto gore u tekstu, naći način da se broj ulaznih kombinacija smanji. Postavlja se pitanje kako to uraditi. Možda se ovo pitanje može činiti teškim, no odgovor je ipak vrlo jednostavan i glasi — pažljivije pogledati spisak pravila. Na primer, prvo pravilo kaže da se ček prihvata ukoliko klijent ima vozačku dozvolu i garantnu karticu. To znači da, ukoliko su ova dva uslova ispunjena, nije bitno da li je bilo koji od ostalih uslova ispunjen, to neće uticati na krajnji rezultat. Stoga se čak osam kombinacija koje sadrže ispunjena dva navedena uslova može zapisati kao jedna jedina kombinacija. Analogan proces se vrši i za ostale uslove. Takođe, u slučajevima kada je ček odbijen bitno je da bar dva od prva tri uslova ne budu ispunjena, kao ni bar jedan od poslednja dva. Na ovaj način, ukupan broj kombinacija se značajno smanjio.

- **kombinacija 1:** Klijent ima i vozačku dozvolu i garantnu karticu

- **kombiancija 2:** Klijent ima vozačku dozvolu i ček je na manje od 100 dolara
- **kombiancija 3:** Klijent ima garantnu karticu i ček je na manje od 100 dolara
- **kombiancija 4:** Ček je na manje od 100 dolara i dvopartijski je
- **kombiancija 5:** Klijent je "preferirani kupac" i ček je dvopartijski
- **kombiancija 6:** Klijent nema vozačku dozvolu, ček nije dvopartijski i veći je od 100 dolara
- **kombiancija 7:** Klijent nema vozačku dozvolu, nije "preferirani kupac" i ček je veći od 100 dolara
- **kombiancija 8:** Klijent nema vozačku dozvolu ni garantnu karticu, a ček nije dvopartijski
- **kombiancija 9:** Klijent nema vozačku dozvolu ni garantnu karticu, a nije ni "preferirani kupac"
- **kombiancija 10:** Klijent nema garantnu karticu, nije ni "preferirani kupac", a ček nije dvopartijski
- **kombiancija 11:** Klijent nema garantnu karticu, a ček je na više od 100 dolara nije dvopartijski

Korak 4. Nakon redukovanja broja ulaznih uslova, potrebno je još pridružiti odgovarajuće akcije kombinacijama:

- **kombiancija 1:** -> Ček je prihvaćen
- **kombiancija 2:** -> Ček je prihvaćen
- **kombiancija 3:** -> Ček je prihvaćen
- **kombiancija 4:** -> Ček je prihvaćen
- **kombiancija 5:** -> Ček je prihvaćen
- **kombiancija 6:** -> Ček nije prihvaćen
- **kombiancija 7:** -> Ček nije prihvaćen
- **kombiancija 8:** -> Ček nije prihvaćen
- **kombiancija 9:** -> Ček nije prihvaćen
- **kombiancija 10:** -> Ček nije prihvaćen
- **kombiancija 11:** -> Ček nije prihvaćen.

Na kraju je ostalo još tabelirati odgovarajuće rezultate. Radi veće preglednosti zbog i dalje velikog broja slučajeva, rezultati će biti podeljeni u dve tabele. Tabela 7 će sadržati slučajeve kada je ček prihvaćen, dok se u tabeli 8 mogu naći slučajevi kada je ček odbijen.

Tabela 7: Tabela odlučivanja za primer sa prihvaćenim čekovima

	U1	U2	U3	U4	U5
Uslovi					
<100\$	-	Y	Y	Y	-
Posедуje vozačku	Y	Y	-	-	-
Ima garantnu karticu	Y	-	Y	-	-
Ček je dvopartijski	-	-	-	-	Y
"Preferirani kupac"	-	-	-	Y	Y
Akcije					
Ček je prihvaćen	Y	Y	Y	Y	Y

Tabela 8: Tabela odlučivanja za primer sa odbijenim čekovima

	U1	U2	U3	U4	U5	U6
Uslovi						
<100\$	N	N	N	N	-	-
Poseduje vozačku	N	N	-	-	N	N
Ima garantnu karticu	-	-	N	N	N	N
Ček je dvopartijski	N	-	N	-	N	-
"Preferirani kupac"	-	N	-	N	-	N
Akcije						
Ček nije prihvaćen	Y	Y	Y	Y	Y	Y

4 Zaključak

Kao što je u radu i izloženo, tabele odlučivanja su veoma moćna tehnika za testiranje softvera. Posebno je značajna jer omogućava testiranje više različitih ulaznih uslova istovremeno, što tehnike pre nje nisu mogle. Iako i tabele odlučivanja, kao i sve ostale tehnike, imaju svoje mane moguće ih je lako prevazići. Koliko su tabele odlučivanja zaista bitne i korišćene govori i činjenica da danas postoji softver koji je sposoban da uoči greške koje se u njima nalaze. Naravno, prostora za dalje napredovanje uvek ima, pa bi bilo zanimljivo testirati rad tabela odlučivanja čija je veličina kompresovana na neki mali prostor.

Literatura

- [1] Decision table explanation and examples. Arvon management, 2020. on-line at: <https://www.arvonmanagement.com/wp-content/uploads/2020/11/DECTAB1.pdf>.
- [2] Sabrina Chand. Simplifying decision tables, 2019. on-line at: <https://www.studocu.com/row/document/the-university-of-the-south-pacific/project-management/example-6-simplifying-decision-tables/14515578>.

- [3] Vineet Nanda. Learn decision table testing with example, 2023. on-line at: <https://www.tutorialspoint.com/learn-decision-table-testing-with-example>.
- [4] Udo W. Pooch. Translation of decision tables. *ACM Computing Surveys*, 6:125–151, 1974.
- [5] Lakshay Sharma. Decision table testing, 2021. on-line at: <https://www.toolsqa.com/software-testing/istqb/decision-table-testing/>.
- [6] What is decision table in software testing? tryQA, 2023. on-line at: <http://tryqa.com/what-is-decision-table-in-software-testing/>.
- [7] Milena Vujošević Jančić. *Verifikacija softvera*. Matematički fakultet, Univerzitet u Beogradu, 2023.