

Sistem za automatizaciju testova za proveru znanja baziran na transformaciji predikatskih iskaza

Ulfeta Marovac, Department for Technical Sciences, State University of Novi Pazar, Serbia

Aldina Avdić, Department for Technical Sciences, State University of Novi Pazar, Serbia

Apstrakt— Obradom prirodnog jezika omogućava se da računari uspostave komunikaciju sa čovekom, da razumeju jezik čoveka, ali i da ga transformišu i da se čoveku obrate na njemu prirodnom jeziku. Problem istinitosne vrednosti iskaza izrečenih prirodnim jezikom je teško utvrditi zbog bogatstva rečnika i dvosmislenosti značenja. Preslikavanjem iskaza prirodnog jezika u iskaze predikatske logike moguće je utvrditi odnose među različitim predikatskim iskazima kao i izvršiti transformaciju iz jednog oblika pojavljivanja u drugi. Jedna od primena ovakvog znanja može biti u automatizovanom kreiranju testova znanja.

Ključne reči— obrada prirodnog jezika, predikatska logika, automatizovani testovi znanja, srpski jezik

I. UVOD

Testiranje je važna komponenta učenja, jer omogućava da se vidi da li je ispitanik usvojio informacije koje su mu izložene. Sve prisutnije e-učenje zahteva digitalizaciju okruženja za predavanja, ali i ispitivanje. Postoji puno dostupnih softverskih alata koji olakšavaju izradu testova. Njihove prednosti su brojne kao:

- Ušteda vremena u kreiranju testova, jer nema potrebe za tradicionalnim metodama
- Uvek su dostupni na mreži, deljivi su i lakše im pristupaju ispitanici.

Međutim ovi alati se uglavnom koriste kako bi ubrzali proces kreiranja testa i njegovu implementaciju, ali ne i u kreiranju baze znanja za test, i ne poseduju mogućnost provere ispravnosti pitanja koja su data kao slobodan tekst.

Obrada prirodnog jezika (Natural Language Processing - NLP [1]) je oblast veštačke inteligencije koja olakšava komunikaciju čoveka i računara. Naravno da je u procesu obrazovanja uloga nastavnika nezamenljiva, ali se novim tehnologijama može olakšati njihov posao. Ključni faktor u formiranju baze znanja za jedan kurs je uvek nastavnik, jer on daje činjenice koje su relevantne za problem koji se obrađuje. Utvrđivanje tačnosti iskaza koji su napisani na prirodnom jeziku nije jednostavno zbog njihove dvosmislenosti i složenosti, tako da se automatizacija testova svodi na postavljanje pitanja i adekvatnih tačnih i netačnih odgovora koje ispitanik treba da izabere. U ovom procesu određivanje tačnosti nekog pitanja vrši nastavnik. Da bi se pomoglo

Ulfeta Marovac is with Department for Technical Sciences, State University of Novi Pazar, Vuka Karadžića bb, 36300 Novi Pazar, Serbia (e-mail: umarovac@np.ac.rs), <https://orcid.org/0000-0001-7232-3755>

Aldina Avdić is with Department for Technical Sciences, State University of Novi Pazar, Vuka Karadžića bb, 36300 Novi Pazar, Serbia (e-mail: apljaskovic@np.ac.rs), <https://orcid.org/0000-0003-4312-3839>

nastavniku u kreiranju veće količine testova od početnog tačnog ili netačnog iskaza, primenom pravila iskazne logike mogu se dobiti iskazi u drugom obliku čija je istinitosna vrednost poznata. Sličan postupak se može vršiti i u obrnutom smeru gde bi se provera datog odgovora ispitanika u obliku slobodnog teksta pokušala izjednačiti sa nekim od poznatih odgovora.

Obrada prirodnog jezika je teška zbog specifičnosti jezika na kome se primenjuje. Uvođenjem formalnih jezika formiraju se rečenice na osnovu unapred definisanih pravila. Formalni jezici koriste logičke veznike, kvantifikatore, promenljive (termine), oznake za svojstva i odnose (predikate) i pomoćne znakove (zagrade) za izražavanje rečenica. Jedan primer formalne gramatike koja proizvodi predikatske iskaze na hrvatskom jeziku dat je u diplomskom radu autora Stanić [2].

U ovom radu biće prikazana obrada rečenica datih na srpskom jeziku. Da bi se izvršila obrada ovih rečenica moraju se koristiti pravila srpskog jezika kao i specijalni jezički resursi kojima će se umanjiti pojavljivanje sinonima.

Rad je organizovan na sledeći način: u drugom poglavlju opisani su predikatski iskazi i njihovi odnosi; u trećem poglavlju su prikazani algoritmi za transformaciju predikatskih iskaza unutar logičkog kvadrata; model za automatizaciju testova znanja prikazan je u četvrtom poglavlju i u poslednjem poglavlju dat je zaključak i ideje za proširenje modela.

II. PREDIKATSKI ISKAZI I NJIHOV ODNOS

Iskazna logika je precizan matematički alat za računanje logičkih vrednosti složenih iskaznih formula kao i za donošenje zaključaka iz njih. Iskazna logika se bavi rečenicama u celini i ne zalazi u njihovu unutrašnju strukturu. Predikatska logika razmatra smisao polaznih iskaza.

U predikatskoj logici razlikujemo:

- **Term (T)**- objekta o kome se govori
- **Predikat (P)** – osobina objekta ili veza među objektima.

Term može biti konstanta (konkretan objekat) ili promenljiva (nedovoljno određen objekat). Atomični predikatski iskaz sastoji se od terma i predikata.

Neki iskazi sadrže neodređen objekat pa ne mogu imati jedinstvenu vrednost. Upotrebom negacije, veznika i kvantifikatora formira se predikatska formula [3].

Kvantifikacija je tema koja spaja lingvistiku, logiku i filozofiju. Kvantifikatori su osnovni alati pomoću kojih, u jeziku ili logici, upućujemo na količinu stvari [4]. Upotrebom kvantifikatora izražava se koliko je neko tvrđenje tačno i

dobijaju se rečenice koje imaju jedinstvenu logičku vrednost. Univerzalni kvantifikator (\forall) se koristi da prikaže izraze koji su tačni za sve vrednosti terma. Sa druge strane, egzistencijalni kvantifikator se koristi da prikaže da postoji term za koji je tvrđenje tačno.

Predikatske rečenice se mogu podeliti prema kvantitetu na:

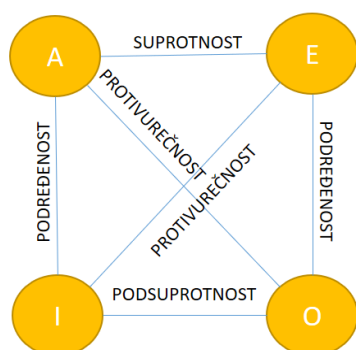
- Univerzalne - opisuju osobinu koju imaju svi pripadnici nekog skupa
- Partikularne - opisuju osobinu koju ima bar jedan element nekog skupa
- Singularne - opisuju osobinu jednog člana nekog skupa.

Predikatske rečenice po kvalitetu se dele na:

- afirmativne - opisuju da subjekat ima neku osobinu,
- negativne - opisuju da subjekat nema neku osobinu,
- limitativne - opisuju da subjekat ima neku osobinu opisanu negacijom.

Izdvojimo četiri tipa predikatskih iskaza koja su prikazana u logičkim kvadratu suprotnosti (Slika 1):

- $(\forall T)P(T)$ - univerzalno-afirmativni iskaz (A)
- $(\forall T)\neg P(T)$ - univerzalno-negativni iskaz (E)
- $(\exists T)P(T)$ - partikularno-afirmativni iskaz (I)
- $(\exists T)\neg P(T)$ - partikularno-negativan iskaz (O).



Slika 1. Logički kvadrat

Predikatske rečenice prikazane u logičkom kvadratu su povezane i ukoliko se zna istinitost jednog tipa može se zaključiti o istinitosti bar još jednog od ostala tri tipa [5].

Prikazani logički kvadrat je motivacija za kreiranje modela za transformaciju jednog predikatskog iskaza za koji je poznata logička vrednost u iskaz kome se može zaključiti tačna logička vrednost.

Može se uočiti povezanost između predikatske logike i prirodnog jezika i to: termi igraju ulogu sličnu onoj koju u prirodnom jeziku igraju imenice i zamenice, a predikati ulogu sličnu glagolima. Prirodni jezik je mnogo bogatiji i nisu dovoljno precizni. Problem prirodnog jezika je i u tome što se isti iskaz može izraziti na više načina.

III. AUTOMATIZOVANO PRESLIKAVANJE PREDIKATSKIH ISKAZA NA SRPSKOM JEZIKU UNUTAR LOGIČKOG KVADRATA

Rečenice u srpskom jeziku mogu se podeliti na proste i složene [6]. Proste rečenice se mogu predstaviti pomoću dva tipa:

1. subjekta (S) i glagolskog predikata (GP)
2. subjekat (S), pomoćni glagol (PG) i imenski predikat (IP).

Prosta rečenica srpskog jezika može se povezati sa atomičnim predikatskim iskazom. Predikatska forma na prirodnom jeziku izražava se upotrebom veznika, negacije i kvantifikatora. U srpskom jeziku se negacija, veznici i kvantifikatori mogu javiti u različitim oblicima. U radu [7] prikazana je veza između predikatske logike i srpskog jezika, kao i pravila prelaza predikatskih formula izraženih na srpskom jeziku unutar logičkog kvadrata.

U ovom radu je dat algoritam za automatizaciju preslikavanja predikatskih iskaza na srpskom jeziku unutar logičkog kvadrata zasnovan na pravilima definisanim u [7]. Za ove potrebe napravljeni su rečnici sa različitim signalima negacije (**D_SN**) kao i rečnik sa potvrdnim (**P_PG**) i odričnim oblicima (**O_PG**) pomoćnih glagola (PG) „imam“, „biti“ i „hteti“ pomoću kojih se može graditi negacija (**D_PG**).

Formiran je takođe:

- **D_UKP**-rečnik različitih termina kojima se izražava univerzalni kvantifikator u potvrdnom obliku (UKP);
- **D_UKO**-rečnik različitih termina kojima se izražava univerzalni kvantifikator u odričnom obliku (UKO);
- **D_EK**-rečnik različitih termina kojima se izražava egzistencijalni kvantifikator UEK.

Kvantifikatori u srpskom jeziku se menjaju po licima, rodu i broju pa se ove informacije nalaze u odgovarajućim rečnicima (L,R,B).

U partikularnim rečenicama uz neke kvantifikatore (na primer „postoji“) se pojavljuje odnosna zamenica (OZ) iza subjekta pa je napravljen i rečnik odnosnih zamenica **D_OZ** sa odgovarajućim informacijama (L,R,B). Za neke oblike egzistencijalnog kvantifikatora se ne upotrebljava odnosna zamenica (na primer „neki“), stoga uz svaki egzistencijalni kvantifikator stoji i opis („tip“) da li ide uz odnosnu zamenicu.

Dalje će biti prikazan algoritmi kojim se od iskaza A dobijaju drugi iskazi logičkog kvadrata. Dobijenim iskazima se može pridružiti i njihova logička vrednost (T-tačno, N-netačno, NaN-nepoznata logička vrednost).

Sva četiri tipa predikatskih iskaza u srpskom jeziku se mogu naći u jednom od dva tipa rečenica prethodno navedenih i to:

Univerzalno-afirmativni iskaz (iskaz_A)

- TIP1 - UKP+S+GP
- TIP2 - UKP+S+PG+IP

Univerzalno-negativni iskaz (iskaz_E)

- TIP1 - UKP+S+SN+GP
- TIP2 - UKO+S+O_PG+IP

Partikularno-afirmativni iskaz (iskaz_I)

- TIP1 - EK+S [+ OZ] +GP

- TIP2 - EK+S [+OZ]+P_PG+IP
- Partikularno-negativan iskaz (iskaz_O)
- TIP1 - EK+S [+ OZ] +SN +GP
 - TIP2 - EK+S [+OZ]+O_PG+IP.

Algoritam 1 (**UP_transformacija**) vrši transformaciju univerzalnih iskaza (A i E) u partikularne (I i O). Inverzna funkcija (**PU_transformacija**) može se slično predstaviti i ona vrši transformaciju partikularnih iskaza (I i O) u univerzalne iskaze (A i E).

Algoritam 1: Preslikavanje univerzalnih iskaza u partikularne

Input: (\$univerzalni_iskaz, \$tip_rečenice,)

Output: (\$niz_partikularnih_iskaza)

UP_transformacija

1. // Odrediti lice, rod i broj
 2. (\$L,\$R,\$B)= **SELECT** L,R,B **FROM** D_UKP
 3. **WHERE** UKP=\$UKP)
 4. // Skup egzistencijalnih kvantifikatora
 5. \$Skup_EK= **SELECT** EK, TIP **FROM** D_EK
 6. **WHERE** (L,R,B)=(\$L, \$R, \$B)
 7. **for** \$EK, \$tip **in** \$Skup_EK **do**
 8. \$p_iskaz=\$univerzalni_iskaz (\$UKP<-\$EK)
 9. **if** (\$tip="uz odnosnu zamenicu")
 10. /*Dodaje se odnosna zamenica u odgovarajućem
 11. *licu , rodu i broju
 12. *ako egzistencijalni kvantifikator zahteva
 13. */
 14. \$Skup_OZ= **SELECT** OZ **FROM** D_OZ
 15. **WHERE** (L,R,B)=(\$L, \$R, \$B)
 16. **for** \$OZ **in** \$Skup_OZ **do**
 17. \$p_iskaz=\$p_iskaz (\$\$<-(\$\$+\$OZ))
 18. **end_for**
 19. **end_if**
 20. \$niz_partikularnih_iskaza += p_iskaz
 21. **end_for**
 22. **RETURN** \$niz_partikularnih_iskaza
-

Algoritam 2 (**AN_transformacija**) vrši transformaciju afirmativnih iskaza (A i I) u negativne (E i O). Inverzna funkcija (**NA_transformacija**) može se slično predstaviti i ona vrši transformaciju negativnih iskaza (E i O) u afirmativne (A i I).

Algoritam 2: Preslikavanje afirmativnih iskaza u negativne

Ulaz: (\$afirmativni_iskaz, \$tip_rečenice)

Izlaz: (\$niz_negativnih_iskaza)

AN_transformacija

1. /* TIP1: iskaz_A(\$UKP+\$S+\$GP)
2. * iskaz_I(\$EK+\$S+\$OZ+\$GP)
3. * Dodavanje singala negacije negacije "ne"
4. */
5. **if** (\$tip_rečenice== TIP1)
6. \$SN="ne"

7. \$n_iskaz=\$afirmativni_iskaz (\$GP<-\$SN+\$GP)
 8. \$niz_negativnih_iskaza += \$n_iskaz
 9. **end_if**
 10. /* TIP2: iskaz_A(\$UKP+\$S+\$P_PG+\$IP)
 11. * iskaz_I(\$EK+\$S+\$OZ+\$P_PG+\$IP)
 12. * Zamena potvrdnog pomoćnog glagola sa
 13. * odričnim.
 14. */
 15. **if** (\$tip_rečenice== TIP2)
 16. \$O_PG=**SELECT** O_PG **FROM** D_PG
 17. **WHERE** P_PG=\$P_PG
 18. \$n_iskaz=\$afirmativni_iskaz (\$P_PG<-\$O_PG)
 19. /* Ako je kvantifikator univerzalni predstavlja se
 20. * negativnim rečenicama odričnim
 21. * kvantifikatorom
 22. */
 23. \$kvantifikator= kvantifikator(n_iskaz)
 24. **if** \$kvantifikator **in** DUKP
 25. // Odrediti lice, rod i broj
 26. (\$L,\$R,\$B)= **SELECT** L,R,B **FROM** D_UKP
 27. **WHERE** UKP=\$kvantifikator
 28. //Skup odričnih kvantifikatora
 29. \$Skup_UKO= **SELECT** UKO **FROM** D_UKO
 30. **WHERE** (L,R,B)=(\$L, \$R, \$B)
 31. **for** \$UKO **in** \$Skup_UKO **do**
 32. \$n_iskaz=\$n_iskaz (\$UKP<-\$UKO)
 33. \$niz_negativnih_iskaza += n_iskaz
 34. **end_for**
 35. **else**
 36. \$niz_negativnih_iskaza += n_iskaz
 37. **end_if**
 38. **end_if**
 39. **RETURN** \$niz_negativnih_iskaza
-

Očigledno je da kombinacijom datih transformacija možemo iz svakog tipa predikatske rečenice dobiti preostala tri, i to u više oblika u zavisnosti od oblika kvantifikatora koji se izabere.

IV. MODEL ZA AUTOMATIZACIJU TESTOVA ZA PROVERU ZNANJA

Prirodni jezici nisu precizni pa predloženi model transformacije zavise od jasnoće ulaza. Većina tvrdnji može se uklopiti u jedan od tipova predikatskih rečenica i ukoliko je njena istinitost poznata može izvršiti transformaciju polazne tvrdnje u skup iskaza koji imaju istu ili suprotnu tačnost. Tabela 1 prikazuje ulazne parametre, primenjene transformacije i rezultat kada se na ulazu nađu iskazi tipa A i E. Slično se može odrediti i za iskaze tipa I i O za koje ukoliko su netačni možemo zaključiti istinitosne vrednosti o ostala tri tipa predikatskih iskaza koje možemo dobiti odgovarajućim transformacijama ili ukoliko su tačni možemo ih transformisati u netačne iskaze tipa (E i A).

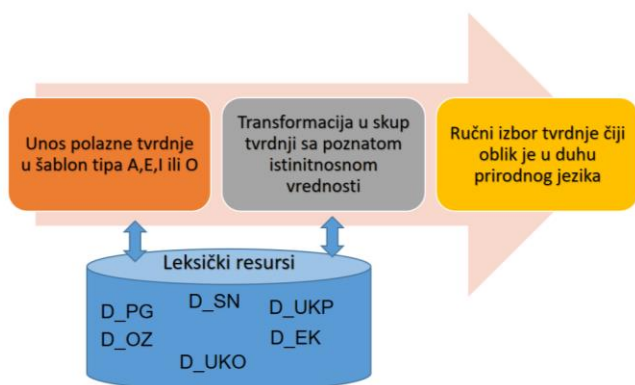
TABELA I
OPIS MOGUĆEG TIPA ULAZA, PRIMENJENIH TRANSFORMACIJA I IZLAZA

Tip ulaznog iskaza	Tačnost	Primenjene transformacije	Tip skupa izlaznih iskaza	
A	Tačno	AN	E	Netačno
A	Tačno	UP	I	Tačno
A	Tačno	AN+UP	O	Netačno
A	Netačno	AN+UP	O	Tačno
E	Tačno	NA	A	Netačno
E	Tačno	UP+NA	I	Netačno
E	Tačno	UP	O	Tačno
E	Netačno	UP+NA	I	Tačno

Postoje dva slučaja koja se pri obradi testova znanja mogu tretirati:

1. postavljanje pitanja
2. provera ispravnosti odgovora.

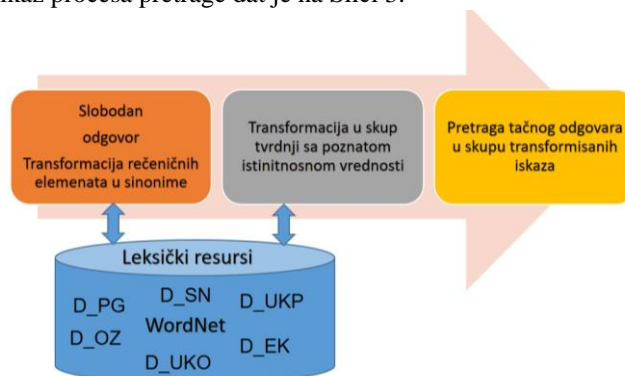
Kod postavljanja pitanja popularna su zaokruživanja više tačnih odgovora tako da se dubina razumevanja materije može upravo proveriti postavljanjem istog odgovora u nekom od njegovih transformacija unutar logičkog kvadrata. Slika 2. prikazuje proces formiranja liste tvrdnji sa poznatom istinitnosnom vrednošću koje mogu da učestvuju u ispitnim pitanjima.



Slika 2. Proces dopune baze sa tvrdnjama za ispitivanje

Provera ispravnosti odgovora se najlakše rešava zaokruživanjem što je na osnovu poznatih istinitnosnih vrednosti procesom formiranja pitanja moguće. Ukoliko se ispitaniku dozvoli unos slobodnog odgovora u šablon tipa A, E,I ili O mora se proveriti ekvivalentnost odgovora sa datim ključem (tačnim odgovorom zadatim od strane ispitivača). Zbog postojanja različitih oblika za izražavanje iste stvari ovaj proces će biti malo teži. Ovde se mogu koristiti napravljeni leksički resursi kako bi se dobili različiti oblici kvantifikatora, simbola negacije, odnosnih zamenica čijom zamenom se neće promeniti odgovor već njegova forma. Takođe za neke dalje korake može se koristiti WordNet za srpski jezik [8] za

pretragu sinonima i za druge rečenične delove. Ispravan odgovor može biti dat i u nekoj od drugih predikatskih formi pa zato ukoliko se transformacijama može dobiti odgovor koji se poklapa sa ključem i pritom je tačan pa se može prihvatiti. Prikaz procesa pretrage dat je na Slici 3.



Slika 3. Proces provere da li dati odgovor ekvivalentan rešenju

Primenu predloženih algoritama demonstriraćemo na primeru transformacije tačnog iskaza "Svaki prirodan broj je pozitivan.", koji se može ubaciti u šablon tipa A. Tabela 2 prikazuje rezultate primene različitih transformacija na ovaj polazni iskaz. Kao rezultat dobijaju se tačni iskazi tipa I kao što su na primer:

- Neki prirodan broj je pozitivan.
- Bar jedan prirodan broj je pozitivan.
- Postoji prirodan broj takav da je pozitivan.
- Postoji prirodan broj koji je pozitivan.

Primenom transformacije AN(+UP) dobijamo netačne iskaze tipa E i O kao što su:

- Svaki prirodan broj nije pozitivan.
- Neki prirodan broj nije pozitivan.
- Postoji prirodan broj takav da nije pozitivan.

TABELA II
PRIMER PRIMENE PRIKAZANOG ALGORITMA NA ISKAZ TIPA A

REČENIČNI ČLAN	ULAZ	IZLAZ			
		TRANSFORMACIJA			
		AN	UP (BEZ OZ)	UP (SA OZ)	AN+UP
KVANTIFIKATOR	Svaki	Svaki	Neki, Bar jedan,...	Postoji, Postoji jedan,...	Neki, Postoji*, ...
SUBJEKAT	prirodan broj	prirodan broj			
ODNOSNA ZAMENICA					
SIMBOL NEGACIJE				takav da, koji	[takav da, koji]*
POMOĆNI GLAGOL	je	nije	je	je	nije
PREDIKAT	pozitivan.	pozitivan.			
LOGIČKA VREDNOST ISKAZA	T	N	T	T	N

IV. ZAKLJUČAK

U ovom radu je predstavljen sistem za automatizaciju testova za proveru znanja na srpskom jeziku baziran na transformaciji predikatskih iskaza. Predikatski iskazi iz logičkog kvadrata se predloženim transformacijama mogu preslikati jedni u druge. Ova osobina je primenjena na tvrdnje u testovima znanja koje se transformacijama mogu pretvoriti u druge oblike sa poznatom istinitnosnom vrednosti i kao takvi mogu se uključiti u testove. Doprinos ovog rada su i kreirani rečnici kvantifikatora u različitim oblicima pojavljivanja kao i metode za transformaciju. Sinonimi su jedan od velikih problema pri obradi prirodnog jezika. Primena obrade sinonima prilikom validacije slobodnih odgovora jeste jedan od sledećih puteva razvoja sistema. Primena De Morganovih pravila i drugih tautologija takođe može proširiti skup ekvivalentnih iskaza.

ZAHVALNICA

Ovaj rad je delemično podržan od strane Ministarstva prosvete, nauke i tehnološkog razvoja Republike Srbije po projektu III44007.

REFERENCES

- [1] Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.
- [2] Stanić, R. Kvantifikacija i negaciju u logici i hrvatskom jeziku, diplomski rad. 2014.
- [3] Nebojša Ikodinović, Uvod u matematičku logiku, Beograd 2015
- [4] Peters, S., & Westerståhl, D. (2006). Quantifiers in language and logic. OUP Oxford.
- [5] Nermin Okačić, Tautologije i valjane formule kao principi zaključivanja, Prirodno matematički fakultet, Tuzla, 2015. godine.
- [6] Ivan Klajn, Gramatika srpskog jezika, Zavod za udžbenike i nastavna sredstva, 2005.
- [7] Marovac, U. Avdić, A. Čuljević, N. Metoda za obradu predikatskih iskaza na srpskom jeziku, Kopaonik, 2022.
- [8] Krstev, C. Pavlović-Lažetić, G. and Obradović, I. "Using textual and lexical resources in developing serbian wordnet." *Romanian Journal of Information Science and Technology* 7.1-2 (2004): 147-161.

ABSTRACT

By processing natural language, we enable computers to establish communication with humans, to understand human language, but also to transform it and to address human in natural language. The problem of the truth value of statements uttered in natural language is difficult to determine due to the richness of vocabulary and ambiguity of meaning. By mapping the statements of natural language into the statements of predicate logic, it is possible to determine the relationships between different predicate statements as well as to perform the transformation from one form of appearance to another. One of the applications of this knowledge can be in the automated creation of knowledge tests.

**System for automation of knowledge verification tests
based on transformation of predicate statements**

Ulfeta Marovac, Aldina Avdić