Informatički praktikum 2 — vežbe

Miloš Utvić

Sadržaj

Sadržaj **i**

I Rad sa formulama u programu MS Excel 1

1 Osnovni koncepti **3**
1.1 "Naivna" definicija formula 3
1.2 Osnovne računske radnje 4
1.3 Stepenovanje i korenovanje 5
1.4 Reference 6
1.5 Funkcije 9
1.6 Tipovi adresiranja. Kopiranje formula 13
1.7 Imena 24
1.8 Studija slučaja: evidentiranje pohađanja nastave 24

2 Tipovi podataka **25**
2.1 Numerički tip podataka 25
2.2 Tekstualni tip podataka 27
2.3 Logički tip podataka 31
2.4 Datum i vreme kao tipovi podataka 31

3 Izrazi **33**
3.1 Aritmetički izrazi 33
3.2 Relacijski izrazi 34
3.3 Logički izrazi 34
3.4 Tekstualni izrazi 35
3.5 Izrazi i formule 35

4 Pregled funkcija **37**
4.1 Aritmetičke funkcije 38
4.2 Logičke funkcije 38

4.3 IF-funkcije 40
4.4 IF-funkcija 40
4.5 Funkcije za rad sa tekstom 43
4.6 Funkcije za rad sa datumom i vremenom 43
4.7 Kompozicija funkcija 43

Deo I

Rad sa formulama u programu MS Excel

Glava 1

Osnovni koncepti

1.1 "Naivna" definicija formula

Formule predstavljaju osnovno sredstvo koje MS Excel koristi kao program za tabelarna izračunavanja. Složenost formula u Excel-u (u daljem tekstu: formula) varira od zapisa koji su po obliku i značenju nalik uobičajenim matematičkim formulama koje se uče u osnovnoj školi, do izuzetno složenih formula koje istovremeno koriste koncepte iz matematike i računarstva poput promenljivih, tipova podataka, izraza, poziva funkcija, kompozicije funkcija, kontrole toka izračunavanja (nalik IF-naredbi u programskim jezicima), itd. S jedne strane, složenost formula omogućava širok spektar mogućnosti i primena u raznim oblastima, uključujući ne samo matematiku i prirodne nauke, već i društvene nauke, pogotovo one društvene nauke koje se oslanjaju na statistiku kao pomoćnu disciplinu. S druge strane, nije jednostavno dati opštu definiciju formula upravo zbog njihove složenosti.

Umesto deduktivnog pristupa, kojim se kreće od sveopšte definicije, a potom se razmatraju konkretni primeri, za početnika je bolje primeniti induktivni pristup: postepeno upoznavanje mogućnosti Excel-a i raznih oblija koje njegove formule mogu uzeti, od jednostavnijih ka složenijim, da bi se tek na kraju čitalac suočio sa strogo formalnom definicijom koja pokušava da opiše proizvoljnu formulu u Excel-u.

Uzimajući u obzir navedene prednosti, u ovom priručniku je primenjen induktivni pristup. U poglavlju 1 su ilustrovane osnovne formule, kao i koncepti koje koristi Excel (reference i funkcije). U poglavlju 2 je dat pregled osnovnih tipova podataka. Pokušaj opšte i formalne definicije izraza i formula u Excel-u je predstavljen u poglavlju 3. Naposletku, u poglavlju 4 su opisane neke od najčešće korišćenih funkcija.

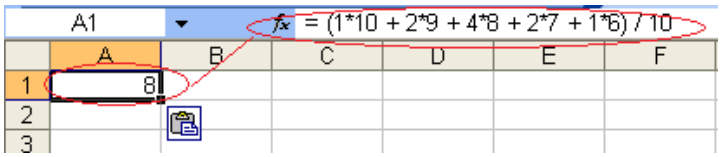
Zapis, formatiranje i automatsko izračunavanje formula

Prilikom zapisivanja formule u ćeliji tabele neophodno je sledeće:

(F1) Zapis formule u ćeliji mora početi znakom jednakosti ('=').

(F2) Ćelija mora biti formatirana tako da pripada opštoj kategoriji (eng. General).

Uslovi (F1) i (F2) su bitni kako bi Excel prepoznao da je sadržaj ćelije formula, a ne podatak (na primer, običan tekst) i da bi formulu (ukoliko je ispravno zapisana) automatski izračunao. Naime, kada se formula upiše u datu ćeliju u skladu sa navedenim uslovima i prestane uređivanje ćelije, Excel će u datoj ćeliji, umesto unete formule, prikazati vrednost dobijenu automatskim izračunavanjem formule, dok se sama formula se može videti i ažurirati u posebnom polju iznad tabele, označenom sa f_x (v. Primer 1.2.1, Slika 1.1). Ukoliko se započne ažuriranje sadržaja ćelije, umesto vrednosti, ponovo će biti prikazana uneta formula.



Slika 1.1: Prikaz formule i njene vrednosti u tabeli.

Automatsko izračunavanje naročito dolazi do izražaja ukoliko formula u jednoj ćeliji zavisi od vrednosti drugih ćelija (v. odeljak 1.4). Naime, kada se promene vrednosti ćelija od kojih zavisi formula, Excel automatski ponovo izračunava vrednost formule u skladu sa nastalim promenama i prikazuje novu vrednost u odgovarajućoj ćeliji formule.

1.2 Osnovne računske radnje

Najjednostavnije formule u Excel-u se svode na upotrebu konkretnih brojeva i osnovnih računskih radnji, što omogućava da se program koristi i kao običan kalkulator.

Osnovne računske radnje, sabiranje, oduzimanje, množenje i deljenje, se redom realizuju pomoću binarnih operatora $+$, $-$, $*$ i $/$ (Tabela 1.1). Oznaka za operator oduzimanja $(-)$ se takođe koristi kao oznaka za unarni operator promene znaka. Kao i u matematici, najveći prioritet ima operator promene znaka, za njim slede operatori množenja i deljenja, a najmanji

prioritet među računskim radnjama imaju operatori sabiranja i oduzimanja. Redosled prioriteta se zaobilazi upotrebom zagrada.

operator	broj operanada	značenje	primer	prioritet
$-$	1 (unarni)	promena znaka	-7	1
$*$	2 (binarni)	množenje	$12 * 3$	2
$/$	2 (binarni)	deljenje	$15/4$	2
$+$	2 (binarni)	sabiranje	$2 + 3$	3
$-$	2 (binarni)	oduzimanje	$14 - 9$	3

Tabela 1.1: Osnovne računske radnje u Excel-u.

Primer 1.2.1. Pretpostavimo da je student položio deset ispita i da ima jednu desetku, dve devetke, četiri osmice, dve sedmice i jednu šesticu. Njegov prosek se može izračunati formulom (1.1):

```
= (1*10 + 2*9 + 4*8 + 2*7 + 1*6) / 10          (1.1)
```

♢

1.3 Stepenovanje i korenovanje

Stepenovanje se u Excel-u realizuje pomoću operatora ^(„kapica").

Primer 1.3.1. Jedan od načina da se izračuna četvrti stepen broja 16, tj. 16^4, je primenom formule (1.2):

```
= 16 ^4                                        (1.2)
```

♢

Operator ^je levo asocijativan, tj. za brojeve x, y i z važi da je vrednost izraza $x\hat{}y\hat{}z$ jednaka vrednosti izraza $(x\hat{}y)\hat{}z$.

Korenovanje se takođe realizuje pomoću operatora ^, pri čemu se koristi algebarsko svojstvo da za proizvoljan pozitivan prirodan broj n i broj x iz odgovarajućeg domena[1] važi:

$$\sqrt[n]{x} = x^{\frac{1}{n}}.$$

Primer 1.3.2. Primenom formule (1.3):

```
= 16 ^(1/4)                                    (1.3)
```

se računa četvrti koren broja 16, tj. $\sqrt[4]{16}$. ♢

[1]U slučaju da je n paran prirodan broj, x mora biti pozitivan broj, itd.

1.4 Reference

Uobičajeni kalkulatori raspolažu memorijom u koju korisnici smeštaju međurezultate ukoliko se proces računanja odvija u više koraka. Excel koristi ćelije tabela za smeštanje ulaznih podataka i međurezultata računanja. Kada korisnik želi da iskoristi međurezultat koji je prethodno izračunao, to može da uradi na dva načina:

(i) navođenjem konkretne vrednosti prethodnog međurezultata (prekucavanjem ili kopiranjem) ili

(ii) referisanjem na ćeliju u kojoj je međurezultat smešten, tj. navođenjem njen adrese. Ako se prilikom izračunavanja koriste reference (adrese ćelija), one se zamenjuju odgovarajućim vrednostima izraza u ćelijama na koje se referiše.

Relativna adresa pojedinačne ćelije

Svaka ćelija tabele u Excel-u se može adresirati na dva osnovna načina koji se još nazivaju **stilovi adresiranja**. Stilovi adresiranja u Excel-u nose naziv po odgovarajućoj adresi prve ćelije, te se tako razlikuju stil `A1` i stil `R1C1`. Excel je podešen tako da uvek koristi isključivo jedan stil adresiranja. Podrazumevani stil adresiranja posle instalacije Excel-a je u većini slučajeva stil `A1`, pa će se u ovom priručniku, osim ako se ne kaže drugačije, pretpostavljati da je Excel podešen da koristi stil `A1`.

Međutim, čak i kada se izabere stil `A1` kao stil adresiranja, postoje četiri različita načina da se ćeliji pridruži adresa. Radi jednostavnosti izlaganja, u ovom odeljku razmotrićemo samo **relativno adresiranje**, dok će ostali tipovi adresiranja biti objašnjeni u odeljku 1.6.

Svakoj ćeliji u tabeli može se pridružiti jedinstvena relativna adresa koja je određena:

(i) nazivom radne knjige u kojoj se ćelija nalazi,

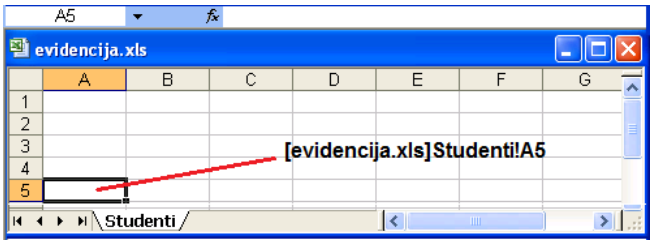
(ii) nazivom tabele (radnog lista, radne sveske) u kojoj se ćelija nalazi,

(iii) nazivom kolone kojoj pripada ćelija i

(iv) rednim brojem reda kome pripada ćelija.

Navedeni elementi adrese (i)–(iv) se moraju navesti upravo u redosledu u kom su nabrojani, pri čemu se naziv radne knjige (i) navodi između srednjih zagrada (karakteri ‚[' i ‚]'), dok se naziv tabele (ii) odvaja od naziva kolone (iii) uzvičnikom (‚!').

Primer 1.4.1. Neka radna knjiga `evidencija.xls` sadrži tabelu `Studenti` (Slika 1.2).



Slika 1.2: Adresa ćelije u koloni A i 5. redu tabele `Studenti` u datoteci (radnoj knjizi) `evidencija.xls`.

Adresa ćelije koja se nalazi u koloni `A` i 5. redu tabele `Studenti` je:

```
[evidencija.xls]Studenti!A5
```
♢

Adresiranje omogućava da se prilikom izračunavanja neke formule referiše na vrednost određene ćelije navođenjem njene adrese. Ukoliko se u formuli referiše na ćeliju koja se nalazi u istoj datoteci kao i sama formula, u adresi se ne mora navoditi ime datoteke. Takođe, ako se u formuli referiše na ćeliju koja je i u istoj datoteci i u istoj tabeli kao i sama formula, onda nije potrebno navoditi ni ime datoteke ni ime tabele, već je dovoljno naznačiti samo naziv kolone i redni broj reda u kome se nalazi ćelija.

Primer 1.4.2. Neka je u tabeli `Studenti` datoteke `evidencija.xls` potrebno sabrati vrednosti prvih dvaju ćelija kolone A i rezultat smestiti u treću ćeliju kolone A iste tabele, dovoljno je u trećoj ćeliji kolone A tabele `Studenti` otkucati formulu:

```
= A1 + A2
```

Isti rezultat daje i formula

```
[evidencija.xls]Studenti!A1 + [evidencija.xls]Studenti!A2
```

ali je ona neuporedivo složenija za zapisivanje. ♢

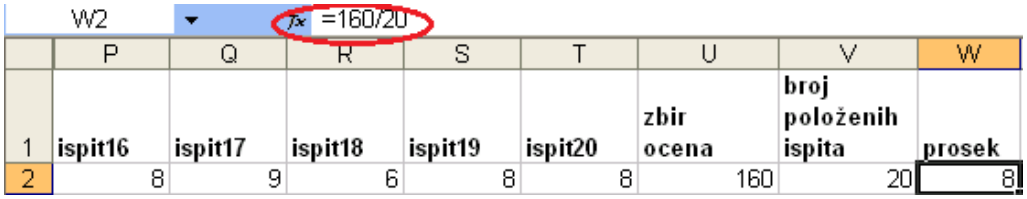
S obzirom da je najčešći slučaj da se u formulama referiše na ćelije koje se nalaze u istoj datoteci i istoj tabeli, u najvećem broju primera koji slede pretpostavićemo da je upravo to slučaj i zbog jednostavnijeg zapisivanja formule nećemo u referencama navoditi ni naziv datoteke ni naziv tabele odgovarajućih ćelija na koje se referiše.

Takođe, radi jednostavnijeg izlaganja, kad god je iz konteksta jasno o čemu se govori, poistovetićemo ćeliju sa njenom adresom, pa ćemo tako, umesto „3. ćelija u koloni `A`", uglavnom govoriti samo „ćelija `A3`".

Primer 1.4.3. Pretpostavimo da je student N. N. diplomirao i da je tokom studija položio ukupno 20 različitih predmeta. Prosečna ocena studenta se računa kao količnik zbira ocena i broja ocena, odnosno broja položenih ispita. Pretpostavimo da smo već izračunali zbir ocena i broj položenih ispita i da se te vrednosti redom nalaze u ćelijama `U2` (na primer 160) i `V2` (na primer 20). Tada se prosečna ocena studenta može izračunati u ćeliji `W2`:

(i) bilo navođenjem konkretnih vrednosti prethodnih međurezultata (formula (1.4), Slika 1.3):

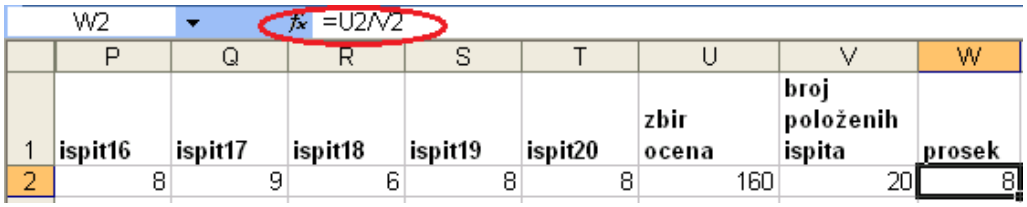
```
= 160 / 20
```
(1.4)



Slika 1.3: Navođenje konkretnih vrednosti prethodnih međurezultata.

(ii) bilo referisanjem na ćelije sa međurezultatima (formula (1.5), Slika 1.4):

```
= U2 / V2
```
(1.5)



Slika 1.4: Referisanje na ćelije sa međurezultatima.

Iako obe formule, (1.4) i (1.5), daju isti rezultat, druga formula može da se iskoristi ponovo za izračunavanje sa drugim ulaznim podacima (ocenama), što ćemo detaljnije ilustrovati u Primeru 1.5.4. ♢

Opseg. Relativno adresiranje opsega

Pod **blokom** ili **opsegom** (eng. **range**) u programu MS Excel se podrazumeva pravougaoni podskup ćelija tabele. Adresa opsega je određena:

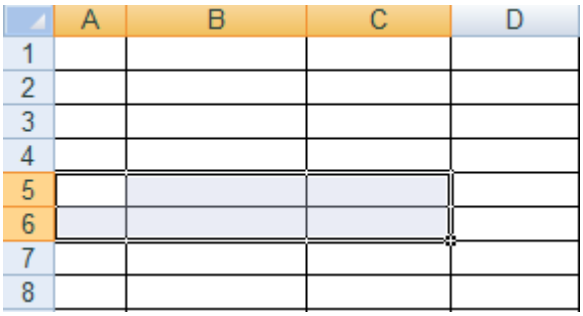
(i) adresom gornje leve ćelije (gornji levi kraj opsega) i

(ii) adresom donje desne ćelije (donji desni kraj opsega).

Prilikom navođenja adrese opsega, gornji levi kraj i donji desni kraj opsega se razdvajaju karakterom ‚:'.

Primer 1.4.4. Neka je r opseg ćelija određen presekom kolona `A`, `B` i `C` i redova sa rednim brojem 5 i 6. Tada je ćelija `A5` gornji levi kraj opsega r, ćelija `C6` je donji desni kraj opsega r, dok je `A5:C6` — adresa opsega r (Slika 1.5). ♢

Iz istih razloga kao u odeljku 1.4, kad god je iz konteksta jasno o čemu se govori, poistovetićemo opseg sa njegovom adresom, pa ćemo tako, umesto „opseg čiji je gornji levi kraj ćelija `A5`, a donji desni kraj ćelija `C6`", uglavnom govoriti samo „opseg `A5:C6`".



Slika 1.5: Opseg `A5:C6` kao presek kolona `A`, `B` i `C` i redova 5 i 6.

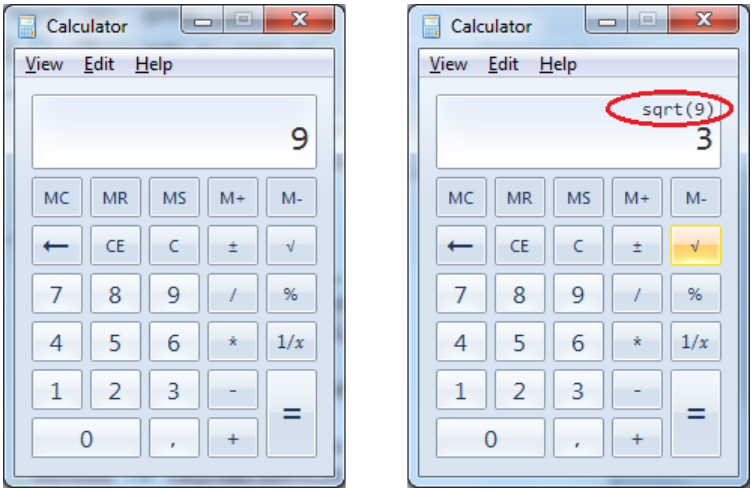
1.5 Funkcije

Napredniji kalkulatori, pored osnovnih računskih radnji i memorije za međurezultate računanja, raspolažu i nizom matematičkih funkcija (eksponencijalna i logaritamska, trigonometrijske i inverzne trigonometrijske funkcije, statističke funkcije, itd.).

Excel takođe koristi funkcije, pri čemu postoje dve bitne razlike u odnosu na funkcije kalkulatora.

Prva razlika je u broju funkcija koje su na raspolaganju: Excel raspolaže daleko većim brojem funkcija u odnosu na najnaprednije kalkulatore.

Druga bitna razlika između kalkulatora i programa MS Excel je notacija koja se koristi prilikom poziva funkcija, tj. redosled navođenja imena funkcije i njenih ulaznih podataka. Ulazni podaci funkcije se još nazivaju **argumenti funkcije**. Kod kalkulatora se najčešće najpre navodi prvi argument funkcije, a potom se aktivira taster koji predstavlja poziv funkcije, posle čega se opciono navode ostali argumenti, a na samom kraju kalkulator ispisuje vrednost izračunavanja. Tipičan primer opisanog poziva funkcije na kalkulatoru je izračunavanje kvadratnog korena zadatog broja (Slika 1.6).



Slika 1.6: Izračunavanje kvadratnog korena zadatog broja u softverskom simulatoru kalkulatora (MS Calculator). Crvenom bojom je uokvirena interna notacija za predstavljanje poziva funkcija, identična sa notacijom koju koristi Excel.

S druge strane, Excel za pozivanje funkcija koristi prefiksnu notaciju preuzetu iz matematike:

$$imeFunkcije(argument_1, argument_2, \ldots, argument_n) \qquad (1.6)$$

Broj n u (1.6) predstavlja broj ulaznih podataka (argumenata) funkcije i još se naziva **arnost funkcije**. Funkcije se, prema broju argumenata, nazivaju **unarnim** ($n = 1$), **binarnim** ($n = 2$), **ternarnim** ($n = 3$), itd. U opštem slučaju se za funkciju koja ima n argumenata kaže da je **n-arna funkcija**.

Iako različite funkcije mogu imati više argumenata, prilikom poziva funkcije uvek se kao rezultat izračunavanja dobija jedna jedina vrednost — **vrednost funkcije**.

Ukoliko funkcija ima bar dva argumenta, neophodan je poseban simbol kojim se oni razdvajaju — separator argumenata. Separator argumenata nije fiksiran karakter, već njegova vrednost zavisi od podešavanja opcija operativnog sistema Windows koja se odnose na jezik i pravopis. Prilikom instalacije operativnog sistema korisnik bira podrazumevani jezik (srpski / ćirilica, srpski / latinica, engleski, itd.) i taj izbor utiče na formatiranje prikaza datuma i vremena, ali i na formatiranje numeričkih vrednosti, kao i na karakter koji se koristi kao separator argumenata funkcija u programu Excel. Ovde ćemo razmotriti separator argumenata funkcije samo u slučajevima kada je izabrana neka varijanta srpskog jezika, odnosno kada je izabran engleski jezik.

U slučaju izbora neke varijante srpskog jezika, na formatiranje brojeva se primenjuju pravila pravopisa za srpski, pa se zapeta koristi za razdvajanje celog dela od decimala, dok se tačka koristi kao separator grupa od po tri cifre (hiljada) u celom delu broja (na primer, 2.124.678,99 je zapis broja „dva miliona sto dvadeset četiri hiljade šesto sedamdeset osam celih i devedeset devet stotih" po srpskom pravopisu). Pošto je u ovom slučaju zapeta iskorišćena kao separator celog broja i decimala u zapisu broja, kao separator argumenata funkcije u programu MS Excel se koristi tačka zapeta (';').

U slučaju da je izabran engleski jezik, pravopisna pravila za formatiranje broja su obrnuta u odnosu na srpski: tačka se koristi za razdvajanje celog dela od decimala, dok se zapeta koristi kao separator grupa od po tri cifre (hiljada) u celom delu broja (na primer, 2,124,678.99 je zapis broja „dva miliona sto dvadeset četiri hiljade šesto sedamdeset osam celih i devedeset devet stotih" po engleskom pravopisu). Takođe, pri izboru engleskog jezika, kao separator argumenata funkcije u programu MS Excel se koristi zapeta.

U primerima koji slede pretpostavićemo da je podrazumevani jezik srpski, a ne engleski, te će se shodno tome koristiti pravila srpskog pravopisa za formatiranje brojeva, odnosno tačka zapeta kao separator argumenata funkcije u programu MS Excel.

Primer 1.5.1. U Excel-u, kao i kod uobičajenih kalkulatora, postoji posebna unarna funkcija za računanje kvadratnog korena zadatog nenegativnog realnog broja — `SQRT`. Da bi se izračunao kvadratni koren broj 729 pomoću funkcije `SQRT`, dovoljno je upotrebiti formulu (1.7):

$$= \texttt{SQRT}(729) \tag{1.7}$$

Isti efekat se može postići i upotrebom binarne funkcije `POWER` koja je samo prefiksna verzija infiksnog operatora stepenovanja ^, tj. izraz

`POWER(729; 1/2)` ima istu vrednost kao izraz 729^(1/2)[2]. ♢

Iako se funkcija neretko poistovećuje sa pravilom izračunavanja, neophodno je uzeti u obzir i druge dve komponente koje se spominju u matematičkoj definiciji funkcije: domen i kodomen funkcije.

Domen funkcije predstavlja skup vrednosti koje mogu uzeti argumenti funkcije, tj. skup vrednosti za koje ima smisla primeniti izračunavanje koje sprovodi funkcija. **Kodomen funkcije** predstavlja skup iz koga mogu biti vrednosti funkcije.

Primer 1.5.2. Unarna funkcija `SQRT` zahteva jedan argument koji mora biti nenegativan realni broj. Dakle, domen funkcije `SQRT` je skup nenegativnih realnih brojeva.

Rezultat koji izračunava funkcija `SQRT` je takođe nenegativan realni broj, pa je kodomen funkcije `SQRT` istovetan sa njenim domenom. ♢

Primer 1.5.3. Unarna funkcija `LOWER` zahteva argument tekstuelnog tipa, tj. nisku (v. odeljak 2.2, str. 27).

Rezultat koji izračunava funkcija `LOWER` je takođe niska, dobijena od argumenta funkcije zamenom pojavljivanja svih velikih slova odgovarajućim malim slovima. Na primer, poziv funkcije `LOWER` u formuli (1.8) ima za rezultat nisku `"crna gora"`:

```
= LOWER("Crna Gora")                                    (1.8)
```

♢

Argument funkcije može biti i referenca, tj. adresa ćelije, pod uslovom da se u ćeliji na koju se referiše nalazi vrednost iz odgovarajućeg domena. Prilikom poziva funkcije, ako je neki argument referenca, argument se zamenjuje vrednošću izraza u ćeliji na koju se referenca odnosi. Takođe, argument pojedinih funkcija može biti i opseg, odnosno niz referenci.

Primer 1.5.4. Razmotrimo ponovo izračunavanje prosečne ocene studenta N. N. (v. Primer 1.4.3). U opsegu `A2:T2` se nalaze konačne ocene studenta N. N. Zbir ocena se može izračunati formulom 1.9 u ćeliji `U2`, broj položenih ispita — formulom 1.10 u ćeliji `V2`, a prosečna ocena studenta — formulom 1.11 u ćeliji `W2`.

[2]Razlika između **prefiksne** i **infiksne** notacije se odnosi na položaj imena funkcije, odnosno oznake operatora, u odnosu na argumente funkcije, odnosno operande. U slučaju prefiksne notacije, ime funkcije se piše pre argumenata, dok se kod infiksne notacije oznaka operatora navodi između operanada.

$$= \texttt{SUM(A2:T2)} \tag{1.9}$$
$$= \texttt{COUNT(A2:T2)} \tag{1.10}$$
$$= \texttt{U2 / V2}. \tag{1.11}$$

Funkcije `SUM` i `COUNT` su primeri funkcija čiji argument može biti opseg, u ovom slučaju `A2:T2`. Funkcija `SUM` sabira vrednosti izraza u ćelijama opsega `A2:T2`, dok funkcija `COUNT` prebrojava koliko u istom opsegu ima nepraznih ćelija čija je vrednost broj. Pošto su sve ćelije opsega `A2:T2` neprazne i sadrže numeričku vrednost (broj), izraz `COUNT(A2:T2)` zapravo prebrojava koliko u istom opsegu ima ćelija, što se u ovom primeru svodi na broj ocena, odnosno broj predmeta koje je student N. N. položio.

Umesto tri odvojene formule (1.9–1.11), prosečna ocena studenta N. N. se može izračunati jednom formulom (1.12):

$$= \texttt{SUM(A2:T2) / COUNT(A2:T2)} \tag{1.12}$$

♢

Pošto je računanje proseka, odnosno aritmetičke sredine datog niza vrednosti čest zadatak u statistici, Excel poseduje posebnu funkciju `AVERAGE` koja za zadati skup vrednosti (opseg) računa prosečnu vrednost, tako da su formule (1.12) i (1.13) istog značenja:

$$= \texttt{AVERAGE(A2:T2)} \tag{1.13}$$

♢

1.6 Tipovi adresiranja. Kopiranje formula

U odeljku 1.4, kada je bilo reči o stilu `A1` kao stilu adresiranja, rečeno je da se svakoj ćeliji može dodeliti adresa na četiri različita načina, pri čemu je objašnjeno samo relativno adresiranje. Razlozi zbog kojih postoje četiri različita tipa adresiranja za isti stil adresiranja odnose se na četiri moguća efekta koje korisnici očekuju kada kopiraju formule (Primeri 1.6.1–1.6.10)

Primer 1.6.1. Neka je tabela u Excel-u (Slika 1.7a) popunjena na sledeći način:

- svaki red numerisan od 1 do 100 sadrži podatke o nekom studentu;

- podaci o pojedinačnom studentu su odvojeni po kolonama sledećim redom: prezime (kolona A), ime (kolona B), broj indeksa (kolona C), broj poena ostvaren na predispitnim obavezama (kolona D) i broj poena osvojen na pismenom ispitu (kolona E).

Pretpostavimo da je potrebno popuniti kolonu F ukupnim brojem poena koje je student stekao na predispitnim obavezama i pismenom ispitu, pri čemu je ideja da sav račun obavi Excel. Ukupan broj poena za prvog studenta se može izračunati formulom 1.14 u ćeliji `F1`:

```
= D1 + E1
```
(1.14)

Jasno da će slična formula izračunati ukupan broj poena za svakog studenta posebno, ali se postavlja pitanje da li postoji način da se izbegne mukotrpno prekucavanje ili kopiranje teksta formule 1.14 i prepravljanje referenci `D1` i `E1` u odgovarajuće (`D2` i `E2` za drugog studenta, `D3` i `E3` za trećeg studenta, itd.). Potreban je mehanizam koji, uz pomoć Excel-a, „preslikava" formulu 1.14 u preostale ćelije opsega `F2:F100`, tako da se ukupan broj poena za svakog studenta izračuna sabiranjem vrednosti odgovarajućih ćelija sa poenima stečenim na predispitnim obavezama i pismenom ispitu. Takav mehanizam postoji i svodi se na kopiranje kompletne ćelije `F1`, a ne njenog sadržaja (teksta formule), i realizuje se na sledeća dva načina (pod pretpostavkom da se u ćeliji `F1` već nalazi formula 1.14):

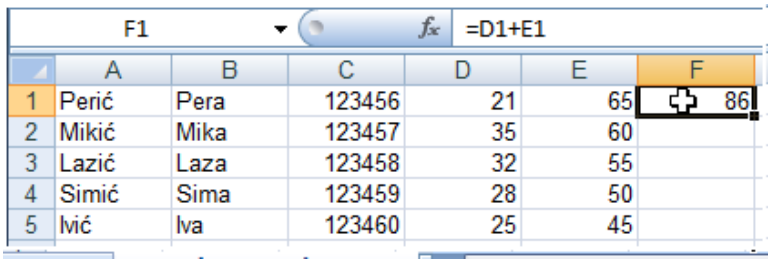
(i) Izabere se ćelija `F1` (Slika 1.7a). Pozicioniranjem miša u donji desni ugao ćelije `F1` se menja izgled kursora od belog punog krsta u tanki crni krst (Slika 1.7b). Promena izgleda kursora je signal da može da se pristupi kopiranju ćelije `F1` tako što se klikne levi taster miša, drži i miš se prevuče preko desnih donjih uglova ćelija opsega `F2:F100` (Slike 1.7c i 1.7d).

(ii) Izabere se ceo opseg `F1:F100`, a potom se aktivira kombinacija tastera Ctrl i D (Ctrl+D).

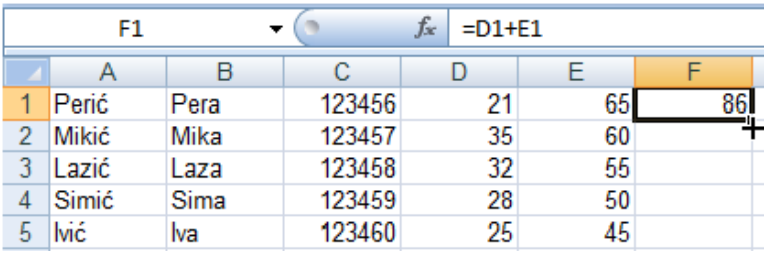
U oba slučaja se formula ćelije `F1` kopira u ćelije opsega `F2:F100`, pri čemu se adrese `D1` i `E1` redom zamenjuju u odgovarajuće (`D2` i `E2` za drugog studenta, `D3` i `E3` za trećeg studenta, itd.). ♢

Da bismo razumeli zašto relativno adresiranje omogućava željeni efekat u Primeru 1.6.1, neophodno je da se upoznamo i sa drugim stilom adresiranja, stilom `R1C1`.

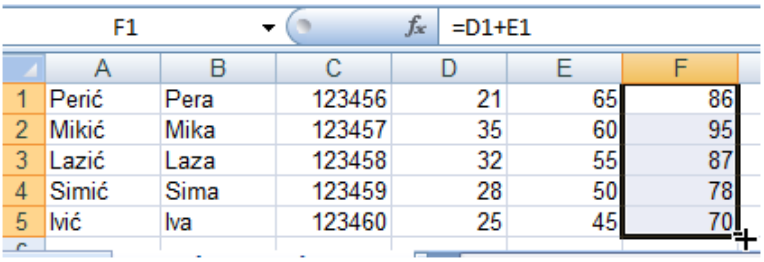
Pre toga ćemo definisati još dva tipa adresa u stilu `A1`: apsolutne i kombinovane adrese.



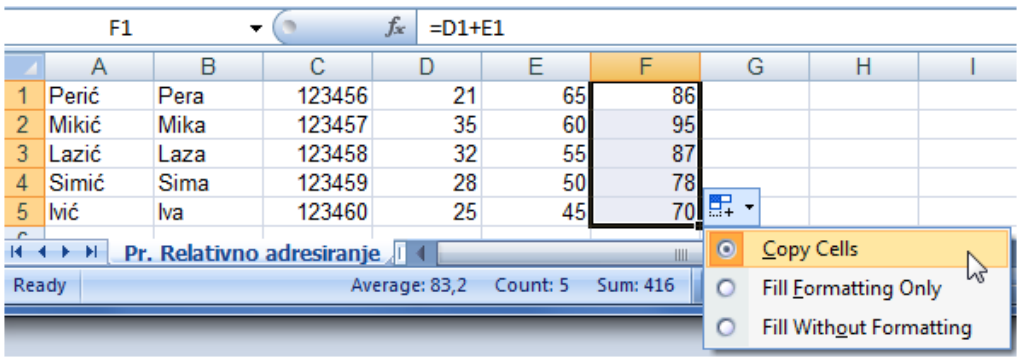
(a) Izbor ćelije `F1`. Kursor je u obliku širokog belog krsta.



(b) Pozicioniranje miša na donji desni ugao ćelije `F1`. Kursor se menja u tanki crni krst.



(c) Prevlačenje miša duž desne ivice opsega `F2:F5`.



(d) Konačan rezultat kopiranja ćelije `F1` u ostale ćelije opsega `F2:F5`.

Apsolutne adrese ćelija u stilu `A1`

U stilu `A1` apsolutna adresa ćelije se dobija dodavanjem znaka za dolar ('$') ispred oznake kolone i reda u odgovarajućoj relativnoj adresi.

Primer 1.6.2. Ćeliji `A1` odgovara apsolutna adresa `$A$1`. Slično, ćeliji `D6` odgovara apsolutna adresa `$D$6`. ♢

Apsolutne adrese u stilu `A1` se prilikom kopiranja ćelije ne menjaju.

Primer 1.6.3. Neka se u ćeliji `C1` nalazi formula 1.15:

```
=$A$1+$B$1
```
(1.15)

Ukoliko kopiramo ćeliju `C1` u bilo koju drugu ćeliju tabele, sadržaj te druge ćelije će takođe biti formula 1.15. ♢

Kombinovane adrese čelija u stilu `A1`

U stilu `A1` kombinovana adresa ćelije se dobija dodavanjem znaka za dolar ('$') ili ispred oznake kolone ili ispred oznake reda u odgovarajućoj relativnoj adresi.

Primer 1.6.4. Ćeliji `A1` odgovaraju dve kombinovane adrese: `$A1` i `A$1`. Slično, ćeliji `D6` odgovaraju dve kombinovane adrese: `$D6` i `D$6`. ♢

Prilikom kopiranja ćelije sa kombinovanom adresom u neku drugu ćeliju, posle kopiranja će druga ćelija sadržati formulu koja će biti modifikovana u odnosu na polaznu formulu, ali isključivo u delu adrese koji ispred sebe nije imao znak za dolar ('$'). Modifikacija se obavlja po istim pravilima kao i za relativno adresiranje, pa je zato neophodno je da se upoznamo i sa drugim stilom adresiranja, stilom `R1C1`, kako bismo razumeli šta se zapravo dešava prilikom kopiranja ćelija sa formulama koje sadrže relativne i kombinovane adrese.

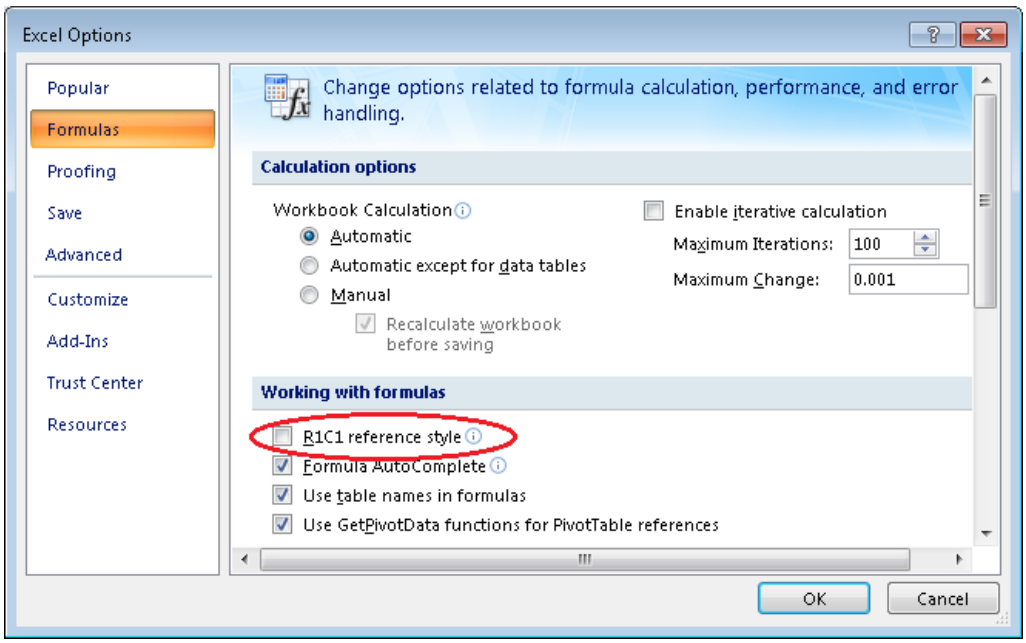
Stil `R1C1`

Stil adresiranja `R1C1` se svodi na uobičajenu numeraciju koja se u matematici koristi za elemente matrica. Prvi softverski proizvodi kompanije Microsoft koji su, kao preteča programa Excel, razvijeni kao aplikacije za tabelarna izračunavanja, su koristili stil `R1C1` za adresiranje ćelija tabele, da bi tek kasnije od popularne konkurencije preuzeli stil `A1`[3]. Excel i danas

[3]Detalji se mogu naći, na primer, na adresi `https://excelmate.wordpress.com/2013/04/22/excel-r1c1-reference-style-vs-a1/`.

ima mogućnost da adresira ćelije u stilu `R1C1`, ali je većina korisnika uglavnom navikla na stil `A1`, te se stil `R1C1` retko koristi, a mnogi korisnici i ne znaju da on postoji.

Excel u svom dijalogu za podešavanja opcija programa omogućava da se pređe sa jednog na drugi stil adresiranja jednim klikom (Slika 1.8). Takođe, privremeno možemo da uključimo opciju kojom Excel, umesto da prikazuje vrednosti formula, prikazuje same formule i da na taj način pratimo šta se dešava prilikom kopiranja ćelija sa formulama (Slike 1.9a, 1.9b 1.9c).



Slika 1.8: Dijalog za podešavanje stila adresiranja u Excel-u.

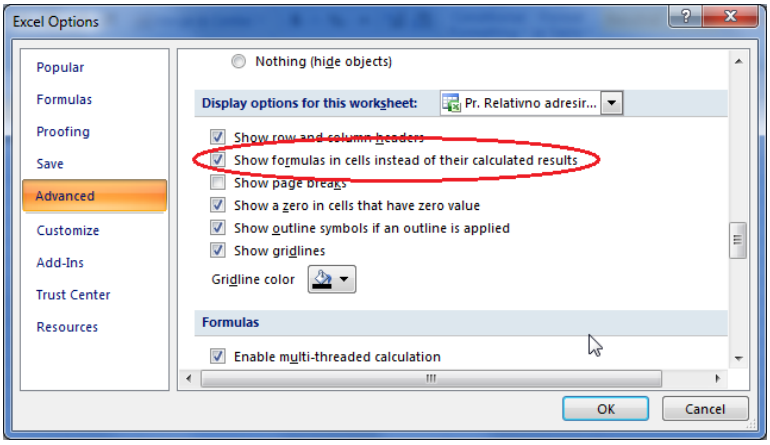
Kao i u slučaju stila `A1`, i adrese u stilu `R1C1` mogu biti apsolutne, relativne i kombinovane, ali koriste drugačiji zapis.

Četiri osnovne karakteristike stila `R1C1`, po kojima se razlikuje od stila `A1`, su:

(i) i redovi i kolone su numerisani prirodnim brojevima u stili `R1C1`;

(ii) prilikom navođenja adrese u stilu `R1C1`, prvo se navodi redni broj reda, a potom redni broj kolone;

(iii) prilikom kopiranja ćelije sa formulom u stilu `R1C1` u neku drugu ćeliju, druga ćelija će sadržati istovetnu formulu u stilu `R1C1` kao i polazna ćelija, bez obzira da li je adresa bila apsolutna, relativna ili kombinovana;



(a) Dijalog za podešavanje stila adresiranja u Excel-u.

A	B	C	D	E	F
Perić	Pera	123456	21	65	=D1+E1
Mikić	Mika	123457	35	60	=D2+E2
Lazić	Laza	123458	32	55	=D3+E3
Simić	Sima	123459	28	50	=D4+E4
Ivić	Iva	123460	25	45	=D5+E5

(b) Prikazivanje formula umesto njihovih vrednosti u stilu adresiranja `A1`.

	1	2	3	4	5	6
1	Perić	Pera	123456	21	65	=R[0]C[-2]+R[0]C[-1]
2	Mikić	Mika	123457	35	60	=R[0]C[-2]+R[0]C[-1]
3	Lazić	Laza	123458	32	55	=R[0]C[-2]+R[0]C[-1]
4	Simić	Sima	123459	28	50	=R[0]C[-2]+R[0]C[-1]
5	Ivić	Iva	123460	25	45	=R[0]C[-2]+R[0]C[-1]

(c) Prikazivanje formula umesto njihovih vrednosti u stilu adresiranja `R1C1`.

(iv) relativna i kombinovana adresa u stilu `R1C1` se ne odnose uvek na istu ćeliju, tj. istoj ćeliji mogu odgovarati različite relativne i kombinovane adrese u stilu `R1C1`, sve u zavisnosti od toga u odnosu na koju ćeliju se računaju.

Prve tri karakteristike stila `R1C1` su jednostavne za razumevanje, dok će četvrta karakteristika biti detaljno objašnjena u pododeljcima Relativne adrese ćelija u stilu `R1C1` i Kombinovane adrese ćelija u stilu `R1C1`.

Apsolutne adrese ćelija u stilu `R1C1`

Svakoj ćeliji u tabeli može se pridružiti jedinstvena **apsolutna adresa** u stilu `R1C1` koja je određena:

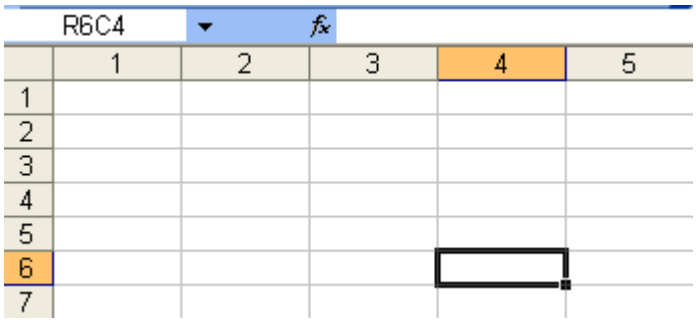
(i) nazivom radne knjige u kojoj se ćelija nalazi,

(ii) nazivom tabele (radnog lista, radne sveske) u kojoj se ćelija nalazi,

(iii) rednim brojem reda kome pripada ćelija i

(iv) rednim brojem kolone kojoj pripada ćelija.

Primer 1.6.5. Ćeliji `A1`, ćeliji u prvoj koloni i prvom redu tabele, u stilu `R1C1` odgovara upravo apsolutna adresa `R1C1`, odnosno apsolutna adresa `$A$1` u stilu `A1`. Slično, ćeliji `D6`, koja se nalazi u četvrtoj koloni i šestom redu, u stilu `R1C1` odgovara apsolutna adresa `R6C4` (Slika 1.10), odnosno apsolutna adresa `$D$6` u stilu `A1`. Prilikom kopiranja ćelija čije formula sadrže apsolutne adrese u bilo kom stilu adresiranja, sadržaj formule u kopiji je identičan formuli u originalnoj ćeliji. ♢



Slika 1.10: Apsolutna adresa ćelije `D6` u stilu adresiranja `R1C1`.

Relativne adrese ćelija u stilu R1C1

Relativne i kombinovane adrese ćelije u stilu R1C1, kao što je već napomenuto, se ne odnose uvek na istu ćeliju, već njihovo pravo značenje zavisi od ćelije u kojoj se nalaze. Kao posledica, istoj ćeliji mogu odgovarati različite relativne i kombinovane adrese, sve u zavisnosti od toga u odnosu na koju ćeliju se računaju.

Relativne adrese u stilu R1C1 su oblika:

$$\texttt{R[X][Y]} \tag{1.16}$$

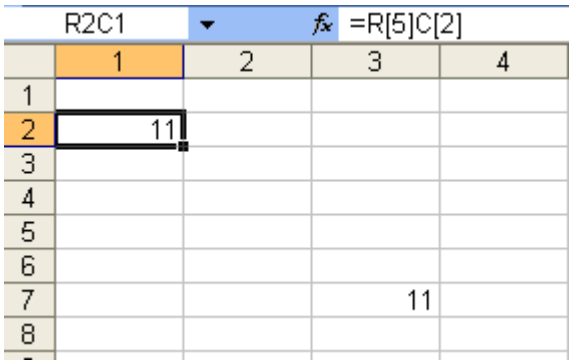
gde X i Y predstavljaju redom broj redova, odnosno broj kolona, za koji se treba pomeriti da bi se iz ćelije u kojoj se nalazi relativna adresa došlo do ćelije na koju se adresa odnosi. Za brojeve X i Y ćemo još koristiti zajedničko ime **pomeraj** (eng. **offset**). Znak pomeraja X i Y opisuje smer u kome se vrši pomeranje:

- ako je X pozitivan broj, pomeranje se vrši nadole za X redova;
- ako je X negativan broj, pomeranje se vrši nagore za |X| redova;
- ako je Y pozitivan broj, pomeranje se vrši udesno za Y redova;
- ako je Y negativan broj, pomeranje se vrši ulevo za |Y| redova.

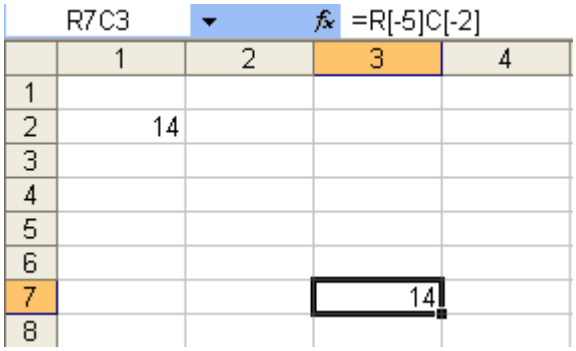
Primer 1.6.6. Razmotrimo ćelije A2 i C7. Njihove apsolutne adrese u stilu R1C1 su R2C1 i R7C3. Ukoliko se nalazimo u ćeliji A2, njena relativna adresa je R[0]C[0]. Srednje zagrade označavaju da se radi o relativnoj adresi, dok brojevi unutar srednjih zagrada ukazuju za koliko redova, odnosno kolona treba da se pomerimo da bismo došli do ćelije A2. Pošto se već nalazimo u ćeliji A2, ti brojevi su jednaki nuli, tj. uopšte nije potrebno da se pomerimo. Međutim, da bismo iz ćelije A2 (odnosno, R2C1) došli do ćelije C7 (odnosno, R7C3), potrebno je da se pomerimo pet redova nadole ($7-2=5$) i dve kolone udesno ($3-1=2$). Ukoliko se dogovorimo da smerovi udesno i nadole budu pozitivni (povećava se redni broj reda, odnosno kolone), a smerovi ulevo i nagore negativni (smanjuje se redni broj reda, odnosno kolone), tada relativnu adresu ćelije C7 u odnosu na ćeliju A2 možemo opisati relativnom adresom R[5]C[2] u stilu R1C1 (Slika 1.11a).

Pretpostavimo sada obrnuto, da se nalazimo u ćeliji C7. Tada je njena relativna adresa R[0]C[0]. Da bismo iz ćelije C7 (odnosno, R7C3) došli do ćelije A2 (odnosno, R2C1), potrebno je da se pomerimo pet redova nagore ($2-7=-5$) i dve kolone ulevo ($1-3=-2$). Poštujući dogovor da su smerovi ulevo i nagore negativni (smanjuje se redni broj reda, odnosno

kolone), tada relativnu adresu ćelije `A2` u odnosu na ćeliju `C7` možemo opisati relativnom adresom `R[-5]C[-2]` u stilu `R1C1` (Slika 1.11b). ♢



(a) Relativna adresa ćelije `C7` u odnosu na ćeliju `A2` u stilu `R1C1`.



(b) Relativna adresa ćelije `A2` u odnosu na ćeliju `C7` u stilu `R1C1`.

Međutim, najvažnija razlika u odnosu na stil `A1` prilikom kopiranja ćelija, relativne adrese u formulama u stilu `R1C1` ostaju nepromenjene.

Primer 1.6.7. Primer 1.6.6 objašnjava šta se zapravo dešava prilikom kopiranja formula u Primeru 1.6.1. Da se podsetimo, u ćeliji `F1` se nalazila formula 1.17:

```
= D1 + E1
```
(1.17)

Ukoliko se prebacimo na stil `R1C1`, u ćeliji `F1` (odnosno, `R1C6`) se nalazi formula formula 1.18:

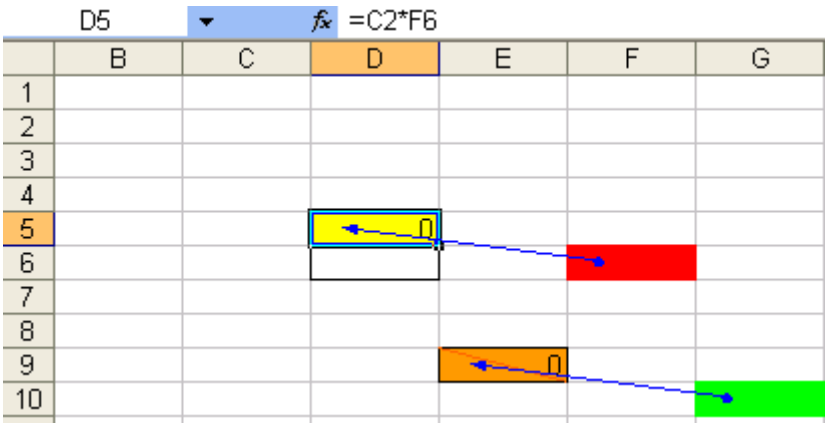
```
= R[0]C[-2] + R[0]C[-1]
```
(1.18)

Kopiranjem ćelije `F1` u ćelije opsega `F2:F100`, u svakoj od njih će se naći ista formula 1.18 (Slika1.9c). Međutim, ako protumačimo formulu 1.18 u ćeliji `F2`, na primer, i pretvorimo je u stil `A1`, videćemo da ćemo dobiti formulu

```
= D2 + E2
```
(1.19)

Naime, ćelije `D2` i `E2` su u istom (drugom) redu kao i ćelija `F2` (`R[0]`), ćelija `D2` je dve kolone levo od ćelije `F2` (`C[-2]`), dok je ćelija `E2` jednu kolonu levo od ćelije `F2` (`C[-1]`) (Slika 1.9b). ♢

Na kraju možemo da zaključimo da se prilikom kopiranja ćelije čija formula sadrži relativne adrese, bez obzira na stil adresiranja, čuvaju relativni odnosi između polazne ćelije sa formulom i ćelija na koje se referiše u samoj formuli. Konkretno, ako formula u ćeliji X (Slika 1.12, žuta ćelija) referiše na ćeliju Y (Slika 1.12, crvena ćelija) relativnom adresom i ako se ćelija X kopira u ćeliju U (Slika 1.12, smeđa ćelija), tada će se u formuli ćelije U zameniti referenca na Y referencom na ćeliju V (Slika 1.12, zelena ćelija), pri čemu su brojevi redova i kolona za koje je potrebno da se pomerimo da bismo došli iz ćelije X u ćeliju Y istovetni kao prilikom pomeranja iz ćelije U da bismo došli do ćelije V.



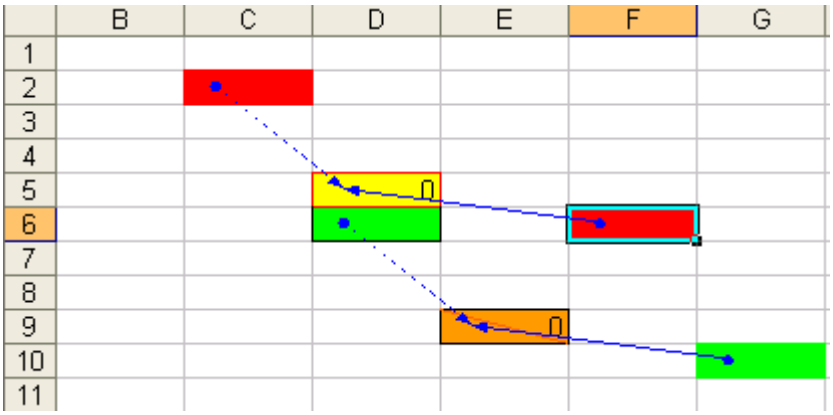
Slika 1.12: Odnos (pozicija) ćelije koja sadrži formulu i ćelije na koju se referiše relativnom adresom u toj formuli se ne menja prilikom kopiranja.

Primer 1.6.8. Neka se u ćeliji `D5` nalazi formula 1.20

```
= C2 * F6
```
(1.20)

Kako će formula 1.20 izgledati u ćeliji `E9` posle kopiranja ćelije `D5` u ćeliju `E9`?

Ćelija `D5` referiše na ćeliju `C2` koja je jednu kolonu levo i tri reda iznad ćelije `D5`. Takođe, ćelija `D5` referiše na ćeliju `F6` koja je dve kolone desno i jedan red ispod ćelije `D5`. Posle kopiranja ćelije `D5` u ćeliju `E9`, ćelija `E9` će takođe referisati na ćeliju koja je jednu kolonu levo i tri reda iznad ćelije `E9` (`D6`), a takođe će referisati i na ćeliju koja je dve kolone desno i jedan red



Slika 1.13: Kopiranje formule iz ćelije `D5` u ćeliju `E9`. Žuta boja označava originalnu ćeliju `D5` sa formulom, a crvena boja — ćelije na koje se u formuli referiše (`C2` i `F6`). Smeđa boja označava ćeliju u koju je originalna ćelija kopirana (`E9`), a zelena — ćelije na koje se referiše u formuli smeđe ćelije posle kopiranja (`D6` i `G10`).

ispod ćelije `E9` (`G10`). Prema tome, posle kopiranja ćelije `D5` u ćeliju `E9`, u ćeliji `E9` biće formula 1.21:

```
= D6 * G10
```
(1.21)

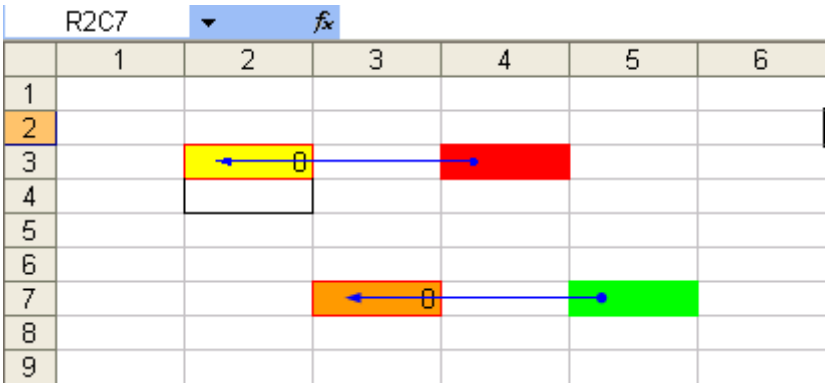
♢

Kombinovane adrese ćelija u stilu `R1C1`

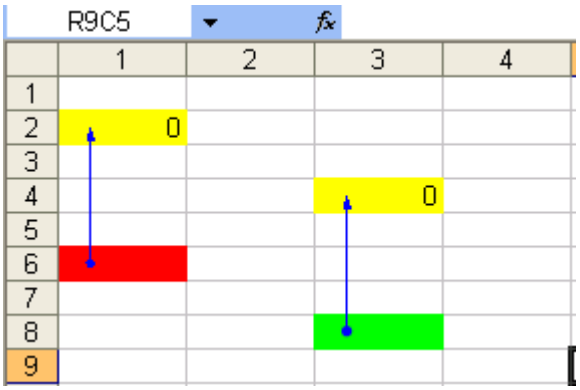
Kombinovane adrese ćelija u stilu `R1C1` su zapravo relativne adrese ćelija u stilu `R1C1` kod kojih je jedan od pomeraja jednak nuli. U primerima koje smo do sada videli (Primeri 1.6.6 i 1.6.7) sve relativne adrese u stilu `R1C1` su kombinovane, za razliku od relativnih adresa u Primeru 1.6.8 koje to nisu. Pošto su kombinovane adrese zapravo specijalan slučaj relativnih adresa, za njih važi ista pravila prilikom kopiranja. Jedina posebnost kombinovanih adresa je sledeće: ako su originalna ćelija sa formulom i ćelija na koju formula referiše bile u istom redu (koloni) pre kopiranja, kopija i ćelija na koju kopijina formula referiše će takođe biti u istom redu (koloni), kao što to ilustruju Slike 1.14 i 1.15.

Primer 1.6.9.

Primer 1.6.10.



Slika 1.14: Odnos (pozicija) ćelije koja sadrži formulu i ćelije na koju se referiše kombinovanom adresom u toj formuli se ne menja prilikom kopiranja.



Slika 1.15: Odnos (pozicija) ćelije koja sadrži formulu i ćelije na koju se referiše kombinovanom adresom u toj formuli se ne menja prilikom kopiranja.

1.7 Imena

1.8 Studija slučaja: evidentiranje pohađanja nastave

Primer: predavanja i vezbe: ukupan broj dolazaka: + i `SUM`
procentni racun
deo = ceo * procenatprocenat
procenat / 100 = deo / ceo
procenat = deo / ceo * 100
Primer: dolazenje na predavanja u procentima dolazenje na vezbe u procentima dolazenje na nastavu u procentima

Glava 2

Tipovi podataka

Pod tipom podataka se podrazumeva konačan skup vrednosti (domen) i operacija koje se mogu primeniti nad vrednostima domena. Iako u Excel-u postoji više tipova podataka, detaljnije ćemo razmotriti:

- numerički tip podataka,
- tekstualni tip podataka,
- logički tip podataka i
- datum i vreme kao tipove podataka.

2.1 Numerički tip podataka

Numerički tip podataka omogućava rad sa brojevima. Za razliku od karakterističnih skupova brojeva u matematici, poput skupa prirodnih ($\mathbb{N}$), celih ($\mathbb{Z}$), racionalnih ($\mathbb{Q}$), realnih ($\mathbb{R}$) i kompleksnih brojeva ($\mathbb{C}$), skup numeričkih vrednosti sa kojima Excel može da operiše je konačan i njegova veličina zavisi od broja bajtova koje Excel koristi za predstavljanje jednog broja. U daljem izlaganju nećemo se baviti kompleksnim brojevima.

Vrednosti svih navedenih skupova brojeva sem kompleksnih ($\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$) sa kojima Excel može da operiše su predstavljene kao realni brojevi, korišćenjem notacije pokretnog zareza za interni zapis broja. Excel pak prikazuje numeričku vrednost u notaciji koju izabere korisnik, što može biti i notacija nepokretnog zareza. Tako se ista vrednost, na primer, ceo broj 6, može prikazati u Excel-u i kao 6 (realan broj sa nula decimala) i kao 6,00 (realan broj sa dve decimale).

U matematici se skupovi celih i realnih brojeva razlikuju po operacijama koje su nad tim skupovima definisane. Zajedničke su im operacije sabiranja,

oduzimanja i množenja koje se u Excel-u realizuju pomoću operatora $+$, $-$ i $*$. U slučaju deljenja postoje značajne razlike. Nad skupom celih brojeva su definisane dve operacije koje odgovaraju **celobrojnom deljenju**, **količnik pri celobrojnom deljenju** i **ostatak pri celobrojnom deljenju**. S druge strane, nad skupom realnih brojeva je definisana samo jedna operacija koja odgovara **realnom deljenju**.

Primer 2.1.1. Razmotrimo deljenje broja 14 brojem 4. Ukoliko je reč o celobrojnom deljenju, realizuju se dve operacije, količnik (rezultat je vrednost 3) i ostatak (rezultat je 2), pošto je $14 = 3 * 4 + 2$. Međutim, ukoliko se radi o realnom deljenju, realizuje se svega jedna operacija i njen rezultat je $14/4 = 3{,}5$. ♢

Količnik i ostatak pri celobrojnom deljenju se tim redom realizuju u Excel-u pomoću binarnih funkcija `QUOTIENT` i `MOD` čiji je prvi argument deljenik, a drugi delilac. Očekivalo bi se da oba argumenta ovih funkcija moraju biti celi brojevi. Međutim, u Excel-u su svi brojevi predstavljeni kao realni, pa Excel dozvoljava da argumenti funkcija `QUOTIENT` i `MOD` budu proizvoljni realni brojevi, ali se prilikom poziva funkcije realni argumenti funkcija konvertuju u ceo broj odbacivanjem svih decimala kako bi izračunavanje ovih funkcija imalo smisla. Isti princip važi i za ostale funkcije čije matematičke definicije zahtevaju celobrojne argumente (na primer, funkcije za računanje faktorijela, broja varijacija i kombinacija, itd.).

Primer 2.1.2. Celobrojno deljenje (Primer 2.1.1) se može realizovati pomoću formula 2.1 (količnik) i 2.2 (ostatak):

$$= \texttt{QUOTIENT(14; 4)} \tag{2.1}$$

$$= \texttt{MOD(14; 4)} \tag{2.2}$$

♢

Procenti

Procenti predstavljaju poseban način zapisivanja realnih brojeva. Notacija za procente oblika $x\%$, gde je x realan broj, je zapravo drugačiji zapis broja $\frac{x}{100}$. Procentima se opisuje odnos nekog dela prema celini na takav način što se celina podeli na 100 jednakih komponenti, a onda se registruje koliko posmatrani deo sadrži tih komponenti. Drugim rečima, neki deo predstavlja $x\%$ celine ukoliko važi 2.3:

$$\frac{deo}{celina} = \frac{x}{100} = x\% \tag{2.3}$$

Iz jednačine 2.3 se jednostavno izvode formule na osnovu kojih se izračunava koliko iznosi deo neke celine na osnovu svog procentnog udela u toj celini (2.4), kao i vrednost celine, ako je poznat neki njen deo i procentualni udeo tog dela u odnosu na celinu (2.5).

$$deo = celina * x\% = celina * \frac{x}{100} \tag{2.4}$$

$$celina = \frac{deo}{x\%} = deo * \frac{100}{x} \tag{2.5}$$

Primer 2.1.3. Neka student treba da prisustvuje na bar 75% ukupno održanih časova da bi dobio 5 poena za pohađanje nastave.

a) Ako je ukupno održano 12 časova, koliko najmanje puta je student morao da prisustvuje nastavi da bi osvojio odgovarajućih 5 poena?

U ovom slučaju $celina = 12$, a $x\% = 75\%$. Na osnovu formule (2.4) sledi da je student morao da pohađa bar

$deo = celina * x\% = 12 * 75\% = 12 * 75/100 = 12 * 3/4 = 3 * 3 = 9$ časova.

b) Ako je student dolazio na svega 6 časova i ima 5 poena za pohađanje nastave, koliko je najviše časova ukupno održano?

U ovom slučaju $deo = 6$, a $x\% \geq 75\%$. Na osnovu formule (2.5) sledi da je ukupno održano najviše

$$celina = \frac{deo}{x\%} \leq \frac{6}{75\%} = 6 * \frac{4}{3} = 2 * 4 = 8 \text{ časova.}$$

♢

2.2 Tekstualni tip podataka

Da bismo preciznije definisali tekstualni tip podataka, moramo najpre uvesti nekoliko osnovnih pojmova teorije formalnih jezika.

Definicija 2.2.1. Pod **(formalnom) azbukom** podrazumevamo proizvoljan konačan skup simbola. Elemente (simbole) azbuke nazivamo **slovima**.

Definicija 2.2.2. Označimo sa Σ proizvoljnu azbuku. **Niska nad azbukom** Σ je konačan niz simbola azbuke Σ, tj. ako je $x = x_1 x_2 \cdots x_n$, gde su $x_1, x_2, \ldots, x_n$ slova azbuke Σ i $n \geq 0$. Broj simbola n predstavlja **dužinu niske** x. Dužinu niske x označavaćemo sa $|x|$.

Definicija 2.2.3. Niska dužine nula se naziva **prazna niska** i označava sa ε, tj. $|\varepsilon| = 0$.

Praznu nisku u Excel-u ćemo još označavati sa "". Između navodnika nema ničega, dakle, ni razmaka. U slučajevima kada je potrebno naglasiti da se na nekoj poziciji u niski nalazi razmak, koristićemo simbol $\lrcorner$. Treba napomenuti da niska koja se sastoji samo od razmaka nije prazna niska, iako na ekranu ćelija koje sadrži praznu nisku i ćelije koje sadrže isključivo niz razmaka izgledaju isto. Naime, niska "$\lrcorner\lrcorner\lrcorner$" (tri razmaka) je dužine tri, dok po definiciji prazna niska ima dužinu 0, tj. ne sadrži nijedno slovo.

Ako je Σ azbuka i d prirodan broj ($d \geq 0$), tada ćemo:

- skup svih niski nad azbukom Σ, čija je dužina tačno d, označavati sa Σ^d;
- skup svih niski nad azbukom Σ označavati sa Σ^*;
- skup svih nepraznih niski nad azbukom Σ označavati sa Σ^+.

Primer 2.2.1. Neka je $\Sigma = \{a,b,c\}$ azbuka. Tada se skup niski nad azbukom Σ može opisati na sledeći način:

$$\left\{\begin{array}{ll} \varepsilon & \} \Sigma^0, \text{ tj. niske dužine 0} \\ a,b,c & \} \Sigma^1, \text{ tj. niske dužine 1} \\ \begin{array}{l} aa,ab,ac, \\ ba,bb,bc, \\ ca,cb,cc \end{array} & \Big\} \Sigma^2, \text{ tj. niske dužine 2} \\ \begin{array}{l} aaa,aab,aac,aba,abb,abc,aca,acb,acc, \\ baa,bab,bac,bba,bbb,bbc,bca,bcb,bcc \\ caa,cab,cac,cba,cbb,cbc,cca,ccb,ccc, \\ \quad \vdots \end{array} & \Big\} \Sigma^3, \text{ tj. niske dužine 3} \end{array}\right.$$

Ovaj primer pokazuje da, iako je azbuka Σ konačna, niski nad tom azbukom ima beskonačno mnogo, jer iako je dužina svake niske konačna, ničim nije ograničena, tj. može biti proizvoljan prirodan broj. $\diamondsuit$

Definicijom 2.2.2 se zapravo precizira i uopštava termin koji nam je dobro poznat — *reč*. Međutim, umesto termina *reč*, radije ćemo koristiti termin *niska*. Postoji više razloga za izbegavanje termina *reč*, pre svega, njegova višeznačnost (reč kao oblik, reč kao leksema, reč kao vrsta reči,

itd.). S druge strane, ako uočimo azbuku koju čine slova srpske abecede i proširimo je simbolom za razmak, tada bi zvučalo nekorektno da se kaže da je tekst poput `"Crna Gora"` *reč*, kada se to inače u jeziku tretira kao tekst sačinjen od dve reči. Jezički izraz ‚niska `"Crna Gora"`' nema tih problema, a pojam *niska* čak i podseća na skraćenicu za izraz *NIZ KAraktera* (uz odgovarajuće jednačenje po zvučnosti).

Domen tekstualnog tipa podataka u Excel-u predstavlja skup niski nad računarskom azbukom, tj. azbukom sastavljenom od svih simbola (karaktera) koji se mogu proizvesti sa tastature[1]. Jedina operacija na ovom domenu je operacija dopisivanja koja se u Excel-u realizuje na dva načina:

- kao binarni operator & i
- kao funkcija `CONCATENATE`.

Binarni operator & se koristi na sličan način kao operator sabiranja (`+`). Prilikom navođenja konkretnih niski u formuli, moraju se koristiti navodnici kako bi se jasno razlikovalo šta su u formuli reference (adrese ćelija ili imena), a šta su konkretne niske (v. Primer 2.2.2). Tekstualni podaci se, kao sadržaj ćelije, navode bez navodnika, sem ako su i sami navodnici deo teksta.

Primer 2.2.2. Neka se u prvih 10 redova tabele nalaze imena i prezimena studenata, pri čemu su imena u koloni A, a prezimena u koloni B. Pretpostavimo da je potrebno u koloni C, u odgovarajućem redu, generisati ime i prezime studenta u formatu `Prezime,␣Ime`. Konkretno, ako se u ćeliji `A1` nalazi niska `"Pera"` (bez navodnika), a u ćeliji `B1` niska `"Mitić"`, tada je potrebno da se u ćeliji `C1` generiše tekst `"Mitić,␣Pera"` (bez navodnika), što se može postići formulom 2.6:

```
= B1 & ",␣" & A1
```
(2.6)

Isti efekat se može postići formulom 2.7:

```
= CONCATENATE(B1 ; ",␣" ; A1)
```
(2.7)

U praksi se uglavnom koristi operator &, dok je funkcija `CONCATENATE` pogodna kada treba spojiti više niski i želi se izbeći ponavljanje oznake operatora &. ♢

[1] U pitanju su zapravo elementi karakterskog skupa Unicode, o kome će biti više reči u okviru predmeta Digitalni tekst 1. Za početnika je dovoljno da zna da azbuku Excel-a čine simboli koji se koriste u svakodnevnoj komunikaciji: slova azbuka prirodnih jezika (srpskog, engleskog,...), cifre, znaci interpunkcije, itd.

Podniske

Neka je Σ azbuka i neka je x niska nad azbukom Σ dužine n. Ako sa x_i označimo simbol niske x na poziciji i $(1 \leq i \leq n)$, tada se niska x može predstaviti u obliku

$$x = x_1 x_2 \cdots x_n.$$

Ako se niska x tretira kao niz n simbola, možemo posmatrati i njene podnizove, tj. nizove oblika

$$x_{j_1} x_{j_2} \cdots x_{j_k},$$

takvih da je $1 \leq k \leq n$ i $1 \leq j_1 < j_2 \cdots < j_k \leq n$. Podniz niza x je zapravo niz sastavljen od nekih (ne nužno svih) elemenata polaznog niza x, koji se u podnizu pojavljuju u istom redosledu kao i u polaznom nizu x.

Poseban značaj imaju podnizovi niske x, sastavljeni isključivo od uzastopnih simbola niske x, a koji se nazivaju **podniske**. Takođe, prazna niska je, po definiciji, podniska svake niske.

Primer 2.2.3. Neka je data niska $x =$ `"Perica"`. Ona se može posmatrati kao niz šest simbola. Neki podnizovi niske x su:

(i) P,	(vi) Prc,	(xi) Pera,
(ii) a,	(vii) Pra,	(xii) eri,
(iii) Per,	(viii) Peri,	(xiii) eric,
(iv) ica,	(ix) rica,	(xiv) eca,
(v) Prica,	(x) rca,	(xv) Perica.

Među navedenim podnizovima (i)–(xv) niske x, samo su (i), (ii), (iii), (iv), (viii), (ix), (xii), (xiii), (xv) istovremeno i podniske, pošto se radi o podnizovima uzastopnih simbola niske x. ♢

Podniska kojom niska x počinje se naziva **prefiks** niske x, dok se podniska kojom se niska x završava naziva **sufiks** niske x. Podniska niske x, takva da x niti počinje niti se završava tom podniskom, se naziva **infiks** niske x.

Podniske se mogu i preciznije definisati.

Definicija 2.2.4. Neka je x niska nad azbukom Σ. Za nisku $v \in \Sigma^*$ kažemo da je:

- **prefiks** ili **levi faktor** niske x, ako postoji niska $w \in \Sigma^*$ takva da je $x = vw$;
- **sufiks** ili **desni faktor** niske x, ako postoji niska $u \in \Sigma^*$ takva da je $x = uv$;
- **podniska** ili **faktor** niske x, ako postoje niske u,$w \in \Sigma^*$ takve da je $x = uvw$;
- **infiks** niske x, ako je v podniska niske x i nije ni prefiks ni sufiks niske x.

Primer 2.2.4. Uočimo ponovo nisku $x =$ `"Perica"`. Njene podniske, prefiksi, sufiksi i infiksi, mogu redom da se opišu skupovima P, S i I:

$$P = \{\varepsilon, \text{P}, \text{Pe}, \text{Per}, \text{Peri}, \text{Peric}, \text{Perica}\} \tag{2.8}$$
$$S = \{\varepsilon, \text{a}, \text{ca}, \text{ica}, \text{rica}, \text{erica}, \text{Perica}\} \tag{2.9}$$
$$I = \{\varepsilon, \text{e}, \text{r}, \text{i}, \text{c}, \text{er}, \text{ri}, \text{ic}, \text{eri}, \text{ric}, \text{eric}\} \tag{2.10}$$

♢

Primetimo da su prazna niska i sama niska x uvek i prefiksi i sufiksi niske x. Prefiksi niske x, koji nisu jednaki niski x, nazivaju se **pravi prefiksi** niske x. Slično, sufiksi niske x, koji nisu jednaki niski x, nazivaju se **pravi sufiksi** niske x.

Excel nudi više funkcija koje se mogu koristiti za izračunavanje podniski (v. odeljak 4.5).

2.3 Logički tip podataka

Domen logičkog tipa podataka čine dve logičke konstante: `TRUE` i `FALSE`. Operacije nad ovim tipom se realizuju kao logičke funkcije `AND` (konjunkcija), `OR` (disjunkcija) i `NOT` (negacija), koje su detaljno opisane u odeljku 4.2.

2.4 Datum i vreme kao tipovi podataka

Glava 3

Izrazi

3.1 Aritmetički izrazi

Aritmetički izrazi su dobili naziv po tome što je rezultat njihovog izračunavanja (vrednost izraza) — numerički podatak (broj).

Definicija 3.1.1. Aritmetički izrazi se definišu na sledeći način:

(i) Celobrojne i realne konstante su aritmetički izrazi.

(ii) Adrese ćelija koje sadrže aritmetičke izraze su aritmetički izrazi.

(iii) Pozivi aritmetičkih funkcija su aritmetički izrazi (v. odeljak 4.1, str. 38).

(iv) Ako su E_1 i E_2 aritmetički izrazi, tada su i

- $E_1 + E_2$ (zbir),
- $E_1 - E_2$ (razlika),
- $E_1 * E_2$ (proizvod),
- E_1/E_2 (količnik pri realnom deljenju),
- $E_1\hat{}E_2$ (stepenovanje),
- (E_1) (izraz u zagradama),
- $+E_1$ (potvrda znaka) i
- $-E_1$ (promena znaka)

takođe aritmetički izrazi.

(v) Ništa osim (i)–(iv) nije aritmetički izraz.

3.2 Relacijski izrazi

Relacijski izrazi, kao što im i naziv govori, uspostavljaju određeni odnos (relaciju) između dva aritmetička izraza.

Definicija 3.2.1. Ako E_1 i E_2 predstavljaju dva aritmetička izraza, tada svaki od narednih izraza predstavlja jedan relacijski izraz:

(i) $E_1 = E_2$ (upoređivanje da li su vrednosti dva aritmetička izraza jednake),

(ii) $E_1 <> E_2$ (upoređivanje da li su vrednosti dva aritmetička izraza različite),

(iii) $E_1 < E_2$ (upoređivanje da li je vrednost prvog aritmetičkog izraza manja od vrednosti drugog),

(iv) $E_1 <= E_2$ (upoređivanje da li je vrednost prvog aritmetičkog izraza manja ili jednaka vrednosti drugog),

(v) $E_1 > E_2$ (upoređivanje da li je vrednost prvog aritmetičkog izraza veća od vrednosti drugog) i

(vi) $E_1 >= E_2$ (upoređivanje da li je vrednost prvog aritmetičkog izraza veća ili jednaka vrednosti drugog).

(vii) Ništa osim (i)–(vi) nije relacijski izraz.

Operatori =, <>, <, <=, >, >= se nazivaju relacijskim operatorima i odgovaraju uobičajenim matematičkim relacijama upoređivanja $=$, $\neq$, $<$, $\leq$, $>$, $\geq$.

Relacijski izrazi se obično koriste prilikom provere ispunjenosti određenih uslova (v. funkciju IF, odeljak 4.4, str. 43). Vrednost relacijskog izraza je jedna od dve moguće logičke vrednosti: `TRUE` („tačno”, „istina”, matematička logička konstanta $\top$) ili `FALSE` („netačno”, „laž”, matematička logička konstanta $\bot$).

3.3 Logički izrazi

Logički izrazi su dobili naziv po tome što je njihova vrednost ili logička vrednost `TRUE` („tačno”, „istina”, matematička logička konstanta $\top$) ili logička vrednost `FALSE` („netačno”, „laž”, matematička logička konstanta $\bot$).

Definicija 3.3.1. Logički izrazi se formalno definišu na sledeći način:

(i) Logičke konstante `TRUE` i `FALSE` su logički izrazi.

(ii) Adrese ćelija čije su vrednosti logički izrazi su takođe logički izrazi.

(iii) Relacijski izrazi su logički izrazi.

(iv) Pozivi logičkih funkcija su logički izrazi (v. odeljak 4.2, str. 38).

(v) Ništa osim (i)–(iv) nije logički izraz.

3.4 Tekstualni izrazi

Tekstualni izrazi imaju za vrednost nisku, odnosno tekst, po čemu su i dobili naziv.

Definicija 3.4.1. Tekstualni izrazi se formalno definišu na sledeći način:

(i) Niske (tekstualne konstante) su tekstualni izrazi.

(ii) Adrese ćelija čije su vrednosti tekstualni izrazi su takođe tekstualni izrazi.

(iii) Ako su T_1 i T_2 tekstualni izrazi, tada je T_1 & T_2 takođe tekstualni izraz.

(iv) Pozivi tekstualnih funkcija čija je vrednost niska su tekstualni izrazi (v. odeljak 4.5, str. 43).

(v) Ništa osim (i)–(iv) nije tekstualni izraz.

3.5 Izrazi i formule

Posle definicija datih u odeljcima 3.1–3.3, konačno možemo dati preciznu definiciju formula.

Definicija 3.5.1. Izrazi u Excel-u se definišu na sledeći način:

(i) Aritmetički izrazi su izrazi.

(ii) Logički izrazi (a time i relacijski izrazi) su izrazi.

(iii) Tekstualni izrazi su izrazi.

(iv) Ništa osim (i)–(iii) nije izraz.

Definicija 3.5.2. Ako je I izraz u Excel-u, tada je svaki zapis oblika

$$= I$$

formula u Excel-u.

Da bi Excel automatski interpretirao formulu, tj. izračunao njenu vrednost, ćelija u kojoj se nalazi zapis formule mora biti formatirana kao ćelija opšte kategorije (eng. General).

Glava 4

Pregled funkcija

U ovom poglavlju biće predstavljene najčešće korišćene funkcije u programu Excel. Funkcije su grupisane u kategorije u zavisnosti:

- od tipa vrednosti koju izračunavaju kao rezultat ili
- od tipa argumenata funkcije, ukoliko se funkcija primenjuje isključivo nad argumentima specifičnog tipa.

Na primer, funkcija `LEN` (v. odeljak 4.5), iako izračunava numeričku vrednost, te se može smatrati za aritmetičku (numeričku) funkciju, uglavnom se ubraja u tekstualne funkcije, pošto se primenjuje isključivo na argument tekstualnog tipa (nisku).

Za svaku funkciju biće navedeno:

- ime funkcije,
- broj i tip argumenata,
- tip vrednosti rezultata koji funkcija izračunava,
- primer upotrebe funkcije.

Broj argumenata pojedinih funkcija nije fiksiran, ali je ograničen odozgo maksimalnim brojem argumenata koje dozvoljava verzija programa Excel. Na primer, verzija iz 2003. godine je dozvoljavala najviše 30 argumenata, dok je verzija iz 2007. godine povećala taj broj na 255, mada su oba ograničenja sasvim dovoljna većini korisnika Excel-a.

U svim primerima je korišćena pretpostavka da je tačka zapeta (';') podrazumevani separator argumenata funkcije.

4.1 Aritmetičke funkcije

Funkcija SUM

Funkcija PROD

Funkcija SQRT

Funkcije za prebrojavanje (COUNT, COUNTA, COUNTB)

Funkcija AVERAGE

4.2 Logičke funkcije

U matematici se sve funkcije čijim pozivom se kao rezultat dobija logička vrednost `TRUE` ili `FALSE`, nazivaju logičke funkcije.

U Excel-u se logičkim funkcijama nazivaju samo uopštenja matematičkih operacija koje su definisane nad iskazima u okviru iskaznog računa (iskazne logike): konjunkcije ($\wedge$), disjunkcije ($\vee$) i negacije ($\neg$).

Ostale funkcije u Excel-u čija je vrednost logička se uglavnom svrstavaju u kategoriju informativnih funkcija (na primer, `ISBLANK`).

Funkcija `AND`

Funkcija `AND` je uopštenje iskazne operacije konjunkcije ($\wedge$).

Argumenti funkcije `AND` su logički izrazi i njihov broj nije fiksiran.

Vrednost funkcije `AND` je logička konstanta `TRUE` ako i samo ako je vrednost svakog njenog argumenta `TRUE`. Ako je vrednost bar jednog argumenta funkcije `AND` jednaka `FALSE`, vrednost funkcije je takođe `FALSE`.

Funkcija `AND` je pogodna za proveru da li je svaki uslov iz zadatog skupa uslova zadovoljen.

Primer 4.2.1. Pretpostavimo da je student položio ispit ukoliko su ispunjeni svi uslovi koji slede:

(i) student je položio praktikum (ima bar 8 od mogućih 15 poena),

(ii) student je položio pismeni (ima bar 20 od mogućih 40 poena),

(iii) student je ukupno ostvario bar 51 poen.

Neka se u ćelijama C1, D1 i E1 nalaze tim redom poeni koje je student ostvario na praktikumu, pismenom ispitu i ukupno. U ćeliji F1 treba da se upiše logička vrednost TRUE ili FALSE u zavisnosti od toga da li je student položio ispit ili ne. Koristeći funkciju AND, to se može realizovati formulom 4.1:

```
= AND(C1 >= 8 ; D1 >= 20 ; E1 >= 51)          (4.1)
```

♢

Funkcija OR

Primer 4.2.2. Ako ponovo razmotrimo Primer 4.2.1, tada možemo reći da je student pao ispit ukoliko je ispunjen bar jedan od uslova koji slede:

(i) student je pao praktikum (ima manje od 8 poena),

(ii) student nije položio pismeni (ima manje od 20 poena),

(iii) student je ukupno ostvario broj poena manji od 51.

Neka se u ćelijama C1, D1 i E1 i dalje nalaze tim redom poeni koje je student ostvario na praktikumu, pismenom ispitu i ukupno. U ćeliji G1 treba da se upiše logička vrednost TRUE, ako je student pao ispit, odnosno FALSE, ako je student položio ispit. Koristeći funkciju OR, to se može realizovati formulom 4.2:

```
= OR(C1 < 8 ; D1 < 20 ; E1 < 51)              (4.2)
```

♢

Funkcija NOT

Funkcija NOT odgovara iskaznoj operaciji negacije ($\neg$). Ima tačno jedan argument, logički izraz. Vrednost funkcije je logička konstanta, TRUE ili FALSE, koja je jednaka suprotnoj (komplementarnoj) vrednosti ulaznog argumenta. Jednostavno rečeno, ako je vrednost argumenta funkcije NOT jednaka TRUE, vrednost funkcije je FALSE, odnosno, ako je vrednost argumenta funkcije NOT jednaka FALSE, vrednost funkcije je TRUE.

U praksi se funkcija NOT uglavnom izbegava, jer se relacijski izrazi, kao najprostiji logički izrazi, uvek mogu preformulisati tako da ne koriste funkciju NOT:

- NOT(A1 = B1) ima isto značenje kao i A1 <> B1;

- `NOT(A1 <> B1)` ima isto značenje kao i `A1 = B1`;
- `NOT(A1 > B1)` ima isto značenje kao i `A1 <= B1`;
- `NOT(A1 < B1)` ima isto značenje kao i `A1 >= B1`;
- `NOT(A1 >= B1)` ima isto značenje kao i `A1 < B1`;
- `NOT(A1 <= B1)` ima isto značenje kao i `A1 > B1`.

4.3 IF-funkcije

IF-funkcijama nazivaćemo grupu funkciju koje koriste uslovno izračunavanje. Uslovno izračunavanje podrazumeva da se vrednost funkcije u opštem slučaju može izračunati na više načina, pri čemu jedan argument funkcije, logički izraz, predstavlja uslov koji svojom vrednošću (`TRUE` ili `FALSE`) utiče kako će biti izabran načina izračunavanja vrednosti funkcije.

4.4 IF-funkcija

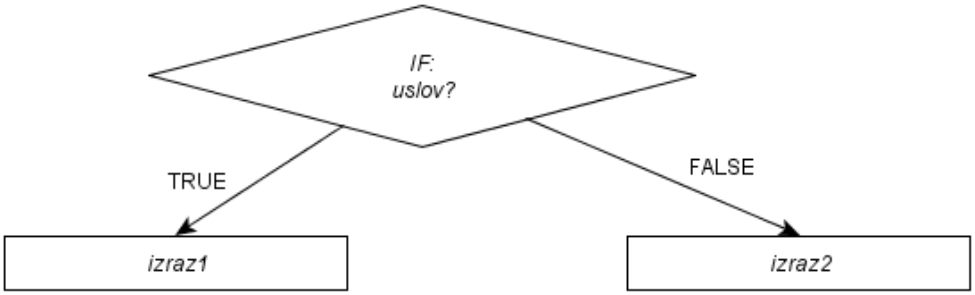
Funkcija `IF` ima tri argumenta. U opštem slučaju, poziv funkcije `IF` ima oblik:

$$\texttt{IF}(uslov;\ izraz_1;\ izraz_2) \tag{4.3}$$

Prvi argument funkcije `IF` *uslov* je logički izraz (v. odeljak 3.3) od čije vrednosti zavisi vrednost kompletnog izraza 4.3. Ako je vrednost argumenta *uslov* `TRUE`, tada je vrednost izraza 4.3 jednaka vrednosti drugog argumenta $izraz_1$. U protivnom, ako je vrednost argumenta *uslov* `FALSE`, tada je vrednost izraza 4.3 jednaka vrednosti trećeg argumenta $izraz_2$.

Izračunavanje vrednosti funkcije `IF` se može predstaviti razgranatom strukturom (Slika 4.1). U zavisnosti od vrednosti izraza *uslov*, bira se grana koja definiše kako će se izračunati vrednost funkcije. Prilikom rešavanja problema sa uslovnim izračunavanjem, ponekad je lakše prvo problem rešiti grafički, pa tek onda napisati odgovarajuću formulu (v. Primer 4.4.2).

Primer 4.4.1. Neka se u prvih 10 redova tabele nalaze podaci o studentima, pri čemu svaki red sadrži podatke jednog studenta. Neka je u koloni `E` za svakog studenta dat ukupan broj časova na kojima je student prisustvovao tokom jednog semestra. U koloni `F` je potrebno izračunati poene koje student dobija na osnovu pohađanja nastave po sledećem pravilu: ako je student bio prisutan na 75% ukupno održanih časova, dobija 5 poena, inače



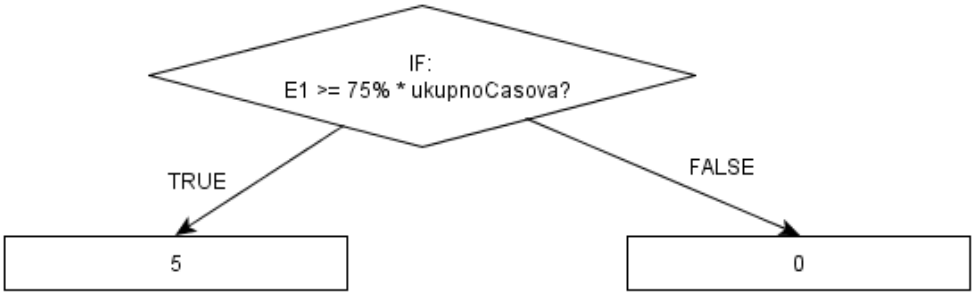
Slika 4.1: Grafički prikaz uslovnog izračunavanja u slučaju funkcije `IF`.

dobija 0 poena. Pretpostavimo da je u istoj radnoj knjizi u Excel-u definisano ime (konstanta) `ukupnoCasova` čija je vrednost jednaka ukupnom broju održanih časova. Tada se zadati problem za prvog studenta može rešiti formulom 4.4 u ćeliji `F1`, a za ostale studente kopiranjem te formule u ćelije opsega `F2:F10`:

$$= \texttt{IF(E1 >= 75\% * ukupnoCasova ; 5 ; 0)} \tag{4.4}$$

♢

Grafičko rešenje problema ilustruje Slika 4.2.



Slika 4.2: Uslovno izračunavanje poena za pohađanje nastave.

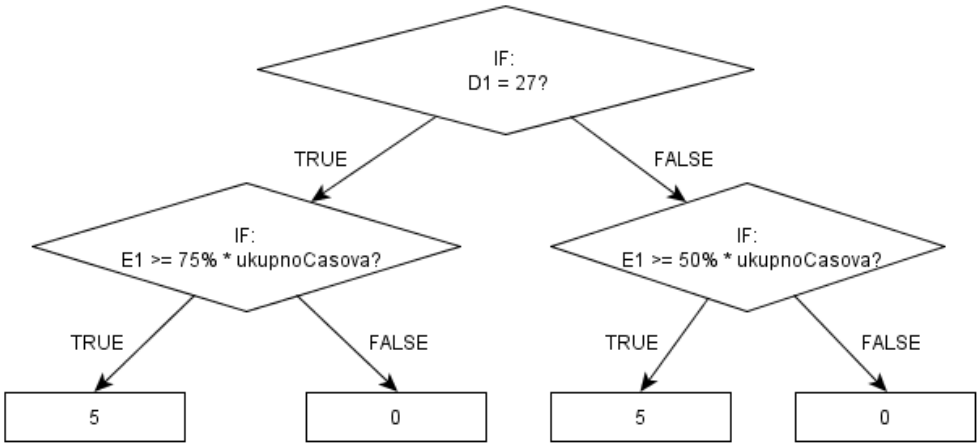
Primer 4.4.2. Razmotrimo ponovo zadatak iz Primera **??** uz sledeću izmenu. Neka se, pored već navedenih podataka, u koloni `D` nalaze brojevi koji predstavljaju neku od grupa na Filološkom fakultetu u Beogradu (na primer, broj 27 predstavlja grupu Bibliotekarstvo i informatika). Neka je i dalje potrebno izračunati poene koje student dobija na osnovu pohađanja nastave u koloni `F`, ali na osnovu izmenjenog pravila:

- ako je student sa grupe 27, dobija 5 poena ako je bio prisutan na 75% ukupno održanih časova, inače dobija 0 poena;

- ako je student sa neke druge grupe na Filološkom fakultetu, dobija 5 poena ako je bio prisutan na 50% ukupno održanih časova, inače dobija 0 poena;

Takođe, ponovo pretpostavimo da je u istoj radnoj knjizi u Excel-u definisano ime (konstanta) `ukupnoCasova` čija je vrednost jednaka ukupnom broju održanih časova.

S obzirom na složenost zadatka, u ovom slučaju je jednostavnije da se problem najpre reši grafički (Slika 4.3). Problem je daleko složeniji od Primera 4.4.1, pošto se posle prvog grananja (u zavisnosti od toga da li je studentova grupa 27 ili ne) ponovo pojavljuje grananje (u zavisnosti da li student ima minimum dolazaka na nastavu), kako bi se konačno izračunalo koliko student treba da dobije poena za pohađanje nastave. U ovom primeru postoje četiri načina da se izračuna vrednost funkcije `IF`, pri čemu su drugi i treći argument prvog poziva funkcije `IF` takođe (ugnježdeni) pozivi funkcije `IF` (naravno, sa drugim argumentima).



Slika 4.3: Uslovno izračunavanje poena za pohađanje nastave, posebno za bibliotekare (grupu 27) i posebno za ostale grupe.

Na osnovu Slike 4.3 sledi se zadati problem za prvog studenta može rešiti formulom 4.5 u ćeliji `F1`, a za ostale studente kopiranjem te formule u ćelije opsega `F2:F10`. Radi jasnijeg pregleda, formula 4.5 je razbijena u

više redova, a u Excel-u se svakako mora otkucati u jednom redu.

```
= IF(D1 = 27 ;
     IF(E1 >= 75% * ukupnoCasova ; 5 ; 0) ;
     IF(E1 >= 50% * ukupnoCasova ; 5 ; 0)
    )
```
(4.5)

♢

Formula 4.5 je, takođe, primer kompozicije funkcija, tj. drugi i treći argument spoljašnjeg poziva funkcije `IF` su takođe pozivi te iste funkcije sa drugim argumentima.

Funkcije SUMIF i SUMIFS

Funkcije COUNTIF i COUNTIFS

Funkcija IF

4.5 Funkcije za rad sa tekstom

Dopisivanje (funkcija CONCATENATE i operator &)

Funkcija LEN

Funkcije LEFT, RIGHT i MID

Funkcija SEARCH

Funkcije LOWER, UPPER i PROPER

4.6 Funkcije za rad sa datumom i vremenom

(DATE, TODAY, DAY, MONTH, YEAR) (TIME, NOW, HOUR, MINUTE, SECOND)

4.7 Kompozicija funkcija

Primena u kombinatorici i statistici

FACT(faktorijel) Funkcije PERMUT (varijacije bez ponavljanja i permutacije) COMBIN (kombinacije bez ponavljanja)