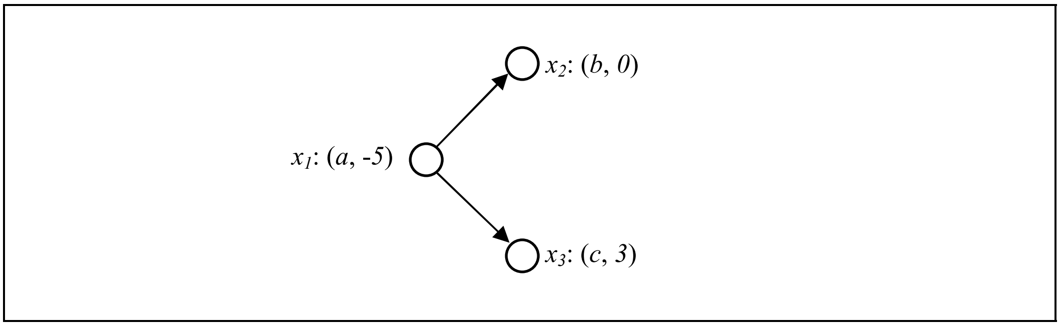
Glava 1.

Klasifikacija struktura podataka

Struktura podataka je orijentisani graf sa funkcijama koje njegovim čvorovima i ivicama pridružuju semantiku. Za graf $G = (S, R)$, gde je S konačan skup, a R relacija u tom skupu, se kaže da ima ***semantički određene čvorove*** nad skupom V, ako postoji funkcija $f : S \to V$. Isto tako, graf ima ***semantički određene ivice*** nad skupom V', ako postoji funkcija $g : R \to V'$. Može postojati više funkcija pridruživanja semantike čvorovima, ili ivicama grafa nad različitim skupovima, na primer $f_1 : S \to V_1, f_2 : S \to V_2, ..., f_n : S \to V_n$.

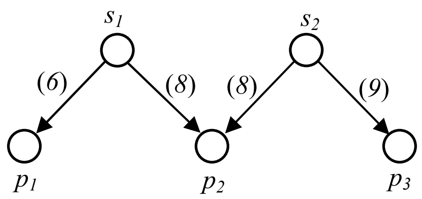
U geometrijskim ilustracijama grafova sa pridruženom semantikom, čvor se predstavlja kao n - torka pridruženih vrednosti uključenih u zagrade, tj. $x : (f_1(x), f_2(x), ..., f_n(x))$, gde je x ***identifikator*** čvora i $x \in S$. Slično, semantika pridružena ivici (x_i, x_j), upisuje se u zagradu pored ivice (x_i, x_j).

Primer 1.1. Na slici 1.1 prikazan je usmereni graf $G = (S, R)$, gde je $S = \{x_1, x_2, x_3\}$, $R = \{(x_1, x_2), (x_1, x_3)\}$, sa dve funkcije pridruživanja semantike čvorovima nad skupovima V_1 i V_2, gde je V_1 skup slova abecede, a V_2 skup celih brojeva. □



Slika 1.1.

Primer 1.2. Na slici 1.2 prikazan je usmereni graf $G = (S, R)$, gde je: $S = \{s_1, s_2, p_1, p_2, p_3\}$, $R = \{(s_1, p_1), (s_1, p_2), (s_2, p_2), (s_2, p_3)\}$, sa funkcijom f, koja ivicama grafa pridružuje vrednosti iz skupa $\{6, 7, 8, 9, 10\}$. Graf na slici 1.2 može da se interpretira na sledeći način: s_1 i s_2 su studenti, p_1, p_2 i p_3 predmeti, a funkcija f pridružuje studentu s ocenu iz predmeta p. □



Slika 1.2.

Relacija R, koja skup S snabdeva strukturom, najčešće predstavlja relaciju strogog poretka ili relaciju ekvivalencije, mada može predstavljati i neku drugu relaciju. U daljem tekstu će se smatrati da je R ***tranzitivna relacija***, mada se mogu naći retki kontraprimeri ovom tvrđenju, što naredni primer i ilustruje.

Primer 1.3. U skupu studenata jednog univerziteta relacija "pohađa isti fakultet" nije tranzitivna ako postoji bar jedan student koji istovremeno pohađa dva fakulteta i bar jedan student koji pohađa samo jedan od ta dva fakulteta. □

Pri crtanju geometrijske reprezentacije strukture podataka, uobičajeno je da se iz crteža izostave sve petlje[*)] i sve one ivice koje se mogu izvesti na osnovu tranzitivnosti relacije R. Ako je relacija tranzitivna u skupu S i ako između čvorova x_i, $x_j \in S$ postoji put dužine veće od 1, ivica (x_i, x_j) se ne crta. Graf, koji zadovoljava opisana pravila naziva se ***osnovnim grafom***[**)]. Korišćenjem osnovnog grafa povećava se preglednost njegove geometrijske reprezentacije. Međutim, ekonomičnost koja se postiže korišćenjem osnovnog grafa bitna je, pre svega, za oblikovanje njegove memorijske reprezentacije.

Primer 1.4. Na slikama 1.3 i 1.4 prikazane su dve reprezentacije iste relacije parcijalnog poretka. Na slici 1.4 prikazan je osnovni graf grafa na slici 1.3. □

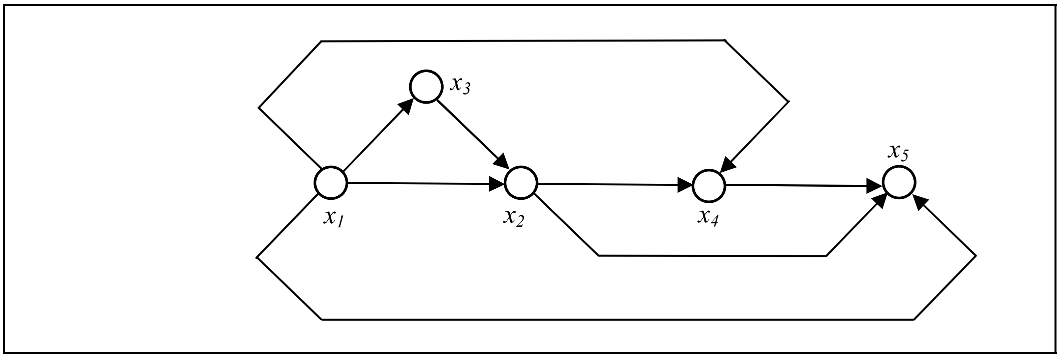
Osnovni graf strukture podataka je ili jako povezan, ili jednostrano povezan, ili slabo povezan.

Ako čvorovima x_i i x_j u osnovnom grafu odgovara put $p(x_i, x_j)$ dužine 1, x_i se naziva direktnim prethodnikom x_j, a x_j direktnim sledbenikom x_i.

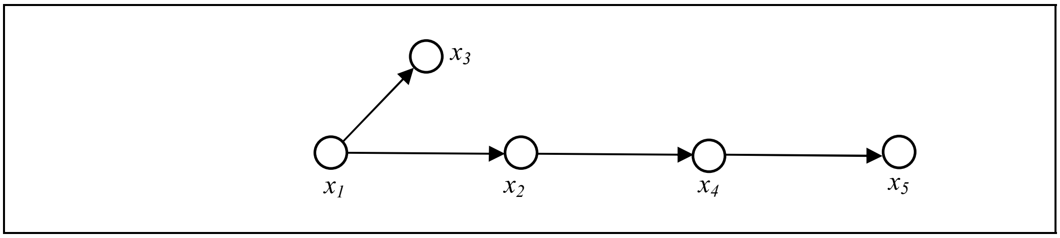
---

*) Smatra se da pojam petlje u strukturama podataka nema smisla. [Pf]

**) Ciklični grafovi imaju više osnovnih grafova, dok aciklični grafovi imaju samo jedan osnovni graf.



Slika 1.3.



Slika 1.4.

Kada se osnovni graf predstavlja putem matrice susedstva $\left\|a_{ij}\right\|_1^N$ glavna dijagonala matrice je nula dijagonala. Ako je relacija R tranzitivna, tada se, za $a_{ij} = 1$ i $a_{jk} = 1$, postavlja $a_{ik} = 0$. Nenula elementi matrice susedstva nose informaciju o direktnom prethođenju i direktnom sleđenju elemenata skupa S.

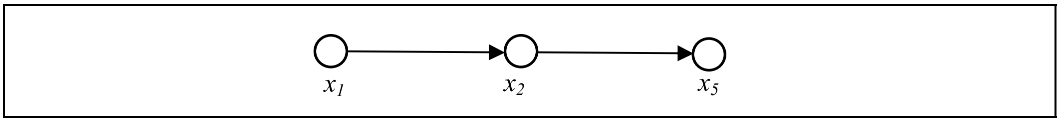
Korišćenje osnovnog grafa za predstavljanje struktura podataka zahteva primenu posebnog postupka za traženje podgrafa osnovnog grafa. Neka je dat osnovni graf $G = (S, R)$, pri čemu je R tranzitivna relacija. Potrebno je naći graf $H = (S', R')$, gde je $S' \subseteq S$. Jedan od mogućnih postupaka je sledeći:

1⁰ Pronaći tranzitivno zatvorenje R^T relacije R, tako što će se za svako (x, y) i (y, z) iz R, R unirati sa $\{(x, z)\}$, tj. $R^T = R \cup \{(x, z) \mid (x, y), (y, z) \in R\}$.

2⁰ Formirati graf $H' = (S', R^T|_{S'})$, gde je $R^T|_{S'}$ projekcija relacije R^T na skup S'.

3⁰ Od grafa H' formirati osnovni graf $H = (S', R')$.

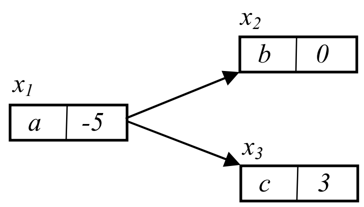
Primer 1.5. Da bi se od osnovnog grafa sa slike 1.4 formirao podgraf $H = (S', R')$ za $S' = \{x_1, x_2, x_5\}$, prvo bi se našlo tranzitivno zatvorenje $R^T = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_1, x_5), (x_2, x_4), (x_4, x_5), (x_2, x_5)\}$, zatim njegova projekcija na S', $R^T|_{S'} = \{(x_1, x_2), (x_1, x_5), (x_2, x_5)\}$, da bi se konačno dobio osnovni graf na slici 1.5. □



Slika 1.5.

U literaturi posvećenoj strukturama podataka, često se sreće geometrijska reprezentacija struktura podataka, koja se razlikuje od one, uvedene na početku ovog poglavlja. Razlike su u sledećem. Čvor se, umesto kružićem, predstavlja pravougaonikom. U taj pravougaonik se upisuju semantičke komponente, pridružene čvoru. Identifikator čvora se upisuje iznad čvora ili se uloga identifikatora delegira nekoj od semantičkih komponenata čvora. Ako semantika nije bitna, tada se identifikator čvora upisuje u pravougaonik. U daljem tekstu će se često koristiti i ova tehnika geometrijske reprezentacije struktura podataka.

Primer 1.6. Na slici 1.6 prikazan je alternativni način crtanja osnovnog grafa sa slike 1.1. □



Slika 1.6.

Strukture podataka se mogu klasifikovati uz pomoć više kriterijuma. Za projektovanje informacionih sistema važni su sledeći:
- dozvoljeni broj neposrednih prethodnika i sledbenika jednog čvora strukture i
- nivo apstraktnosti skupa, čiji elementi snabdevaju čvorove i ivice grafa semantikom.

S obzirom na prvi kriterijum, razlikuju se:

- linearne strukture,
- strukture stabla i
- mrežne strukture.

S obzirom na drugi kriterijum, razlikuju se:

- strukture nad skupom obeležja i
- strukture nad skupom podataka.

U slučaju klasifikacije po prvom kriterijumu strukture podataka se posmatraju kao strukture bez konteksta (semantike). Važne su samo karakteristike grafa. Pri tome, graf može biti ***linearan***, ***stablo*** ili ***mreža***. U slučaju klasifikacije po drugom kriterijumu, bitan je nivo apstraktnosti semantike pridružene grafu.

Strukture nad skupom obeležja (ili kratko strukture obeležja) nazivaju se i ***logičkim strukturama***, jer predstavljaju pojam na visokom nivou apstraktnosti. Strukture nad skupom podataka (ili kratko strukture podataka) dele se na logičke i fizičke strukture podataka.

1.1 LINEARNE STRUKTURE PODATAKA

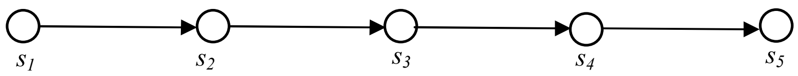
Za linearne strukture podataka je karakteristično da svaki čvor grafa može imati najviše jednog direktnog prethodnika i najviše jednog direktnog sledbenika. Linearne strukture podataka se dele na aciklične i ciklične. Aciklične se još nazivaju i ***lancima***, otvorenim ili prostim ***listama*** i ***nizovima*** (jednostruko indeksirani nizovi). Ciklične linearne strukture se još nazivaju i zatvorenim listama ili ***prstenom***.

Definicija 1.1. Osnovni graf $G = (S, R)$ je aciklična linearna struktura ako:

1° postoji samo jedan najmanji čvor,
2° ulazni i izlazni stepen svakog čvora nije veći od jedan,
3° graf G je jednostrano povezan graf. □□

Kao posledice definicije 1.1 mogu se navesti i sledeće činjenice. U acikličnoj linearnoj strukturi postoji samo jedan najveći elemenat. Relacija strogog poretka snabdeva skup S linearnom strukturom.

Primer 1.7. Na slici 1.7 prikazana je geometrijska reprezentacija jedne aciklične linearne strukture nad skupom $S = \{s_1,..., s_5\}$ i odgovarajuća matrica susedstva. Lako se proverava da graf sa slike 1.7 zadovoljava uslove definicije 1.1.



$$\left\| \begin{matrix} 0&1&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&1&0 \\ 0&0&0&0&1 \\ 0&0&0&0&0 \end{matrix} \right\|$$

Slika 1.7.

Na slici 1.8 prikazan je još jedan alternativan postupak predstavljanja struktura podataka, karakterističan samo za linearne strukture. Dobijen je izostavljanjem ivica i upisivanjem elemenata skupa S u susedne pravougaonike. Susednost pravougaonika nosi informaciju o susednosti čvorova. □

s_1	s_2	s_3	s_4	s_5

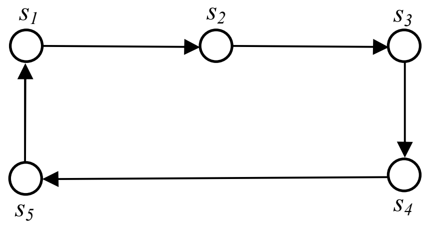
Slika 1.8.

Definicija 1.2. Osnovni graf $G = (S, R)$ je ciklična linearna struktura ako:

1° ulazni i izlazni stepen svakog čvora iznosi tačno jedan i
2° graf G je jako povezan graf. □

Kao posledice definicije 1.2 mogu se navesti i sledeće činjenice. U cikličnoj linearnoj strukturi nema čvora bez prethodnika niti čvora bez sledbenika. Relacija ekvivalencije snabdeva skup S cikličnom linearnom strukturom.

Primer 1.8. Na slici 1.9 prikazan je osnovni graf jedne ciklične linearne strukture nad skupom $S = \{s_1,..., s_5\}$, kao i odgovarajuća matrica susedstva. □



$$\left\|\begin{matrix} 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&0&0 \end{matrix}\right\|$$

Slika 1.9.

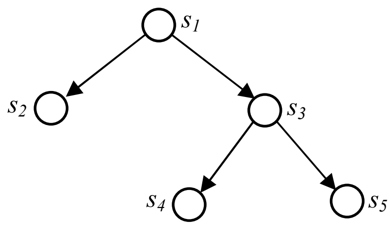
1.2 STRUKTURE TIPA STABLA

U strukturama tipa stabla, svaki čvor može imati najviše jednog direktnog prethodnika i do n direktnih sledbenika, gde je $0 \leq n \leq N - 1$, a N kardinalni broj skupa koji se snabdeva strukturom.

Definicija 1.3. Osnovni graf $G = (S, R)$ predstavlja strukturu stabla ako:

1° postoji samo jedan najmanji čvor i
2° ulazni stepen svakog čvora, osim najmanjeg, iznosi tačno 1. □

Posledice definicije 1.3 su i sledeće činjenice. Strukture stabla su aciklične strukture. Postoji put od najmanjeg do svakog drugog čvora. U opštem slučaju, to su slabo povezani grafovi. Postoji bar jedan maksimalan čvor. Broj ivica grafa iznosi $N - 1$.



$$\left\|\begin{matrix} 0&1&1&0&0\\ 0&0&0&0&0\\ 0&0&0&1&1\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{matrix}\right\|$$

Slika 1.10.

Primer 1.9. Na slici 1.10 prikazan je osnovni graf i matrica susedstva strukture stabla definisane nad skupom $S = \{s_1,..., s_5\}$. □

Određenim čvorovima strukture tipa stabla dodeljuju se posebna imena. Najmanji čvor, odnosno, čvor u koji ne dolazi nijedan poteg, naziva se ***korenom*** stabla. Maksimalni čvor, odnosno, čvor iz kojeg ne polazi nijedan poteg, naziva se ***listom***. Svaki čvor predstavlja koren jednog podstabla posmatranog stabla.

Među čvorovima stabla postoji ***nivovska hijerarhija***. Koren predstavlja čvor prvog nivoa hijerarhije. Proizvoljan čvor s_i nalazi se na k - tom nivou hijerarhije ($k \in \{1, 2,..., h\}$, gde je h broj nivoa hijerarhije stabla), ako se nalazi na kraju puta dužine k - 1, a put počinje u korenu stabla. Dužina puta se meri brojem potega između dva posmatrana čvora. Broj nivoa hijerarhije h naziva se ***visinom*** stabla. Visinu h stabla predstavlja broj, koji je za jedan veći od dužine puta između korena i najudaljenijeg lista.

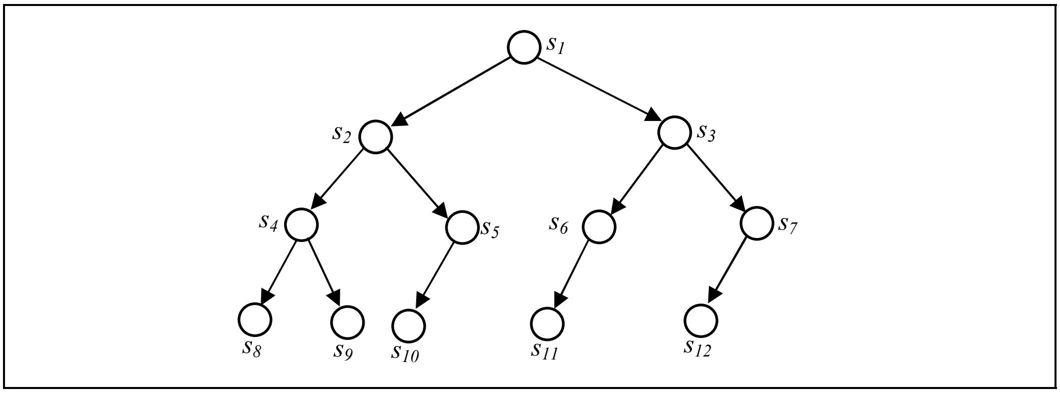
Ako postoji put iz čvora s_i u čvor s_j, kaže se da je čvor s_i ***nadređen*** čvoru s_j, odnosno, da je čvor s_j ***podređen*** čvoru s_i. Ako put između čvorova s_i i s_j ima dužinu 1, reč je o direktnoj nadređenosti, odnosno, podređenosti.

S obzirom da strukture stabla imaju veliki značaj za strukture podataka, u daljem tekstu je dato još nekoliko definicija koje bliže objašnjavaju osobine pojedinih karakterističnih struktura tipa stabla.

RED STABLA

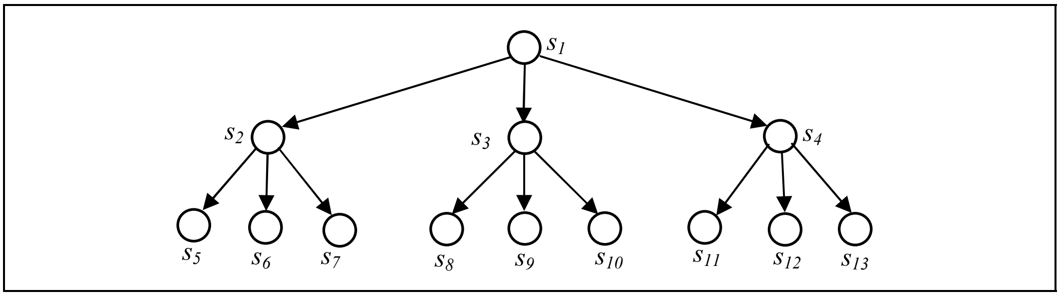
Definicija 1.4. Za stablo se kaže da je ***n - arno***, odnosno ***reda n***, ako svakom čvoru koji nije list odgovara maksimalno n direktno potčinjenih čvorova ($1< n <N$). □□

U specijalnom slučaju, kada je $n = 2$, reč je o binarnom stablu. U daljem tekstu će se smatrati da, ako to nije drugačije naglašeno, maksimalni broj direktno podređenih čvorova geometrijske reprezentacije strukture stabla određuje njegov red.



Slika 1.11.

Primer 1.10. Stablo na slici 1.11 je binarno, a na slici 1.12 je prikazano stablo reda $n = 3$. □



Slika 1.12.

PUNO STABLO

Definicija 1.5. Za stablo se kaže da je ***puno***, ako se svi listovi nalaze na istom odstojanju od korena, odnosno, ako od korena svakom listu odgovara put dužine h - 1. □

KOMPLETNO STABLO

Definicija 1.6. Za stablo reda n se kaže da je ***kompletno***, ako svi njegovi čvorovi, koji ne predstavljaju listove, imaju svih n odlaznih potega, odnosno svih n direktno podređenih čvorova. □

Broj čvorova C kompletnog punog stabla reda n iznosi

(1.1) $$C = \sum_{i=1}^{h} n^{i-1} .$$

Za datu vrednost C, visina punog kompletnog stabla reda n iznosi

(1.2) $$h = \log_n(1 + (n - 1)C).$$

Primer 1.11. Stablo na slici 1.10 je kompletno, ali nije puno, dok je stablo na slici 1.12 i kompletno i puno. □

BALANSIRANO STABLO

Definicija 1.7. Za stablo se kaže da je ***balansirano***, ako za svaki čvor važi da se broj čvorova u svakom njegovom podstablu ne razlikuje za više od jedan. □

Pri datom broju čvorova, balansirana stabla poseduju minimalnu visinu. Istu osobinu, uz manje strože uslove, poseduje i optimalno balansirano stablo istog reda.

OPTIMALNO BALANSIRANO STABLO

Definicija 1.8. Za stablo reda n čiji su svi čvorovi na nivoima od 1 do h - 2 kompletni, kaže se da je ***optimalno balansirano***.

Čvor je kompletan ako poseduje svih n direktno podređenih čvorova. □

Optimalno balansirano stablo poseduje $C_i = n^{i-1}$ čvorova na nivoima $i = 1, 2, ..., h-1$ i

$$C_h = C - \frac{n^{h-1} - 1}{n - 1}$$

listova. Broj čvorova C optimalno balansiranog stabla uzima vrednosti iz poluotvorenog intervala

$$\left(\frac{n^{h-1} - 1}{n - 1}, \frac{n^h - 1}{n - 1} \right].$$

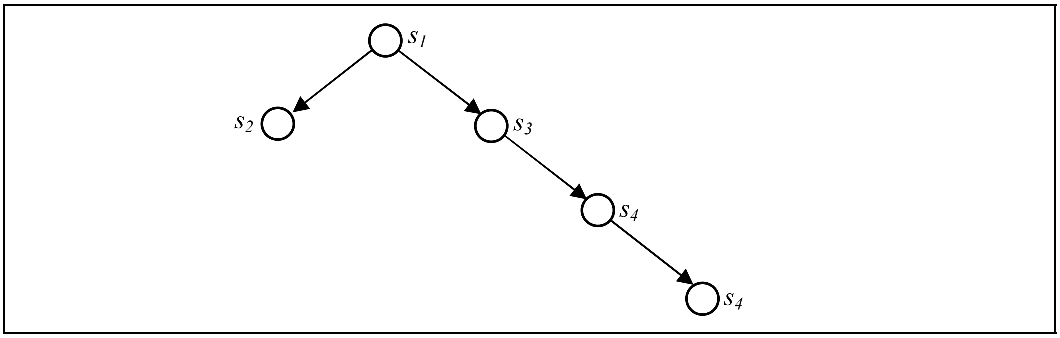
Pri datom broju čvorova C, visina optimalno balansiranog stabla iznosi

(1.3) $$h = \lceil \log_n(1 + (n - 1)C) \rceil$$

gde $\lceil x \rceil$ predstavlja minimalan ceo broj ne manji od x. Treba naglasiti da, pri datom n i C, visina stabla je minimalna ako je ono optimalno balansirano.

Primer 1.12. Kompletno i puno stablo predstavlja specijalan slučaj balansiranog stabla, te je stablo na slici 1.12 balansirano. Stablo na slici 1.10 nije balansirano. Na slici 1.11 je prikazano jedno nekompletno, puno balansirano stablo.

Saglasno definiciji 1.8, stablo na slici 1.10 je optimalno balansirano, a takođe i stabla na slikama 1.11 i 1.12. Na slici 1.13 nacrtano je jedno binarno stablo, koje nije optimalno balansirano. Treba zapaziti da stablo na slici 1.13 poseduje visinu $h = 4$, mada ima isti broj čvorova kao i stablo na slici 1.10. □



Slika 1.13.

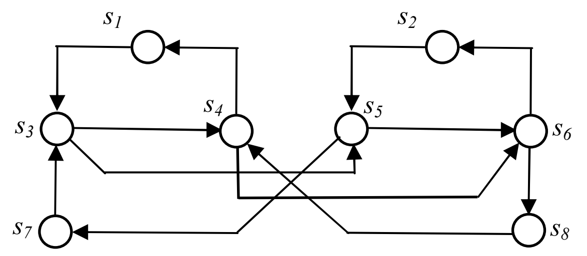
Definicija 1.9. Za stablo koje nema korena kaže se da je prazno. □

Definicija 1.10. Disjuktni skup stabala naziva se šumom. □

1.3 MREŽNA STRUKTURA PODATAKA

U ***mrežnoj strukturi*** podataka svaki čvor može imati do N[*)] direktnih prethodnika i do N direktnih sledbenika. Mrežne strukture podataka su često ciklične strukture. Struktura podataka se smatra mrežnom ako postoji čvor koji poseduje više od jednog direktnog prethodnika.

Primer 1.13. Na slici 1.14 prikazana je geometrijska i matrična reprezentacija jedne strukture mreže nad skupom $S = \{s_1,..., s_8\}$. Interesantno je ukazati na jednu od mogućih semantika ove strukture. Na primer, čvorovi s_1 i s_2 mogu reprezentovati studente, čvorovi s_7 i s_8 predmete, čvorovi s_3 i s_4 ocene studenta s_1, redom, iz predmeta s_7 i s_8, a čvorovi s_5 i s_6 ocene studenta s_2 iz predmeta s_7 i s_8. Lako se da zamisliti kakvu kompleksnost bi posedovala odgovarajuća struktura, kada bi broj studenata i predmeta bio veći. □



$$\left\| \begin{matrix} 0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&1&1&0&0&0 \\ 1&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&1&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \end{matrix} \right\|$$

Slika 1.14.

1.4. OSNOVNI POJMOVI LOGIČKIH STRUKTURA PODATAKA

U ovoj tački su izložene definicije, ilustrovane primerima, jednog broja osnovnih pojmova, vezanih za logičke strukture nad skupom obeležja i logičke strukture nad skupom podataka. U tom smislu, definisani su pojmovi: obeležja, podatka, tipa i pojave entiteta, odnosno sloga, datoteke i ključa.

OBELEŽJE I PODATAK

Zadatak automatizovanih informacionih sistema predstavlja prikupljanje, memorisanje, obrada, prenos i prezentiranje podataka o ***entitetima*** [*)] raznih klasa nekog realnog sistema. Skup sličnih entiteta naziva se ***klasom***. Klase entiteta predstavljaju: klase subjekata, kao što su

---

[*)] Ako se prihvati da petlje u strukturama podataka nemaju smisla, tada do $N - 1$.

[*)] Engleska reč entity se, po pravilu, ne prevodi na srpski jezik. Približno značenje joj je jedinica posmatranja.

radnici jedne radne organizacije ili studenti nekog univerziteta, klase objekata, kao što su proizvodi, zgrade ili organizacije, klase događaja, kao što su uplata na štednu knjižicu ili upis u školu, klase raznih pojmova i pojava. Treba zapaziti da kriterijum sličnosti igra značajnu ulogu u definisanju klase entiteta. U zavisnosti od usvojenog kriterijuma sličnosti, dva entiteta se mogu naći u istoj ili u dve različite klase. Isto tako, isti entitet može pripadati različitim klasama.

Primer 1.14. Studenti različitih fakulteta istog univerziteta mogu pripadati klasama entiteta studenata odgovarajućih fakulteta ili klasi entiteta studenata univerziteta. □

Svi entiteti jedne klase poseduju određene zajedničke osobine, kao što su: naziv, boja, vrednost, trajanje i slično. Ove osobine nazivaju se ***obeležjima*** (atributima). Obeležja se označavaju velikim slovima latinske azbuke, skraćenim nazivom (mnemonikom) ili punim nazivom. Na primer, A, X, MBR ili *MATIČNI_BROJ_RADNIKA*.

Svakom od obeležja odgovara jedan skup svih mogućnih vrednosti koje to obeležje, u konkretnim slučajevima, može imati. Taj skup vrednosti se naziva ***domenom obeležja***. Domen obeležja A se obeležava sa $dom(A)$. Domen se naziva i ***tipom podataka***.

Primer 1.15. Za obeležje *BOJA_AUTOMOBILA* skup vrednosti je

$$dom(BOJA\_AUTOMOBILA) = \{bela, žuta, crna, plava, ...\}. \ \square$$

U strukturama podataka, pojam domena se koristi u smislu skupa vrednosti iz kojeg semantički definisani objekti, kao što su tip entiteta i obeležje, uzimaju vrednosti.

S obzirom da, u opštem slučaju, obeležje uzima pojedine vrednosti iz svoga domena sa različitim verovatnoćama, obeležje se može smatrati semantički definisanom slučajnom veličinom. Tada se vrednosti, koje obeležje uzima, nazivaju njegovim konkretizacijama.

Konkretizacija obeležja predstavlja ***podatak*** o određenom entitetu i samo konkretizacija obeležja može predstavljati podatak.

Primer 1.16. Ako se posmatra obeležje *MATIČNI_BROJ_RADNIKA* i jedna njegova konkretizacija *110451*, tada *110451* predstavlja podatak o nekom radniku jedino ako se unapred zna da *110451* predstavlja konkretizaciju obeležja *MATIČNI_BROJ_RADNIKA*. U suprotnom *110451* ne predstavlja podatak, jer mu je smisao neodređen. □

Formalno, neka je $E = \{e_i \mid i = 1, ..., m\}$ klasa entiteta, a A jedno od obeležja te klase. Obeležje A predstavlja funkciju $A: E \rightarrow dom(A)$. Drugim rečima, obeležje A pridružuje svakom entitetu $e_i \in E$ jednu vrednost iz $dom(A)$, tako da $A(e_i)$ predstavlja podatak o e_i s obzirom na A.

Obeležje, koje se dalje ne može dekomponovati, ili koje se u posmatranom slučaju dalje ne dekomponuje na komponente, koje takođe predstavljaju obeležja, naziva se ***elementarnim obeležjem***. Skup, niz, ili logički proizvod elementarnih obeležja predstavlja ***složeno obeležje***. Tom nizu obeležja se može pridružiti neko ime.

Primer 1.17. Obeležja *NAZIV_PROIZVODA*, *BOJA_AUTOMOBILA*, *IME_STANOVNIKA* predstavljaju elementarna obeležja različitih klasa entiteta. Složena obeležja predstavljaju, na primer, $ADRESA = \{MESTO, ULICA, BROJ\}$, $\{IME, PRZ, MESTO\}$, ili $DATUM\_UPLATE = \{DAN, MESEC, GODINA\}$. □

Složena obeležja se, često, označavaju slovima sa kraja abecede, na primer X ili Y ili Z, elementarna slovima sa početka abecede, na primer A ili B ili C. Saglasno rečenom, složeno obeležje je $X = \{A_1, A_2,..., A_k\}$, gde su A_i, $1 \leq i \leq k$, elementarna obeležja. Konkretizacija elementarnog obeležja predstavlja elementarni podatak, a konkretizacija složenog obeležja predstavlja složeni podatak. Ako je $X = \{A_1,..., A_k\}$, tada $dom(X) \subseteq dom(A_1)\times...\times dom(A_k)$. Obeležje čije se vrednosti dobijaju primenom nekog algoritma na vrednosti drugih obeležja naziva se ***izvedenim obeležjem***, a njegove vrednosti ***izvedenim podacima***.

Često se javlja potreba da se, pri evidentiranju podataka o nekom entitetu e, zabeleži da vrednost za neko obeležja A nedostaje. U tom cilju se u domen obeležja A uvodi fiktivna ***nula vrednost***. Najčešće, nula vrednost ima jednu od sledeće dve semantike: “postojeća ali trenutno nepoznata vrednost”, ili “obeležje A predstavlja neprimereno svojstvo za entitet e”. Nula vrednost se može zameniti bilo kojim elementom domena obeležja. U daljem tekstu, nula vrednost će se označavati sa ω.

TIP ENTITETA I POJAVA TIPA ENTITETA

Sa tačke gledišta zadataka informacionog sistema, nisu sva obeležja klase entiteta jednako važna. Od obeležja, bitnih za realizaciju zadataka informacionog sistema, gradi se ***model*** realne klase entiteta. Model klase entiteta naziva se ***tipom entiteta***. Tip entiteta se označava sa $N(A_1,..., A_n)$, gde je N naziv tipa entiteta, A_i $(1 \leq i \leq n)$ predstavljaju odabrana obeležja klase entiteta E. Kao i svaki model, tip entiteta predstavlja samo približnu sliku klase entiteta realnog sistema. Neki put se za reprezentaciju klase entiteta, umesto oznake za tip entiteta $N(A_1,..., A_n)$ koristi samo naziv N.

Pošto niz $N(A_1,..., A_n)$ predstavlja složeno obeležje, tip entiteta $N(A_1,..., A_n)$ predstavlja imenovano složeno obeležje.

Klasa entiteta poseduje konačno mnogo osobina zajedničkih svim realnim entitetima. Neka je $\{A_1,..., A_m\}$ skup osobina klase entiteta E. Tada je skup obeležja $\{A_1,..., A_n\}$ odabranih za izgradnju tipa entiteta kao modela klase E, pravi ili nepravi podskup skupa obeležja $\{A_1,..., A_m\}$.

Primer 1.18. Tip entiteta *STUDENT* (*BROJ_INDEKSA*, *IME*, *PREZIME*, *NAZIV_FAKULTETA, BROJ_ISPITA*) reprezentuje sve studente jednog univerziteta. □

Svaka klasa entiteta predstavlja skup sličnih entiteta. Svakom entitetu posmatrane klase odgovaraju konkretne vrednosti obeležja klase entiteta. U okviru informacionog sistema, svaki entitet se predstavlja modelom koji sadrži konkretne vrednosti onih obeležja, uz pomoć kojih je opisana klasa entiteta. ***Uređenje podataka*** u modelu entiteta diktirano je ***uređenjem skupa obeležja*** u tipu entiteta. Obeležjima u tipu entiteta odgovaraju konkretizacije tih obeležja (podaci) u modelu entiteta istim redom.

Model jednog entiteta naziva se ***pojavom*** odgovarajućeg tipa entiteta. Tip entiteta i pojava entiteta predstavljaju apstraktne opise konkretnih realnih entiteta. Tip entiteta predstavlja model entiteta na višem nivou apstrakcije nego pojava tipa entiteta.

Formalno, za dati tip entiteta $N(A_1,..., A_n)$ skup funkcija $\{A_i : E \rightarrow dom(A_i) \mid i = 1,..., n\}$ jednoznačno određuje funkciju $(A_1,..., A_n) : E \rightarrow dom(A_1)\times ... \times dom(A_n)$, gde je E klasa entiteta, a $dom(A_1)\times...\times dom(A_n) = \{(a_1,..., a_n) \mid a_i \in dom(A_i), i = 1,..., n\}$. Funkcija $(A_1,..., A_n)$ je definisana sa $(A_1,..., A_n)(e) = (A_1(e),..., A_n(e))$, gde je $e \in E$ entitet. Drugim rečima, $(a_1,..., a_n)$ predstavlja pojavu tipa entiteta $N(A_1,..., A_n)$, ako je zadovoljena kvantifikatorska formula

$$\mathcal{F}((a_1,..., a_n)) = (\exists e \in E)(i \in \{1,..., n\})(a_i = A_i\ (e)),$$

a skup P pojava tipa entiteta je

$$P = \{(a_1,..., a_n) \mid \mathcal{F}((a_1,..., a_n))\}.$$

Uređenje podataka u n - torki $(a_1,..., a_n)$ je bitno, jer nosi informaciju o tome da je $a_i \in dom(A_i)$.

Primer 1.19. Za tip entiteta *STUDENT* iz primera 1.18, mogućne pojave predstavljaju sledeće petorke:

(*159*, *Ivo*, *Ban*, *Fakultet tehničkih nauka*, *15*),
(*313*, *Eva*, *Tot*, *Ekonomski fakultet*, *23*),
(*505*, *Aco*, *Kon*, *Pravni fakultet*, ω). □

Treba zapaziti da za studenta *Aco Kon*, obeležje *BROJ_ISPITA* ima nula vrednost. Moguća semantika te nula vrednosti je „postojeća ali trenutno nepoznata vrednost“. □

KLJUČ

Definicija pojma entiteta kao „jedinice posmatranja“ ukazuje na potrebu da se entiteti posmatrane klase mogu razlikovati. To zahteva da i modeli dva entiteta budu različiti. Neka su e_1 i e_2 entiteti klase E, a $N(A_1,..., A_n)$ tip entiteta, tada mora važiti $(A_1,..., A_n)(e_1) \neq (A_1,..., A_n)(e_2)$. Znači, mora postojati neprazan skup obeležja $X \subseteq \{A_1,..., A_n\}$ takav da je $X(e_1) \neq X(e_2)$.

Definicija 1.11. Neka je $P = \{p_i \mid i = 1,..., k\}$ skup svih pojava tipa entiteta $N(A_1,..., A_n)$, a $p[X]$ restrikcija pojave p na obeležje X. Obeležje X predstavlja ***ključ*** tipa entiteta $N(A_1,..., A_n)$ ako i samo ako važe sledeća tri uslova:

1° $(\forall p_i, p_j \in P)(p_i \neq p_j \Rightarrow p_i[X] \neq p_j[X])$,
2° $(\forall X' \subset X)(\neg \mathbf{1^o})$ i
3° $(\forall p \in P)(\forall A \in X)(p[A] \neq \omega)$. □

Uslov **1°** se naziva uslovom jedinstvenosti i on ukazuje da sve pojave tipa entiteta N moraju imati različite vrednosti ključa X. Uslov **2°** se naziva uslovom minimalnosti i on ukazuje da ključ mora sadržati samo neophodna obeležja. Putem uslova **2°** se sprečava nepotrebna kompleksnost ključa. Konačno, uslov **3°** ukazuje da nijedna komponenta ključa ne sme imati nula vrednost. Uslov **3°** dopunjuje uslov **1°** u cilju obezbeđenja jedinstvenosti vrednosti ključa, jer nula vrednost predstavlja moguću zamenu za svaku vrednost iz domena obeležja. Ako X zadovoljava samo uslove **1°** i **3°**, naziva se ***superključem*** tipa entiteta N.

Svaki tip entiteta poseduje bar jedan ključ. Neka je $X = \{A_1,..., A_n\}$ i važe uslovi **1°** i **3°**. Tada je X superključ tipa entiteta N. Ako važi i uslov **2°**, X predstavlja ključ tipa entiteta $N(A_1,..., A_n)$. Inače, mora postojati neprazan podskup skupa $\{A_1,..., A_n\}$ za koji važi **1°**, **2°** i **3°**. Ovo razmatranje, ujedno, sugeriše i jedan od mogućih postupaka za određivanje ključa tipa entiteta.

Jedan tip entiteta može posedovati više ključeva. Nazivaju se ***ekvivalentnim***. Jedan od ekvivalentnih ključeva se bira za ***primarni***. U okviru notacije za tip entiteta sva obeležja jednog ključa podvlače se jednom, kontinualnom linijom. Uobičajeno je da se, u definiciji tipa entiteta, sva obeležja - komponente jednog ključa podvuku kontinualnom linijom.

Primer 1.20. Tip entiteta *STUDENT*(*BROJ_INDEKSA*, *IME*, *PREZIME*, *ADRESA*, *MATIČNI_BROJ_STANOVNIKA*) poseduje dva ključa. To su *BROJ_INDEKSA* i *MATIČNI_BROJ_STANOVNIKA*. Ako je reč o tipu entiteta, definisanom u projektu fakultetskog informacionog sistema, obeležje *BROJ_INDEKSA* bi, nejverovatnije, bilo izabrano za primarni kluč.

Posmatra se tip entiteta *ISPIT*(*BROJ_INDEKSA*, *IDBROJ_PREDMETA*, *OCENA*). Neka u realnom sistemu važe sledeća ograničenja:

1. jedan student može imati ocene iz različitih predmeta,
2. veći broj studenata može imati ocene iz istog predmeta i
3. svaki student ima najviše jednu ocenu iz jednog predmeta.

Skup obeležja {*BROJ_INDEKSA*, *IDBROJ_PREDMETA*, *OCENA*} predstavlja superključ tipa entiteta *ISPIT*. Na osnovu uslova 2. se zaključuje da obeležje *BROJ_INDEKSA* predstavlja obaveznu komponentu ključa, jer može postojati veći broj pojava tipa entiteta sa istim {*IDBROJ_PREDMETA*, *OCENA*} vrednostima. Na osnovu uslova 1. se zaključuje da i obeležje *IDBROJ_PREDMETA* predstavlja obaveznu komponentu ključa, jer može postojati veći broj pojava tipa entiteta sa istim {*BROJ_INDEKSA*, *OCENA*} vrednostima. Na osnovu uslova 3. se zaključuje da se obeležje *OCENA* može izostaviti iz superključa, te se zaključuje da ključ predstavlja složeno obeležje {*BROJ_INDEKSA*, *IDBROJ_PREDMETA*}.

Pretpostavimo da obeležje *IDBROJ_PREDMETA* sme imati nula vrednosti. Tada skup pojava tipa entiteta *ISPIT* može sadržati sledeće dve pojave (*159*, *Matematika*, *9*) i (*159*, ω, *9*). Pošto nula vrednost predstavlja zamenu za svaku vrednost iz domena obeležja, ove dve pojave se, strogo govoreći, ne mogu razlikovati.

Tip entiteta *POVERAVANJE*(*IDBROJ_NASTAVNIKA*, *IDBROJ_PREDMETA*) opisuje odnos između nastavnika i predmeta u realnom sistemu, gde se nastavniku poverava izvođenje predavanja iz predmeta. Ako se za ključ tipa entiteta *POVERAVANJE* odredi obeležje *IDBROJ_NASTAVNIKA*, tada tip entiteta opisuje odnos, prema kojem se svakom nastavniku poverava samo jedan predmet. Ako se za ključ odredi *IDBROJ_PREDMETA*, tip entiteta opisuje realnu situaciju u kojoj nastavu iz svakog predmeta izvodi samo jedan nastavnik.

Ako se pretpostavi da se istom nastavniku može poveriti izvođenje nastave iz više predmeta, a isti predmet može poveriti većem broju nastavnika, tada sva obeležja tipa entiteta *POVERAVANJE* (*IDBROJ_NASTAVNIKA*, *IDBROJ_PREDMETA*) predstavljaju ključ. Može se zaključiti da semantika modela klase entiteta značajno zavisi od odabranog ključa. □

TIP SLOGA I POJAVA TIPA SLOGA

Bez pretenzije na strogost definicije, smatraće se da pojmovi tip sloga i pojava tipa sloga predstavljaju sinonime, redom, za tip entiteta i pojavu tipa entiteta. Saglasno tome i skup *P* pojava tipa entiteta predstavlja skup pojava tipa sloga definisanog nad istim skupom obeležja kao i tip entiteta.

DATOTEKA

Generalno govoreći, datoteke se mogu podeliti na:

- programske datoteke, koje sadrže programe bilo u izvornom ili mašinskom kodu i
- datoteke, koje sadrže podatke.

Datoteke koje sadrže podatke se mogu dalje deliti na:

- datoteke sa podacima u slobodnom formatu (polustrukturirane i nestrukturirane datoteke) i
- datoteke sa veoma strogim formatom (strogo definisanom strukturom).

Samo datoteke sa strogim formatom podataka predstavljaju predmet izučavanja u ovoj knjizi. Saglasno tom opredeljenju se, u daljem tekstu, pojam datoteke i definiše.

Datoteka *F* predstavlja strukturu nad skupom *P* pojava tipa sloga $N(A_1,..., A_n)$. Relacija *R*, koja uređuje skup *P*, najčešće se definiše s obzirom na vrednosti primarnog ključa tipa sloga *N*. Saglasno rečenom, $F = (P, R)$. Datoteka poseduje ime i nalazi se memorisana na medijumu nekog memorijskog uređaja. Najčešće, datoteci se dodeljuje naziv odgovarajućeg tipa sloga.

Tip sloga i datoteka predstavljaju modele klase entiteta. Pri tome, tip sloga predstavlja model na višem nivou apstrakcije. Istovremeno, tip sloga predstavlja i apstraktni model odgovarajuće datoteke.

Primer 1.21. Sledeće pojave tipa sloga

ISPIT(<u>*BROJ_INDEKSA*, *PREDMET*</u>, *PREZIME*, *OCENA*):

((*013*, *Matematika*, *Marić*, *9*),
(*013*, *Mehanika*, *Marić*, *7*),
(*117*, *Matematika*, *Marić*, *7*),
(*159*, *Fizika*, *Ban*, *6*),
(*159*, *Matematika*, *Ban*, *7*))

reprezentuju jednu od mogućnih datoteka nad tim tipom sloga. □

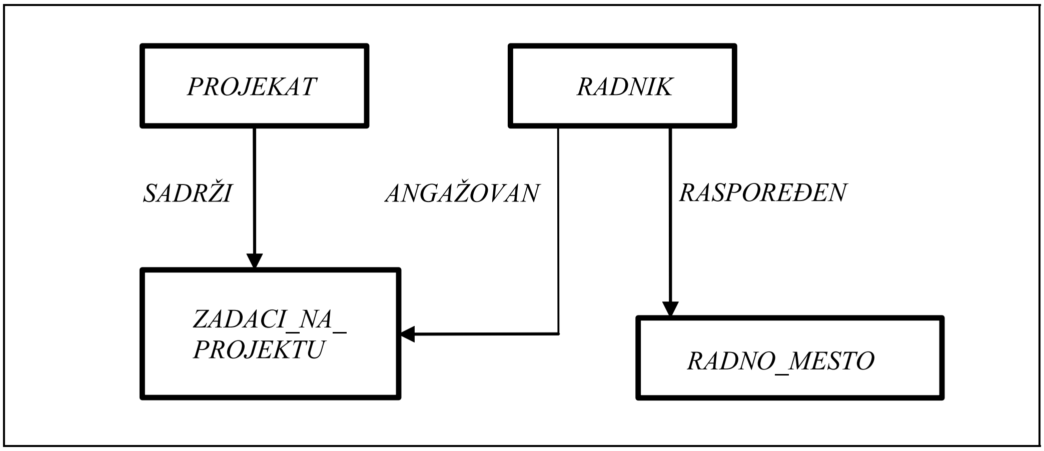
1.5. LOGIČKE STRUKTURE NAD SKUPOM OBELEŽJA

Projektovanje automatizovanog informacionog sistema vrši se na osnovu prethodno definisanih zahteva, koje taj informacioni sistem treba da zadovolji. Automatizovani informacioni sistem se projektuje sa ciljem obezbeđenja uslova za efikasnije upravljanje razvojem i operativnim funkcionisanjem nekog realnog sistema. Taj realni sistem može predstavljati neka radna organizacija, društveno-politička zajednica, neka druga asocijacija građana ili neki njihov deo. Kada je reč o projektovanju organizacije podataka tog informacionog sistema, postoji više karakterističnih grupa zahteva. Na ovom mestu će biti ukazano samo na dve. To su:

- grupa zahteva koja ukazuje na to, o kojim klasama entiteta realnog sistema treba da se iskazuju podaci u izlazima informacionog sistema (grupa zahteva Z1) i
- grupa zahteva koja ukazuje na to, koji podaci o entitetima pojedinih klasa treba da se iskazuju (grupa zahteva Z2).

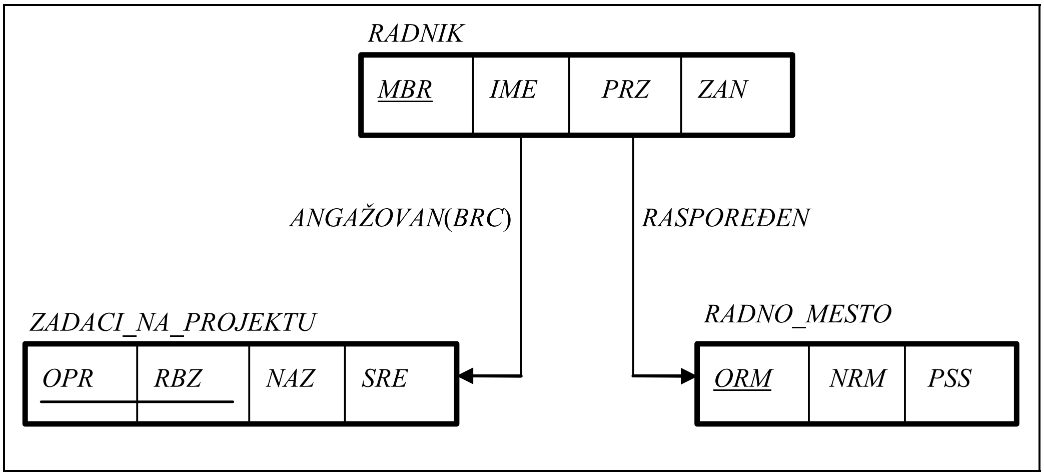
Na osnovu zahteva iz grupe Z1 gradi se, u postupku projektovanja organizacije podataka, graf $G = (S, R)$, gde je $S = \{E_i \mid i = 1,..., n\}$ skup klasa entiteta sa nazivima (oznakama) $E_1,..., E_n$, a $R = \{(E_i, E_j) \mid E_i, E_j \in S$ i E_i je u vezi sa $E_j\}$ relacija, koja ukazuje da između entiteta klase E_i i klase E_j u realnom sistemu postoji neka veza. Pri crtanju grafa G, nazivi klasa entiteta predstavljaju identifikatore čvorova grafa, a često se i ivicama (granama) grafa pridružuju identifikatori koji opisuju vezu između entiteta povezanih klasa.

Primer 1.22. Na slici 1.15 je nacrtan graf nad skupom klasa entiteta jedne hipotetične projektne organizacije. Graf sa slike 1.15 se može opisati sledećim rečenicama. Radnici su raspoređeni na radna mesta i angažovani na zadacima u okviru nekog projekta. Projekti sadrže zadatke. □



Slika 1.15.

Neka je U skup svih obeležja realnog sistema. Na osnovu zahteva iz grupe Z2, identifikuju se ona obeležja skupa U koja su bitna za realizaciju zadataka informacionog sistema. Ova identifikacija predstavlja jedan od koraka metodologije projektovanja organizacije podataka. Jedan drugi korak ove metodologije, recimo korak L, pridružuje svakom čvoru i svakoj ivici grafa G jedan podskup skupa obeležja U. Drugim rečima, L predstavlja funkciju koja preslikava S i R u partitivni skup skupa U, tj $L : S \to \mathcal{P}(U)$ i $L : R \to \mathcal{P}(U)$. Pri tome, L pridružuje čvorovima iz S neprazne podskupove skupa U, dok nekim ivicama iz R funkcija L može pridružiti i prazan skup. Očigledno, funkcija L pridružuje čvorovima i ivicama grafa G semantiku.



Slika 1.16.

Primer 1.23. Na slici 1.16 prikazan je jedan podgraf grafa sa slike 1.15 čijim čvorovima i ivicama je pridružena semantika. Čvoru *RADNIK* pridružena su obeležja: matični broj radnika (*MBR*), ime (*IME*), prezime (*PRZ*) i zanimanje (*ZAN*). Čvoru *ZADACI_NA_PROJEKTU* pridružena su obeležja: oznaka projekta (*OPR*), redni broj zadataka na projektu (*RBZ*), naziv zadatka (*NAZ*), sredstva na zadatku (*SRE*). Čvoru *RADNO_MESTO* pridružena su obeležja: oznaka radnog mesta (*ORM*), naziv radnog mesta (*NRM*) i potrebna stručna sprema (*PSS*). Ivici *ANGAŽOVAN* pridruženo je obeležje broj časova rada radnika na zadatku (*BRC*), a ivici *RASPOREĐEN* nije pridruženo nijedno obeležje. □

U skup obeležja jednog čvora, ili ivice, često se uvodi poredak. Taj poredak ne nosi nikakvu semantiku. Uvođenjem poretka, svaki čvor sa pridruženom semantikom postaje tip entiteta. Bitan korak metodologije projektovanja organizacije podataka predstavlja i određivanje ključeva tipova entiteta, o čemu je već bilo reči.

Svaka ivica grafa *G* sa pridruženom semantikom predstavlja imenovani objekat čije komponente su povezane klase entiteta i skup pridruženih obeležja *Q*, u oznaci $N((E_i, E_j), Q)$. Ovaj objekat se naziva ***tipom poveznika***. Kao i u slučaju tipa entiteta, uobičajeno je da se u skup *Q* uvede poredak.

Primer 1.24. Na slici 1.16 su prikazana dva tipa poveznika *ANGAŽOVAN* ((*RADNIK*, *ZADACI_NA_PROJEKTU*), *BRC*) i *RASPOREĐEN*(*RADNIK*, *RADNO_MESTO*). □

Pošto tipovi entiteta i tipovi poveznika predstavljaju imenovana složena obeležja, graf *G* snabdeven semantikom na nivou apstrakcije obeležja se naziva i ***logičkom strukturom nad skupom obeležja***. Takođe, ovaj graf predstavlja statički model realnog sistema i model buduće baze podataka. Svaki tip entiteta predstavlja, model jedne klase entiteta, a tip poveznika model odnosa između entiteta dve klase. Tip entiteta i tip poveznika predstavljaju i logičke strukture nad skupom obeležja i modele odgovarajućih datoteka. Graf *G* kod kojeg je kardinalitet skupa *S* veći od *1* predstavlja statički model realnog sistema i baze podataka informacionog sistema tog realnog sistema.

1.6 LOGIČKE STRUKTURE NAD SKUPOM PODATAKA

Svaki tip entiteta kao funkcija preslikava odgovarajuću klasu entiteta u skup koji predstavlja Dekartov proizvod domena obeležja tipa entiteta, tj.

$$(A_1,..., A_n) : E \rightarrow dom(A_1) \times ... \times dom(A_n),$$

gde je E klasa entiteta. Time je određen skup P pojava tipa entiteta $N(A_1,..., A_n)$.

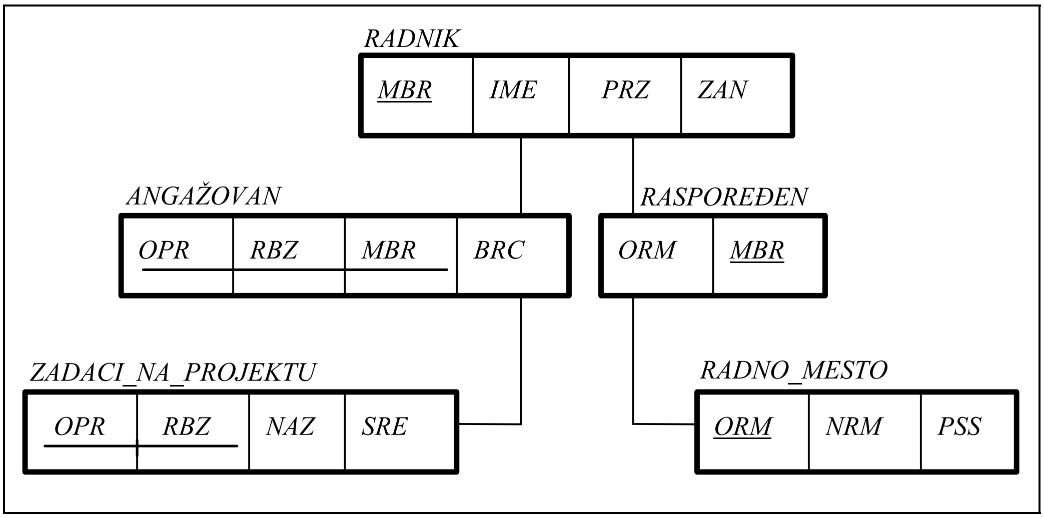
Ako su X i Y klase entiteta, a $R = \{(x, y) \mid x \in X, y \in Y$ i x je u vezi sa $y\}$, tada tip poveznika $N(X, Y, Q)$, gde je $Q = (A_1,..., A_k)$, eventualno, prazan niz obeležja, pridružuje semantiku svakom uređenom paru (x, y) iz R. ***Tip poveznika*** $N(X, Y, Q)$ predstavlja funkciju koja preslikava R u skup torki reda $k + 2$, tj.

$$(X, Y, Q) : R \rightarrow X \times Y \times dom(A_1) \times ... \times dom(A_k).$$

Torka $(x, y, a_1,..., a_k)$, gde $a_i \in dom(A_i)$, predstavlja ***pojavu tipa poveznika*** $N(X, Y, Q)$. Pri tome, x i y su entiteti.

Neka je K primarni ključ tipa entiteta $N(A_1,..., A_n)$, pridruženog klasi entiteta E. Preslikavanje $K : E \rightarrow P$, gde je P skup pojava tipa entiteta, je bijektivno. Saglasno tome, kao reprezent svakog entiteta e iz E može se koristiti vrednost ključa iz odgovarajuće pojave tipa entiteta. Dalje, pojava tipa poveznika $(x, y, a_1,..., a_k)$ može se transformisati u torku $(k_x, k_y, a_1,..., a_k)$, gde su k_x i k_y vrednosti ključeva tipova entiteta pridruženih, redom, klasama entiteta X i Y.

Ako se pretpostavi da u jednoj logičkoj strukturi nad skupom obeležja ne postoje dva tipa entiteta sa istim ključem, tada se i tip poveznika $N(X, Y, A_1,..., A_k)$ može transformisati u $N(K_x, K_y, A_1,..., A_k)$, gde su K_x i K_y primarni ključevi tipova entiteta pridruženih redom klasama X i Y. Ključ tipa poveznika $N(K_x, K_y, A_1,..., A_k)$ je $K \subseteq K_x \cup K_y$.



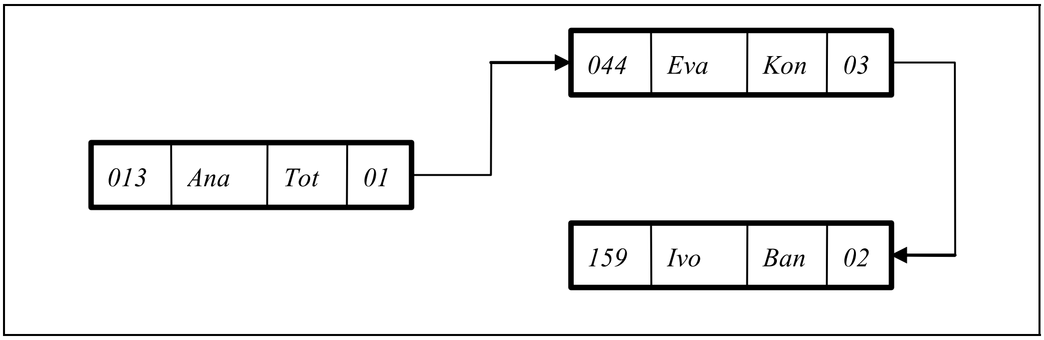
Slika1.17.

Primer 1.25. Na slici 1.17 je nacrtana ista logička struktura nad skupom obeležja kao i na slici 1.16, samo je za predstavljanje tipova poveznika upotrebljena geometrijska reprezentacija koja se koristi za predstavljanje tipa entiteta. Pod pretpostavkom da jedan radnik može biti raspoređen na samo jedno radno mesto, ključ tipa poveznika *RASPOREĐEN* je samo *MBR*. Neorijentisanim potezima na slici 1.17 nije pridružena semantika. □

Definicija 1.12. Kada se u skup *P* pojava tipa entiteta $N(A_1,..., A_n)$ ili tipa poveznika $N(K_x, K_y, A_1,..., A_k)$ uvede relacija strogog poretka sa semantikom "ima veću (manju) vrednost ključa od", dobija se ***logička struktura podataka datoteke***. □

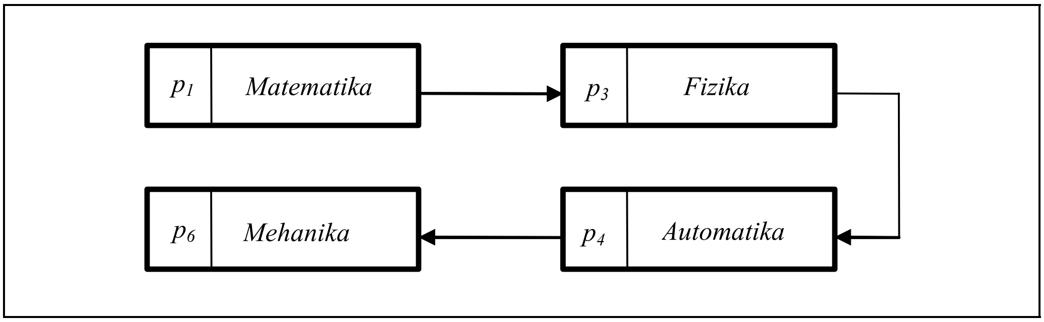
Znači, logičku strukturu podataka datoteke predstavlja graf *G* = (*S*, *R*) sa pridruženom semantikom, gde je *S* skup pojava jednog tipa entiteta, ili skup pojava jednog tipa poveznika, a *R* relacija strogog poretka, koja skup *S* uređuje saglasno rastućim (ili opadajućim) vrednostima ključa.

Primer 1.26. Neka je *S* skup pojava tipa entiteta *STUDENT*(<u>*BRI*</u>, *IME*, *PRZ*, *BPI*). Na slici 1.18 je prikazana logička struktura podataka datoteke *STUDENT*. □



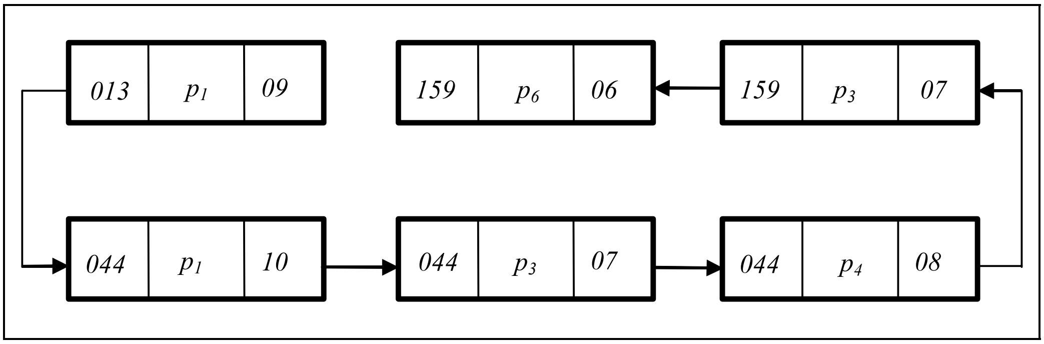
Slika 1.18.

Primer 1.27. Neka je *S* skup pojava tipa entiteta *PREDMET*(<u>*OZP*</u>, *NAZ*). Na slici 1.19 je prikazana logička struktura podataka datoteke *PREDMET*. □



Slika 1.19.

Primer 1.28. Neka je *X* skup pojava tipa entiteta *STUDENT* iz primera 1.26, *Y* skup pojava tipa entiteta *PREDMET* iz primera 1.27, a *ISPIT*(<u>*BRI*, *OZP*</u>, *OCE*) tip poveznika. Na slici 1.20 je prikazana logička struktura podataka datoteke *ISPIT*. □

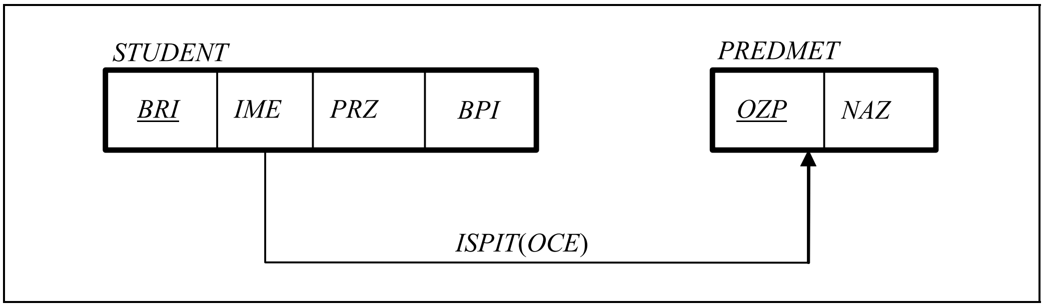


Slika 1.20.

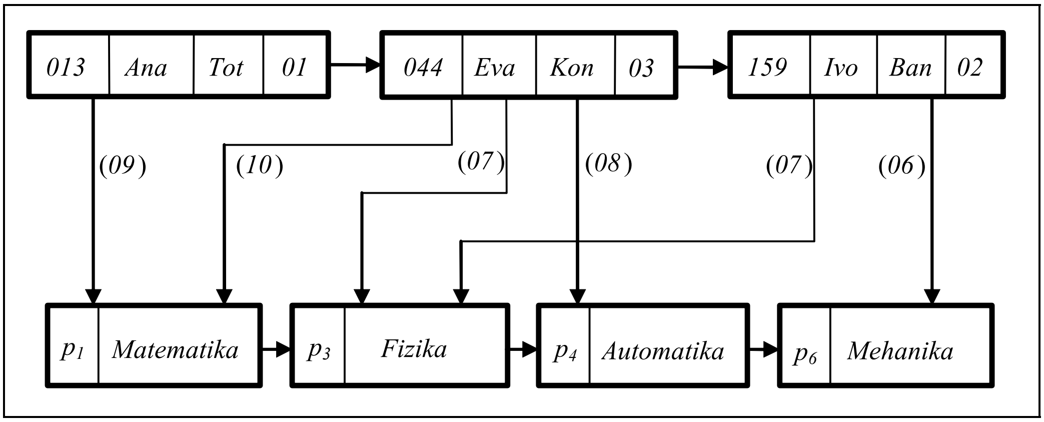
Neka je *LSO* logička struktura nad skupom obeležja, dobijena u koraku *L* projektovanja organizacije podataka.

Definicija 1.13. Logičku strukturu nad skupom podataka baze podataka predstavlja graf $G = (S, R)$ čijim čvorovima i ivicama je putem *LSO* pridružena semantika. □

Primer 1.29. Logička struktura nad skupom obeležja sa slike 1.21 definiše logičku strukturu podataka baze podataka, prikazanu na slici 1.22. □



Slika 1.21.



Slika 1.22.

Opisane strukture nad skupom podataka nazivaju se logičkim, jer predstavljaju sliku dela realnog sveta koja, po pravilu, egzistira samo u mislima projektanta, programera i korisnika datoteke ili baze podataka. Logičke strukture nad skupom podataka predstavljaju modele realnih sistema i njihovih baza podataka na nivou apstrakcije podataka.

Sama logička struktura nad skupom podataka je vremenski promenljiva struktura, ona prati promene u realnom sistemu. Za razliku od nje, logička struktura nad skupom obeležja se može smatrati uslovno statičnom. Tek značajne strukturalne promene u realnom sistemu dovode do promena i u logičkoj strukturi nad skupom obeležja.

Za prikazivanje logičkih struktura nad skupom podataka, pored usmerenih grafova, često se koriste i ***tabele sa homogenim kolonama***. Za skup pojava svakog tipa entiteta ili tipa poveznika koristi se posebna tabela. Iznad tabele se upisuje naziv tipa entiteta ili poveznika, u zaglavlje tabele se upisuju obeležja tipa entiteta ili poveznika, a u vrste tabele unose se pojave.

Primer 1.30. Na slici 1.23 je data tabelarna predstava logičke strukture nad skupom podataka iz primera 1.29. □

BRI	*IME*	*PRZ*	*BPI*
159	*Ivo*	*Ban*	*02*
013	*Ana*	*Tot*	*01*
044	*Eva*	*Kon*	*03*

OZP	*NAZ*
p_1	*Matematika*
p_3	*Fizika*
p_6	*Mehanika*
p_4	*Automatika*

BRI	*OZP*	*OCE*
159	p_3	*07*
013	p_1	*09*
044	p_3	*07*
044	p_1	*10*
159	p_6	*06*
044	p_4	*08*

Slika 1.23.

1.7 FIZIČKE STRUKTURE PODATAKA

Kada se semantika, pridružena čvorovima i ivicama logičke strukture nad skupom podataka smesti na medijum memorijskog uređaja, dobija se fizička struktura podataka. Pri tome se, za predstavljanje podataka, umesto prirodne koristi binarna azbuka. Pored podataka koje sadrži logička struktura, fizička struktura podataka se proširuje i podacima svojstvenim postupcima memorisanja na konkretnom uređaju, podacima koji opisuju određene osobine pojava tipova entiteta (na primer: dužina pojave izražena brojem znakova prirodne azbuke, pripadnost tipu entiteta i slično). Takođe, postoje različiti postupci za predstavljanje ivica logičke strukture nad skupom podataka na memorijskom medijumu.