UNIVERZITET U BEOGRADU

MAŠINSKI FAKULTET

Goran M. Lazović

PROJEKTOVANJE I ISPITIVANJE STRUKTURA BAZA PODATAKA U UPRAVLJANJU ODRŽAVANJEM VAZDUHOPLOVNIH SISTEMA

doktorska disertacija

Beograd, 2012

Mentor:

Dr Slobodan Radojević, vanredni profesor, Univerzitet u Beogradu, Mašinski Fakultet

Komentor:

Dr Slobodan Stupar, redovni profesor, Univerzitet u Beogradu, Mašinski Fakultet

Članovi komisije:

Dr Zlatko Petrović, redovni profesor, Univerzitet u Beogradu, Mašinski Fakultet

Dr Ivan Aranđelović, vanredni profesor, Univerzitet u Beogradu, Mašinski Fakultet

Dr Milan Božić, vanredni profesor, Univerzitet u Beogradu, Matematički Fakultet

Datum odbrane:

Projektovanje i ispitivanje struktura baza podataka u upravljanju održavanjem vazduhoplovnih sistema

Rezime

Visoki zahtevi koji se postavljaju sistemu za održavanje vazduhoplova između ostalog zahtevaju temeljnu analizu postojeće strategije održavanja i njeno značajno unapređenje. Savremeni sistemi održavanja obuhvataju i upravljanje održavanjem, tj. zasnovani su na automatizovanoj podršci održavanju primenom računara (*Computer Integrated Maintenance Management*). U osnovi automatizovane podrške održavanju primenom računara je model podataka. Veoma je važno da je model podataka dovoljno bogat da bi mogao, na relativno lak način prihvatiti i u sebe uključiti različite koncepte realnog sistema zajedno sa svojim atributima i međusobnim vezama. Hijerarhijske strukture podataka (*strukture podataka tipa stablo*) su posebno interesantne u sistemu održavanja vazduhoplova i one su predmet istraživanja u ovom radu. Postavlja se sledeći zahtev. Obezbediti takvu implementaciju hijerarhijske strukture koja pruža mogućnost njene efikasne obrade. Rekurzija koja je u osnovi hijerarhijske strukture podataka odnosno stabla koji je reprezent ovakve strukture podataka, može biti problem u efikasnom upravljanju sistemom održavanja vazduhoplova. U radu su date nove karakterizacije stabla, koje daju osnov za nerekurzivne modele hijerarhijske strukture podataka.

U kontekstu ograničenih resursa u sistemu za održavanje vazduhoplova posebno su interesantne optimalne strategije održavanja. Bitan segment sistema za upravljanje održavanjem je planiranje održavanja. Jedan od osnovnih problema između ostalog je neizvesnost događaja koji značajno utiču na samo planiranje. Reč je o događajima koji su po svojoj prirodi slučajni. Posebno analizira se planiranje održavanja u slučaju kada skup aktivnosti ima hijerarhijsku strukturu i pri tome je trajanje pojedinih aktivnosti iz datog skupa aktivnosti slučajna veličina. Pod pretpostavkom da aktivnost na višem nivou hijerarhije ne može započeti pre nego što su završene sve aktivnosti na nižem nivou hijerarhije i da su poznati gubici koji nastaju u slučaju da se sa nekim aktivnostima kasni, odnosno da su poznati gubici u slučaju da su neke aktivnosti preuranjene, potrebno je formirati optimalnu strategiju aktiviranja aktivnosti iz datog skupa. U slučaju hijerarhije visine jedan problem se svodi na poznati *newsboy problem*, ali je u opštem slučaju isuviše komplikovan. U radu se predlaže heuristički pristup kojim se do rešenja dolazi iterativnom primenom jednostavnog rešenja problema na dvononiovskoj hijerarhiji.

Vazduhoplovi su savremena saobraćajna sredstva vrhunske tehnologije sa velikim brojem komponenti i kompleksnom strukturom. Pojedine komponente vazduhoplova se koriste na različit način, trpe različita opterećenja i održavaju po programu koji je njima najbolje prilagođen. Jedan od načina savladavanja složenosti je takozvana hijerarhijska dekompozicija. Složeni koncept se na jednom nivou apstrakcije posmatra kao jedinstvena celina. Na nižem nivou apstrakcije se posmatra kao koncept koji se sastoji od delova (komponenti).

Uzastopnom primenom opisanog postupka dobija se takozvana hijerarhijska *sastavnica*. Dakle jedan od bitnih koncepata u sistemu održavanja vazduhoplova jeste sastavnica. Poseban problem je obezbediti kvalitetan sadržaj hijerarhijske strukture podataka, odnosno u sistemu za upravljanje održavanjem vazduhoplova treba obezbediti kvalitetnu sastavnicu. U radu je predložena fleksibilna struktura sastavnica koja u velikoj meri podržava njeno sadržajno raširenje.

Ključne reči: održavanje vazduhoplova, hijerarhijska struktura podataka, stablo, newsboy problem, sastavnica

Naučna oblast: tehničke nauke, mašinstvo
Uža naučna oblast: vazduhoplovstvo, računarske nauke
UDK: 004.6:629.7(043.3)

Design and investigation of database structures in aircraft maintenance management systems

Abstract

The high demands that are placed to aircraft maintenance systems among other things require thorough analysis of existing maintenance strategy and its significant improvement. Modern maintenance systems include the management of maintenance, i.e. they are based on automated maintenance support by computer - *Computer Integrated Maintenance Management*. The basic component in automated maintenance support account is data model. It is important that the data model is rich enough to accept in a relatively easy way a variety of real system concepts together with their attributes and interrelations. Hierarchical data structure (textit tree type data structure) are of special interest in the aircraft maintenance systems, and they are the subject of this study. This raise the following request. Implement hierarchical data structure which allows efficient processing. Recursion which is essential in implementation of hierarchical data structure, i.e., of the tree, which is typical representative of these data structures, can be a problem in efficient aircraft maintenance management. This study gives a new characterizations of tree, which provide a basis for nonrecursive models of hierarchical data structures.

In the context of limited resources in the aircraft maintenance systems, optimal maintenance strategies are of special interest. An important segment of the maintenance management system is maintenance planning. One of the major problems is uncertainty of events which significantly affect maintenance planning. These events are random by its nature. In particular, the planning of maintenance is analyzed in the case when the set of activities has a hierarchical structure and duration, of each activity from a given set of activities, is random value. Assuming that the activity at a higher level of hierarchy can not begin before the completion of all activities at a lower level of hierarchy, and that losses incurred in the case of delayed activities, i.e., losses incurred in the case of premature activity are known, it is necessary to establish optimal activating strategy for each activity from given set of activities. In the case of one level hierarchy, problem is reduced to the well known *newsboy problem*, but it is too complicated in general. In this study, a heuristic approach is proposed for the solution of general problem that is based on iterative application of the solution of a simple two levels hierarchy problem.

Aircrafts are modern high-tech means of transport with a large number of components and complex structure. Components of aircraft are used in different ways, they suffer different loads and are maintained by the programs that are best suited for them. One way to overcome this complexity is so-called hierarchical decomposition. Complex concept at one level of abstraction is seen as unified. At the lower level of abstraction it is seen as a concept that consists of parts (other components). Repeated application of described procedure gives the so-called hierarchical *bill of material*. So one of the important concepts in the aircraft maintenance systems is bill of material. A particular problem is to provide quality hierarchical data structures to the aircraft maintenance management system, i.e., to provide quality bill of material. This study proposes a flexible structure of the bill of material which greatly supports disseminate its content.

Keywords: aircraft maintenance, hierarchical data structure, tree, newsboy problem, bill of material

Scientific area: technical sciences, mechanical engineering
Scientific subarea: aeronautical engineering, computer science
UDC:004.6:629.7(043.3)

Sadržaj

Slike

Tabele

Glava 1

Uvod

Vazduhoplovi su savremena saobraćajna sredstva vrhunske tehnologije sa velikim brojem komponenti i kompleksnom strukturom, pa se svi radovi na održavanju moraju sistematizovati. Pod održavanjem se podrazumevaju sve akcije čja je svrha zadržavanje ili povratak elementa sistema u stanje u kojem može izvoditi predviđenu funkciju. Ove akcije između ostalog uključuju detekciju otkaza, opravku i zamenu delova koji su doveli do otkaza.
Prema [15], od tridesetih godina prošlog veka razvoj održavanja se odvijao u tri faze:

I) Prva faza obuhvata period do II Svetskog rata u kome industrija nije bila visoko mehanizovana, tako da otkazi opreme nisu bili značajni. U toj fazi nije davan poseban prioritet prevenciji otkaza. U isto vreme oprema je bila predimenzionisana, vrlo pouzdana i bila je laka za opravljanje tako da nije bilo potrebe za sistematskim održavanjem.

II) Druga faza je započela dramatičnim promenama u II Svetskom ratu, porastom zahteva za različitim robama i naglim porastom mehanizacije, odn. sve brojnije i sve složenije mašine izazivaju sve oštrije izražene pojave otkaza. Rađa se ideja da se otkazi mogu sprečiti prevencijom i javlja se koncept preventivnog održavanja. U šezdesetim se ovaj koncept ogleda u primeni preventivnog održavanja u fiksnim intervalima - održavanje po unapred utvrđenom resursu, koji može biti:

    – sati rada
    – pređeni kilometri
    – ciklusi funkcionisanja
    – motočasovi itd.

    Troškovi održavanja rastu rapidno u odnosu na ostale eksploatacione troškove, što dovodi do razvoja sistema planiranja i upravljanja održavanjem.

III) Treća faza počinje sredinom sedamdesetih godina i okarakterisana je porastom očekivanja od održavanja i pojavom novih istraživanja i novih tehnika održavanja. Dok se u prethodnoj fazi, od održavanja očekivalo da snizi troškove, poveća raspoloživost postrojenja i produži životni vek opreme, sada se od održavanja očekuje da, pored ranijih očekivanja, ispuni i sledeće zahteve:

- povećanje sigurnosti opreme kako za rukovaoce tom opremom, tako i za okolinu
- povećanje pouzdanosti opreme
- povećanje kvaliteta proizvoda
- povećanje efektivnosti investicija u održavanje

Ovako visoki zahtevi koji se postavljaju sistemu za održavanje između ostalog zahtevaju temeljnu analizu postojeće strategije održavanja i njeno značajno unapređenje. Pre svega razmotrimo postojeću strategiju održavanja vazduhoplova. U početku se pretpostavljalo da su resursi koji su na raspolaganju sistemu održavanja vazduhoplova neograničeni. U današnje vreme to već nije slučaj, ne samo da su resursi ograničeni već su i u pojedinim situacijama nedostupni. Sve ovo je uticalo na razvoj strategija održavanja u kontekstu ograničenih resursa. Od interesa su strategije održavanja kojima se obezbeđuje optimalna pouzdanost sistema uz minimalne troškove. Takve strategije održavanja zovemo optimalne strategije održavanja. Što se tiče aktivnosti koje se sprovode i vremenskih trenutaka kada se one sprovode [24], postoje dve glavne kategorije u sistemu održavanja vazduhoplova:

- *Korektivno održavanje*
- *Preventivno održavanje*

Pod korektivnim održavanjem (*run to failure*) se podrazumevaju aktivnosti koje se preduzimaju jedino u slučaju pojave otkaza, a čiji je cilj povratak u prvobitno stanje. U ovoj kategoriji održavanja se polazi od pretpostavke da je efikasnije izvršiti opravku odnosno zamenu nekog elementa tek kada otkaže. Dakle strategija održavanja je *ne diraj dok radi*. Očigledne prednosti ovakvog pristupa su:

- minimalni troškovi održavanja
- maksimalno iskorišćenje opreme (koristi se dok god funkcioniše)

Sa druge strane nedostaci su:

- nemogućnost planiranja održavanja imajući u vidu neizvesnost trenutka otkaza
- pouzdanost sistema je dovedena u pitanje

Korektivno održavanje je i dalje dominantna strategija održavanja.

Preventivno održavanje uključuje aktivnosti održavanja koje se sprovode pre nego što dođe do pojave otkaza, odnosno ove aktivnosti imaju zadatak da spreče ili odlože pojavu otkaza. U klasičnom preventivnom održavanju preduzima se preventivna zamena ili opravka na osnovu poznavanja prosečnog trajanja opreme. Polazna pretpostavka je da za svaki element sistema postoji pravi trenutak kada nad njim treba sprovesti određene aktivnosti preventivnog održavanja i da će se na ovaj način smanjiti učestalost otkaza. Međutim ispostavilo se da ovakav pristup ne umanjuje uvek učestalost otkaza. Pokazalo se da je ovako koncipirano preventivno održavanje pogodno samo za elemente sistema kod kojih učestalost otkaza poseduje izvesnu pravilnost. S obzirom na ovu činjenicu razlikovaćemo dve vrste preventivnog održavanja [24]. U prvom slučaju, preventivno održavanje se sprovodi, isključivo, na osnovu informacija o pouzdanosti, tj. na osnovu raspodele vremena rada do pojave otkaza za posmatrani sistem, odnosno njegov element. U drugom slučaju, posmatra se i prati neki pokazatelj, tj. parametar koji reprezentuje stanje posmatranog sistema. Obe ove vrste preventivnog održavanja mogu da se vežu za neki određeni vremenski period ili za vremenski trenutak koji se ne određuje unapred, već se tokom rada sistema podešava, tj. adaptira konstatovanom stanju, odnosno informacijama o izabranim parametrima stanja koji se prikupljaju i analiziraju Sa stanovišta karaktera postupaka održavanja, preventivno održavanje može biti realizovano kao:

- osnovno održavanje
- preventivne zamene
- održavanje prema stanju

U okvir osnovnog održavanja spadaju sve aktivnosti opsluživanja, snabdevanja gorivom, mazivom, podmazivanje, kao i osnovni pregledi stanja pojedinih elemenata sistema. Preventivna zamena uključuje aktivnosti zamene ili opravke elemenata i to samo u slučaju ako se na taj način, sasvim izvesno, doprinosi povećanju pouzdanosti, odnosno ako se time neposredno smanjuje verovatnoća pojave otkaza. Ideja kod održavanja prema stanju jeste preduzimati zamenu u optimalno vreme na osnovu stvarnog stanja opreme. Stvarno stanje se utvrđuje dijagnostičkim pregledima, a da se prethodno ne mora vršiti rastavljanje po principu *otvori i izmeri*. Prednosti ovakvog pristupa održavanju su:

- jednostavnije planiranje održavanja
- obezbeđuje se veća pouzdanost sistema

dok su nedostaci:

- povećani troškovi održavanja
- manja iskorišćenost opreme (oprema ne mora biti raspoloživa u toku pojedinih aktivnosti održavanja)

Savremeni sistemi održavanja obuhvataju i upravljanje održavanjem, tj. zasnovani su na automatizovanoj podršci održavanju primenom računara (*Computer Integrated Maintenance Management*). Dakle reč je o sistemu, pa ćemo u analizi sistema održavanja poći od pojma sistema. *Sistem* se, najopštije definiše kao skup objekata (*entiteta*) i njihovih medjusobnih veza. *Informacioni sistem* je sistem u kome se veze izmedju objekata i veze sistema sa okolinom ostvaruju razmenom informacija. Osnovu svakog informacionog sistema čini *baza podataka*, koja se može definisati kao kolekcija medjusobno povezanih podataka koja modelira (reprezentuje) objekte, veze objekata i atribute objekata posmatranog realnog sistema.
Na ovaj način informacioni sistem je model realnog sitema, a postupak projektovanja informacionog sistema se svodi na neku vrstu modeliranja realnog sistema. Rezultat faze projektovanja informacionog sistema su dva podmodela:

- *model podataka*
- *model procesa*

Model podataka obuhvata reprezentaciju objekata realnog sistema, njihovih atributa i njihovih medjusobnih veza. Dakle statičke karakteristike realnog sistema se opisuju modelom podataka. Sa druge strane dinamika realnog sistema se opisuje modelom procesa. Ovde ćemo se baviti samo modelom podataka.
Model podataka obuhvata reprezentaciju koncepata realnog sistema, njihovih atributa i njihovih medjusobnih veza. Veoma je važno da je model podataka dovoljno bogat da bi mogao, na relativno lak način prihvatiti i u sebe uključiti različite koncepte realnog sistema zajedno sa svojim atributima i međusobnim vezama. Ove zavisnosti se, u modelu podataka, mogu predstaviti bilo strukturom podataka, bilo skupom ograničenja na vrednosti podataka. Pored toga, neophodno je definisati i skup operacija modela podataka, da bi se preko njih, u modelima procesa, mogla opisati i dinamika realnog sistema. Dakle model podataka uključuje tri osnovne komponente:

1) *struktura podataka*
2) ograničenja
3) operacije

Kvalitet modela podataka se meri time u kojoj meri su obuhvaćeni koncepti realnog sistema i jednostavnošću implementacije struktura podataka, ograničenja i operacija na računaru. Strukture podataka se mogu klasifikovati uz pomoć više kriterijuma. Jedan od kriterijuma može biti karakteristika veze (relacije) među podacima. U odnosu na ovaj kriterijum struktura podataka može biti [Mogin]:

- *linearna struktura podataka*
- *nelinearna struktura podataka*

U slučaju linearne strukture podataka, skup podataka je *linearno uređen.* Drugim rečima skup podataka formira niz, odnosno svaki podatak (sem dva istaknuta, koje zovemo prvi odnosno poslednji podatak) je povezan sa tačno dva podatka, koje zovemo prethodni odnosno naredni podatak. Podatak za koga ne postoji prethodni je prvi podatak, a podatak za koga ne postoji naredni je poslednji podatak. Strukture podataka u kojima skup podataka nije linearno uređen zovemo nelinearne strukture podataka. Prema tome koliki je broj podataka koji prethode datom podatku, odnosno koliki je broj podataka koji su naredni u u odnosu na dati podatak, nelinearne strukture podataka dalje mogu biti:

- *hijerarhijska struktura podataka*
- *mrežna struktura podataka*

Hijerarhijske strukture podataka ili (*strukture tipa stablo*) su takve strukture podataka u kojima za svaki podatak (sem istaknutog podatka koga zovemo koren hijerarhije) postoji tačno jedan podatak koji mu prethodi, pri tome može postojati nula ili više podataka koji su u odnosu na njega naredni podaci. U slučaju hijerarhijske strukture podataka umesto termina prethodni odnosno naredni podatak koiste se termini nadređeni odnosno podređeni podatak. Koren hijerarhije nema nadređeni podatak. Nelinearne strukture podataka koje nisu hijerarhijske su mrežne strukture podataka. Hijerarhijske strukture podataka su posebno interesantne u sistemu održavanja vazduhoplova i one će biti predmet istraživanja u ovom radu. Vazduhoplov je sistem visoke složenosti, pojedine komponente vazduhoplova koriste na različit način, trpe različita opterećenja i održavaju se po programu koji je njima najbolje prilagođen. Jedan od načina savladavanja pomenute složenosti je takozvana hijerarhijska dekompozicija. Složeni koncept se na jednom nivou apstrakcije posmatra kao jedinstvena celina. Na nižem nivou apstrakcije se posmatra kao koncept koji se sastoji od delova (komponenti). Uzastopnom primenom opisanog postupka dobija se takozvana hijerarhijska *sastavnica*. Dakle jedan od bitnih koncepata u sistemu održavanja vazduhoplova jeste sastavnica. U sistemu održavanja se koriste termini sklop, podsklop nadsklop koji istovremeno označavaju i uzajamni odnos u datoj hijerarhiji. Dakle u sistemu za upravljanje održavanjem imaćemo sastavnice sklopova koje se dobijaju hijerarhijskom dekompozicijom elemenata sistema nad kojima se izvode akcije održavanja.
Koliko je ovaj koncept važan govori i činjenica da su se prve *hijerarhijske baze podataka* šezdesetih godina prošlog veka razvijene kako bi se direktno podržale sastavnice u prizvodnim organizacijama. Dakle model podataka informacionog sistema koji bi bio podrška sistemu održavanja mora obezbediti efikasnu implementaciju hijerarhijske strukture podataka i operacija nad hijerarhijskim strukturama podataka. Obično se hijerarhijske strukture podataka implementiraju rekurzivnim strukturama koje dalje nameću obradu rekurzivnim procedurama, a poznato je da rekurzivna rešenja mogu značajno degradirati performanse sistema. Problemi vezani za implementaciju hijerarhijskih struktura podataka i neka rešenja biće izneta u Glavi 2.
U delu planiranja održavanja, jedan od osnovnih problema između ostalog je neizvesnost događaja koji značajno utiču na efikasnost održavanja. Reč je o događajima koji su po svojoj prirodi slučajni. Ovi slučajni događaji su pored ostalog vezani za:

- vremenski trenutak otkaza rezervnog dela
- vremenski trenutak zahteva za nekim delom
- vreme isporuke, odnosno vreme potrebno za sklapanje (pripremu) rezervnog dela

Posebno, podrazumevalo se da je vreme isporuke odnosno vreme potrebno za pripremu rezervnog dela nula (rezervni deo je neposredno dostupan), a to je pretpostavka neograničenih resursa što je uglavnom redak slučaj. Kako mnogo faktora utiče na ovo vreme, ono po pravilu uvek ima neko slučajano odstupanje. Za neke analize vreme isporuke odnosno vreme sklapanja rezervnog dela se može ignorisati, ali u analizi kompletnog sitema održavanja ova odstupanja bitno degradiraju performanse sistema. Pod vremenom isporuke (*lead time*) za dati rezervni deo se podrazumeva:

- vreme koje je potrebno da bi se rezervni deo dopremio iz nekog spoljnjeg izvora
- vreme koje je potrebno da bi se rezervni deo sklopio prema utvrđenoj proceduri sklapanja
- vreme da bi se poluproizvod obradio do nivoa gotovog rezervnog dela

Na ovo vreme utiče mnoštvo faktora. Neki od faktora su

- problemi u transportu
- kvar na mašinama
- problemi vezani za kvalitet

Ovi faktori su po svojoj prirodi slučajne veličine, pa je i ovo vreme isporuke po pravilu slučajna veličina. Jedan od pristupa kojim se može umanjiti uticaj ovih slučajnih veličina je skladištenje gotovih rezervnih delova, odnosno održavanje zaliha gotovih rezervnih delova. U slučaju da je ovaj nivo iznad odgovarajućeg javiće se nepotrebni troškovi skladištenja. Sa druge strane ako je nivo zaliha ispod odgovarajućeg nivoa javiće se troškovi kašnjenja. Cilj je održavati nivo zaliha kojim se obezbeđuju minimalni troškovi skladištenja sa jedne strane a pri tome se garantuje visok nivo raspoloživosti gotovih delova kako bi se minimizovali troškovi kašnjenja. Za uspešno rešavanje ovog problema neophodno je razviti odgovarajuće matematičke modele koji će na odgovarajući način tretirati neizvesnost pojedinih faktora u sistemu održavanja. Posebno problem određivanja nivoa zaliha rezervnih delova u kontekstu slučajnog broja zahteva za rezervnim delovima je problem koji je najviše izučavan i analiziran. Kako je naručivanje i prodaja dnevnih novina ilustrativna metafora za ovaj tip problema, za probleme ovog tipa koristi se termin *newsboy problem*, a za odgovarajući model se koristi termin *newsboy model*. Newsboy problemi nisu samo problemi iz oblasti upravljanja zalihama, oni se javljaju u svim situacijama kada treba izvršiti ocenu neke slučajne promenljive, a ta ocena je rezultat kompromisa između gubitaka u slučaju da je vrednost slučajne promenljive precenjena i gubitaka koji su posledica potcenjene vrednosti slučajne promenljive. Posebno ovaj problem se može iskoristiti u situacijama kada treba odrediti optimalni vremenski trenutak kada treba započeti neku aktivnost održavanja.

U planiranju održavanja česta je sledeća situacija. Neki skup aktivnosti održavanja ima hijerarhijsku strukturu i pri tome je trajanje svake aktivnosti slučajna veličina. Pod pretpostavkom da aktivnost na višem nivou hijerarhije ne može započeti pre nego što su završene sve aktivnosti na nižem nivou hijerarhije i da su poznati gubici koji nastaju u slučaju da se sa nekim aktivnostima kasni, odnosno da su poznati gubici u slučaju da su neke aktivnosti preuranjene, potrebno je formirati optimalnu strategiju aktiviranja aktivnosti iz datog skupa. Upravo to će biti predmet istraživanja u Glavi 3.

Primetili smo da je jedan od bitnih koncepata u sistemu održavanja vazduhoplova sastavnica. Sastavnica je najprostije rečeno, lista komponenti koji su sastavni delovi posmatranog objekta. Pri tome se vodi računa o vezama između sastavnih komponenti. Sastavnica dakle daje strukturu posmatranog objekta, još više sastavnica je opis hijerarhijske strukture tog objekta. Prema tome da li se sastavnica koristi u projektovanju, proizvodnji ili održavanju, sastavnica može biti:

– *inženjerska sastavnica*

– *proizvodna sastavnica*

– *servisna sastavnica*

i u tom smislu će biti prilagođen sadržaj kojim će biti snabdevene. Posebno sastavnice koje će se koristiti u sistemu održavanja će između ostalog uključivati i sledeće podatke:

– količina pojedinih podsklopova koja je neophodna za sklapanje finalnog sklopa

– vremena neophodna za ugradnju pojedinih podsklopova u procesu sklapanja finalnog sklopa

– redosled ugradnje pojedinih podsklopova u procesu sklapanja finalnog sklopa

Kvalitetan sistem za upravljanje održavanjem vazduhoplova mora obezbediti sadržajno kvalitetne sastavnice i to će biti predmet istraživanja u Glavama 4 i 5.

Glava 2

Hijerarhijske strukture podataka

2.1 Uvodna razmatranja

Pod strukturom podataka podrazumeva se skup međusobno povezanih podataka. Strukture podataka se mogu klasifikovati na osnovu više kriterijuma. Jedan od kriterijuma može biti karakteristika veze (relacije) među podacima. U odnosu na ovaj kriterijum struktura podataka može biti [14]:

– *linearna struktura podataka*

– *nelinearna struktura podataka*

U slučaju linearne strukture podataka, skup podataka je *linearno uređen*. Drugim rečima skup podataka formira niz, odnosno svaki podatak (sem dva istaknuta, koje zovemo prvi odnosno poslednji podatak) je povezan sa tačno dva podatka, koje zovemo prethodni odnosno naredni podatak. Podatak za koga ne postoji prethodni je prvi podatak, a podatak za koga ne postoji naredni je poslednji podatak. Strukture podataka u kojima skup podataka nije linearno uređen zovemo nelinearne strukture podataka. Prema tome koliki je broj podataka koji prethode datom podatku, odnosno koliki je broj podataka koji su naredni u u odnosu na dati, podatak nelinearne strukture podataka dalje mogu biti:

– *hijerarhijska struktura podataka*

– *mrežna struktura podataka*

Hijerarhijske strukture podataka ili (*strukture podataka tipa stablo*) su takve strukture podataka u kojima za svaki podatak (sem istaknutog podatka koga zovemo koren hijerarhije) postoji tačno jedan podatak koji mu prethodi, pri tome može postojati nula ili više podataka koji su u odnosu na njega naredni podaci. U slučaju hijerarhijske strukture podataka umesto termina prethodni odnosno naredni podatak koiste se termini nadređeni odnosno podređeni podatak. Koren hijerarhije nema nadređeni podatak. Nelinearne strukture podataka koje nisu hijerarhijske su mrežne strukture podataka. U daljem tekstu ćemo se detaljnije baviti hijerarhijskom strukturom podataka odnosno strukturom podataka tipa stablo. Kao što sam naziv kaže u osnovi hijerarhijske strukture podataka je stablo.

Engleski matematičar Kejli je 1857. godine uveo pojam stabla [27]. Koncept stabla je korišćen i ranije, ali Kejli je bio prvi koji je iskoristio ovaj pojam u pisanom radu. Skoro u isto vreme (oko 1859.) otkrivene su strukturne formule hemijskih jedinjenja. Kejli je našao vezu između ova dva pojma (povezao je stabla i strukturne formule alkana - jedinjenja formule C_nH_{2n+2}) i u radu *O matematičkoj teoriji izomera* iz 1874. godine je postavio kamen temeljac naučne discipline Hemijske teorije grafova. Kejli je pokušao da pronađe broj izomera I_n alkana C_nH_{2n+2}, nije uspeo u tome, ali je dao nekoliko važnih rezultata vezanih za prebrojavanje stabala. Sa druge strane Fizičar Kirhof je rešavajući problem izračunavanja jačina električnih struja u granama nekog električnog kola, došao do značajnih rezultata vezanih za stabla.
Što se tiče računarskih nauka, stablo takođe ima veoma značajnu ulogu, između ostalog koristi se na sledeći način:

- algoritmi pretrage koriste stabla (binarno stablo pretrage) za uređivanje podataka
- kompajleri koriste stabla (stablo izraza) za reprezentaciju aritmetičkih izraza
- algoritmi za kompresiju podataka koriste stabla (stablo kodiranja)
- operativni sistemi koriste hijerarhijsku strukturu datotečnog sistema

U ovom radu stablo ćemo koristiti za reprezentaciju hijerarhijske strukture podataka. Iz očiglednih razloga (ograničeni resursi) bavićemo se samo konačnim stablima. Postavlja se sledeći zahtev. Obezbediti takvu implementaciju stabla koja pruža mogućnost efikasne obrade hijerarhijske strukture podataka. Najčešće se hijerarhijske strukture podataka implementiraju rekurzivnim strukturama koje dalje nameću rekurzivnu implementaciju operacija nad ovim strukturama podataka. Rekurzivna rešenja su sa jedne strane jednostavna i efektna, ali sa druge strane mogu značajno degradirati performanse sistema. U potrazi za alternativnim rešenjima biće neophodna detaljnija analiza samog pojma stabla. Ako se stablo uvede rekurzivnom definicijom što je najčešći slučaj u računarskim naukama, onda se prirodno nameću rekurzivna rešenja kako implementacije tako i procedura za njihovu obradu. U teoriji skupova se stablo uvodi kao klasa parcijalno uređenih skupova. Ako zanemarimo šta su elementi tih skupova, problem se svodi na problem implementacije binarne relacije (parcijalnog uređenja), a ovo se standardno rešava takozvanom *matricom susedstva*. Za neka nova rešenja problema implementacije stabla, neophodne su neke nove karakterizacije konačnog stabla. Dalje se iz izvedenih karakterizacija konačnog stabla mogu izvući modeli implementacije stabla (hijerarhijske strukture podataka) [5] [7]

- *model sa listom susedstva*
- *model sa izlistanim putanjama*
- *model sa ugnježdenim skupovima*

i njihovi varijeteti:

- *model sa tranzitivnim zatvorenjem*
- *Dewey-jeva decimalna notacija*
- *model sa ugnježdenim intervalima*

2.2 Stablo

2.2.1 Rekurzivna definicija stabla

Najčešće korišćena definicija stabla je [11]:

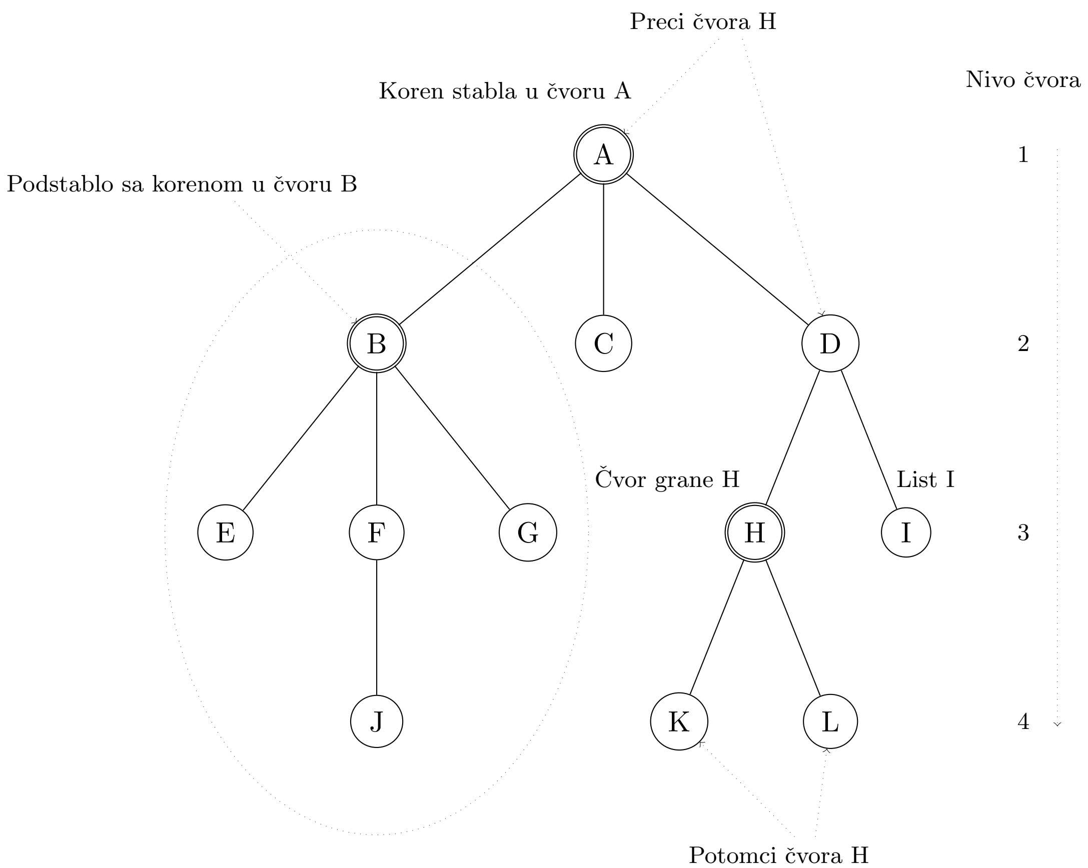
Definicija 2.2.1. *Stablo* $\mathcal{T}$ je neprazan skup čvorova T takav da važi:

a) postoji tačno jedan istaknuti *čvor* u T koga zovemo *koren* stabla $\mathcal{T}$ i označavamo sa root ($\mathcal{T}$);

b) preostali čvorovi su elementi disjunktnih skupova $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m, \quad m \geq 0$ i svaki od ovih skupova je stablo.

◇

Primer stabla sa najvažnijim elementima dat je na slici (2.1).



Slika 2.1: Stablo sa najvažnijim elementima

Stabla $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m$ zovemo *podstablima* korena stabla $\mathcal{T}$. Iz definicije (2.2.1) sledi da je svaki čvor koren nekog podstabla polaznog stabla $\mathcal{T}$. Broj podstabala nekog čvora zovemo *stepen čvora*. Čvor stepena nula zovemo *terminalni čvor* ili *list* , inače je *neterminalni čvor* ili *čvor grane*. *Nivo čvora* u stablu $\mathcal{T}$ se definiše na sledeći način:

Definicija 2.2.2. Neka je $\mathcal{T}$ stablo, tada:

a) root $(\mathcal{T})$ ima nivo jedan;

b) čvor koji nije koren stabla $\mathcal{T}$ ima nivo koji je za jedan veći od nivoa koga ima u podstablu $\mathcal{T}_j$, $j \in \{1, 2, \ldots, m\}$ u kome se nalazi.

$\diamond$

Visina stabla je maksimalni nivo na kome se neki čvor nalazi. Ako postoji uređenje podstabala $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m$, onda kažemo da je stablo $\mathcal{T}$ uređeno . Za korene podstabala $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m$ kažemo da su *potomci* korena stabla $\mathcal{T}$, odnosno koren stabla $\mathcal{T}$ je *predak* korena njegovih podstabala $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_m$. Svaki čvor ima najviše jednog pretka, a nula ili više potomaka. Stablo $\mathcal{T}$ je *binarno stablo*, ako je svaki čvor koren najviše dva podstabla koje zovemo levo, odnosno desno podstablo i označavamo sa left$(\mathcal{T})$, right$(\mathcal{T})$ respektivno. Šuma je skup (obično uređen skup) skup disjunktnih stabala. Svaka šuma stabala se na prirodan način može reprezentovati binarnim stablom [11].

Definicija 2.2.3. Neka je $\mathcal{F} = (\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n)$ šuma stabala i neka su $\mathcal{T}_{11}, \mathcal{T}_{12}, \ldots, \mathcal{T}_{1m}$ podstabla korena krajnjeg levog podstabla $\mathcal{T}_1$. Binarno stablo $\mathcal{B}(\mathcal{F})$ određeno sa:

– ako je $n = 0$, $\mathcal{B} = \emptyset$

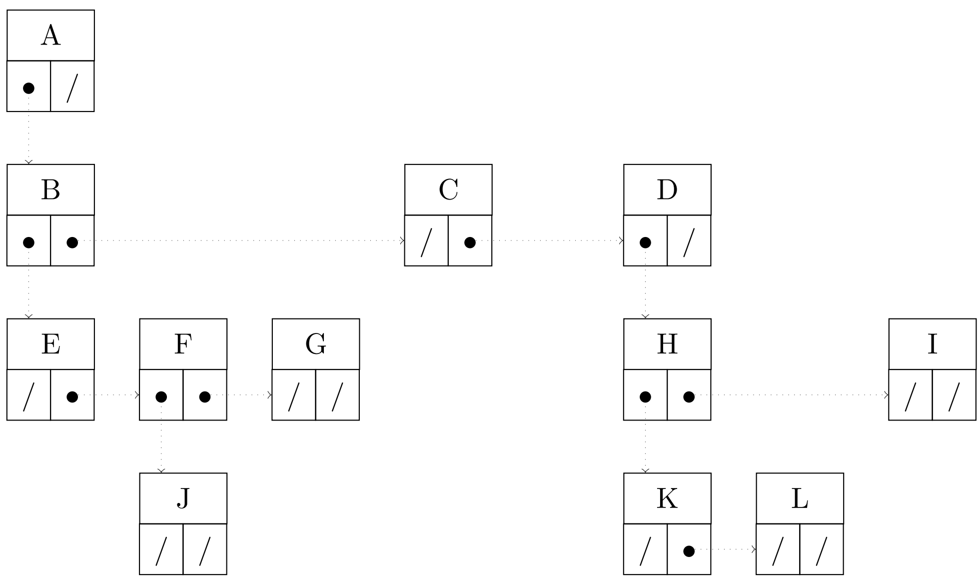
– ako je $n > 0$,

- $\text{root}(\mathcal{B}(\mathcal{F})) = \text{root}(\mathcal{T}_1)$
- $\text{left}(\mathcal{B}(\mathcal{F})) = \mathcal{B}(\mathcal{T}_{11}, \mathcal{T}_{12}, \ldots, \mathcal{T}_{1m})$
- $\text{right}(\mathcal{B}(\mathcal{F})) = \mathcal{B}(\mathcal{T}_2, \mathcal{T}_3, \ldots, \mathcal{T}_n)$

je binarno stablo koje odgovara šumi stabala $\mathcal{F}$

$\diamond$

Suština navedene reprezentacije (2.2):



Slika 2.2: Implementacija stabla sa slike (2.1)

je sledeća:

- koren levog podstabla je krajnji levi potomak - *first child*
- koren desnog podstabla je sledeći potomak nadređenog čvora - *next sibling*.

Rekurzivna definicija (2.2.1) prirodno nameće kako implementaciju rekurzivnim strukturama (2.1):

Algoritam 2.1 Rekurzivna struktura podataka

```
typedef type_of_data_in_the_node info_t;
struct node
{
    info_t info;
    struct node* first_child;
    struct node* next_sibling;
};
typedef node *nodep;
```

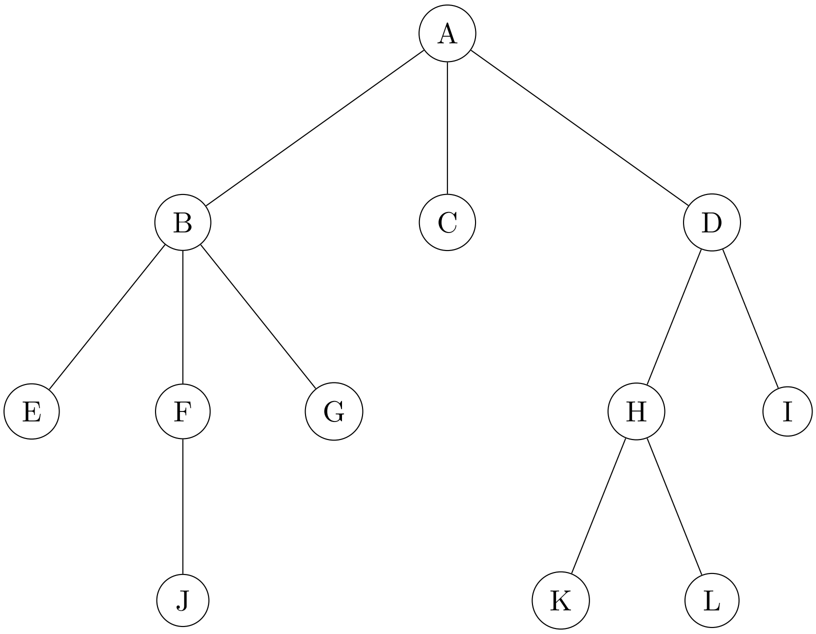
tako i obradu rekurzivnim procedurama. Procedure za obilaženje čvorova stabla u odgovarajućem redosledu su posebno značajne. Kako se radi o obradi rekurzivne strukture podataka, odgovarajuća rekurzivna rešenja su jednostavna i efektna. Jedna od standardnih procedura za *pre-order* obilaženje stabla i njena rekurzivna verzija je sledećeg oblika (2.2):

Algoritam 2.2 Rekurzivna obrada

```
preorder(nodep rt)
{
    nodep temp

    output(Info(rt))
    temp = firstChild(rt)
    while(temp is not NULL)
    {
        preorder(temp)
        temp = nextSibling(temp)
    }
}
```

U pre-order obilasku stabla [11], prvo se obilazi čvor, a zatim njegovi potomci od krajnjeg levog do krajnjeg desnog. U slučaju stabla sa slike (2.1), rezultat pre-order obilaska je (2.3). Rekurzivnost strukture podataka i procedura za njihovu obradu ne moraju biti direktno podržani modelom podataka u kome se ove strukture i pripadajuće procedure implementiraju.



ABEFJGCDHKLI

Slika 2.3: Pre-Order obilazak stabla sa slike (2.1)

Nivo ozbiljnosti ovog ograničenja, još je veći ako se zna da rekurzivna rešenja mogu značajno degradirati performanse sistema. Sva ova ograničenja opravdavaju istraživanja modela koja proističu iz nerekurzivnih karakterizacija stabla.

2.2.2 Nerekurzivna definicija stabla

Stablo je skup čvorova na kome je definisana binarna relacija *predak*. Kako je relacija predak *relacija poretka*, poći ćemo od parcijalno uređenih skupova [13].

Definicija 2.2.4. Neka je $\leq$ relacija poretka na skupu P. Uređen par $(P, \leq)$ zovemo *parcijalno uređen skup*.

$\diamond$

Standardni primeri parcijalno uređenih skupova su:

- $(\mathbb{P}(T), \subseteq)$ gde je T proizvoljan skup, $\mathbb{P}(T)$ *partitivni skup* skupa T i $\subseteq$ relacija podskup. Još više proizvoljna kolekcija $\mathbb{X}$ podskupova skupa T je parcijalno uređen skup ako se uredi relacijom *podskup* (*nadskup*)
- proizvoljan skup reči nad datom *azbukom* A koji sadrži *praznu reč*, je parcijalno uređen skup ako se uredi relacijom *prefiks* (*sufiks*, *infiks*)

Uređenje $\leq$ na skupu P je *linearno* ili *totalno uređenje* ako pored uslova koji definišu uređenje ispunjava i uslov linearnosti:

$$p, q \in P \Rightarrow p \leq q \vee q \leq p \tag{2.1}$$

U tom slučaju par $(P, \leq)$ se naziva *linearno uređen skup*, *totalno uređen skup* ili *lanac*. To znači da je parcijalno uređen skup linearno uređen ako su svaka dva njegova elementa uporediva. Neka je $(P, \leq)$ parcijalno uređen skup, tada:

– za element $p \in P$ kažemo da je *minimalan* ako je:

$$(\neg \exists q \in P)\, q \neq p \wedge q \leq p \tag{2.2}$$

odnosno p je minimalan ako u P ne postoji strogo manji element od njega. Slično se uvodi *maksimalni* element parcijalno uređenog skupa.

– za element $p \in P$ kažemo da je *najmanji* ako važi:

$$q \in P \Rightarrow p \leq q \tag{2.3}$$

odnosno p je najmanji element u P ako je manji od svakog drugog elementa iz P. Slično se uvodi *najveći* element parcijalno uređenog skupa.

Ukoliko parcijalno uređen skup $(P, \leq)$ ima najmanji element tada je on jedinstven. Uređeni skup može imati više minimalnih elemenata i tada ne može imati najmanji element. Za parcijalno uređen skup $(P, \leq)$ kažemo da je *dobro uređen* ako je linearno uređen i svaki njegov neprazan podskup ima najmanji elemenat. Očigledno je svaki konačan linearno uređen skup dobro uređen. Sada možemo dati sledeću definiciju stabla [13]:

Definicija 2.2.5. Stablo $\mathcal{T}$ je parcijalno uređenje $(T, \leq)$ koje ima najmanji element i pri tome je:

$$(\forall x \in T) \uparrow(x) = \{y \in T | y \leq x\} \tag{2.4}$$

skup $\uparrow(x)$ dobro uređen relacijom $\leq$.

$\diamond$

Od interesa su nam konačna stabla, odnosno konačni parcijalno uređeni skupovi T, sa najmanjim elementom takvim da je za svako $x \in T$, $\uparrow(x)$ linearno uređen skup odnosno lanac. Najmanji element stabla je koren, a maksimalne elemente zovemo listovima. Visina stabla je određena najdužim lancem:

$$H = \max_{x \in T} \{|\uparrow(x)|\} \tag{2.5}$$

2.3 Modeli stabla

2.3.1 Model sa listom susedstva

Kako svaki čvor stabla sem korena ima jedinstvenog direktnog pretka (možemo reći da je koren predak samom sebi) onda na skupu čvorova stabla postoji preslikavanje (direktni predak) sa tačno jednom fiksnom tačkom. Ova osobina nameće prvu karakterizaciju stabla i data je sledećom teoremom:

Teorema 2.3.1. *Konačan skup T na kome je definisana funkcija $f : T \to T$ sa osobinama:*

a) postoji tačno jedna fiksna tačka r funkcije f ,

$$(\exists_1 r \in T)\, f(r) = r \tag{2.6}$$

b) za svako $n \in \mathbb{N}$

$$(\neg\exists x \in T)\, x \neq r \wedge f^n(x) = x \tag{2.7}$$

je konačno stablo.

Dokaz. *Definišimo relaciju $\leq$ na T na sledeći način:*

$$(x, y \in T)\, y \leq x \Leftrightarrow (\exists k \in \mathbb{N}_0)\, y = f^k(x) \tag{2.8}$$

pri čemu je $f^0 = I_T$ identičko preslikavanje skupa T.
Očigledno je $\leq$ relacija poretka. Neka je $n \in \mathbb{N}$, označimo sa:

$$F_n(x) = \left\{ f^k(x) | k \in \mathbb{N}_0 \wedge k < n \right\} \tag{2.9}$$

Kako je T konačan skup svakom $x \in T$ možemo pridružiti:

$$n_x = min\,\{n \in \mathbb{N} | f^n(x) \in F_n(x)\} \tag{2.10}$$

Mora biti $f^{n_x}(x) = r$, jer u suprotnom

$$f^{n_x}(x) = u \in F_{n_x}(x) \Rightarrow f^{n_x}(x) = f^k(x) \tag{2.11}$$

za neko $k < n_x$, odnosno

$$f^{n_x - k}(u) = f^{n_x - k}\left(f^k(x)\right) = f^k(x) = u \tag{2.12}$$

što je kontradikcija sa pretpostavkom o funkciji f, dakle $f^{n_x}(x) = r$ odnosno r je najmanji element. Očigledno je:

$$\uparrow(x) = \{y \in T | y \leq x\} = \left\{ f^k(x) | k < n_x \right\} = F_{n_x}(x) \tag{2.13}$$

linearno uređen skup pri tome konačan, dakle dobro uređen skup.

◇

Neka je $(T, \leq)$ konačno stablo sa korenom r, funkcija (direktni predak) f iz navedene karakterizacije konačnog stabla je:

$$f(x) = \begin{cases} r & , x = r \\ max\,\{y | y \lneq x\} & , x \neq r \end{cases} \tag{2.14}$$

Iz navedene karakterizacije stabla izvodi se i najčešće korišćeni model [5]. Za svaki čvor x beleži se njegov jedinstveni direktni predak $y = f(x)$, odnosno beleže se grane stabla. Kako je koren stabla predak samom sebi, odnosno jedini čvor koji nema drugog pretka, sem samog sebe, odgovarajuća grana se obično izostavlja iz modela. Ovo je takozvani *model sa listom susedstva* (*adjacency list model*) (2.1).

Tabela 2.1: Lista susedstva stabla sa slike (2.3)

čvor	roditelj
B	A
C	A
D	A
E	B
F	B
G	B
H	D
I	D
J	F
K	H
L	H

Očigledno je da se u ovom modelu mogu efikasno:

– dodavati novi čvorovi - funkcija direktni predak je raširenje postojeće funkcije

– brisati čvorovi - funkcija direktni predak je suženje postojeće funkcije

Nedostatak ovog modela je neefikasno generisanje podstabla sa korenom u datom čvoru. Neefikasnost je posledica činjenice da se odnos predak/potomak ne može neposredno ustanoviti u opštem slučaju. Neposredno se može ustanoviti odnos između čvora i njegovih direktnih potomaka. U opštem slučaju ovaj odnos se proverava na sledeći način:

– čvor je *potomak* datog čvora ako je:

  · njegov direktni potomak, ili je

  · potomak njegovog direktnog potomka;

– čvor je *predak* datog čvora ako je:

  · njegov direktni predak, ili je

  · predak njegovog direktnog pretka.

i zato se ovaj model naziva *rekurzivni model*. Kako se odnos neposredno proverava samo ako su čvorovi na uzastopnim nivoima stabla, odnos između čvorova koji nisu na uzastopnim nivoima se proverava izračunavanjem f^k, $k = 1, 2, \ldots$.

2.3.2 Model sa tranzitivnim zatvorenjem

Jedno poboljšanje modela sa listom susedstva se sastoji u sledećem [Celko]. Umesto funkcije direktni predak beleži se njeno *refleksivno* i *tranzitivno* zatvorenje. Problem tranzitivnog zatvorenja je poseban problem koji se ovde neće detaljno obrađivati. Na ovaj način se iz funkcije direktni f predak reprodukuje kompletna relacija poretka $\leq$:

$$\leq = \bigcup_{k\in\mathbb{N}_0} f^k \tag{2.15}$$

gde je:

$$\leq = \{(y,x)|(x,y\in T)y\leq x\} \tag{2.16}$$

$$f^k = \left\{(y,x)|(x,y\in T)y = f^k(x)\right\} \tag{2.17}$$

U ovom modelu za svaki čvor x beleže se svi njegovi prethodnici $y = f^k(x)$, $k = 1, 2, ...$, odnosno beleži se niz grana od čvora do korena. Ovo je takozvani *model sa tranzitivnim zatvorenjem* (*transitive closure model*) (2.2).

Tabela 2.2: Tranzitivno zatvorenje liste susedstva (2.1)

čvor	predak
A	A
B	A
B	B
C	A
C	C
D	A
D	D
E	A
E	B
E	E
F	A
F	B
F	F
G	A
G	B
G	G
H	A
H	D
H	H
I	A
I	D
I	I
J	A
J	B
J	F
J	J
K	A
K	D
K	H
K	K
L	A
L	D
L	H
L	L

Na ovaj način se efikasno može testirati odnos između bilo koja dva čvora. Ako se sa svakim zapisom (y, x) gde je $y \leq x$ odnosno $y = f^k(x)$ za neko $k \in \mathbb{N}$, beleži i k, koje predstavlja nivo čvora[1] x u podstablu sa korenom u čvoru y, odnosno dužina jedinstvene putanje od čvora x do čvora y, neposredno je dostupan i lanac prethodnika čvora x (2.3).

[1] Preciznije nivo čvora je $k + 1$

Tabela 2.3: Uređenje na tranzitivnom zatvorenju (2.2)

čvor	predak	nivo
A	A	1
B	A	2
B	B	1
C	A	2
C	C	1
D	A	2
D	D	1
E	A	3
E	B	2
E	E	1
F	A	3
F	B	2
F	F	1
G	A	3
G	B	2
G	G	1
H	A	3
H	D	2
H	H	1
I	A	3
I	D	2
I	I	1
J	A	4
J	B	3
J	F	2
J	J	1
K	A	4
K	D	3
K	H	2
K	K	1
L	A	4
L	D	3
L	H	2
L	L	1

Nedostatak ovog modela je što za razliku od modela sa listom susedstva kojim se stablo sa n čvorova reprezentuje sa $n-1$ zapisa, broj zapisa u ovom modelu je reda $n \cdot h$ gde je h prosečan nivo čvora u stablu. Pored ovog nedostatka očigledni su problemi:

– dodavanja novog čvora

– brisanja postojećeg čvora

2.3.3 Efikasnost modela

Razmotrimo sada (neformalno) pitanje efikasnosti modela implementacije stabla, pri čemu ćemo efikasnost modela meriti efikasnošću standardnih procedura za njihovu obradu. Efikasnost ispitivanja odnosa, odnosno izračunavanja vrednosti izraza $y \leq x$ na čvorovima stabla $(T, \leq)$ je svakako posebno važna. U modelu sa listom susedstva testiranje odnosa između čvorova se u opštem slučaju svodi na rekurzivnu proveru postojanja niza zapisa:

$$(y, u_1), (u_1, u_2), \ldots (u_{n-1}, u_n), (u_n, x)$$

što je očigledno neefikasno. U modelu sa tranzitivnim zatvorenjem testiranje odnosa između čvorova se svodi na efikasniju proveru postojanja odgovarajućeg zapisa (y, x), ali je nedostatak što broj zapisa može ići do reda n^2 u slučaju da je stablo sa n čvorova lanac (prostor pretrage može biti značajno veliki). Problem efikasnog izračunavanja vrednosti izraza $y \leq x$ na čvorovima stabla $(T, \leq)$ se može rešiti na sledeći način. Uoči se neki parcijalno uređen skup X, takav da se odnos između njegovih elemenata može efikasno utvrditi. Elemente skupa X ćemo zvati obeležja. Zatim se konstruiše *izomorfizam* φ skupa čvorova T datog stabla na skup $\varphi(T) \subseteq X$. Na ovaj način se složena provera odnosa dva čvora $x, y \in T$ svodi na efikasnu proveru odnosa dva obeležja:

$$\varphi(x), \varphi(y) \in f(T) \subseteq X$$

U modelima koji slede neće biti potrebno da se beleže grane, niti putanje od čvora do korena stabla kao što je to bio slučaj u prethodna dva modela, već će se samo beležiti obeležja čvorova stabla. Na ovaj način će se u modelima koji slede efikasno:

- dodavati novi čvorovi
- brisati postojeći čvorovi

Dakle stablo će uvek biti reprezentovano sa brojem zapisa koji je jednak broju čvorova stabla. Posebno, svakom čvoru $x \in T$ možemo pridružiti obeležje koje je skup njegovih predaka $\uparrow (x)$ i umesto stabla $(T, \leq)$ posmatraćemo strukturu $(\uparrow T, \leq_\uparrow)$ gde je:

$$\uparrow T = \{\uparrow (x) | x \in T\}\,, \tag{2.18}$$

i pri tome je $\leq_\uparrow$ pogodno izabrana relacija na $\uparrow T$ tako da su $(T, \leq)$ i $(\uparrow T, \leq_\uparrow)$ *izomorfne strukture*. U tom slučaju, odnos između čvorova $x, y \in T$ možemo efikasno testirati na pridruženim obeležjima $\uparrow (x), \uparrow (y)$ jer važi:

$$y \leq x \Leftrightarrow \uparrow (y) \leq_\uparrow \uparrow (x) \tag{2.19}$$

Slično, svakom čvoru $x \in T$ možemo pridružiti obeležje koje je skup njegovih potomaka $\downarrow (x)$ i umesto stabla $(T, \leq)$ možemo posmatrati strukturu $(\downarrow T, \leq_\downarrow)$ gde je:

$$\downarrow T = \{\downarrow (x) | x \in T\} \tag{2.20}$$

i pri tome je $\leq_\downarrow$ pogodno izabrana relacija na $\downarrow T$ tako da su $(T, \leq)$ i $(\downarrow T, \leq_\downarrow)$ izomorfne strukture. U tom slučaju, odnos između čvorova $x, y \in T$ možemo testirati na pridruženim obeležjima $\downarrow (x), \downarrow (y)$ jer važi:

$$y \leq x \Leftrightarrow \downarrow (y) \leq_\downarrow \downarrow (x) \tag{2.21}$$

Za efikasnu implementaciju predložene metodologije neophodne su nam pogodnije karakterizacije stabla.

2.3.4 Model sa izlistanim putanjama

Za karakterizaciju konačnog stabla koja sledi, neophodne su sledeće definicije.

Definicija 2.3.1. Neka je T konačan skup i neka je $\mathcal{C}$ kolekcija podskupova od T. $\mathcal{C}$ je $\subseteq$-hijerarhija na skupu T ako važi:

a) $T = \bigcup \{C | C \in \mathcal{C}\} = \bigcup \mathcal{C}$

b) $U, V \in \mathcal{C} \Rightarrow U \cap V \in \mathcal{C}$

c) $U, V, W \in \mathcal{C} \wedge U, V \subseteq W \Rightarrow U \subseteq V \vee V \subseteq U$

$\diamond$

Definicija 2.3.2. $\subseteq$-hijerarhija $\mathcal{C}$ na skupu T je $\mathcal{T}_{\subseteq}$-hijerarhija na skupu T ako važi:

a) za sve $x, y \in T$, $x \neq y$, postoji $U \in \mathcal{C}$ tako da tačno jedan od x, y pripada U

b) unija kolekcije elementa hijerarhije $\mathcal{C}$ je elemenat hijerarhije $\mathcal{C}$ akko je jednaka nekom članu kolekcije

$\diamond$

Definicija 2.3.3. $\subseteq$-hijerarhija $\mathcal{C}$ na skupu T je povezana $\subseteq$-hijerarhija na skupu T ako važi:

– postoji istaknuti element $r \in T$ takav da važi $U \in \mathcal{C} \Rightarrow r \in U$.

$\diamond$

Sada možemo dati sledeću karakterizaciju konačnog stabla.

Teorema 2.3.2. *Povezana $\mathcal{T}_{\subseteq}$-hijerarhija $\mathcal{C}$ na konačnom skupu T je konačno stablo.*

Dokaz. *Neka je $\mathcal{C}$ povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T. Primetimo da $\{r\}$ mora biti elemenat povezane $\mathcal{T}_{\subseteq}$-hijerarhije $\mathcal{C}$, gde je r istaknuti element skupa T. Neka to nije slučaj i neka je $U \in \mathcal{C}$ skup sa najmanjim brojem elemenata (ili jedan od njih). Kako je $\mathcal{C}$ povezana $\subseteq$-hijerarhija mora biti $r \in U$, odnosno $U = \{r, x_1, x_2, \ldots, x_k\}$, $k \geq 1$. $\mathcal{C}$ je $\mathcal{T}_{\subseteq}$-hijerarhija i $r \neq x_1$ pa postoji $V \in \mathcal{C}$ takav da tačno jedan od r, x_1 pripada V, to svakako mora biti r jer je $\mathcal{C}$ povezana $\subseteq$-hijerarhija dakle $x_1 \notin V$. Uočimo skup $U \cap V \in \mathcal{C}$ koji je pravi podskup skupa U, što je suprotno pretpostavci o skupu U. Dakle $\{r\} \in \mathcal{C}$.*
Sada možemo definisati funkciju (relaciju) direktni predak $f : \mathcal{C} \to \mathcal{C}$ na sledeći način:

$$f(U) = \begin{cases} \{r\} & U = \{r\} \\ \bigcup \mathcal{C}_U & U \neq \{r\} \end{cases} \tag{2.22}$$

gde je $\mathcal{C}_U = \{V \in \mathcal{C} | V \subsetneq U\}$.
Neka je $U \neq \{r\}$ elemenat povezane $\mathcal{T}_{\subseteq}$-hijerarhije $\mathcal{C}$ na skupu T, pokazaćemo da mora biti $f(U) \subsetneq U$. Kolekcija $\mathcal{C}_U$ je neprazna jer $\{r\} \in \mathcal{C}_U$, dalje kako su svi elementi kolekcije $\mathcal{C}_U$ podskupovi od $U \in \mathcal{C}$ i pri tome je $\mathcal{C}$ konačna $\subseteq$-hijerarhija, postoji $V_0 \in \mathcal{C}_U$ takav da za svako $V \in \mathcal{C}_U$ važi $V_i \subseteq V_0$, odnosno postoji $V_0 \in \mathcal{C}_U$ tako da važi $V_0 = \bigcup \mathcal{C}_U = f(U)$. Kako $V_0 \in \mathcal{C}_U$ zaključujemo $f(U) = V_0 \subsetneq U$.
Još više, neka je $U \neq \{r\}$ elemenat povezane $\mathcal{T}_{\subseteq}$-hijerarhije $\mathcal{C}$ na skupu T tada važi: $f^k(U) \subsetneq U$ za sve $k \in N$. Iz prethodnog možemo zaključiti da funkcija f ima tačno jednu fiksnu tačku $\{r\}$ i pri tome f^k $k = 2, 3, \ldots$ nema drugih fiksnih tačaka sem tačke $\{r\}$, pa je na osnovu teoreme (2.3.1) povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T stablo.

$\diamond$

Funkcija f iz prethodne teoreme, definisana na povezanoj $\mathcal{T}_{\subseteq}$-hijerarhiji $\mathcal{C}$ na konačnom skupu T, ima sledeću osobinu. Neka su $U, V \in \mathcal{C}$ i $k \in N_0$, tada važi:

$$V = f^k(U) \Leftrightarrow V \subseteq U \tag{2.23}$$

odnosno relacija podskup $\subseteq$ na $\mathcal{C}$ je refleksivno i tranzitivno zatvorenje ovako definisane funkcije (relacije) direktni predak f.

$$\subseteq = \bigcup_{k \in N_0} f^k \tag{2.24}$$

gde je:

$$\subseteq = \{(V,U)|(U,V \in \mathcal{C})V \subseteq U\} \tag{2.25}$$

$$f^k = \left\{(V,U)|(V,U \in \mathcal{C})V = f^k(U)\right\} \tag{2.26}$$

Drugim rečima povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T je stablo ako se uredi relacijom podskup.
Neka je $(T,\leq)$ konačno stablo, pokažimo da se može konstruisati povezana $\mathcal{T}_{\subseteq}$-hijerarhija $\mathcal{C}$ na skupu T. Pre svega primetimo da će u ovoj karakterizaciji svakom čvoru $x \in T$ odgovarati skup njegovih predaka, odnosno skup čvorova kojima je on potomak. U tom smislu definišimo preslikavanje $\uparrow: T \to \mathbb{P}(A)$ na sledeći način:

$$\uparrow(x) = \{z \in T | z \leq x\} \tag{2.27}$$

gde je $\mathbb{P}(A)$ partitivni skup skupa čvorova T. Sada nije teško pokazati sledeći stav.

Teorema 2.3.3. *Neka je $(T,\leq)$ konačno stablo. Kolekcija*

$$\mathcal{C} = \{\uparrow(x)|x \in T\} = \uparrow T \tag{2.28}$$

je povezana $\mathcal{T}_{\subseteq}$-hijerarhija na skupu T.

Dokaz. *Dokažimo prvo da je $\mathcal{C}$ $\subseteq$-hijerarhija. Primetimo $x \in \uparrow(x)$ za sve $x \in T$ pa je $T = \bigcup\{\uparrow(x)|x \in T\} = \bigcup\mathcal{C}$. Neka su $U,V \in \mathcal{C}$, tada je $U = \uparrow(x), V = \uparrow(y)$, za neke $x,y \in T$. Kako je $U \cap V \subseteq U = \uparrow(x)$ i pri tome je $\uparrow(x)$ konačan linearno uređen skup pa postoji $z \in U \cap V$ takav da je on najveći element skupa $U \cap V$, odnosno važi $U \cap V = \uparrow(z) \in \mathcal{C}$. Neka su dalje $U,V,W \in \mathcal{C}$ i neka važi $U,V \subseteq W$ tada je $U = \uparrow(x), V = \uparrow(y), W = \uparrow(z)$, za neke $x,y,z \in T$. Kako važi $x \in U = \uparrow(x)$, $y \in V = \uparrow(y)$ i $U = \uparrow(x), V = \uparrow(y) \subseteq W = \uparrow(z)$ biće $x,y \in W = \uparrow(z)$ pri tom je $\uparrow(z)$ linearno uređen skup pa važi bar jedan od uslova:*

- *$y \leq x$ odnosno $V = \uparrow(y) \subseteq U = \uparrow(x)$,*
- *$x \leq y$ odnosno $U = \uparrow(x) \subseteq V = \uparrow(y)$*

odnosno važi $U \subseteq V \vee V \subseteq U$. Dakle $\mathcal{C} = \{\uparrow(x)|x \in T\}$ je $\subseteq$-hijerarhija.
Pokažimo dalje da je $\mathcal{C}$ povezana $\subseteq$-hijerarhija. Neka je r koren stabla T, u tom slučaju za svaki čvor $x \in T$ važi $r \leq x$ odnosno $r \in \uparrow(x)$ za sve $x \in T$. Kako je $\mathcal{C} = \{\uparrow(x)|x \in T\}$ sledi da je $\mathcal{C}$ povezana $\subseteq$-hijerarhija.
Dokažimo još da je $\mathcal{C}$ $\mathcal{T}_{\subseteq}$-hijerarhija. Neka su $x,y \in T$ i $x \neq y$, jasno $x \in \uparrow(x)$, odnosno $y \in \uparrow(y)$. Neka je $y \in \uparrow(x)$, u protivnom ako $y \notin \uparrow(x)$ skup $U = \uparrow(x)$ razdvaja čvorove

x, y. U tom slučaju $x \notin \uparrow(y)$ jer bi u suprotnom bilo $x = y$, pa skup $U = \uparrow(y)$ razdvaja čvorove x, y. Dakle svaka dva različita čvora stabla T se mogu razdvojiti nekim skupom U iz kolekcije $\mathcal{C}$. Ostalo je da se dokaže da je unija kolekcije elemenata $\subseteq$-hijerarhije $\mathcal{C}$ elemenat $\subseteq$-hijerarhije $\mathcal{C}$ samo ako je jednaka nekom elementu kolekcije. Neka je $\{C_1, C_2, \ldots, C_k\}$ kolekcija elemenata $\subseteq$-hijerarhije $\mathcal{C}$ i $k \in N$, tada je $C_i = \uparrow(x_i)$ za neke $x_i \in T$ $i = 1, 2, \ldots, k$. Pretpostavimo $\bigcup C_i = \bigcup \uparrow(x_i) \in \mathcal{C}$, odnosno neka postoji čvor $x \in T$ takav da je $\bigcup C_i = \bigcup \uparrow(x_i) = \uparrow(x) \in \mathcal{C}$. Dokažimo da je $\uparrow(x) = \uparrow(x_i)$ za neko $i \in \{1, 2, \ldots, k\}$. Kako $x_i \in \uparrow(x_i)$ sledi $x_i \in \bigcup \uparrow(x_i) = \uparrow(x)$, odnosno $x_i \leq x$ za sve $i = 1, 2, \ldots, k$. Sa druge strane je $x \in \uparrow(x) = \bigcup \uparrow(x_i)$ odnosno $x \in \uparrow(x_i)$ za neko $i \in \{1, 2, \ldots, k\}$, pa je $x \leq x_i$ za neko $i \in \{1, 2, \ldots, k\}$, kako je pri tome $x_i \leq x$ zaključujemo da je $x = x_i$ odnosno $\uparrow(x) = \uparrow(x_i)$ za neko $i \in \{1, 2, \ldots, k\}$ što je i trebalo dokazati.

$\diamond$

Važi i obrat, ako je $\mathcal{C}$ povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T tada se skup T može urediti nekom relacijom $\leq$ tako da je $(T, \leq)$ stablo i svaki element kolekcije $\mathcal{C}$ je skup predaka nekog čvora stabla T. Pre svega pokažimo da povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T ima isti broj elemenata kao i konačan skup T.

Teorema 2.3.4. *Neka je $\mathcal{C}$ povezana $\mathcal{T}_{\subseteq}$-hijerarhija na konačnom skupu T, tada postoji bijekcija $\uparrow : T \to \mathcal{C}$*

Dokaz. *Problem je odrediti čvor $x \in T$ koji odgovara elementu $C \in \mathcal{C}$. Kako je $T = \bigcup\{C | C \in \mathcal{C}\} = \bigcup \mathcal{C}$, za svaki čvor $x \in T$ postoji $U \in \mathcal{C}$ tako da važi $x \in U$, odnosno svaki čvor je elemenat bar jednog podskupa skupa čvorova T iz kolekcije $\mathcal{C}$. Neka je $x \in T$ označimo sa $\mathcal{C}_x$ kolekciju elemenata povezane $\mathcal{T}_{\subseteq}$-hijerarhije $\mathcal{C}$ koji sadrže čvor x:*

$$\mathcal{C}_x = \{U \in \mathcal{C} | x \in U\} \tag{2.29}$$

Na osnovu prethodnog razmatranja $\mathcal{C}_x$ je neprazna kolekcija za sve $x \in T$. Sada možemo formirati skupove:

$$\cap_x = \bigcap \{U \in \mathcal{C} | x \in U\} = \bigcap \mathcal{C}_x \tag{2.30}$$

Očigledno $x \in \cap_x$ za sve $x \in T$. Kako je $\mathcal{C}$ $\subseteq$-hijerarhija presek elemenata kolekcije $\mathcal{C}$ je element kolekcije $\mathcal{C}$, odnosno $\{\cap_x | x \in T\} \subseteq \mathcal{C}$. Dakle za $x \in T$, $\cap_x$ je najmanji elemenat kolekcije $\mathcal{C}$ koji sadrži x, odnosno za $x \in T$ i $U \in \mathcal{C}$ važi:

$$x \in U \Rightarrow \cap_x \subseteq U \tag{2.31}$$

Pokažimo da je ovim povezana $\mathcal{T}_{\subseteq}$-hijerarhija $\mathcal{C}$ iscrpljena, odnosno $\mathcal{C} = \{\cap_x | x \in T\}$. Pretpostavimo suprotno, neka postoji $C \in \mathcal{C}$ i za svako $x \in T$, $C \neq \cap_x$. Neka je $C = \{x_1, x_2, \ldots, x_k\} \subseteq T$ i $k \in N$. Kako je $x_i \in C$ mora biti $\cap_{x_i} \subseteq C$ za sve $i = 1, 2, \ldots, k$ odnosno $\bigcup \cap_{x_i} \subseteq C$. Sa druge strane za svako $x_i \in C$ važi $x_i \in \cap_{x_i}$ pa važi $C \subseteq \bigcup \cap_{x_i}$, dakle $\bigcup \cap_{x_i} = C$. Kako je $\mathcal{C}$ $\mathcal{T}_{\subseteq}$-hijerarhija i pri tome je $\bigcup \cap_{x_i} = C \in \mathcal{C}$ mora biti $\bigcup \cap_{x_i} = C = \cap_{x_i}$ za neko $i \in \{1, 2, \ldots, k\}$ što je suprotno pretpostavci, dakle $\mathcal{C} = \{\cap_x | x \in T\}$.

Pokažimo još da u $\mathcal{C} = \{\cap_x | x \in T\}$ *nema istih, odnosno da različitim čvorovima* $x, y \in T$ *odgovaraju različiti* $\cap_x, \cap_y \in \mathcal{C}$. *Neka su* $x, y \in T$ *i neka je* $x \neq y$, *pretpostavimo* $\cap_x = \cap_y \in \mathcal{C}$. *Kako* $x, y \in \cap_x = \cap_y$ *i* $\mathcal{C}$ *je* $\mathcal{T}_{\subseteq}$*-hijerarhija, postoji* $C \in \mathcal{C}$ *tako da tačno jedan od* x *i* y *pripada* C. *Neka* $x \in C$ *i* $y \notin C$. *U ovom slučaju iz* $x \in C$ *sledi* $\cap_x \subseteq C$ *pa i* $y \in C$ *što je nemoguće, dakle* $\cap_x \neq \cap_y$. *Konačno preslikavanje* $\uparrow: T \to \mathcal{C}$ *definisano sa* $\uparrow(x) = \cap_x$ *je dobro definisano i pri tome je bijekcija.*

◇

Teorema 2.3.5. *Neka je* $\mathcal{C}$ *povezana* $\mathcal{T}_{\subseteq}$*-hijerarhija na konačnom skupu* T, *tada postoji uređenje* $\leq$ *skupa* T *tako da je* $(T, \leq)$ *stablo i svaki element kolekcije* $\mathcal{C}$ *je skup predaka nekog čvora stabla* T *u odnosu na tako uvedeno uređenje* $\leq$.

Dokaz. *Preslikavanje* $\uparrow: T \to \mathcal{C}$ *definisano sa:*

$$\uparrow(x) = \cap_x = \bigcap \{U \in \mathcal{C} | x \in U\} = \bigcap \mathcal{C}_x \tag{2.32}$$

je bijekcija na osnovu prethodnog stava. Uredimo skup T *na sledeći način:*

$$y \leq x \Leftrightarrow \cap_y \subseteq \cap_x \tag{2.33}$$

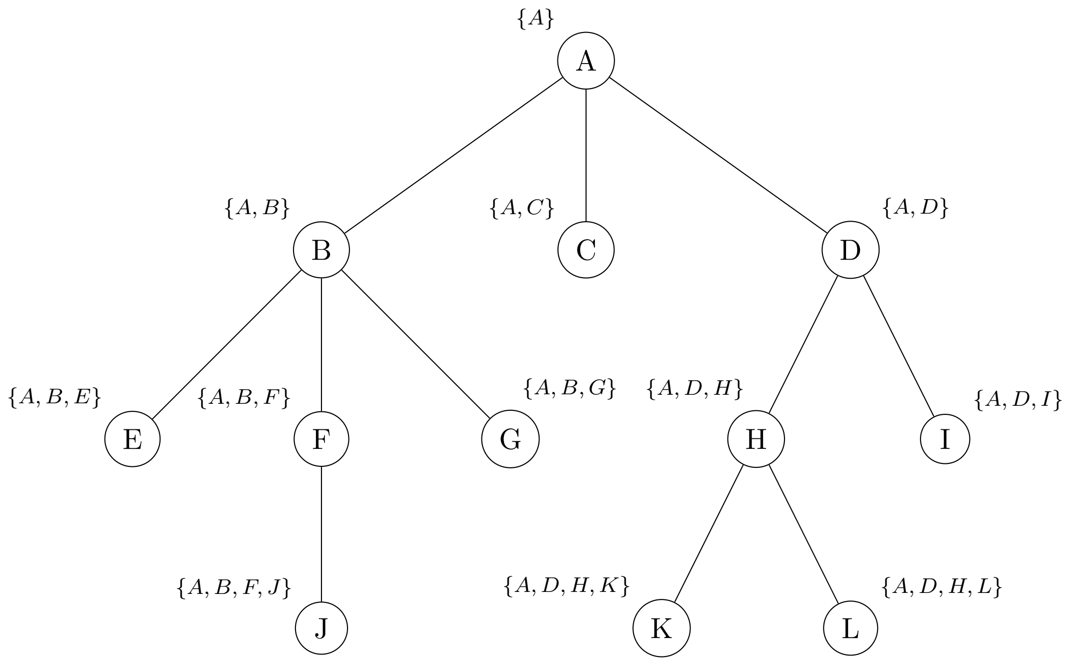
Na ovaj način $\uparrow$ *je izomorfizam struktura* $(T, \leq)$ *i* $(\mathcal{C}, \subseteq)$, *pa je struktura* $(T, \leq)$ *stablo, jer je to slučaj sa strukturom* $(\mathcal{C}, \subseteq)$. *Dokažimo sada da je pri ovakvom uređenju čvorova stabla* T, *svaki elemenat povezane* $\mathcal{T}_{\subseteq}$*-hijerarhije* $\mathcal{C}$ *skup predaka nekog čvora* x *stabla* T. *Dovoljno je dokazati da za* $x \in T$ *važi* $\{y \in T | y \leq x\} = \cap_x$.
Neka je $y \in \{y \in T | y \leq x\}$, *odnosno neka je* $y \leq x$. *S obzirom kako je uvedeno uređenje* $\leq$ *biće* $\cap_y \subseteq \cap_x$. *Kako je* $y \in \cap_y$ *mora biti* $y \in \cap_x$. *Dakle* $\{y \in T | y \leq x\} \subseteq \cap_x$. *Dokažimo i obrat* $\cap_x \subseteq \{y \in T | y \leq x\}$. *Pretpostavimo da je* $y \in \cap_x$ *tada je* $\cap_x \in \mathcal{C}_y$ *pa mora biti* $\cap_y \subseteq \cap_x$ *odnosno s obzirom na definiciju relacije* $\leq$ *važi* $y \leq x$ *odnosno* $y \in \{y \in T | y \leq x\}$. *Dakle* $\uparrow(x) = \cap_x = \{y \in T | y \leq x\}$, *što je i trebalo dokazati.*

◇

Iz navedenog zaključujemo da nema razlike između konačnog stabla $(T, \leq)$ i povezane $\mathcal{T}_{\subseteq}$-hijerarhije $\mathcal{C}$ na skupu T, odnosno to su izomorfne strukture. Dakle umesto stabla $(T, \leq)$ možemo posmatrati povezanu $\mathcal{T}_{\subseteq}$-hijerarhiju $\mathcal{C}$ na skupu T, pri tome se odnos između čvorova stabla $(T, \leq)$ preslikava na odnos između elemenata kolekcije $\mathcal{C}$ na sledeći način:

$$y \leq x \Leftrightarrow \uparrow(y) \subseteq \uparrow(x) \tag{2.34}$$

gde su $x, y \in T$, $\uparrow(x), \uparrow(y) \in \mathcal{C}$ odgovarajući elementi struktura $(T, \leq)$ i $(\mathcal{C}, \subseteq)$. U ovom modelu čvor $x \in T$ se reprezentuje skupom čvorova $\uparrow(x)$ kojima je on potomak, odnosno skupom predaka (2.4).



Slika 2.4: Skupovi predaka

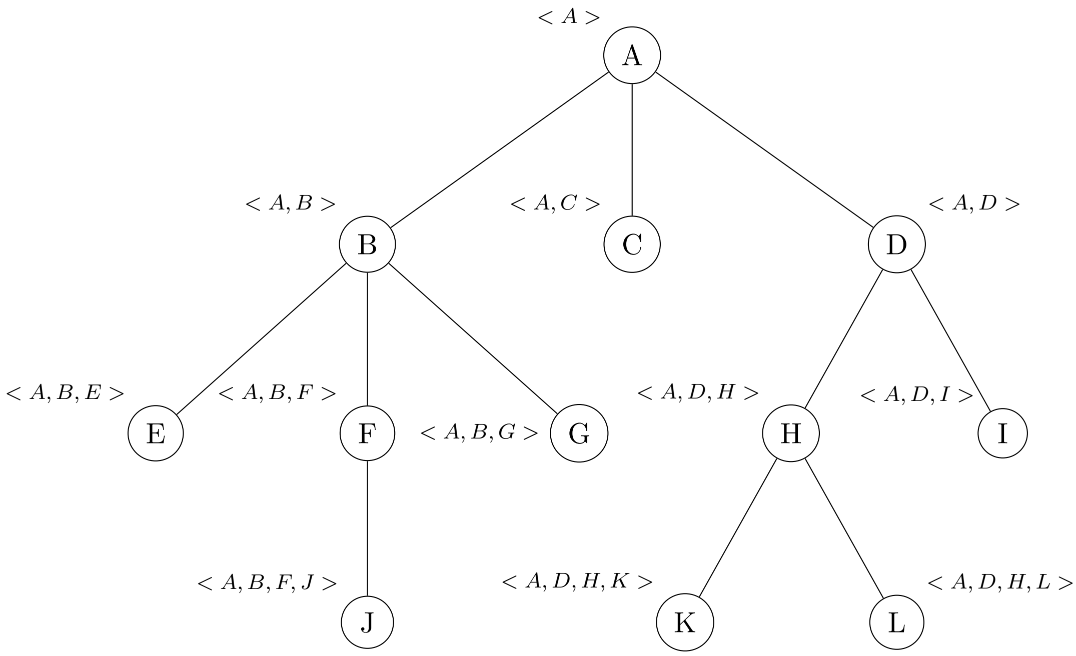
Dakle za obeležje čvora $x \in T$ uzimamo skup njegovih predaka $\uparrow(x)$. Kako je skup predaka dobro uređen, obeležje čvora x će biti niz čvorova na jedinstvenoj putanji od korena r stabla T do čvora x:

$$\begin{aligned} <\uparrow(x)> &= <r, c_1, c_2, \ldots, c_k, x> \\ &= \{\{r\}, \{r, c_1\}, \ldots \{r, c_1, \ldots, c_k, x\}\} \end{aligned}$$

gde su $c_1, c_2, \ldots, c_k \in T$ čvorovi na jedinstvenoj putanji od korena stabla r do čvora x. Nije teško pokazati da važi:

$$<\uparrow(y)> \subseteq <\uparrow(x)> \Leftrightarrow <\uparrow(y)> \leq_p <\uparrow(x)> \tag{2.35}$$

gde je $\leq_p$ relacija prefiks na skupu nizova $\{<\uparrow(x)> | x \in T\}$. Ovo je takozvani *model sa izlistanim putanjama* (*path enumeration model*) (2.5).



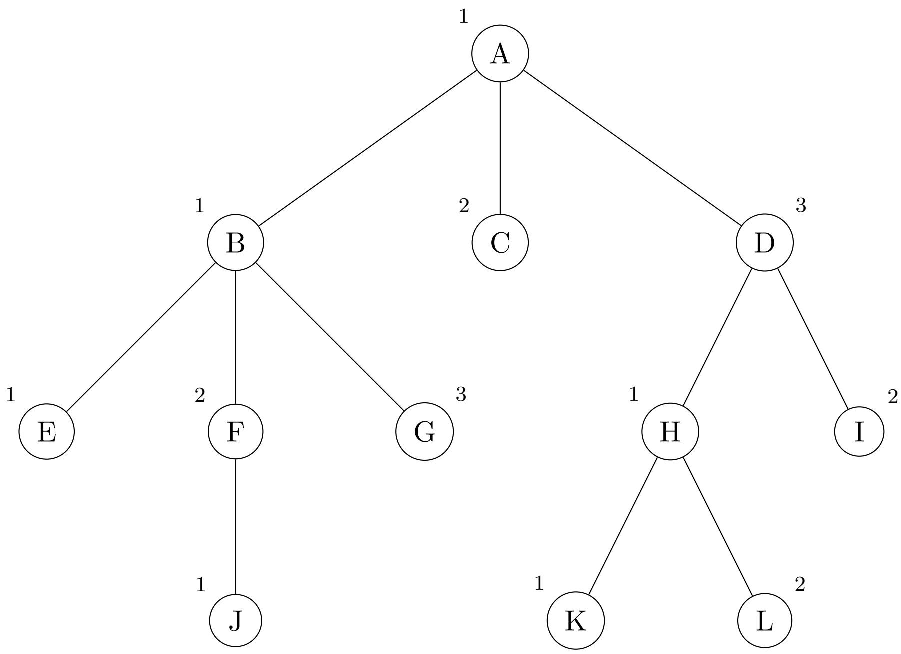
Slika 2.5: Izlistane putanje-Nizovi predaka

Predloženi model stabla se dalje može uprostiti ako svaki čvor indeksiramo vrednostima iz skupa prirodnih brojeva N. Ovo indeksiranje čvorova se može izvršiti na sledeći način:

1) Koren stabla ima indeks 1

2) Potomci indeksiranog čvora se indeksiraju vrednostima $1, 2, \dots, m$; $m \in N$

Pri tome se pogodnim indeksiranjem potomaka može postići željeno uređenje stabla, kao na slici (2.6).



Slika 2.6: Indeksiranje čvorova stabla

Sada svakom čvoru stabla $x \in T$, možemo na jednoznačan način pridružiti niz $< \text{path}_x >=< 1, i_1, i_2, \ldots, i_k >$ čiji su elementi indeksi čvorova na jedinstvenoj putanji od korena stabla do datog čvora x. Ovo pridruživanje se može izršiti na sledeći način:

1) Korenu r stabla T se pridružuje niz indeksa $< \text{path}_r >=< 1 >$ dužine 1

2) Neka je čvoru x pridružen niz indeksa $< \text{path}_x >$. Potomku čvora x se indeksom $i \in \{1, 2, \ldots, m\}$, pridružuje se niz $< \text{path}_x, i >$.

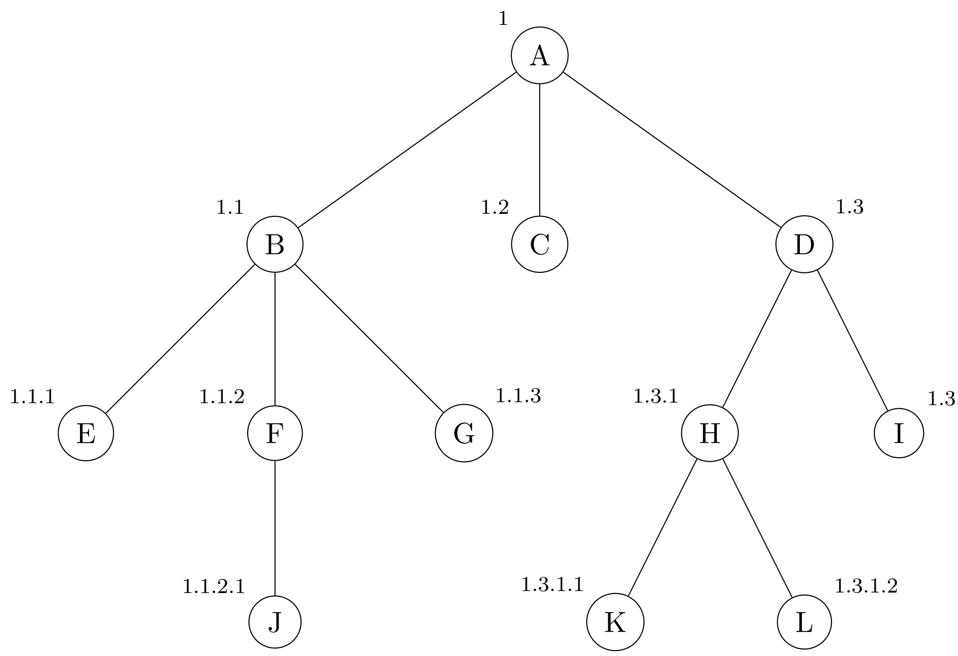
Označimo sa T^* skup nizova indeksa pridruženih čvorovima stabla T:

$$T^* = \{x^* | x \in T\} = \{< \text{path}_x > | x \in T\}$$

Neka su $x, y \in T$, očigledno važi:

$$y \leq x \Leftrightarrow y^* \leq_p x^* \tag{2.36}$$

gde je $\leq_p$ relacija prefiks na skupu nizova indeksa T^*, odnosno $(T, \leq)$ i $(T^*, \leq_p)$ su izomorfne strukture.
Ako za obeležje čvora umesto niza indeksa $< \text{path}_x >=< 1, i_1, i_2, \ldots, i_k >$, uzmemo odgovarajuću reč $\text{path}_x = 1.i_1. \ldots .i_k$ nad azbukom $A = \{0, 1, \ldots, 9, .\}$ relacija $\leq_p$ postaje relacija *prefiks* na skupu reči nad azbukom A (2.7).



Slika 2.7: Dewey-jeva decimalna notacija

Ovo je takozvana *Dewey-jeva decimalna notacija* [11]. Primetimo da se u ovom modelu za zadani čvor može neposredno zaključiti koji su njegovi preci. Na primer preci čvora sa obeležjem 1.3.1.2 su čvorovi sa obeležjima redom 1.3.1, 1.3 i 1.

2.3.5 Model sa ugnježdenim skupovima

U prethodnoj karakterizaciji stabla čvoru stabla je odgovarao skup njegovih prethodnika. Kolekcija skupova prethodnika se uređuje relacijom podskup $\subseteq$. U karakterizaciji koja sledi, čvoru stabla će odgovarati skup potomaka, a odgovarajuća kolekcija skupova potomaka se uređuje relacijom nadskup $\supseteq$. Pre svega uvedimo sledeće definicije.

Definicija 2.3.4. Neka je T konačan skup i neka je $\mathcal{C}$ kolekcija podskupova od T. $\mathcal{C}$ je $\supseteq$-hijerarhija na skupu T ako važi:

a) $T \in \mathcal{C}$

b) za sve $U, V \in \mathcal{C}$ i $U \neq V$ važi tačno jedan od uslova $U \subset V \vee V \subset U \vee U \cap V = \emptyset$.

$\diamond$

Definicija 2.3.5. $\supseteq$-hijerarhija $\mathcal{C}$ na skupu T je $\mathcal{T}_{\supseteq}$-hijerarhija na skupu T ako važi:

a) za sve $x, y \in T$, $x \neq y$, postoji $U \in \mathcal{C}$ tako da tačno jedan od x, y pripada U

b) unija kolekcije elementa hijerarhije $\mathcal{C}$ je elemenat hijerarhije $\mathcal{C}$ akko je jednaka nekom članu kolekcije

$\diamond$

Definicija 2.3.6. $\supseteq$-hijerarhija $\mathcal{C}$ na skupu T je povezana $\supseteq$-hijerarhija na skupu T ako važi:

- postoji element $r \in T$ takav da je jedini element U hijerarhije $\mathcal{C}$ takav da $r \in U$, polazni skup T.

$\diamond$

Sada možemo dati sledeću karakterizaciju konačnog stabla.

Teorema 2.3.6. *Povezana $\mathcal{T}_{\supseteq}$-hijerarhija $\mathcal{C}$ na konačnom skupu T je konačno stablo.*

Dokaz. *Neka je $\mathcal{C}$ povezana $\mathcal{T}_{\supseteq}$-hijerarhija na konačnom skupu T. Primetimo da prazan skup $\emptyset$ ne može biti elemenat $\supseteq$-hijerarhije $\mathcal{C}$, jer u protivnom $\emptyset, T \in \mathcal{C}$ i $\emptyset \neq T \ni r$ i pri tome važi $\emptyset \subset T \wedge \emptyset \cap T = \emptyset$, pa $\mathcal{C}$ ne može biti $\supseteq$-hijerarhija.*
Definišimo funkciju (relaciju) direktni predak $f : \mathcal{C} \to \mathcal{C}$ na sledeći način:

$$f(U) = \begin{cases} T & U = T \\ \bigcap \mathcal{C}_U & U \neq T \end{cases} \tag{2.37}$$

gde je $\mathcal{C}_U = \{V \in \mathcal{C} | U \subsetneq V\}$.
Neka je $U \neq T$ elemenat povezane $\mathcal{T}_{\supseteq}$-hijerarhije $\mathcal{C}$ na skupu T, pokazaćemo da mora biti $U \subsetneq f(U)$. Kolekcija $\mathcal{C}_U$ je neprazna jer $T \in \mathcal{C}_U$, dalje kako svi elementi kolekcije $\mathcal{C}_U$ imaju neprazan presek i pri tome je $\mathcal{C}$ konačna $\supseteq$-hijerarhija, postoji $V_0 \in \mathcal{C}_U$ takav da za svako $V \in \mathcal{C}_U$ važi $V_0 \subseteq V$, odnosno postoji $V_0 \in \mathcal{C}_U$ takvo da važi $V_0 = \bigcap \mathcal{C}_U = f(U)$. Kako

$V_0 \in \mathcal{C}_U$ zaključujemo $U \subsetneq V_0 = f(U)$. Još više, neka je $U \neq T$ elemenat povezane $\mathcal{T}_{\supseteq}$-hijerarhije $\mathcal{C}$ na skupu T tada važi $U \subsetneq f^k(U)$ za sve $k \in N$. Iz prethodnog možemo zaključiti da funkcija f ima tačno jednu fiksnu tačku T i pri tome f^k $k = 2, 3, \ldots$ nema drugih fiksnih tačaka sem tačke T, pa je povezana $\mathcal{T}_{\supseteq}$-hijerarhija na konačnom skupu T stablo.

$\diamond$

Funkcija f iz prethodne teoreme, na povezanoj $\mathcal{T}_{\supseteq}$-hijerarhiji $\mathcal{C}$ na konačnom skupu T, ima sledeću osobinu. Neka su $U, V \in \mathcal{C}$, tada važi: i $k \in N_0$

$$V = f^k(U) \Leftrightarrow V \supseteq U \tag{2.38}$$

odnosno relacija nadskup $\supseteq$ na $\mathcal{C}$ je refleksivno i tranzitivno zatvorenje ovako definisane funkcije (relacije) direktni predak f.

$$\supseteq = \bigcup_{k \in N_0} f^k \tag{2.39}$$

gde je:

$$\supseteq = \{(V, U) | (U, V \in \mathcal{C}) V \supseteq U\} \tag{2.40}$$

$$f^k = \left\{(V, U) | (V, U \in \mathcal{C}) V = f^k(U)\right\} \tag{2.41}$$

Drugim rečima povezana $\mathcal{T}_{\supseteq}$-hijerarhija na konačnom skupu T je stablo ako se uredi relacijom nadskup.
Neka je $(T, \leq)$ konačno stablo, pokažimo da se može konstruisati povezana $\mathcal{T}_{\supseteq}$-hijerarhija $\mathcal{C}$ na skupu T. Pre svega primetimo da u ovoj karakterizaciji svakom čvoru $x \in T$ odgovara skup njegovih potomaka, odnosno skup čvorova kojima je on predak. U tom smislu definišimo preslikavanje $\downarrow : T \to \mathbb{P}(A)$ na sledeći način:

$$\downarrow(x) = \{z \in T | x \leq z\} \tag{2.42}$$

gde je $\mathbb{P}(A)$ partitivni skup skupa čvorova T. Sada nije teško pokazati sledeći stav.

Teorema 2.3.7. *Neka je $(T, \leq)$ konačno stablo. Kolekcija $\mathcal{C} = \{\downarrow(x) | x \in T\} = \downarrow T$, je povezana $\mathcal{T}_{\supseteq}$-hijerarhija na skupu T.*

Dokaz. *Primetimo $x \in \downarrow(x)$ za sve $x \in T$. Pokažimo prvo da je $\mathcal{C}$ $\supseteq$-hijerarhija na T. Neka je r koren stabla T, očigledno $\downarrow(r) = T \in \mathcal{C}$. Neka su $U, V \in \mathcal{C}$ i $U \neq V$, tada je $U = \downarrow(x), V = \downarrow(y)$, za neke $x, y \in T$. Mora biti $x \neq y$, jer bi u suprotnom važilo $\downarrow(x) = \downarrow(y)$ odnosno $U = V$ što je suprotno pretpostavci $U \neq V$. Neka je $U = \downarrow(x) \cap V = \downarrow(y) \neq \emptyset$, odnosno neka postoji $z \in T$ tako da $z \in U = \downarrow(x) \wedge z \in V = \downarrow(y)$, odnosno čvorovi x, y su preci čvora z, dakle $x, y \in \uparrow(z)$. $(T, \leq)$ je stablo pa je skup predaka $\uparrow(z)$ dobro uređen pa važi tačno jedan od uslova:*

- *$y \lneq x$ odnosno $V = \downarrow(y) \supsetneq U = \downarrow(x)$*

– $x \lneq y$ *odnosno* $U =\downarrow(x) \supsetneq V =\downarrow(y)$

odnosno važi tačno jedan od uslova $U \cap V = \emptyset \vee U \subset V \vee V \subset U$. *Dakle* $\mathcal{C} = \{\uparrow(x)|x \in T\}$ *je* $\supseteq$*-hijerarhija. Očigledno da je to povezana* $\supseteq$*-hijerarhija jer koren* r *stabla* T *je potomak jedino sebe samog, odnosno važi* $r \in\downarrow(x) \Leftrightarrow r \in\downarrow(r) = T$.
Dokažimo još da je $\mathcal{C}$ $\mathcal{T}_{\supseteq}$*-hijerarhija. Neka su* $x, y \in T$ *i* $x \neq y$*, jasno* $x \in\downarrow(x)$*, odnosno* $y \in\downarrow(y)$. *Neka je* $y \in\downarrow(x)$*, u protivnom ako* $y \notin\downarrow(x)$ *skup* $U =\downarrow(x)$ *razdvaja čvorove* x, y. *U tom slučaju* $x \notin\downarrow(y)$ *jer bi u suprotnom bilo* $x = y$*, pa skup* $U =\downarrow(y)$ *razdvaja ćvorove* x, y. *Dakle svaka dva različita čvora stabla* T *se mogu razdvojiti nekim skupom* U *iz kolekcije* $\mathcal{C}$. *Dokažimo još da je unija kolekcije elemenata* $\supseteq$*-hijerarhije* $\mathcal{C}$ *elemenat hijerarhije* $\mathcal{C}$ *samo ako je jednaka nekom elementu kolekcije. Neka je* $\{C_1, C_2, \ldots, C_k\}$ *kolekcija elemenata* $\supseteq$*-hijerarhije* $\mathcal{C}$ *i* $k \in N$*, tada je* $C_i =\downarrow(x_i)$ *za neke* $x_i \in T$ $i = 1, 2, \ldots, k$. *Pretpostavimo* $\bigcup C_i = \bigcup \downarrow(x_i) \in \mathcal{C}$*, odnosno neka postoji čvor* $x \in T$ *takav da je* $\bigcup C_i = \bigcup \downarrow(x_i) =\downarrow(x) \in \mathcal{C}$. *Dokažimo da je* $\downarrow(x) =\downarrow(x_i)$ *za neko* $i \in \{1, 2, \ldots, k\}$. *Iz* $x_i \in\downarrow(x_i)$ *sledi* $x_i \in \bigcup \downarrow(x_i) =\downarrow(x)$*, odnosno* $x \leq x_i$ *za sve* $i = 1, 2, \ldots, k$. *Sa druge strane je* $x \in\downarrow(x) = \bigcup \downarrow(x_i)$ *odnosno* $x \in\downarrow(x_i)$ *za neko* $i \in \{1, 2, \ldots, k\}$*, pa je* $x_i \leq x$ *za neko* $i \in \{1, 2, \ldots, k\}$*, kako je pri tome* $x \leq x_i$ *zaključujemo da je* $x = x_i$ *odnosno* $\downarrow(x) =\downarrow(x_i)$ *za neko* $i \in \{1, 2, \ldots, k\}$ *što je i trebalo dokazati.*

$\diamond$

Važi i obrat, ako je $\mathcal{C}$ povezana $\mathcal{T}_{\supseteq}$-hijerarhija na konačnom skupu T tada se skup T može urediti nekom relacijom $\leq$ tako da je $(T, \leq)$ stablo i svaki element kolekcije $\mathcal{C}$ je skup potomaka nekog čvora stabla T. Pre svega pokažimo da povezana $\mathcal{T}_{\supseteq}$-hijerarhija na konačnom skupu T ima isti broj elemenata kao i konačan skup T.

Teorema 2.3.8. *Neka je* $\mathcal{C}$ *povezana* $\mathcal{T}_{\supseteq}$*-hijerarhija na konačnom skupu* T*, tada postoji bijekcija* $\downarrow: T \to \mathcal{C}$

Dokaz. *Problem je odrediti čvor* $x \in T$ *koji odgovara elementu* $C \in \mathcal{C}$. *Kako je* $\mathcal{C}$ *povezana* $\supseteq$*-hijerarhija, postoji istaknuti element* $r \in T$*, koji ćemo pridružiti elementu* $T \in \mathcal{C}$. *Neka je* $x \in T$ *i* $x \neq r$ *proizvoljan čvor. Kako je* $\mathcal{C}$ $\mathcal{T}_{\supseteq}$*-hijerarhija postoji* $U \in \mathcal{C}$ *tako da tačno jedan od* x *i* r *pripada* U*, očigledno je da je jedina mogućnost da* $x \in U$. *Dakle za svaki čvor* $x \in T$ *postoji* $U \in \mathcal{C}$ *tako da važi* $x \in U$*, odnosno svaki čvor je elemenat bar jednog podskupa skupa čvorova* T *iz kolekcije* $\mathcal{C}$. *Neka je* $x \in T$ *označimo sa* $\mathcal{C}_x$ *kolekciju elemenata povezane* $\mathcal{T}_{\supseteq}$*-hijerarhije* $\mathcal{C}$ *koji sadrže čvor* x*:*

$$\mathcal{C}_x = \{U \in \mathcal{C}|x \in U\} \tag{2.43}$$

Na osnovu prethodnog razmatranja $\mathcal{C}_x$ *je neprazna kolekcija za sve* $x \in T$. *Sada možemo formirati skupove:*

$$\cap_x = \bigcap \{U \in \mathcal{C}|x \in U\} = \bigcap \mathcal{C}_x \tag{2.44}$$

Očigledno $x \in \cap_x$. *Kako skupovi iz kolekcije* $\mathcal{C}_x = \{U \in \mathcal{C}|x \in U\}$ *imaju neprazan presek i pri tome je* $\mathcal{C}$ *konačna* $\supseteq$*-hijerarhija postoji* $U_0 \in \mathcal{C}_x$ *tako da je* $\cap_x = \bigcap \mathcal{C}_x = U_0 \in \mathcal{C}$. *Dakle* *za sve* $x \in T$ $\cap_x$ *je elemenat kolekcije* $\mathcal{C}$*, odnosno* $\{\cap_x | x \in T\} \subseteq \mathcal{C}$*. Posebno* $\cap_x$ *je najmanji elemenat kolekcije* $\mathcal{C}$ *koji sadrži* x*, odnosno za* $x \in T$ *i* $U \in \mathcal{C}$ *važi:*

$$x \in U \Rightarrow \cap_x \subseteq U \tag{2.45}$$

Pokažimo da je ovim povezana $\mathcal{T}_{\supseteq}$*-hijerarhija* $\mathcal{C}$ *iscrpljena, odnosno* $\mathcal{C} = \{\cap_x | x \in T\}$*. Pretpostavimo suprotno, neka postoji* $C \in \mathcal{C}$ *i za svako* $x \in T$*,* $C \neq \cap_x$*. Neka je* $C = \{x_1, x_2, \dots, x_k\} \subseteq T$ *i* $k \in N$*. Kako je* $x_i \in C$ *mora biti* $\cap_{x_i} \subseteq C$ *za sve* $i = 1, 2, \dots, k$ *odnosno* $\bigcup \cap_{x_i} \subseteq C$*. Sa druge strane za svako* $x_i \in C$ *važi* $x_i \in \cap_{x_i}$ *pa važi* $C \subseteq \bigcup \cap_{x_i}$*, dakle* $\bigcup \cap_{x_i} = C$*. Kako je* $\mathcal{C}$ $\mathcal{T}_{\supseteq}$*-hijerarhija i pri tome je* $\bigcup \cap_{x_i} = C \in \mathcal{C}$ *mora biti* $\bigcup \cap_{x_i} = C = \cap_{x_i}$ *za neko* $i \in \{1, 2, \dots, k\}$ *što je suprotno pretpostavci, dakle* $\mathcal{C} = \{\cap_x | x \in T\}$*.*
Pokažimo još da u $\mathcal{C} = \{C_x | x \in T\}$ *nema istih, odnosno da različitim čvorovima* $x, y \in T$ *odgovaraju različiti* $\cap_x, \cap_y \in \mathcal{C}$*. Neka su* $x, y \in T$ *i neka je* $x \neq y$*, pretpostavimo* $\cap_x = \cap_y \in \mathcal{C}$*. Kako* $x, y \in \cap_x = \cap_y$ *i* $\mathcal{C}$ *je* $\mathcal{T}_{\supseteq}$*-hijerarhija, postoji* $C \in \mathcal{C}$ *tako da tačno jedan od* x *i* y *pripada* C*. Neka* $x \in C$ *i* $y \notin C$*. U ovom slučaju iz* $x \in C$ *sledi* $\cap_x \subseteq C$ *pa i* $y \in C$ *što je suprotno pretpostavci, dakle* $\cap_x \neq \cap_y$*. Dakle preslikavanje* $\downarrow : T \to \mathcal{C}$ *definisano sa* $\downarrow(x) = \cap_x$ *je dobro definisano i pri tome je bijekcija.*

◇

Teorema 2.3.9. *Neka je* $\mathcal{C}$ *povezana* $\mathcal{T}_{\supseteq}$*-hijerarhija na konačnom skupu* T*, tada postoji uređenje* $\leq$ *skupa* T *tako da je* $(T, \leq)$ *stablo i svaki element kolekcije* $\mathcal{C}$ *je skup potomaka nekog čvora stabla* T *u odnosu na tako uvedeno uređenje* $\leq$*.*

Dokaz. *Preslikavanje* $\downarrow : T \to \mathcal{C}$ *definisano sa:*

$$\downarrow(x) = \cap_x = \bigcap \{U \in \mathcal{C} | x \in U\} = \bigcap \mathcal{C}_x \tag{2.46}$$

je bijekcija na osnovu prethodnog stava. Uredimo skup T *na sledeći način:*

$$y \leq x \Leftrightarrow \cap_y \supseteq \cap_x \tag{2.47}$$

Na ovaj način $\downarrow$ *je izomorfizam struktura* $(T, \leq)$ *i* $(\mathcal{C}, \supseteq)$*, pa je struktura* $(T, \leq)$ *stablo, jer je to slučaj sa strukturom* $(\mathcal{C}, \supseteq)$*. Dokažimo sada da je pri ovakvom uređenju čvorova stabla* T*, svaki elemenat povezane* $\mathcal{T}_{\supseteq}$*-hijerarhije* $\mathcal{C}$ *skup potomaka nekog čvora* x *stabla* T*. Dovoljno je dokazati da za* $x \in T$ *važi* $\{y \in T | x \leq y\} = \cap_x$*.*
Neka je $y \in \{y \in T | x \leq y\}$*, odnosno neka je* $x \leq y$*. S obzirom kako je uvedeno uređenje* $\leq$ *biće* $\cap_x \supseteq \cap_y$*. Kako je* $y \in \cap_y$ *mora biti* $y \in \cap_x$*. Dakle* $\downarrow(x) \subseteq \cap_x$*. Dokažimo i obrat* $\cap_x \subseteq \{y \in T | x \leq y\}$*. Pretpostavimo da je* $y \in \cap_x$*, tada je* $\cap_x \in \mathcal{C}_y$ *pa mora biti* $\cap_y \subseteq \cap_x$ *odnosno s obzirom na definiciju relacije* $\leq$ *važi* $x \leq y$ *dakle* $y \in \{y \in T | x \leq y\}$*. Dakle* $\downarrow(x) = \cap_x = \{y \in T | x \leq y\}$*, što je i trebalo dokazati.*

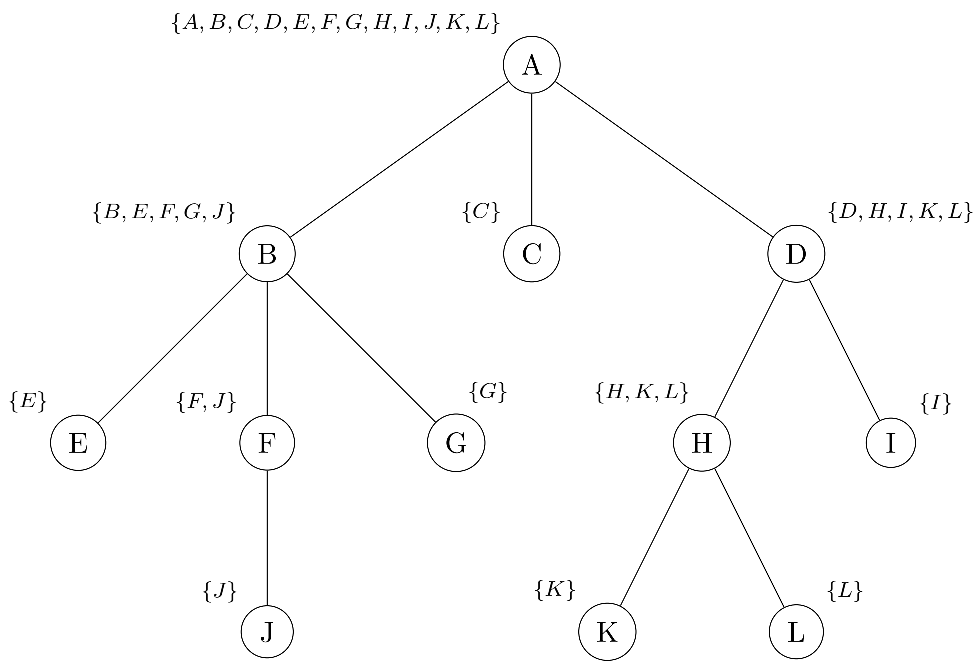
◇

Iz navedenog zaključujemo da nema razlike između konačnog stabla $(T, \leq)$ i povezane $\mathcal{T}_{\supseteq}$-hijerarhije $\mathcal{C}$ na skupu T, odnosno to su izomorfne strukture. Dakle umesto stabla $(T, \leq)$

možemo posmatrati povezanu $\mathcal{T}_{\supseteq}$-hijerarhiju $\mathcal{C}$ na skupu T, pri tome se odnos između čvorova stabla $(T, \leq)$ preslikava na odnos između elemenata kolekcije $\mathcal{C}$ na sledeći način:

$$y \leq x \Leftrightarrow \downarrow (y) \supseteq \downarrow (x) \tag{2.48}$$

gde su $x, y \in T$, $\downarrow (x), \downarrow (y) \in \mathcal{C}$ odgovarajući elementi struktura $(T, \leq)$ i $(\mathcal{C}, \supseteq)$. U ovom modelu čvor $x \in T$ se reprezentuje skupom čvorova $\downarrow (x)$ kojima je on predak, odnosno čvorovima podstabla kojima je on koren. Ovo je takozvani *model sa ugnježdenim skupovima* (*nested sets model*) (2.8).



Slika 2.8: Ugnježdeni skupovi-Skupovi potomaka

Dakle za obeležje čvora $x \in T$ uzimamo skup njegovih potomaka $\downarrow (x)$. Cilj je opisati skup $\downarrow (x)$ u kompaktnoj formi. Za razliku od modela sa izlistanim putanjama, obeležje čvora u ovom slučaju skup potomaka, nije u opštem slučaju dobro uređen. Kompaktnu formu skupa potomaka možemo obezbediti ako na pogodan način indeksiramo čvorove stabla, a to se može postići ako su čvorovi svakog podstabla indeksirani uzastopnim prirodnim brojevima. U tom slučaju je obeležje čvora skup indeksa njegovih potomaka:

$$\downarrow (x) = \{i, i+1, \ldots, i+k\} \tag{2.49}$$

Kako se radi o uzastopnim indeksima, skup $\downarrow (x)$ će biti interval prirodnih brojeva $[i, i+k]$, pri čemu je i indeks samog čvora x, $i+k$ je indeks njegovog poslednjeg indeksiranog potomaka, a k je broj njegovih potomaka. Dakle obeležje čvora $x \in T$ je uređen par $(\text{left}_x, \text{right}_x)$ koji čine leva i, odnosno desna $i+k$ granica intervala prirodnih brojeva. Ovakvom reprezentacijom stabla neposredno se proverava da li je dati čvor x predak čvora y na sledeći način:

$$
\begin{aligned}
y \leq x &\Leftrightarrow \downarrow(y) \supseteq \downarrow(x) \\
&\Leftrightarrow [\mathrm{left}_y, \mathrm{right}_y] \supseteq [\mathrm{left}_x, \mathrm{right}_x] \\
&\Leftrightarrow \mathrm{left}_y \leq \mathrm{left}_x \wedge \mathrm{right}_x \leq \mathrm{right}_y
\end{aligned}
$$

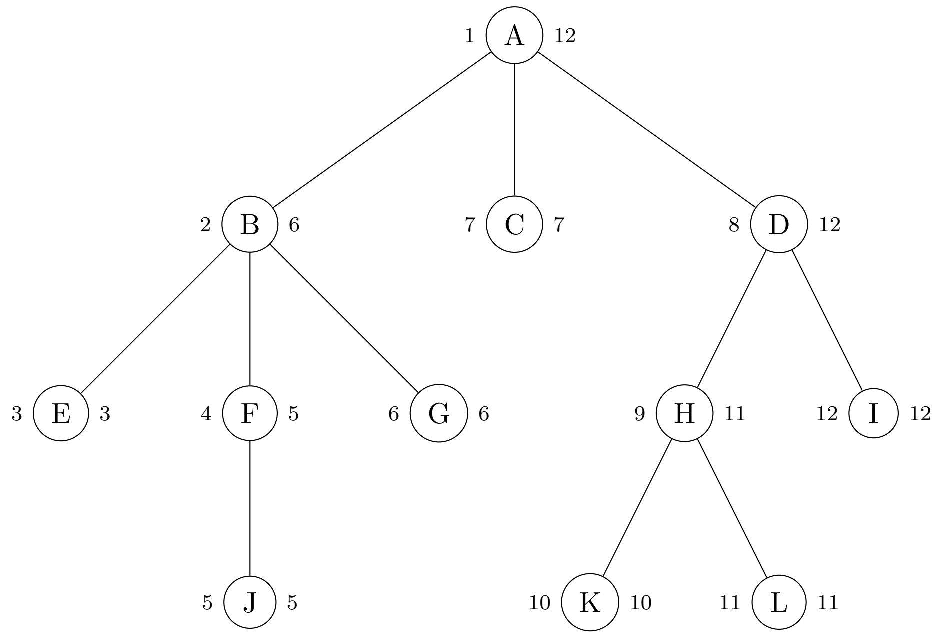
iskoristimo li osobinu $\supseteq$-hijerarhije na T, imaćemo:

$$y \leq x \Leftrightarrow \mathrm{left}_y \leq \mathrm{left}_x \leq \mathrm{right}_y \tag{2.50}$$

ili simetrično:

$$y \leq x \Leftrightarrow \mathrm{left}_y \leq \mathrm{right}_x \leq \mathrm{right}_y \tag{2.51}$$

Postavlja se pitanje kako rešiti problem indeksiranja čvorova stabla koje bi imalo navedenu osobinu. Primetimo sledeće, prvo se indeksira čvor, a zatim njegovi potomci. Ako zanemarimo indeksiranje, onda se problem rešava tako što prvo obilazimo čvor, a zatim njegove potomke [5]. Ovo je tačno pre-order obilazak stabla (2.2). Dakle čvorovima stabla se mogu pridružiti obeležja, odnosno leva i desna granica tako što se stablo obilazi na pre-order nacin. Prvo se obilazi čvor pa njegova podstabla i to od krajnjeg levog do krajnjeg desnog podstabla. Vrednosti za granice (indeksi čvorova) su uzastopni prirodni brojevi $1, 2, \ldots$. Leva granica (indeks samog čvora) se formira u prvom prolazu kroz čvor i njena vrednost je sledeća vrednost indeksa. Desna granica (indeks poslednjeg indeksiranog potomka) se formira pri povratku u čvor i njena vrednost je tekuća vrednost indeksa (poslednji iskorišćeni indeks) (2.9).



Slika 2.9: Skup potomaka

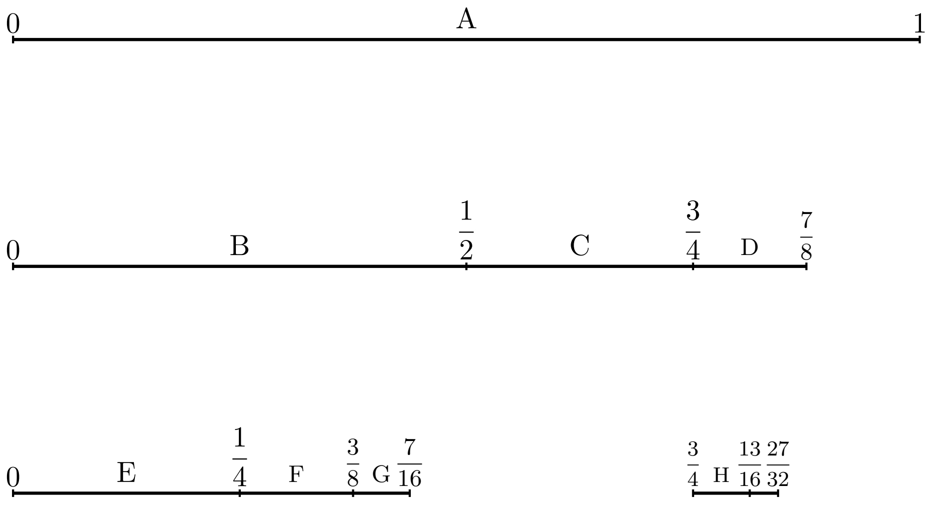
Za razliku od modela sa izlistanim putanjama, u ovom modelu se za zadati čvor ne može neposredno zaključiti koji su njegovi preci, ali se sa druge strane može neposredno odrediti broj potomaka datog čvora.

2.3.6 Model sa ugnježdenim intervalima

U modelu sa ugnježdenim skupovima svakom čvoru se pridružuje obeležje koje je skup njegovih potomaka. Pogodnom numeracijom čvorova stabla, to može biti interval prirodnih brojeva. Osnovni problem je nemogućnost efikasnog umetanja novih čvorova jer interval prirodnih brojeva nije dovoljno gust, pa se svakako mora vršiti ponovna numeracija čvorova da bi se napravio prostor za novi čvor. Ako umesto intervala prirodnih brojeva uzmemo interval racionalnih (realnih) brojeva problem umetanja čvorova će biti rešen. U zavisnosti od toga koji se podinterval intervala pridruženog nadređenom čvoru dodeljuje podređenom čvoru dobićemo različite tipove *modela sa ugnježdenim intervalima*.

Binarni model sa ugnježdenim intervalima

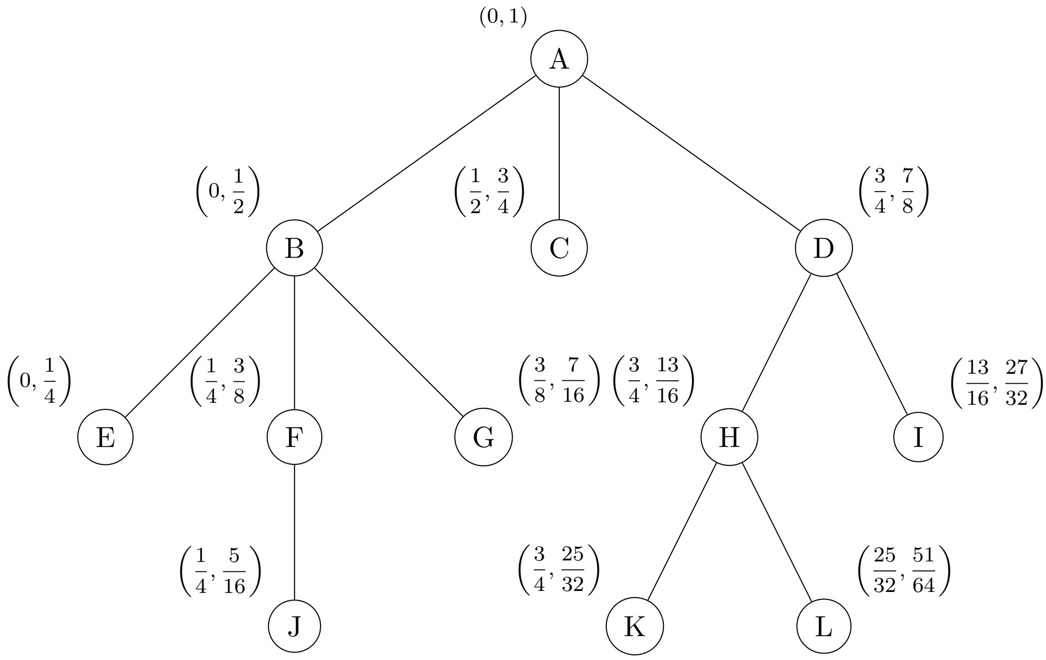
Jedan tip modela sa ugnježdenim intervalima može biti sledeći [7]. Pridružimo korenu stabla neki interval, bez gubitka opštosti to može biti interval $(0, 1)$ i to je na početku slobodni interval korena. Dalje se svakom potomku pridružuje leva polovina slobodnog intervala. Na primer, prvom potomku korena se pridružuje interval $\left(0, \frac{1}{2}\right)$ i nakon ovog pridruživanja slobodan interval korena je interval $\left(\frac{1}{2}, 1\right)$. Sledećem potomku se na isti način pridružuje leva polovina novog slobodnog intervala $\left(\frac{1}{2}, 1\right)$, a to je interval $\left(\frac{1}{2}, \frac{1}{2} + \frac{1}{2^2}\right) = \left(\frac{1}{2}, \frac{3}{4}\right)$ (2.10):



Slika 2.10: Selekcija intervala

Uopšte neka je čvoru stabla pridružen interal (l, r), to je na početku i slobodan interval tog čvora. Potomku čvora se dodeljuje leva polovina njegovog slobodnog intervala, odnosno interval $(l, \frac{l+r}{2})$. Novi slobodni interval uočenog čvora je desna polovina polaznog intervala (l, r), odnosno interval $(\frac{l+r}{2}, r)$. Ovo je takozvani *binarni model sa ugnježdenim intervalima*

(2.11):

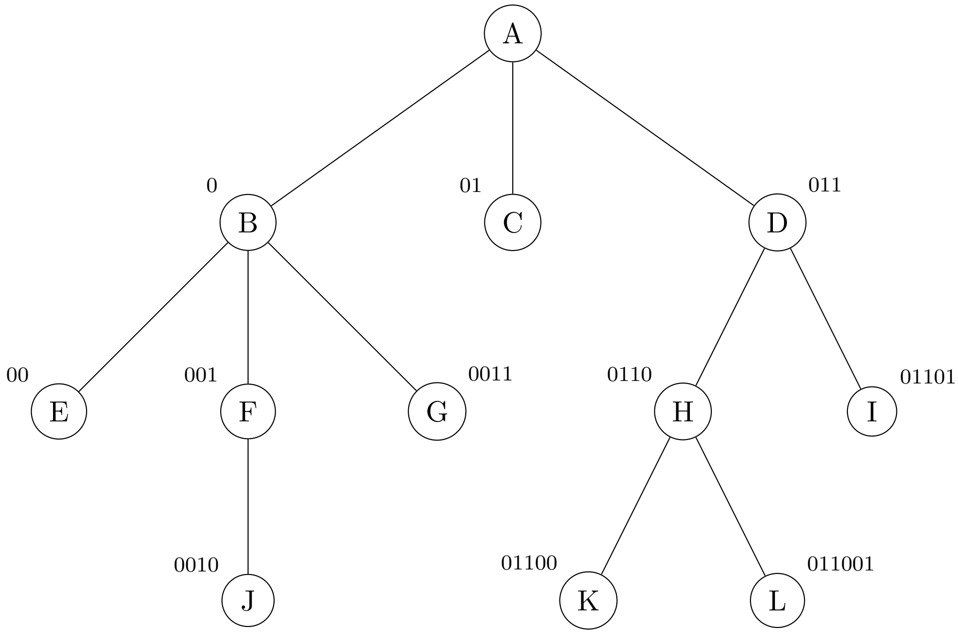


Slika 2.11: Binarni model

Problem kod ovog modela je što obeležja (imenioci granica intervala) eksponencijalno rastu. Ovo se može rešiti na sledeći nacin. Primetimo da je dužina intervala koji se pridružuje čvoru drveta jednaka $\frac{1}{2^k}$ $k \in N$, ako iskoristimo binarni zapis leve odnosno desne granice intervala, intervali koji se pridružuju čvorovima drveta (sem korena kome je pridružen interval $(0,1)$ su sledećeg oblika:

$$(0.a_1a_2\dots a_{k-1}, 0.a_1a_2\dots a_{k-1}1), a_i \in \{0,1\} \tag{2.52}$$

Očigledno je interval pridružen čvoru u potpunosti određen svojom desnom granicom. Dakle dovoljno je za obeležje čvora uzeti desnu granicu pridruženog intervala. Obeležje korena će biti 1, a ostalih čvorova zapis oblika $0.a_1a_2\dots a_{k-1}1$. Zapis desne granice pridruženog intervala dalje mozemo shvatiti kao reč nad binarnom azbukom $A_2 = \{0,1\}$ koja se završava slovom 1. Očigledno možemo izbaciti završno slovo 1 i dobićemo skup obeležja koji čine prazna reč (obeležje korena) i reči binarne azbuke koje počinju slovom 0 (obeležja čvorova koji nisu koren), ovo su praktično zapisi levih granica pridruženih intervala (2.12).



Slika 2.12: Binarni model sa levom granicom

Primetimo da obeležje čvora pri sledećoj interpretaciji:

- · 0 - spusti se na nivo ispod krajnjom levom granom
- · 1 - pređi na sledeći čvor na istom nivou.

predstavlja putanju od korena do datog čvora. Na primer, čvoru L je pridružen interval $\left(\frac{25}{32}, \frac{51}{64}\right)$, zapis desne granice je sledeća reč 0110011 nad binarnom azbukom A_2:

$$\frac{51}{64} = \frac{0}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{0}{2^4} + \frac{1}{2^5} + \frac{1}{2^6}$$

Ako izbacimo završno slovo 1 dobićemo obeležje (binarni zapis leve granice) 011001 čvora L koje se dalje interpretira na sledeći način:

- · 0 - spusti se na nivo ispod krajnjom levom granom (čvor B)
- · 1 - pređi na sledeći čvor na istom nivou (čvor C)
- · 1 - pređi na sledeći čvor na istom nivou (čvor D)
- · 0 - spusti se na nivo ispod krajnjom levom granom (čvor H)
- · 0 - spusti se na nivo ispod krajnjom levom granom (čvor K)
- · 1 - pređi na sledeći čvor na istom nivou (čvor L)

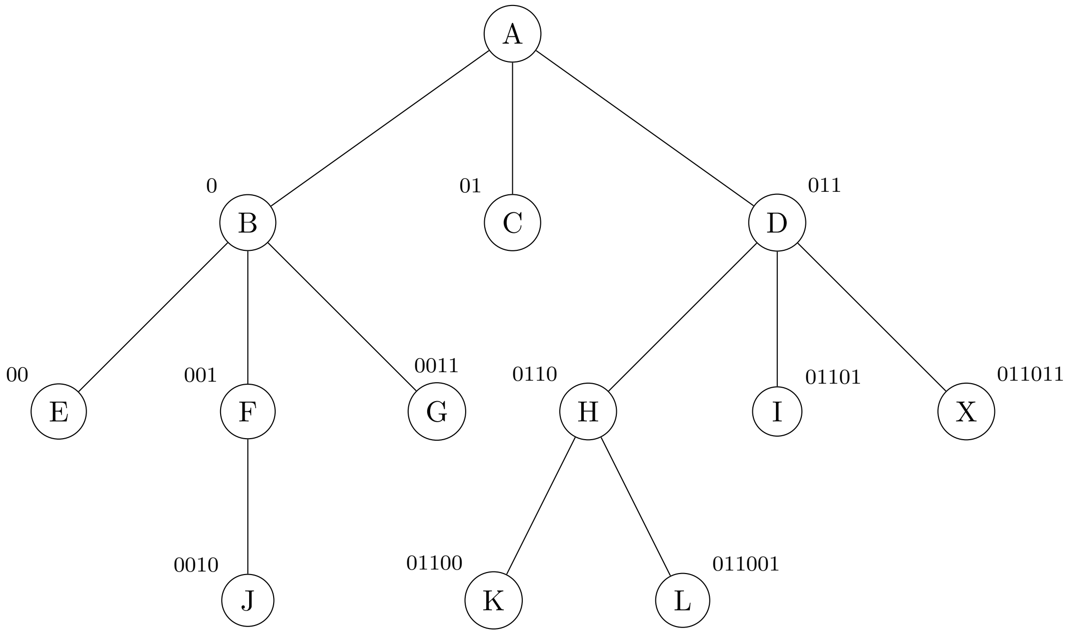
Primetimo sledeće, dužina poodreči koja počinje slovom 0 (prelazak na niži nivo) do sledećeg slova 0 predstavlja indeks čvora na tekućem nivou u modelu sa izlistanim putanjama. Dakle, ako izvršimo opisanu transformaciju obeležja čvora, dobićemo:

- · 011 - treći čvor na prvom nivou (1.3)

· 0 - prvi čvor na drugom nivou (1.3.1)

· 01 - drugi čvor na trećem nivou (1.3.1.2)

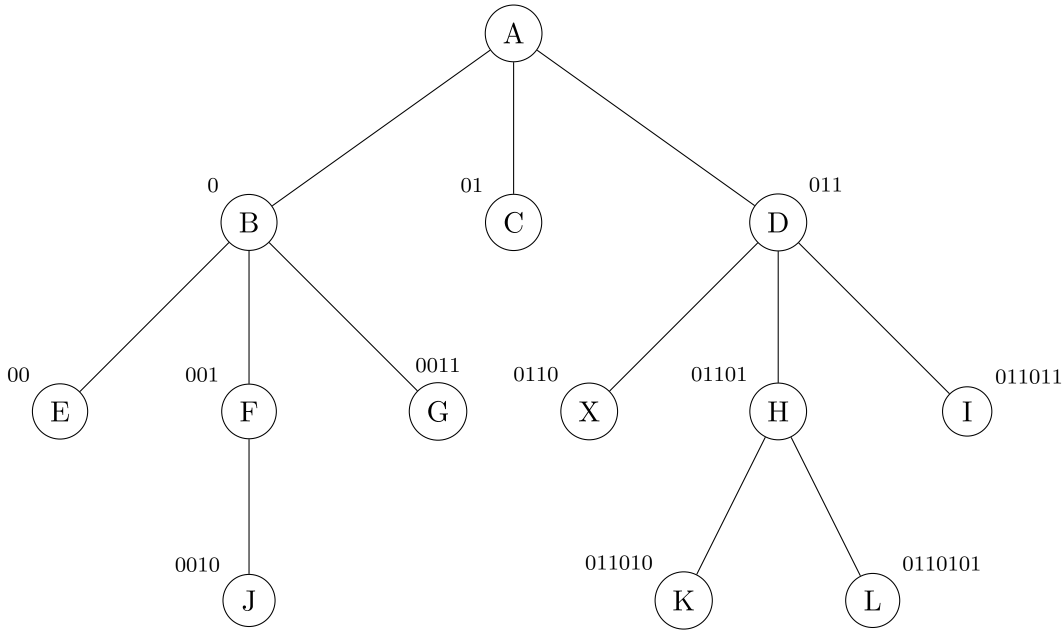
što je obeležje čvora u modelu sa izlistanim putanjama, odnosno Dewey-jeva decimalna notacija. Slično kao u modelu sa izlistanim putanjama i u ovom modelu je problem što se ne može efikasno umetati čvor, odnosno novi čvor se na isti način kao i u modelu sa izlistanim putanjama dodaje uvek na kraj postojećeg niza potomaka na datom nivou. Na primer, novi potomak X čvora D ide na kraj postojeće liste potomaka H, I (2.13)



Slika 2.13: Binarni model insert - $D \leq X$

Problem umetanja čvorova je posledica toga što susedni potomci imaju zajedničku granicu tako da se novi čvor ne može efikasno umetati, a da se pri tome postigne željeno uređenje potomaka. Ovo se između ostalog može rešiti na neki od sledeća dva načina:

1) preoznačavanjem odgoarajućih čvorova stabla. Na primer, dodajmo čvor X koji će biti potomak čvora D, tako da se postigne sledeće uređenje $X < H < I$ potomaka čvora D. Čvorovi u podstablima sa korenom u H i I će biti preoznačeni, odnosno dobiće nova obeležja (2.14)



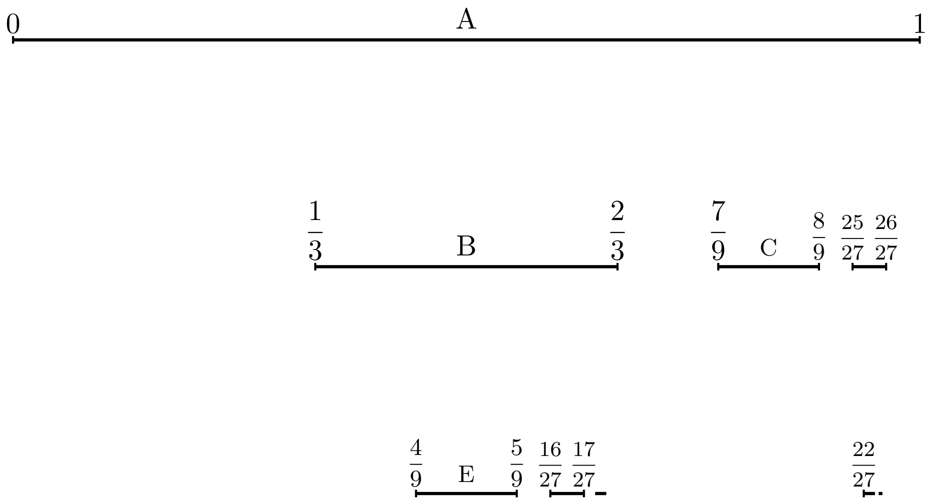
Slika 2.14: Binarni model insert - $X < H$

2) samo pridruživanje podintervala se vrši tako da uvek ostaje slobodan levi odnosno desni podinterval, pa je uvek moguće umetati čvor levo ili desno od postojećeg, odnosno moguće je postići željeno uređenje potomaka. Ovo se može postići ako se sakom novom čvoru dodeljuje neki srednji deo slobodnog intervala nadređenog čvora.

Očigledno je prvo rešenje neefikasno jer se u proseku zahteva ažuriranje obeležja na polovini čvorova stabla. Razmotrićemo dakle drugo rešenje problema umetanja čvorova.

Ternarni model sa ugnježdenim intervalima

Ovaj tip modela sa ugnježdenim intervalima možemo definisati na sledeći nacin. Slično kao što je to urađeno u prethodnom modelu, korenu stabla se pridružuje interval $(0, 1)$ i to je na početku jedini slobodni interval korena. Dalje se svakom potomku datog čvora pridružuje srednja trećina nekog njegovog slobodnog intervala. Na primer, prvom potomku korena pridružujemo interval $\left(\frac{2 \cdot 0 + 1}{3}, \frac{0 + 2 \cdot 1}{3}\right) = \left(\frac{1}{3}, \frac{2}{3}\right)$, Drugi potomak korena možemo smestiti levo od prvog potomka i tada biramo podinterval levog slobodnog intervala $\left(0, \frac{1}{3}\right)$ ili ga možemo smestiti desno kada biramo podinterval desnog slobodnog intervala $\left(\frac{2}{3}, 1\right)$ (2.15).



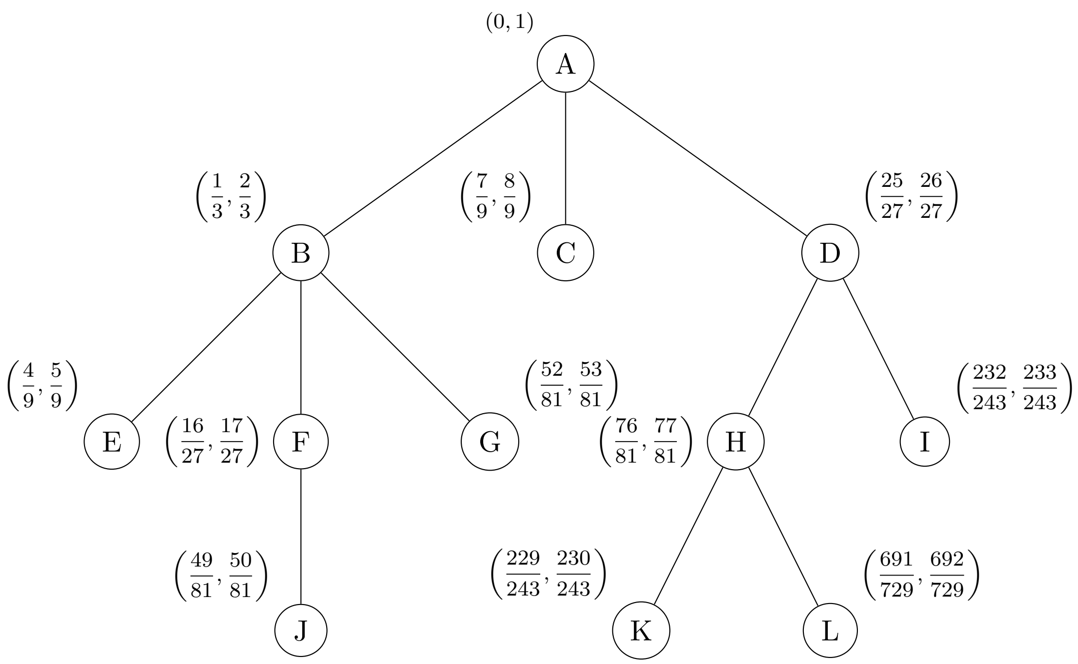
Slika 2.15: Selekcija intervala

Uopšte neka je čvoru stabla pridružen interal (l, r), to je na početku i jedini slobodni interval tog čvora. Potomku čvora se pridružuje podinterval nekog njegovog slobodnog intervala i to na sledeći način. Bira se slobodni interval datog čvora, neka je to interval (l, r). Potomku datog čvora se dodeljuje interval $((2l + r)/3, (l + 2r)/3)$, odnosno srednja trećina izabranog slobodnog intervala. Na ovaj način ostaju dva slobodna intervala:

– *levi podinterval:* $(l, (2l + r)/3)$

– *desni podinterval:* $((l + 2r)/3, r)$

za nove potomke datog čvora. Očigledno se umetanje čvorova može izvesti tako da se postigne željeno uređenje potomaka, nezavisno od redosleda smeštanja čvorova. Ovo je takozvani *ternarni model sa ugnježdenim intervalima* (2.16).



Slika 2.16: Ternarni model

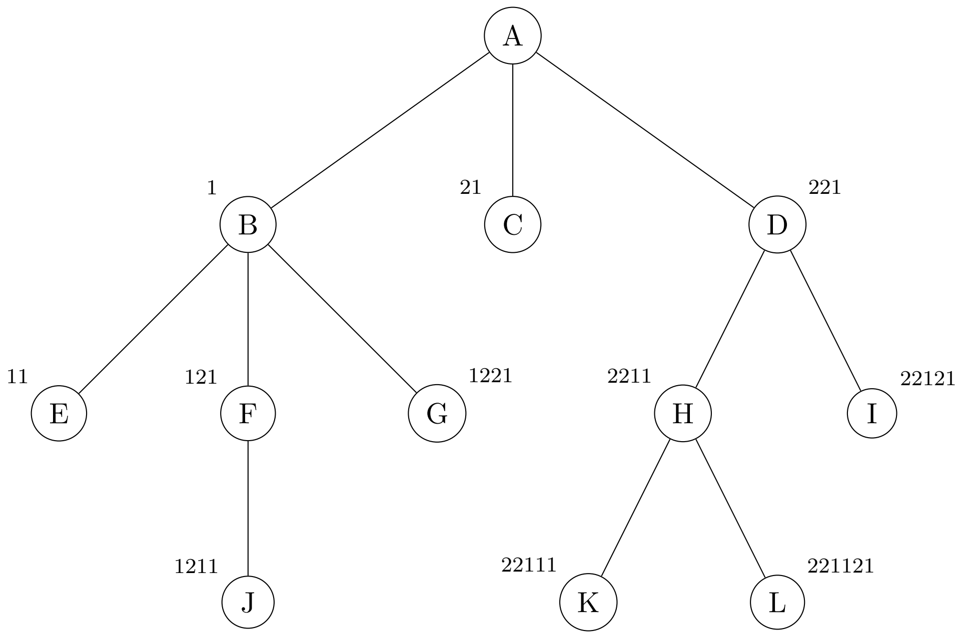
Slično kao i u prethodnom modelu, problem je što obeležja (imenioci granica intervala) eksponencijalno rastu. Po ugledu na rešenje u slučaju binarnog modela sa ugnježdenim intervalima, ovde ćemo koristiti ternarni zapis granica intervala. Primetimo da su uzimajući u obzir način formiranja podintervala, leva odnosno desna granica jedinstvene. Isto tako, dužina intervala koji se pridružuje nekom čvoru stabla jednaka je $\frac{1}{3^k}$ $k \in N$. Nije teško pokazati da je ternarni zapis intervala koji se pridružuju čvorovima stabla sledećeg oblika:

$$(0.a_1a_2\ldots a_{k-1}1, 0.a_1a_2\ldots a_{k-1}2), a_i \in \{0,1,2\} \tag{2.53}$$

Očigledno je interval jedinstveno određen svojom levom (desnom) granicom. Dakle dovoljno je za obeležje čvora uzeti samo na primer, levu granicu i to njen ternarni zapis. Obeležje korena će biti 0, a ostalih čvorova zapis oblika $0.a_1a_2\ldots a_{k-1}1$. Slično kao što je to urađeno u prethodnom modelu, zapis leve granice intervala mozemo shvatiti kao reč nad ternarnom azbukom $A_3 = \{0,1,2\}$ koja počinje slovom 0, a ako je dužine veće od 1 onda se mora završavati slovom 1. Očigledno možemo izbaciti početno slovo 0 i dobićemo skup obeležja koji čine prazna reč (obeležje korena) i reči ternarne azbuke koje se završavaju slovom 1 (obeležja čvorova koji nisu koren). Slova u reči koja je obeležje čvora opisuju izbor intervala koji se pridružuje datom čvoru i to na sledeći način:

- 0 - tekući interval je levi podinterval tekućeg intervala
- 1 - izbor srednjeg podintervala tekućeg intervala (ide se na sledeći nivo)
- 2 - tekući interval je desni podinterval tekućeg intervala

polazeći od intervala $(0,1)$ koji je pridružen korenu stabla (2.17).



Slika 2.17: Ternarni model sa levom granicom

Nije teško pokazati da za čvorove stabla x i y sa obeležjima left_x i left_y u ovom modelu važi:

$$x \leq y \Leftrightarrow \text{left}_x \leq_p \text{left}_y \tag{2.54}$$

gde je $\leq_p$ relacija prefiks na uočenom skupu obeležja. Na primer, čvoru L je pridružen interval $\left(\frac{691}{729}, \frac{692}{729}\right)$, pa je zapis leve granice bez vodeće 0:

$$\frac{691}{729} = \frac{2}{3^1} + \frac{2}{3^2} + \frac{1}{3^3} + \frac{1}{3^4} + \frac{2}{3^5} + \frac{1}{3^6}$$

reč 221121 nad ternarnom azbukom A_3 i to je obeležje čvora L. Prethodnici čvora L su čvorovi sa obeležjem koje je prefiks obeležja čvora L i završavaju se slovom 1:

· 2211 - čvor H

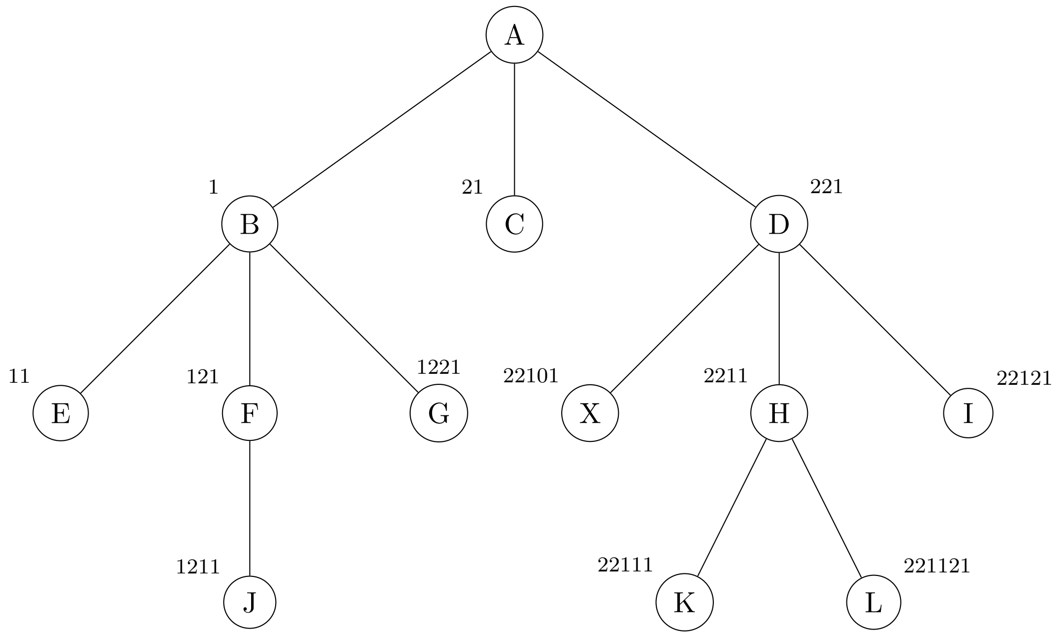
· 221 - čvor D

· ε - čvor A (koren stabla)

gde je ε prazna reč.

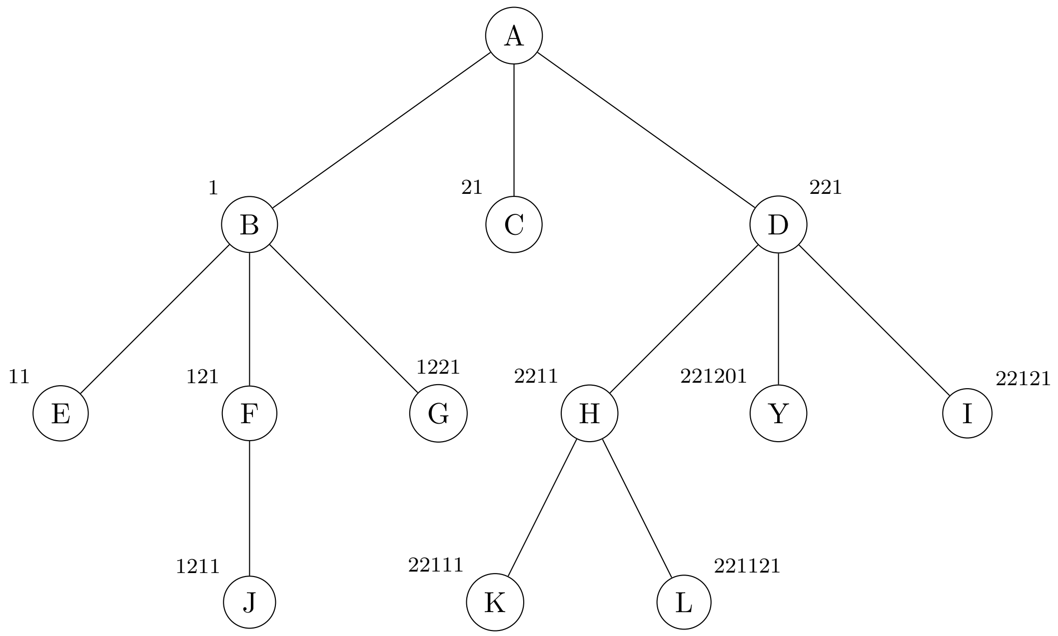
U ovom modelu se za razliku od binarnog modela sa ugnježdenim intervalima, odnosno modela sa izlistanim putanjama može efikasno umetati čvor, a da se pri tome postigne željeno uređenje potomaka datog čvora. Na primer, dodajmo čvorove X, Y, Z koji su potomci čvora D tako da se postigne sledeće uređenje $X < H < Y < I < Z$ potomaka čvora D. Odgovarajuća obeležja su:

X - 221+01 na čvoru D biramo levi podinterval 0 (čvoru H je pridružen srednji podinterval), a zatim se spuštamo na sledeći nivo 1 (2.18)



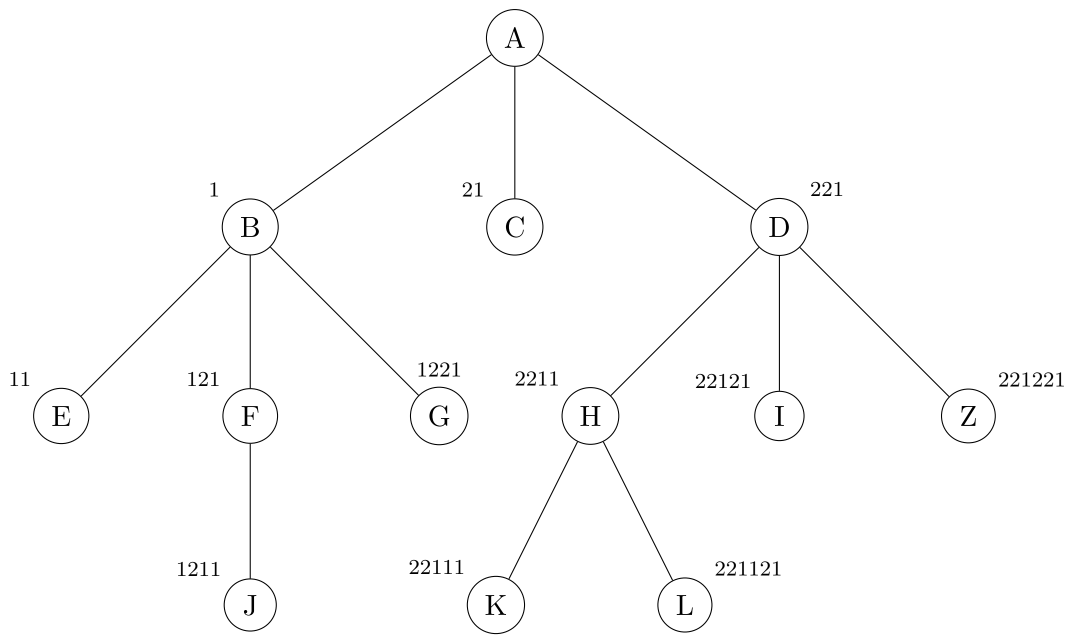
Slika 2.18: Ternarni model - insert $X < H$

Y - 221+201 na čvoru D biramo desni podinterval 2 (čvoru H je pridružen srednji podinterval), sredina desnog podintervala je čvor I zato dalje biramo levi podinterval 0 i na kraju se spuštamo na sledeći nivo 1 (2.19)



Slika 2.19: Ternarni model - insert $H < Y < I$

Z - 221+221 na čvoru D biramo desni podinterval 2 (čvoru H je pridružen srednji podinterval), sredina desnog podintervala je čvor I zato dalje biramo desni podinterval 2 i na kraju se spuštamo na sledeći nivo 1 (2.20)



Slika 2.20: Ternarni model - insert $I < Z$

Glava 3

Planiranje održavanja vazduhoplova

3.1 Uvodna razmatranja

U delu planiranja održavanja, jedan od osnovnih problema između ostalog je neizvesnost događaja koji značajno utiču na efikasnost održavanja. Reč je o događajima koji su po svojoj prirodi slučajni. Ovi slučajni događaji su pored ostalog vezani za:

- vremenski trenutak otkaza rezervnog dela
- vremenski trenutak zahteva za nekim delom
- vreme isporuke, odnosno vreme potrebno za sklapanje (pripremu) rezervnog dela

Posebno, podrazumevalo se da je vreme isporuke odnosno vreme potrebno za pripremu rezervnog dela nula (rezervni deo je neposredno dostupan), a to je pretpostavka neograničenih resursa što je uglavnom redak slučaj. Kako mnogo faktora utiče na ovo vreme, ono po pravilu uvek ima neko slučajano odstupanje. Za neke analize vreme isporuke odnosno vreme sklapanja rezervnog dela se može ignorisati, ali u analizi kompletnog sitema održavanja ova odstupanja bitno degradiraju performanse sistema. Pod vremenom isporuke (*lead time*) za dati rezervni deo podrazumeva se:

- vreme koje je potrebno da bi se rezervni deo dopremio iz nekog spoljnjeg izvora
- vreme koje je potrebno da bi se rezervni deo sklopio prema utvrđenoj proceduri sklapanja
- vreme da bi se poluproizvod obradio do nivoa gotovog rezervnog dela

Na ovo vreme utiče mnoštvo faktora. Neki od faktora su:

- problemi u transportu
- kvar na mašinama
- problemi vezani za kvalitet

Ovi faktori su po svojoj prirodi slučajne veličine, pa je i ovo vreme isporuke po pravilu slučajna veličina. Jedan od pristupa kojim se može umanjiti uticaj ovih slučajnih veličina je skladištenje gotovih rezervnih delova, odnosno održavanje zaliha gotovih rezervnih delova. U slučaju da je ovaj nivo iznad odgovarajućeg javiće se nepotrebni troškovi skladištenja. Sa druge strane ako je nivo zaliha ispod odgovarajućeg nivoa javiće se troškovi kašnjenja. Cilj je održavati nivo zaliha kojim se obezbeđuju minimalni troškovi skladištenja sa jedne strane a pri tome se garantuje visok nivo raspoloživosti gotovih delova kako bi se minimizovali troškovi kašnjenja. Za uspešno rešavanje ovog problema neophodno je razviti odgovarajuće matematičke modele koji će na odgovarajući način tretirati neizvesnost pojedinih faktora u sistemu održavanja. Posebno problem određivanja nivoa zaliha rezervnih delova u kontekstu slučajnog broja zahteva za rezervnim delovima je problem koji je najviše izučavan i analiziran. Kako je naručivanje i prodaja dnevnih novina ilustrativna metafora za ovaj tip problema, za probleme ovog tipa koristi se termin *newsboy problem*, a za odgovarajući model se koristi termin *newsboy model* [28]. Newsboy problemi nisu samo problemi iz oblasti upravljanja zalihama, oni se javljaju u svim situacijama kada treba izvršiti ocenu neke slučajne promenljive, a ta ocena je rezultat kompromisa između gubitaka u slučaju da je vrednost slučajne promenljive precenjena i gubitaka koji su posledica potcenjene vrednosti slučajne promenljive. Posebno ovaj problem se može iskoristiti u situacijama kada treba odrediti optimalni vremenski trenutak kada treba započeti neku aktivnost održavanja.

U planiranju održavanja česta je sledeća situacija. Neki skup aktivnosti održavanja ima hijerarhijsku strukturu i pri tome je trajanje svake aktivnosti slučajna veličina. Pod pretpostavkom da aktivnost na višem nivou hijerarhije ne može započeti pre nego što su završene sve aktivnosti na nižem nivou hijerarhije i da su poznati gubici koji nastaju u slučaju da se sa nekim aktivnostima kasni, odnosno da su poznati gubici u slučaju da su neke aktivnosti preuranjene, potrebno je formirati optimalnu strategiju aktiviranja aktivnosti iz datog skupa. Bez gubitka opštosti možemo pretpostaviti da su te aktivnosti vezane za proces sklapanja (pripreme) rezervnog dela koji je opisan odgovarajućim stablom sklapanja. Dakle u kontekstu sklapanja rezervnih delova problem je sledeći. Zahteva se da neki finalni sklop (rezervni deo) bude sklopljen (raspoloživ) u datom vremenskom trenutku. Neka su aktivnosti sklapanja (pripreme) datog finalnog sklopa opisane stablom sklapanja. Vreme potrebno za sklapanje svakog od podsklopova je slučajna veličina. Sklapanje nadsklopa ne može započeti pre nego što su svi podsklopovi raspoloživi. U slučaju da je neki podsklop sklopljen ranije javiće se troškovi skladištenja. Ako je finalni sklop sklopljen pre utvrđenog roka javiće se troškovi skladištenja, slično ako je sklapanje finalnog sklopa završeno nakon utvrđenog roka javiće se troškovi kašnjenja. Potrebno je odrediti optimalne vremenske trenutke aktiviranja faza sklapanja pojedinih podsklopova tako da se sklapanje finalnog sklopa završi uz minimalne troškove.

Problem o kome je reč je u suštini isuviše komplikovan u opštem slučaju, kada je proces sklapanja opisan stablom sklapanja proizvoljnog nivoa. Problem se značajno uprošćava ako se rešenje problema za stablo visine jedan iterativno primeni na podstblima datog stabla sklapanja proizvoljnog nivoa. Na ovaj način se rešenje jednostavnijeg problema na stablu visine jedan može na odgovarajući način produžiti na stabla sklapanja proizvoljne visine.

Primeri takvih istraživanja su [30], koji obradjuje sličan problem u sistemu sklapanja sa dva odnosno tri nivoa, pri tome je glavna poteškoća u ovom modelu predstavljanje funkcije cilja u konačnom obliku za sisteme sklapanja nivoa većeg od tri. U [8], je razvijena rekurzivna šema za efikasno izračunavanje funkcije cilja u sistemima sklapanja proizvoljnog nivoa. Primeri radova koji analiziraju slične probleme za jednostepene odnosno dvostepne sisteme su [6], [10] i [29]. Slično rešenje koje se sastoji u iterativnoj primeni rešenja jednostavnijeg problema se može naći u [2].

3.2 Newsboy model

U sistemima za održavanje vazduhoplova newsboy model se može direktno koristiti u slučajevima kada treba proceniti vremena isporuke odnosno vremena sklapanja rezervnih delova. Neka je poznat vremenski trenutak t_0 kad je neophodno da rezervni deo bude raspoloživ i neka je slučajna promenljiva L vreme isporuke (*lead time*) rezervnog dela, odnosno vreme koje je potrebno da bi se sklopio zahtevani rezervni deo. U daljem tekstu ćemo podrazumevati da se rezervni delovi sklapaju, odnosno slučajna promenljiva L predstavlja vreme neophodno da bi se sklopio neki rezervni deo i bio raspoloživ za ugradnju. Prirodno je pretpostaviti da je vreme sklapanja L nenegativna ograničena slučajna promenljiva. Potrebno je odrediti vremenski trenutak t_0^* kada treba započeti proces sklapanja rezervnog dela. Neka je vremenski trenutak t_0^* određen na neki način, slučajna promenljiva $t_0^* + L$ predstavlja vremenski trenutak kada je rezervni deo raspoloživ. Neka su pri tom poznati jedinični troškovi:

– h skladištenja rezervnog dela u slučaju da je realizovano vreme sklapanja takvo da je vremenski trenutak u kome je rezervni deo raspoloživ nastupio pre utvrđenog roka (*holding cost*)

– b kašnjenja u slučaju da je realizovano vreme sklapanja takvo da je vremenski trenutak u kome je rezervni deo raspoloživ nastupio nakon utvrđenog roka (*backlogging cost*)

Pretpostavimo da su troškovi skladištenja odnosno kašnjenja proporcionalni odstupanju procene od realizovane vrednosti, odnosno pretpostavimo da su ovi troškovi proporcinalni vremenu čekanja na ugradnju odnosno vremenu kašnjenja ugradnje. Funkcija ukupnih troškova C je u ovom slučaju jednaka:

$$\begin{aligned} C &= C(t_0^*, t_0, L) \\ &= h \cdot (t_0 - (t_0^* + L))^+ + b \cdot ((t_0^* + L) - t_0)^+ \\ &= h \cdot ((t_0 - t_0^*) - L)^+ + b \cdot (L - (t_0 - t_0^*))^+ \end{aligned} \tag{3.1}$$

Označimo sa $\tau_0 = t_0 - t_0^*$, tada je:

$$C = C(\tau_0, L) = h \cdot (\tau_0 - L)^+ + b \cdot (L - \tau_0)^+ \tag{3.2}$$

Ovaj model daje ukupne troškove koji su rezultat procene slučajne promenljive L vrednošću $\tau_0 = t_0 - t_0^*$, što je standardni newsboy model čije je rešenje:

$$F(\tau^*) = \frac{b}{b+h} \tag{3.3}$$

odnosno:

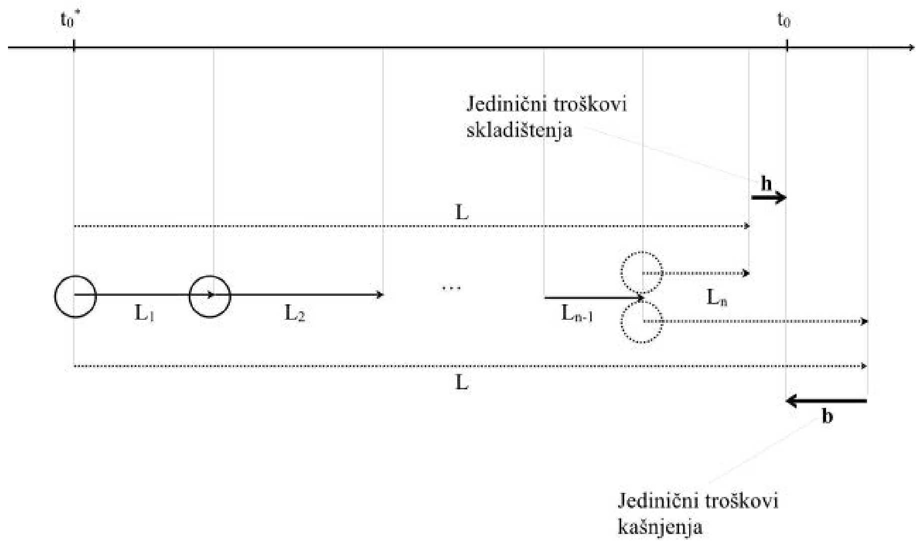
$$\tau^* = F^{-1}\left(\frac{b}{b+h}\right) \tag{3.4}$$

gde je F funkcija raspodele slučajne promenljive L. Optimalni vremenski trenutak, kada treba započeti sklapanje rezervnog dela je:

$$t_0^* = t_0 - F^{-1}\left(\frac{b}{b+h}\right) \tag{3.5}$$

3.2.1 Jedna linija sklapanja

Neka se proces sklapanja sastoji iz $n > 1$ faza koje se sekvencijalno izvršavaju i neka je reč samo o jednoj liniji sklapanja. Pretpostavimo pri tom da nema skladištenja između pojedinih faza sklapanja, odnosno sa završetkom jedne faze aktivira se sledeća faza sklapanja (3.1).



Slika 3.1: Jedna linija sklapanja

Neka je $L_i, i = 1, 2, \ldots, n$ niz slučajnih promenljivih koje predstavljaju vremena trajanja i-te faze respektivno, tada je ukupno vreme sklapanja slučajna promenljiva:

$$L = \sum_{i=1}^{n} L_i \tag{3.6}$$

Rešenje ovog problema je dato sa:

$$t_0^* = t_0 - F_L^{-1}\left(\frac{b}{b+h}\right) \tag{3.7}$$

gde je F_L funkcija raspodele slučajne promenljive (3.6). Problem određivanja raspodele sume slučajnih promenljivih je u opštem slučaju složen, ali se može uprostiti ako uvedemo dodatne pretpostavke o nizu slučajnih promenljivih $L_i, i = 1, 2, \ldots, n$. Pretpostavimo da važi:

$$F^{-1}_{\sum_i L_i} = \sum_i F^{-1}_{L_i} \tag{3.8}$$

što odgovara linearnosti i aditivnosti vremena u procesu održavanja. Uslov (3.8) važi ako postoji slučajna promenljiva X takva da je:

$$L_i = \alpha_i X, \alpha_i \in \mathbb{R}, i = 1, 2, \ldots, n \tag{3.9}$$

i u tom slučaju rešenje problema linijskog procesa sklapanja je dato sa:

$$t_0^* = t_0 - \sum_{i=1}^{n} F_{L_i}^{-1}\left(\frac{b}{b+h}\right) \tag{3.10}$$

Pod određenim uslovima raspodela slučajne promenljive (3.6) može se odrediti korišćenjem *Centralne Granične Teoreme* [1].

3.2.2 Višestruka neodređenost

Neka je poznat vek trajanja datog rezervnog dela i neka je to slučajna promenljiva X i neka je očekivani vek trajanja rezervnog dela $E(X)$. To znači da će u nekom trenutku biti neophodna zamena datog rezervnog dela, odnosno dati rezervni deo mora biti raspoloživ u nekom vremenskom trenutku. Ekonomski je neopravdano istovremeno sa ugradnjom rezervnog dela, obezbediti jedan ili više primeraka koji bi se skladištili do trenutka otkaza ugrađenog dela kada treba izvršiti zamenu. Ako beležimo trenutak ugradnje t_u rezervnog dela tada slučajna promenljiva $T = t_u + X$ predstavlja trenutak otkaza ugrađenog rezervnog dela, odnosno trenutak kada rezervni deo mora biti raspoloživ. Jedna od procena trenutka otkaza je:

$$t_0 = E(T) = t_u + E(X) \tag{3.11}$$

Ako iskoristimo klasičan newsboy problem dobićemo funkciju ukupnih troškova C:

$$C = C(t_0^*, t_0, L) = h \cdot (t_0 - (t_0^* + L))^+ + b \cdot ((t_0^* + L) - t_0)^+ \tag{3.12}$$

Potrebno je odrediti optimalni vremenski trenutak t_0^* kada treba započeti proces sklapanja datog rezervnog dela. Ako dozvolimo da utvrđeni rok t_0 može biti slučajna promenljiva doći ćemo do jednog uopštenja klasičnog newsboy modela. Dakle neka je i utvrđeni rok slučajna promenljiva T, funkcija ukupnih troškova C postaje:

$$\begin{aligned} C &= C(t_0^*, T, L) \\ &= h \cdot (T - (t_0^* + L))^+ + b \cdot ((t_0^* + L) - T)^+ \end{aligned} \tag{3.13}$$

Neka je $t_0 = E(T)$

$$\begin{aligned} C &= C(t_0^*, T, L) \\ &= h \cdot (t_0 + (T - t_0) - (t_0^* + L))^+ + b \cdot ((t_0^* + L) - (t_0 + (T - t_0)))^+ \\ &= h \cdot ((t_0 - t_0^*) - (L - (T - t_0)))^+ + b \cdot ((L - (T - t_0)) - (t_0 - t_0^*))^+ \end{aligned} \tag{3.14}$$

Označimo sa $\tau_0 = t_0 - t_0^*$ i $\Delta = T - t_0$, tada je funkcija ukupnih troškova:

$$\begin{aligned} C &= C(\tau_0, \Delta, L) \\ &= C(\tau_0, L-\Delta) \\ &= h \cdot (\tau_0 - (L-\Delta))^+ + b \cdot ((L-\Delta) - \tau_0)^+ \end{aligned} \tag{3.15}$$

što je klasičan newsboy model . Njegovo rešavanje se svodi na određivanje raspodele slučajne promenljive $L - \Delta$. Kako postoji veza između gustina raspodela razlike nezavisnih slučajnih promenljivih [1]:

$$f_{L-\Delta}(t) = \int\limits_{-\infty}^{+\infty} f_L(t+x) f_\Delta(x) \mathrm{d}x \tag{3.16}$$

može se odrediti funkcija raspodele $F_{L-\Delta}$ razlike nezavisnih slučajnih promenljivih $L - \Delta$

$$F_{L-\Delta}(t) = \int\limits_{-\infty}^{+\infty} F_L(t+x) f_\Delta(x) \mathrm{d}x \tag{3.17}$$

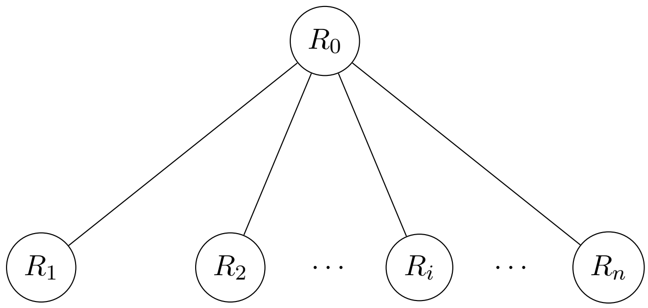
dalje se optimalna vrednost:

$$t_0^* = t_0 - F_{L-\Delta}^{-1}\left(\frac{b}{b+h}\right) \tag{3.18}$$

može odrediti numeričkim metodama.

3.3 Simultani zahtev I

Razmotrimo sada opštiji problem kada $n > 1$ rezervnih delova treba simultano obezbediti u datom vremenskom trenutku. Neka je proces sklapanja rezervnog dela opisan stablom visine dva (3.2).

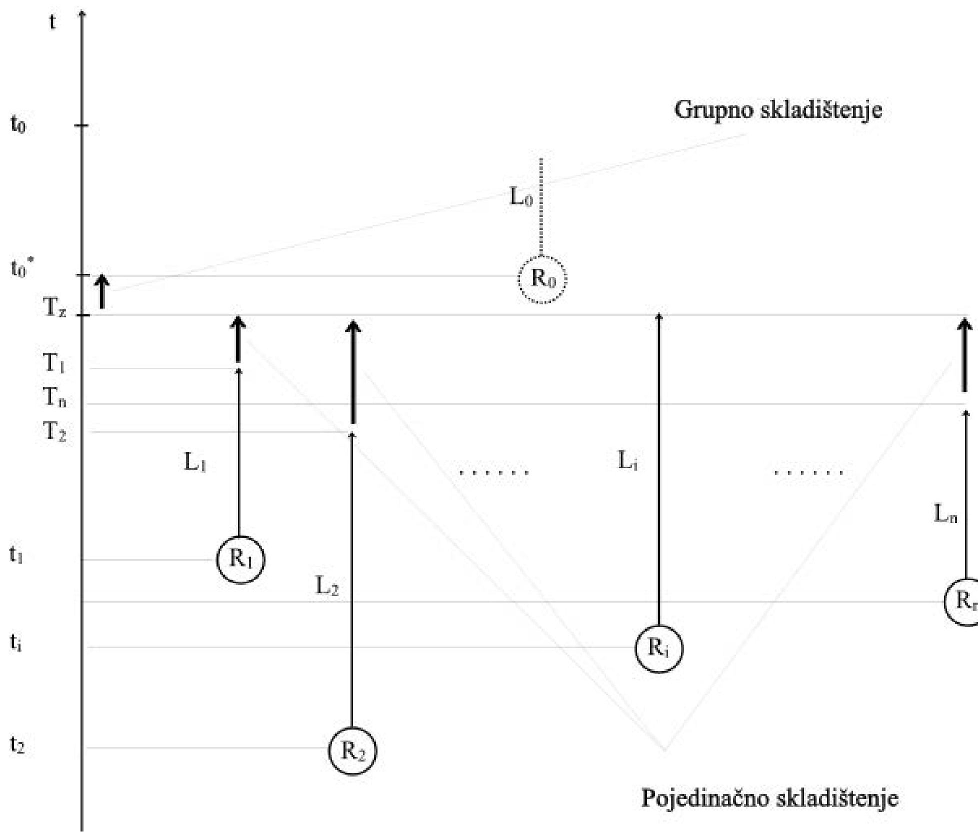


Slika 3.2: Stablo sklapanja visine 2

Kako u ovoj situaciji rezervni deo ima neku strukturu u daljem tekstu ćemo za rezervni deo u zavisnosti od uloge u uočenoj strukturi koristiti termine sklop, podsklop ili nadsklop. Neka je t_0 vremenski trenutak kada finalni sklop mora biti raspoloživ. Neka je vreme sklapanja finalnog sklopa modelirano slučajnom promenljivom L sa funkcijom raspodele F i neka su dalje h i b, troškovi skladištenja odnosno kašnjenja finalnog sklopa. Ovo je klasičan newsboy problem i njegovim rešavanjem se dobija vremenski trenutak t_0^* kada treba započeti sklapanje finalnog sklopa. Ako pretpostavimo da proces sklapanja nadsklopa može započeti tek nakon što su svi podsklopovi raspoloživi, vremenski trenutak t_0^* je simultani rok do koga svi podsklopovi finalnog sklopa moraju biti raspoloživi. Neka je sklapanje i-tog podsklopa započeto u trenutku t_i, $i = 1, 2, \ldots, n$, a vreme potrebno za sklapanje i-tog podsklopa neprekidna slučajna promenljiva L_i, $i = 1, 2, \ldots, n$, sa funkcijom raspodele F_i i funkcijom gustine f_i. Vremenski trenutak kada je i-ti podsklop sklopljen je $T_i = t_i + L_i$, $i = 1, 2, \ldots, n$, a faza sklapanja podsklopova je završena u trenutku kada je i poslednji podsklop sklopljen, odnosno u trenutku $T_z = \max\limits_i \{T_i\}$. Kako bi obezbedili da se proces sklapanja završava u konačnom vremenu, pretpostavićemo da su vremena sklapanja L_i, $i = 1, 2, \ldots, n$ nenegativne ograničene slučajne promenljive. U konačnom trenutku T_z svi podsklopovi su raspoloživi i može započeti faza sklapanja nadsklopa, posebno ako je i-ti podsklop sklopljen u trenutku $T_i < T_z$, $i = 1, 2, \ldots, n$, on se svakako mora skladištiti do trenutka T_z. Pretpostavićemo da su jedinični troškovi skladištenja i-tog podsklopa jednaki h_i i da su proporcionalni periodu skladištenja $(T_z - T_i)$, $i = 1, 2, \ldots, n$. Dakle svaki podsklop se od trenutka T_i kada je sklopljen, do trenutka T_z kada su svi podsklopovi sklopljeni, pojedinačno skladišti i odgovarajuće troškove skladištenja $h_i(T_z - T_i)$, $i = 1, 2, \ldots, n$ ćemo zvati pojedinačni troškovi skladištenja. Zbog prirode procesa sklapanja pojedinačni troškovi kašnjenja ne figurišu u ovom modelu, odnosno ne postoji podsklop čije je sklapanje završeno nakon trenutka T_z kada je moguće započeti sklapanje nadsklopa. Faza sklapanja podsklopova je modelirana grupnom karakteristikom $T_z = \max\limits_i \{T_i\}$, pa u zavisnosti od odnosa ove veličine i predviđenog roka t_0^* javiće se grupni troškovi:

- $T_z < t_0^*$ - podsklopovi su sklopljeni pre predviđenog roka t_0^*. Odgovarajuće troškove ćemo zvati grupni troškovi skladištenja. Slično kao kod pojedinačnih troškova skladištenja, pretpostavimo da su jedinični troškovi grupnog skladištenja jednaki $\hat{h}$ i neka su proporcionalni vremenu grupnog skladištenja $(t_0^* - T_z)$ (3.3).

– $T_z > t_0^*$ - podsklopovi su sklopljeni nakon predviđenog roka t_0^*. Odgovarajuće troškove ćemo zvati grupni troškovi kašnjenja. Neka su jedinični troškovi grupnog kašnjenja jednaki $\hat{b}$ i neka su proporcionalni vremenu grupnog kašnjenja $(T_z - t_0^*)$ (3.4).

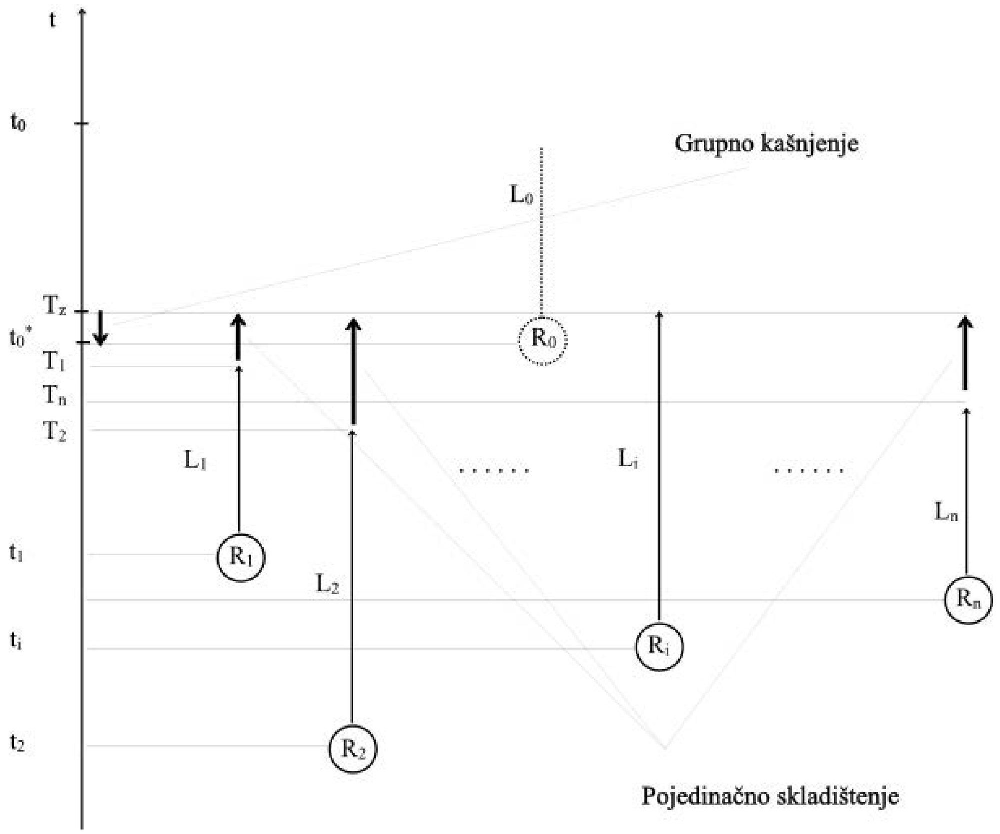


Slika 3.3: Grupno skladištenje

Iz ove analize izvodi se funkcija ukupnih troškova:

$$
\begin{aligned}
C &= C(t_1, \ldots, t_n, t_0^*, L_1, \ldots, L_n) \\
&= \hat{h}(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) + \hat{b}(T_z - t_0^*)^+
\end{aligned} \tag{3.19}
$$

Uzimajući u obzir činjenicu da su troškovi kašnjenja samo grupni troškovi kašnjenja, u daljem tekstu ćemo koristiti termin troškovi kašnjenja podrazumevajući grupne troškove kašnjenja. Slično koristićemo termin troškovi koji može u zavisnosti od konteksta podrazumevati jedinične troškove. Problem je sledeći odrediti vremenske trenutke t_i, $i = 1, 2, \ldots, n$, kada treba započeti proces sklapanja i-tog podsklopa kako bi, kada su svi podsklopovi sklopljeni u trenutku T_z, očekivani ukupni troškovi koji su posledica odstupanja od predviđenog roka t_0^* bili minimalni. Kako bi obezbedili da naš problem ima jedinstveno rešenje uvodićemo neop-



Slika 3.4: Grupno kašnjenje

hodne pretpostavke koje će obezbediti konveksnost funkcije [3] očekivanih ukupnih troškova, a time i jedinstvenost rešenja.
Primetimo da funkcija ukupnih troškova izmedju ostalih uključuje i parametre jediničnih troškova i to:

– jedinične troškove pojedinačnog skladištenja podsklopova h_i, $i = 1, 2, \ldots, n$

– jedinične troškove grupnog skladištenja $\hat{h}$

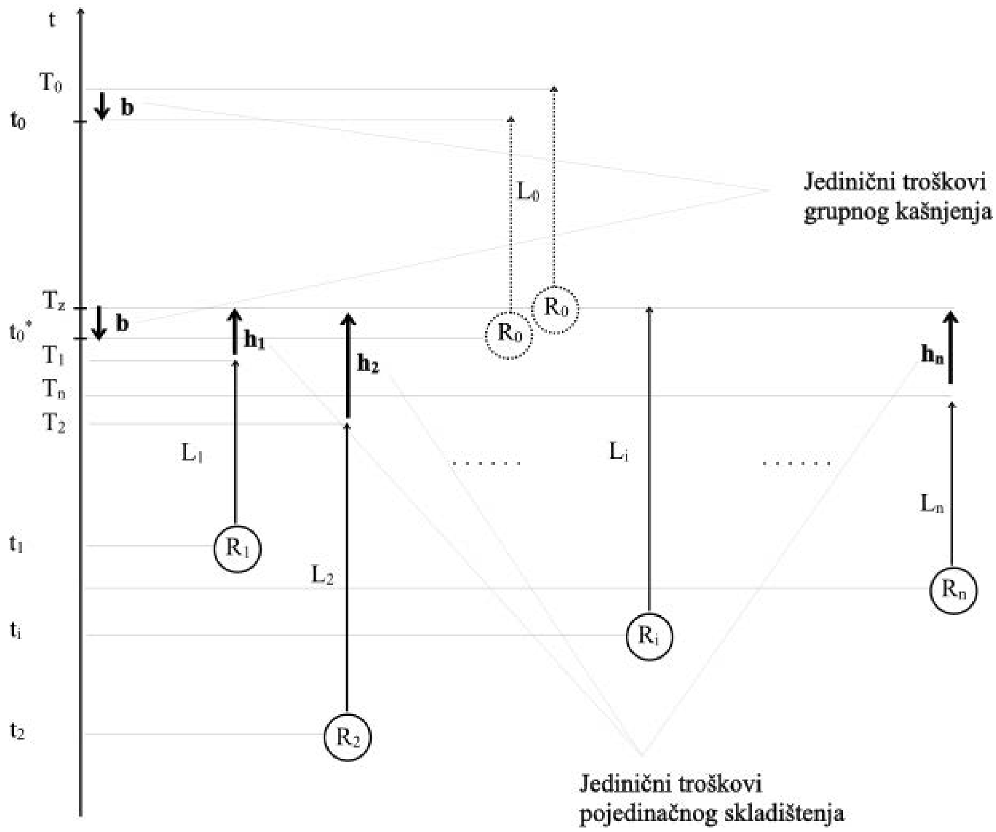
– jedinične troškove grupnog kašnjenja $\hat{b}$.

Prirodno je pretpostaviti da su poznati samo jedinični troškovi pojedinačnog skladištenja podsklopova, a da su parametri grupnih troškova $\hat{h}$ i $\hat{b}$ funkcije jediničnih troškova h i b na nadsklopu (u ovom slučaju radi se o finalnom sklopu). Opravdano je pretpostaviti da će kašnjenje za vremenski period $(T_z - t_0^*)^+$ na podsklopovima proizvesti isto kašnjenje na nadsklopu odnosno, troškovi kašnjenja na podsklopovima proizvode iste troškove kašnjenja na nadredjenom sklopu. Dakle, opravdano je pretpostaviti:

$$\hat{b}(T_z - t_0^*)^+ = b(T_z - t_0^*)^+$$

odnosno jedinični troškovi kašnjenja $\hat{b}$ na neposrednim podsklopovima finalnog sklopa su jednaki jediničnim troškovima kašnjenja b finalnog sklopa (3.5) odnosno:

$$\hat{b} = \hat{b}(L_1, \ldots, L_n, b) = b$$



Slika 3.5: Grupno kašnjenje

Sa ovim pretpostavkama funkcija ukupnih troškova na nivou neposrednih podsklopova finalnog sklopa je:

$$\begin{aligned} C &= C(t_1, \tau_2, \ldots, t_n, t_0^*, L_1, \ldots, L_n) \\ &= \hat{h}(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) + b(T_z - t_0^*)^+ \end{aligned} \tag{3.20}$$

Ostaje da na neki način definišemo jedinične troškove grupnog skladištenja:

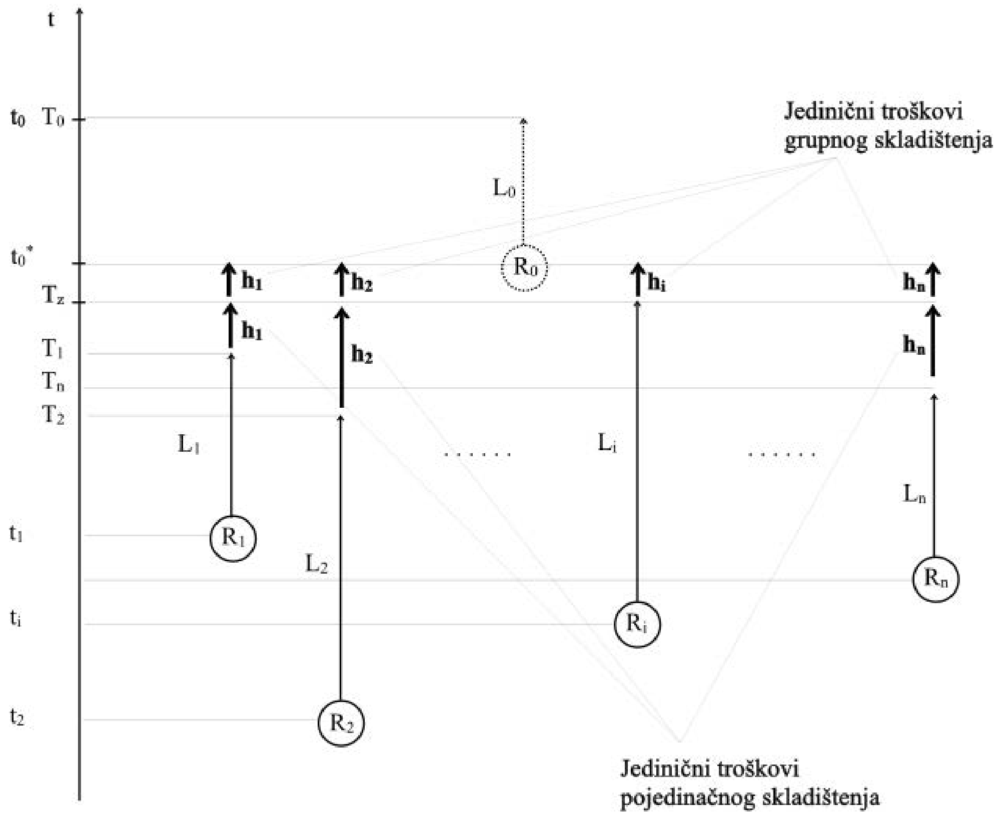
$$\hat{h} = \hat{h}(h_1, h_2, \ldots, h_n, L_1, \ldots, L_n, h)$$

na nivou neposrednih podsklopova finalnog sklopa. Pretpostavimo da se proces sklapanja izvodi na sledeći način, sklopljeni podsklopovi se uvek skladište do predviđenog roka t_0^*, bez obzira što je sklapanje svih podsklopova završeno pre predviđenog roka t_0^*. Drugim rečima nema troškova grupnog skladištenja, sva skladištenja su pojedinačnog tipa tj. svaki sklopljeni podsklop se skladišti do predviđenog roka t_0^* sa svojim jediničnim troškovima h_i (3.6). Dakle pretpostavimo sledeće:

$$\hat{h}(t_0^* - T_z)^+ = \sum_{i=1}^{n} h_i(t_0^* - T_z)^+ \tag{3.21}$$

odnosno:

$$\hat{h} = \hat{h}(h_1, h_2, \ldots, h_n, L_1, \ldots, L_n, h) = \sum_{i=1}^{n} h_i \tag{3.22}$$



Slika 3.6: Grupno skladištenje

Funkcija ukupnih troškova $C = C(t_1, t_2, \ldots, t_n, t_0^*, L_1, \ldots, L_n)$ je u ovom slučaju:

$$\begin{aligned} C &= \sum_{i=1}^{n} h_i (t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i (T_z - T_i) + b(T_z - t_0^*)^+ \\ &= \sum_{i=1}^{n} h_i (t_0^* - T_z) + \sum_{i=1}^{n} h_i (T_z - T_i) + \left(b + \sum_{i=1}^{n} h_i \right) (T_z - t_0^*)^+ \\ &= \sum_{i=1}^{n} h_i (t_0^* - T_i) + \left(b + \sum_{i=1}^{n} h_i \right) (T_z - t_0^*)^+ \end{aligned} \tag{3.23}$$

Kako je $T_i = t_i + L_i$, odnosno $T_z = \max_i \{T_i\}$, dobijamo:

$$C = \sum_{i=1}^{n} h_i (t_0^* - t_i - L_i) + \left(b + \sum_{i=1}^{n} h_i \right) \left(\max_i \{t_i + L_i - t_0^*\} \right)^+ \tag{3.24}$$

Uvedimo sledeću oznaku $\tau_i = t_0^* - t_i$, $i = 1, 2, \ldots, n$ i imaćemo:

$$\begin{aligned} C &= C(\tau_1, \tau_2, \ldots, \tau_n, L_1, \ldots, L_n) \\ &= \sum_{i=1}^{n} h_i(\tau_i - L_i) + \left(b + \sum_{i=1}^{n} h_i\right)\left(\max_i \{L_i - \tau_i\}\right)^+ \end{aligned} \tag{3.25}$$

Očekivani ukupni troškovi $Z = E(C) = Z(\tau_1, \tau_2, \ldots, \tau_n)$ su:

$$Z = \sum_{i=1}^{n} h_i(\tau_i - E(L_i)) + \left(b + \sum_{i=1}^{n} h_i\right) E\left((\max_i \{L_i - \tau_i\})^+\right) \tag{3.26}$$

Iz ove forme funkcije Z se direktno može utvrditi egzistencija globalnog minimuma odnosno važi sledeći stav:

Teorema 3.3.1. *Funkcija očekivanih troškova $Z = E(C) = Z(\tau_1, \tau_2, \ldots, \tau_n)$ data sa (3.26) ima globalni minimum u konačnoj tački $(\tau_1^*, \tau_2^*, \ldots, \tau_n^*)$.*

Dokaz. *Pokažimo prvo da je funkcija ukupnih troškova Z data sa (3.26) konveksna funkcija svojih argumenata $\tau_1, \tau_2, \ldots, \tau_n$. Prvi sabirak:*

$$\sum_{i=1}^{n} h_i \left(\tau_i - E(L_i)\right) \tag{3.27}$$

je linearna dakle konveksna funkcija argumenata $\tau_1, \tau_2, \ldots, \tau_n$, dok konveksnost drugog sabirka:

$$\left(b + \sum_{i=1}^{n} h_i\right) E\left((\max_i \{L_i - \tau_i\}\right)^+) \tag{3.28}$$

sledi iz iz monotonosti i linearnosti očekivanja i konveksnosti maksimuma konveksnih funkcija. Dakle data funkcija ukupnih troškova Z je konveksna funkcija kao zbir konveksnih funkcija. Sa druge strane možemo zaključiti sledeće:

- $\tau_i \to +\infty \Rightarrow Z \sim h_i \tau_i \to +\infty$
- $$\begin{aligned} \tau_i \to -\infty \Rightarrow Z \sim \quad & h_i \tau_i - \tau_i \left(b + \sum_{i=1}^{n} h_i\right) \\ &= -\tau_i \left(b + \left(\sum_{i=1}^{n} h_i\right) - h_i\right) \\ &= -\tau_i \left(b + \sum_{j \neq i}^{n} h_j\right) \to +\infty \end{aligned}$$

odnosno:

$$lim_{\tau_i \to \pm\infty} Z = +\infty \tag{3.29}$$

Navedene osobine konveksne funkcije očekivanih troškova obezbeđuju egzistenciju lokalnog minimuma u konačnoj tački $(\tau_1^*, \tau_2^*, \ldots, \tau_n^*)$, koji će biti i globalni minimum [2] [9] [26].

◇

Razvijmo dalje funkciju očekivanih troškova. Pre svega neophodno je odrediti funkciju raspodele slučajne promenljive D^+:

$$\begin{aligned} D^+ &= \left(\max_i \{L_i - \tau_i\}\right)^+ \\ &= \max\left\{\max_i \{L_i - \tau_i\}, 0\right\} \end{aligned} \tag{3.30}$$

Neka je $t > 0$, funkcija raspodele F_{D^+} slučajne promenljive D^+ je:

$$\begin{aligned} F_{D^+}(t) &= P(D^+ < t) = P\left(\max\left\{\max_i \{L_i - \tau_i\}, 0\right\} < t\right) \\ &= P\left(\max_i \{L_i - \tau_i\} < t\right) \\ &= P\left(\bigcap_i (L_i - \tau_i < t)\right) = \prod_i P(L_i < t + \tau_i) = \prod_i F_i(t + \tau_i) \end{aligned} \tag{3.31}$$

Očekivanje $E(D^+)$ slučajne promenljive D^+ se može odrediti kao očekivanje nenegativne slučajne promenljive [1]:

$$\begin{aligned} E(D^+) &= \int_0^{+\infty} P(D^+ \geq t)\mathrm{d}t = \int_0^{+\infty} \left(1 - P(D^+ < t)\right)\mathrm{d}t \\ &= \int_0^{+\infty} \left(1 - F_{D^+}(t)\right)\mathrm{d}t = \int_0^{+\infty} \left(1 - \prod_i F_i(t + \tau_i)\right)\mathrm{d}t \end{aligned} \tag{3.32}$$

Sada možemo dati funkciju očekivanih ukupnih troškova u razvijenom obliku:

$$Z(\tau_1, \tau_2, \ldots, \tau_n) = \sum_{i=1}^{n} h_i(\tau_i - E(L_i)) + \left(b + \sum_{i=1}^{n} h_i\right) \int_0^{+\infty} \left(1 - \prod_i F_i(t + \tau_i)\right)\mathrm{d}t \tag{3.33}$$

Stacionarna tačka funkcije Z je rešenje sledećeg sistema jednačina:

$$\frac{\partial Z}{\partial \tau_i} = h_i + \left(b + \sum_{i=1}^{n} h_i\right) \frac{\partial}{\partial \tau_i} \left(\int_0^{+\infty} \left(1 - \prod_i F_i(t + \tau_i)\right)\mathrm{d}t \right) = 0,\ i = 1, 2, .., n \tag{3.34}$$

dalje pretpostavimo uslove o funkcijama raspodele F_i, $i = 1, 2, \ldots, n$ tako da parcijalni izvod i integral komutiraju [25] i dobićemo:

$$\frac{\partial Z}{\partial \tau_i} = h_i + \left(b + \sum_{i=1}^{n} h_i\right) \int_0^{+\infty} \frac{\partial}{\partial \tau_i} \left(1 - \prod_i F_i(t+\tau_i)\right) \mathrm{d}t = 0, \ i = 1, 2, .., n$$

$$\frac{\partial Z}{\partial \tau_i} = h_i - \left(b + \sum_{i=1}^{n} h_i\right) \int_0^{+\infty} f_i(t+\tau_i) \prod_{j \neq i} F_j(t+\tau_j) \mathrm{d}t = 0, \ i = 1, 2, .., n \tag{3.35}$$

Konačno stacionarne tačke su rešenja sledećeg sistema nelinearnih jednačina:

$$\int_0^{+\infty} f_i(t+\tau_i) \prod_j F_j(t+\tau_j) \mathrm{d}t = \frac{h_i}{b + \sum\limits_{i=1}^{n} h_i}, \ i = 1, 2, .., n \tag{3.36}$$

Dobijeni sistem jednačina se može rešiti numeričkim metodama i neka su $\tau_1^*, \tau_2^*, \ldots, \tau_n^*$, ove vrednosti. Na ovaj način su određeni optimalni vremenski trenuci $t_i^* = t_0^* - \tau_i$, $i = 1, \ldots, n$, kada treba započeti sklapanje i-tog podsklopa tako da su očekivani ukupni troskovi minimalni.

Primer 3.3.1. Primetimo sledeće, neka je $n = 1$ tada se stacionarna tačka, odnosno optimalno rešenje τ_1^* dobija kao rešenje jednačine:

$$\int_0^{+\infty} f_1(t+\tau_1) \mathrm{d}t = \frac{h_1}{b + h_1} \tag{3.37}$$

kako je: $\int\limits_0^{+\infty} f_1(t+\tau_1)\mathrm{d}t = \int\limits_{\tau_1}^{+\infty} f_1(u)\mathrm{d}u = 1 - F_1(\tau_1)$, gornja jednačina postaje:

$$\begin{aligned} 1 - F_1(\tau_1) &= \frac{h_1}{b+h_1} \\ F_1(\tau_1) = 1 - \frac{h_1}{b+h_1} &= \frac{b}{b+h_1} \end{aligned} \tag{3.38}$$

i njeno rešenje je:

$$\tau_1^* = F_1^{-1}\left(\frac{b}{b+h_1}\right) \tag{3.39}$$

što je rešenje klasičnog newsboy problema na nivou podsklopa. Ovaj rezultat je očekivan jer je pretpostavljeno sledeće:

– lokalno skladištenje podsklopa sa jediničnim troškovima h_1

– kašnjenje na nivou podsklopa će se reprodukovati na finalnom sklopu, dakle jedinični troškovi kašnjenja na nivou podsklopa su b

3.3.1 Stablo sklapanja sa više nivoa I

Neka je proces sklapanja rezervnog dela opisan stablom visine veće od dva. Na opisani način mi mozemo odrediti kritična vremena $t_1^*, \tau_2^*, \ldots, t_n^*$ na nivou neposrednih podsklopova finalnog sklopa. Uočimo podstablo visine dva čiji je koren jedan od neposrednih podsklopova finalnog sklopa, neka je to i-ti ($1 \leq i \leq n$) podsklop. Formirajmo funkciju ukupnih troškova (3.23) koji odgovaraju uočenom podstablu sklapanja:

$$C_i = \sum_j (h_i)_j \left(t_i^* - (T_i)_j\right) + \left(\hat{b}_i + \sum_j (h_i)_j\right) \left((T_i)_z - t_i^*\right)^+ \tag{3.40}$$

gde su:

- t_i^* - utvrđeni rok za simultanu raspoloživost podsklopova i-tog podsklopa
- $(T_i)_j$ - trenutak raspoloživosti pojedinih podsklopova i-tog podsklopa
- $(T_i)_z$ - simultana raspoloživost podsklopova i-tog podsklopa
- $(h_i)_j$ - jedinični troškovi skladištenja pojedinih podsklopova i-tog podsklopa
- $\hat{b}_i$ - grupni troškovi kašnjenja i-tog podsklopa

Da bismo navedni postupak sproveli na ovom stablu sklapanja, neophodno je proceniti jedinične troškove kašnjenja $\hat{b}_i$ i-tog ($1 \leq i \leq n$) podsklopa koji je u korenu ovog podstabla. Jedno rešenje je da se izvrši raspodela troškova kašnjenja $\hat{b}$ nadsklopa u ovom slučaju finalnog sklopa $\hat{b} = b$, na troškove kašnjenja njegovih neposrednih podsklopova. Analizirajmo troškove kašnjenja nadsklopa:

$$B = \hat{b}(T_z - t_0^*)^+ \tag{3.41}$$

Svaki od podsklopova ima svoj udeo $B_i = \hat{b}_i(T_z - t_0^*)^+$ u ukupnim troškovima kašnjenja. Odredimo troškove kašnjenja i-tog podsklopa $\hat{b}_i$. Neka se sklapanje i-tog ($1 \leq i \leq n$) podsklopa pokrene u nekom trenutku $t_i + \Delta t_i$ za proizvoljno mali vremenski period Δt_i. Pretpostavimo $T_z \geq t_0^*$ (u suprotnom nema troškova kašnjenja niti ikakve promene za dovoljno malo Δt_i). U situaciji kao na slici (3.7) promena troškova kašnjenja je:

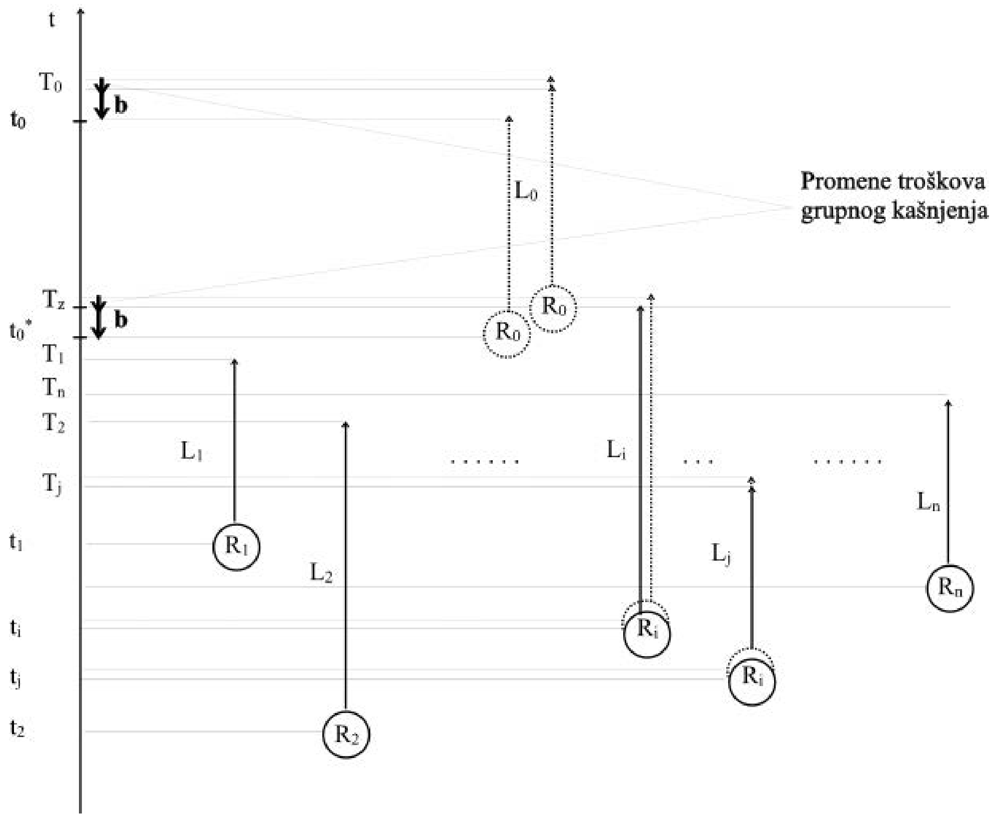
$$\Delta_i B = \Delta t_i \begin{cases} \hat{b}; & T_i = T_z \\ 0; & T_i < T_z \end{cases} \tag{3.42}$$

pa za jedinične troškove kašnjenja $\hat{b}_i$ i-tog ($1 \leq i \leq n$) podsklopa možemo uzeti očekivanje slučajne promenljive $\dfrac{\Delta_i B}{\Delta t_i}$, odnosno $\hat{b}_i = E\left(\dfrac{\Delta_i B}{\Delta t_i}\right) = \hat{b}P(T_i = T_z)$.

Primer 3.3.2. Primetimo sledeće, u slučaju da nadsklop ima tačno jedan podsklop, tj. $n = 1$, kako je $T_1 = T_z$ biće:

$$\hat{b}_1 = \hat{b} \cdot 1 = \hat{b} \tag{3.43}$$

što je i logično, jer u slučaju jedne linije sklapanja kašnjenje u podfazi sklapanja rezultuje istim kašnjenjem u završnoj fazi sklapanja, pa su i jedinični troškovi kašnjenja u podfazama sklapanja jednaki jediničnim troškovima kašnjenja u završnoj fazi sklapanja. Još više, neka se



Slika 3.7: Promena troškova kašnjenja

kasni sa sklapanjem svakog od podsklopova za neki vremenski period Δt. Očigledno sa jedne strane kasniće se sa sklapanjem nadsklopa za isti vremenski period Δt odnosno troškovi kašnjenja će se uvećati za iznos $\hat{b}\Delta t$, sa druge strane predloženi model uključuje uvećanje troškova kašnjenja svakog podsklopa za iznos $\hat{b_i}\Delta t = \hat{b}P(T_i = T_z)$ koji su u zbiru jednaki očekivanom uvećanju $\hat{b}\Delta t$:

$$\sum_i \hat{b_i}\Delta t = \sum_i \hat{b}P(T_i = T_z)\Delta t = \hat{b}\Delta t \sum_i P(T_i = T_z) = \hat{b}\Delta t \tag{3.44}$$

Drugim rečima predloženi model troškova kasnjenja lokalno tretira troškove kašnjenja koji se mogu pojaviti u narednoj fazi procesa sklapanja, a koji u toj fazi ne mogu biti tretirani. Na ovaj način smo pokazali da predloženi model na adekvatan način pokriva navedeni granični slučaj.

Konačno, jedinični troškovi kašnjenja $\hat{b_i}$ i-tog $(1 \le i \le n)$ podsklopa su:

$$\begin{aligned}
\hat{b_i} &= E\left(\frac{\Delta_i B}{\Delta t_i}\right) = \hat{b}P(T_i = T_z) \\
&= \hat{b}P(T_i = T_z) = \hat{b}P\left(T_i = \max_j\{T_j\}\right) = \hat{b}P\left(\bigcap_{j\neq i}(T_i > T_j)\right) \\
&= \hat{b}\left(\prod_{j\neq i}P(T_i > T_j)\right) = \hat{b}\left(\prod_{j\neq i}P(t_i^* + L_i > t_j^* + L_j)\right) \\
&= \hat{b}\left(\prod_{j\neq i}P(L_j - L_i < t_i^* - t_j^*)\right) = \hat{b}\left(\prod_{j\neq i}F_{L_j - L_i}(t_i^* - t_j^*)\right)
\end{aligned} \tag{3.45}$$

gde je $F_{L_j-L_i}$ funkcija raspodele slučajne promenljive $L_j - L_i$. Neka su poznati troškovi kašnjenja svakog pojedinačnog podsklopa $\hat{b_i}$, sada možemo formulisati problem odredjivanja kritičnih vremenskih trenutaka na svakom od podsklopova datog finalnog sklopa odnosno produžiti rešenje problema odredjivanja kritičnih vremenskih trenutaka na procese sklapanja opisanim podstablima korena polaznog stabla sklapanja.

Na ovaj način možemo formulisati rekurzivnu proceduru proračuna vremena za proces sklapanja rezervnog dela koji se opisuje stablom sklapanja proizvoljne visine. Neka svaki čvor stabla sklapanja, koji predstavlja podsklop finalnog rezervnog dela, ima kao karakteristike jedinične troškove skladištenja h, jedinične troškove kašnjenja b, optimalni vremenski trenutak t^* kada treba započeti sklapanje pridruženog podsklopa i neprekidnu slučajnu promenljivu L [1], koja predstavlja vreme sklapanja pridruženog podsklopa. Jasno je da su jedinični troškovi kašnjenja poznati samo za finalni rezervni deo, odnosno postoje samo u korenu stabla sklapanja. Problem je sledeći, za zadani kritični vremenski trenutak t do kog rezervni deo mora biti sklopljen, odrediti optimalne vremenske trenutke t^*svakog od podsklopova finalnog rezervnog dela, kada treba započeti njegovo sklapanje. Optimalni vremenski trenutak t^*, koji odgovara finalnom rezervnom delu odnosno, korenu stabla sklapanja se odredjuje rešavanjem klasičnog newsboy problema, a to je praktično kritični vremenski trenutak za simultanu raspoloživost njegovih direktnih podsklopova. Dakle možemo pretpostaviti da stablo sklapanja rezervnog dela u korenu ima postavljene između ostalih sledeće veličine: optimalni vremenski trenutak t^* i jedinične troškove kašnjenja b. Koren ovakvog stabla sklapanja je ulazni parametar za rekurzivnu proceduru koja rešava problem odredjivanja optimalnih vremenskih trenutaka za stablo sklapanja proizvoljne visine. Koraci pomenute rekurzivne procedure su:

- Ako je dati podsklop koren nekog podstabla sklapanja visine jedan, odnosno ako podsklop nema svojih podsklopova procedura je završena.

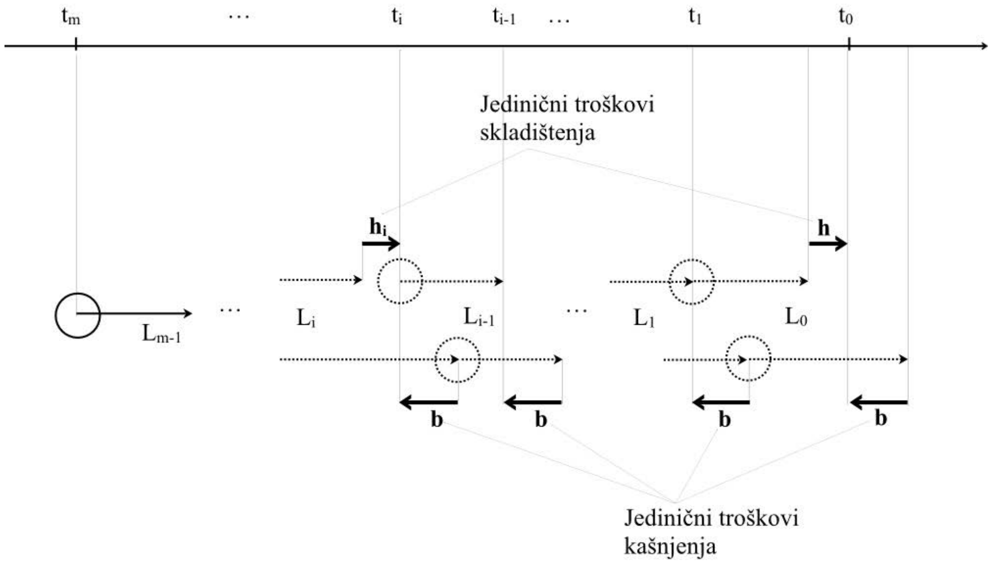
- Za dati podsklop koji je koren nekog podstabla sklapanja, koristeći između ostalog optimalni vremenski trenutak t^* i jedinične troškove kašnjenja $\hat{b}$, rešavanjem sistema jednačina (3.36) odredjuju se optimalni vremenski trenuci $t_1^*, \ldots, t_n^*$ koji odgovaraju njegovim direktnim podsklopovima. Slično koristeći jednačine (3.45) odredjuju se jedinični troškovi kašnjenja $\hat{b_1}, \ldots, \hat{b_n}$ koji odgovaraju njegovim direktnim podsklopovima.

[1] Dovoljno je čvoru pridružiti funkciju raspodele F odnosno gustine f slučajne promenljive L

– Rekurzivno se izvršava procedura na svakom direktnom podsklopu.

Rekurzivnost se može na jednostavan način izbeći. Stablo sklapanja se prolazi po nivoima počevši od korena.

Primer 3.3.3. Razmotrimo problem proračuna vremena u slučaju da postoji samo jedna linija sklapanja, odnosno svako podstablo sklapanja ima tačno jednu granu, u predloženom modelu $n = 1$. Ovaj process se opisuje stablom koje je lanac. Neka je reč o lancu dužine $m > 1$ (3.8).



Slika 3.8: Jedna linija sklapanja

Primetili smo da se u slučaju jedne linije sklapanja troškovi kašnjenja sa finalnog sklopa prenose na podsklopove, odnosno $\hat{b_i} = \hat{b} = b$, $i = 0, 1, 2, \ldots, m-1$. U pojedinim fazama sklapanja $i = 0, 1, \ldots, m-1$ ukupni troškovi su:

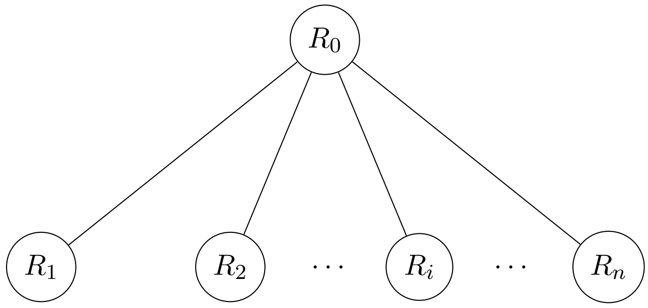
$$C_i = h_i(t_i - T_i)^+ + b(T_i - t_0^*)^+$$

a ovo je klasičan newsboy model, čije je rešenje $t_{i+1} = t_i^* = t_i - F_i^{-1}\left(\frac{b}{b+h_i}\right)$. Konačno rešenje je:

$$\begin{aligned} t_m &= t_{m-1} - F_{m-1}^{-1}\left(\frac{b}{b+h_{m-1}}\right) = t_{m-2} - F_{m-2}^{-1}\left(\frac{b}{b+h_{m-2}}\right) - F_{m-1}^{-1}\left(\frac{b}{b+h_{m-1}}\right) = \ldots \\ &= t_0 - \sum_{i=0}^{m-1} F_i^{-1}\left(\frac{b}{b+h_i}\right) \end{aligned}$$

3.4 Simultani zahtev II

Neka je proces sklapanja rezervnog dela opisan stablom visine dva (3.9).



Slika 3.9: Stablo sklapanja visine 2

Za dati vremenski trenutak t_0 kada finalni rezervni deo treba da bude sklopljen, treba odrediti optimalni vremenski trenutak t_0^*, kada treba započeti proces sklapanja rezervnog dela. Neka su poznati jedinični troškovi skladištenja h, jedinični troškovi kašnjenja b i funkcija raspodele F slučajne promenljive L koja predstavlja vreme sklapanja rezervnog dela. Ovo je klasičan newsboy problem i optimalni vremenski trenutak se određuje na sledeći način:

$$t_0^* = t_0 - F^{-1}\left(\frac{b}{b+h}\right) \tag{3.46}$$

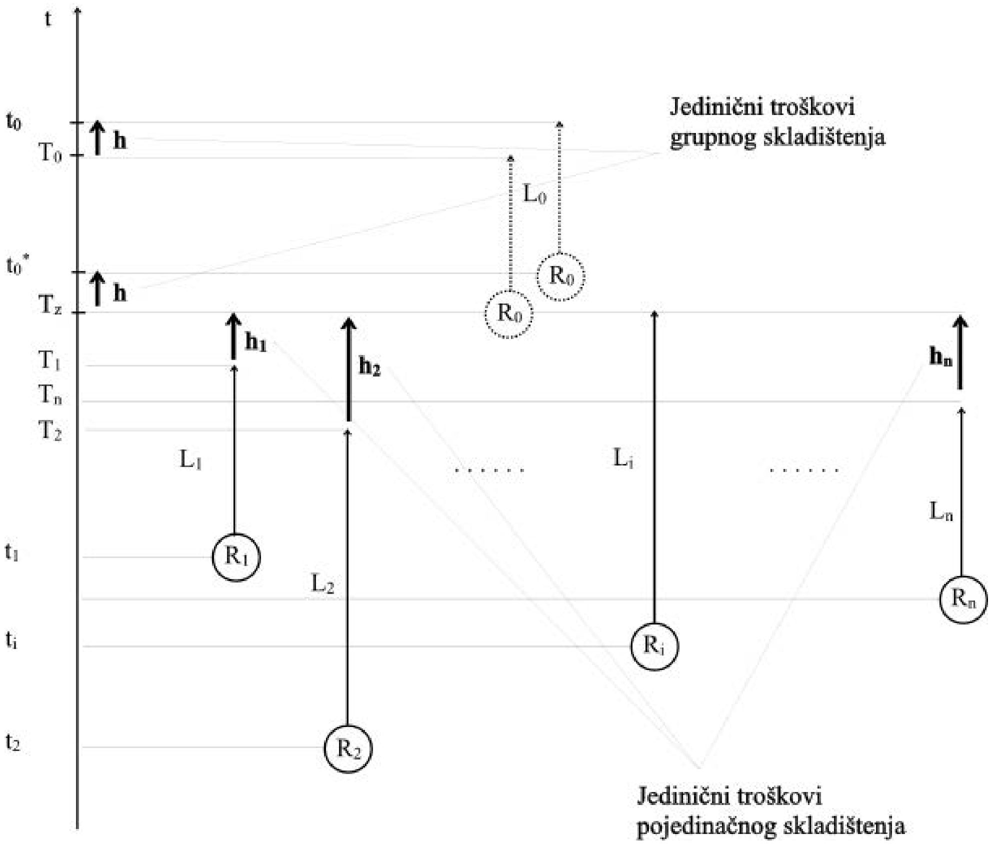
Da bi se proces sklapanja odvijao po optimalnoj putanji, očigledno svi podsklopovi rezervnog dela moraju biti raspoloživi u vremenskom trenutku t_0^*. Na ovaj način dolazimo do problema simultane raspoloživosti. Vreme sklapanja i-tog $i = 1, 2, \ldots, n$ podsklopa je neprekidna slučajna promenljiva L_i. Od trenutka kada je podsklop sklopljen do trenutka kada se započinje sklapanje nadsklopa, podsklop se skladišti. Jedinični troškovi skladištenja i-tog podsklopa su h_i. Treba odrediti optimalne vremenske trenutke $t_1^*, \ldots, t_n^*$, kada treba započeti proces sklapanja svakog pojedinačog podsklopa. Pretpostavimo da sklapanje nadsklopa ne može započeti pre nego što su svi podsklopovi sklopljeni. Funkcija ukupnih troškova je (3.20):

$$\begin{aligned} C &= C(t_1, \ldots, t_n, t_0^*, L_1, \ldots, L_n) \\ &= \hat{h}(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) + b(T_z - t_0^*)^+ \end{aligned} \tag{3.47}$$

gde su sa $\hat{h}$ označeni jedinični troškovi grupnog skladištenja. U prethodnoj analizi (3.3) pretpostavili smo da se podsklopovi pored toga što se skladište do trenutka T_z kada može započeti faza sklapanja nadsklopa, pojedinačno skladiste do predviđenog roka t_0^* ako je sklapanje svih podsklopova završeno pre utvrđenog roka odnosno, $\hat{h} = \sum_{i=1}^{n} h_i$ za $T_z < t_0^*$. Uzimajući ove pretpostavke u obzir funkcija ukupnih troškova je:

$$
\begin{aligned}
C &= \sum_{i=1}^{n} h_i(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) + b(T_z - t_0^*)^+ \\
&= \sum_{i=1}^{n} h_i(t_0^* - T_i) + \left(b + \sum_{i=1}^{n} h_i\right)(T_z - t_0^*)^+
\end{aligned} \tag{3.48}
$$

Pretpostavimo sada drugu krajnost, nema nepotrebnog skladištenja, odnosno u trenutku T_z kada su svi podsklopovi raspoloživi započinje sklapanje nadsklopa (3.10):



Slika 3.10: Grupno skladištenje

Postavlja se pitanje kako ovu situaciju ugraditi u funkciju ukupnih troškova. Kako sklapanje nadsklopa započinje pre predviđenog roka ($T_z < t_0^*$) opravdana je pretpostavka da će nadsklop biti sklopljen pre utvrđenog roka, odnosno prevremeno aktiviranje faze sklapanja nadsklopa će proizvesti troškove skladištenja nadsklopa $h(t_0^* - T_z)$. Dakle pretpostavimo:

$$\hat{h}(t_0^* - T_z) = h(t_0^* - T_z) \tag{3.49}$$

odnosno jedinični troškovi grupnog skladištenja $\hat{h}$ na neposrednim podsklopovima finalnog sklopa su jednaki jediničnim troškovima skladištenja h finalnog sklopa:

$$\hat{h} = \hat{h}(h_1, h_2, \ldots, h_n, L_1, \ldots, L_n, h) = h \tag{3.50}$$

Pretpostavimo još da su sumarni troskovi skladištenja podsklopova veći od troškova skladištenja nadsklopa odnosno:

$$\sum_{i=1}^{n} h_i \geq h \tag{3.51}$$

Dakle uz ove pretpostavke funkcija ukupnih troskova je:

$$C = h(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) + b(T_z - t_0^*)^+ \tag{3.52}$$

Transformišimo gornji izraz u pogodniji oblik, iz koga će se dobiti oblik funkcije očekivanih troškova iz koga se relativno lako može utvrditi njena konveksnost [3] i ustanoviti egzistencija njenog globalnog minimuma:

$$\begin{aligned}
C &= h(t_0^* - T_z) + \sum_{i=1}^{n} h_i(T_z - T_i) + (b+h)(T_z - t_0^*)^+ \\
&= h(t_0^* - T_z) + \sum_{i=1}^{n} h_i\left((T_z - t_0^*) - (T_i - t_0^*)\right) + (b+h)(T_z - t_0^*)^+ \\
&= \left(\sum_{i=1}^{n} h_i - h\right)(T_z - t_0^*) - \sum_{i=1}^{n} h_i(T_i - t_0^*) + (b+h)(T_z - t_0^*)^+ \\
&= \left(\sum_{i=1}^{n} h_i - h\right)(T_z - t_0^*) + \sum_{i=1}^{n} h_i(t_0^* - T_i) + (b+h)(T_z - t_0^*)^+
\end{aligned} \tag{3.53}$$

kako je $T_i = t_i + L_i$, odnosno $T_z = \max_i \{T_i\}$, dobijamo:

$$\begin{aligned}
C &= \left(\sum_{i=1}^{n} h_i - h\right)\left(\max_i \{t_i + L_i - t_0^*\}\right) + \sum_{i=1}^{n} h_i(t_0^* - t_i - L_i) \\
&\quad + (b+h)\left(\max_i \{t_i + L_i - t_0^*\}\right)^+
\end{aligned} \tag{3.54}$$

na kraju, označimo $\tau_i = t_0^* - t_i,\ i = 1, 2, \ldots, n$, pa je funkcija ukupnih troškova:

$$\begin{aligned}
C &= C(\tau_1, \tau_2, \ldots, \tau_n, L_1, \ldots, L_n) \\
&= \left(\sum_{i=1}^{n} h_i - h\right)\max_i \{L_i - \tau_i\} + \sum_{i=1}^{n} h_i(\tau_i - L_i) \\
&\quad + (b+h)\left(\max_i \{L_i - \tau_i\}\right)^+
\end{aligned} \tag{3.55}$$

Očekivani ukupni troškovi $Z = E(C) = Z(\tau_1, \tau_2, \dots, \tau_n)$ su:

$$\begin{aligned} Z = & \left(\sum_{i=1}^{n} h_i - h\right) E\left(\max_i \{L_i - \tau_i\}\right) + \sum_{i=1}^{n} h_i\left(\tau_i - E(L_i)\right) \\ & + (b+h)E\left((\max_i \{L_i - \tau_i\})^+\right) \end{aligned} \tag{3.56}$$

Na isti način kao što je to već urađeno u prethodnoj analizi (3.3.1) možemo zaključiti da funkcija očekivanih ukupnih troškova Z ima globalni minimum u konačnoj tački.

Teorema 3.4.1. *Funkcija očekivanih ukupnih troškova $Z = E(C) = Z(\tau_1, \tau_2, \dots, \tau_n)$ data sa (3.56) ima globalni minimum u konačnoj tački $(\tau_1^*, \tau_2^*, \dots, \tau_n^*)$.*

Dokaz. *Pokažimo prvo da je funkcija ukupnih troškova Z data sa (3.56) konveksna funkcija svojih argumenata $\tau_1, \tau_2, \dots, \tau_n$. Drugi sabirak*

$$\sum_{i=1}^{n} h_i\left(\tau_i - E(L_i)\right) \tag{3.57}$$

je linearna dakle konveksna funkcija argumenata $\tau_1, \tau_2, \dots, \tau_n$ dok konveksnost prvog odnosno trećeg sabirka:

$$\left(\sum_{i=1}^{n} h_i - h\right) E\left(\max_i \{L_i - \tau_i\}\right) \tag{3.58}$$

$$(b+h)E\left((\max_i \{L_i - \tau_i\})^+\right) \tag{3.59}$$

sledi iz iz monotonosti i linearnosti očekivanja i konveksnosti maksimuma konveksnih funkcija. Dakle data funkcija ukupnih troškova Z je konveksna funkcija kao zbir konveksnih funkcija. Sa druge strane možemo zaključiti sledeće:

- $\tau_i \to +\infty \Rightarrow Z \sim h_i\tau_i \to +\infty$
- $$\begin{aligned} \tau_i \to -\infty \Rightarrow Z \sim & \quad -\tau_i\left(\sum_{i=1}^{n} h_i - h\right) + h_i\tau_i - \tau_i(b+h) \\ = & \quad -\tau_i\left(b + \left(\sum_{i=1}^{n} h_i\right) - h_i\right) \\ = & \quad -\tau_i\left(b + \sum_{j\neq i}^{n} h_j\right) \to +\infty \end{aligned}$$

odnosno:

$$lim_{\tau_i \to \pm\infty} Z = +\infty$$

Navedene osobine konveksne funkcije očekivanih troškova obezbeđuju egzistenciju lokalnog minimuma u konačnoj tački $(\tau_1^*, \tau_2^*, \dots, \tau_n^*)$, koji će biti i globalni minimum [2] [9] [26].

◇

Razvijmo dalje funkciju očekivanih troškova. Pre svega neophodno je odrediti funkciju raspodele slučajne promenljive:

$$D = \max_i \{L_i - \tau_i\} \tag{3.60}$$

odnosno:

$$D^+ = \max \left\{ \max_i \{L_i - \tau_i\}, 0 \right\} \tag{3.61}$$

Kako je za određivanje očekivanja nenegativne slučajne promenljive dovoljno poznavanje njene funkcije raspodele [1], iskoristimo osobinu $D = D^+ - D^-$ i imaćemo:

$$\begin{aligned} Z &= \left(\sum_{i=1}^{n} h_i - h\right) E(D) + \sum_{i=1}^{n} h_i \left(\tau_i - E(L_i)\right) + (b+h)E(D^+) \\ &= \left(\sum_{i=1}^{n} h_i - h\right) E(D^+ - D^-) + \sum_{i=1}^{n} h_i \left(\tau_i - E(L_i)\right) + (b+h)E(D^+) \\ &= \left(h - \sum_{i=1}^{n} h_i\right) E(D^-) + \sum_{i=1}^{n} h_i \left(\tau_i - E(L_i)\right) + \left(b + \sum_{i=1}^{n} h_i\right) E(D^+) \end{aligned} \tag{3.62}$$

Analogno kao što je to urađeno za slučajnu promenljivu D^+ nađimo očekivanje slučajne promenljive $D^- = (\max_i \{L_i - \tau_i\})^- = \max \left\{ -\max_i \{L_i - \tau_i\}, 0 \right\}$. Kako se radi o nenegativnoj slučajnoj promenljivoj dovoljno je odrediti $P(D^- \geq t)$. Neka je $t > 0$, imaćemo:

$$\begin{aligned} P(D^- \geq t) &= P(\max \left\{ -\max_i \{L_i - \tau_i\}, 0 \right\} \geq t) = P\left(-\max_i \{L_i - \tau_i\} \geq t\right) \\ &= P\left(\max_i \{L_i - \tau_i\} \leq -t\right) = P\left(\bigcap_i (L_i - \tau_i \leq -t)\right) \\ &= \prod_i P(L_i \leq \tau_i - t) = \prod_i F_i(\tau_i - t) \end{aligned} \tag{3.63}$$

pa je očekivanje nenegativne slučajne promenljive D^-:

$$E(D^-) = \int_0^{+\infty} P(D^- \geq t)\mathrm{d}t = \int_0^{+\infty} \prod_i F_i(\tau_i - t)\mathrm{d}t \tag{3.64}$$

Sada funkcija očekivanih ukupnih troškova Z, ima oblik:

$$
\begin{aligned}
Z &= \left(h - \sum_{i=1}^{n} h_i\right) \int_0^{+\infty} \prod_i F_i(\tau_i - t)\mathrm{d}t + \sum_{i=1}^{n} h_i \left(\tau_i - E(L_i)\right) \\
&+ \left(b + \sum_{i=1}^{n} h_i\right) \int_0^{+\infty} \left(1 - \prod_i F_i(t + \tau_i)\right) \mathrm{d}t
\end{aligned} \tag{3.65}
$$

Stacionarna tačka funkcije Z, je rešenje sledećeg sistema jednačina:

$$
\begin{aligned}
\frac{\partial Z}{\partial \tau_i} &= \left(h - \sum_{i=1}^{n} h_i\right) \frac{\partial}{\partial \tau_i} \left(\int_0^{+\infty} \prod_i F_i(\tau_i - t)\mathrm{d}t \right) + h_i \\
&+ \left(b + \sum_{i=1}^{n} h_i\right) \frac{\partial}{\partial \tau_i} \left(\int_0^{+\infty} \left(1 - \prod_i F_i(t + \tau_i)\right) \mathrm{d}t \right) = 0,\ i = 1, 2, .., n
\end{aligned} \tag{3.66}
$$

dalje pretpostavimo uslove o funkcijama raspodele F_i, $i = 1, 2, \ldots, n$ tako da parcijalni izvod i integral komutiraju [25] i dobićemo:

$$
\begin{aligned}
\frac{\partial Z}{\partial \tau_i} &= \left(h - \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} \frac{\partial}{\partial \tau_i} \prod_i F_i(\tau_i - t)\mathrm{d}t \right) + h_i \\
&+ \left(b + \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} \frac{\partial}{\partial \tau_i} \left(1 - \prod_i F_i(t + \tau_i)\right) \mathrm{d}t \right) = 0,\ i = 1, 2, .., n \\
&= \left(h - \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} f_i(\tau_i - t) \prod_{j \neq i} F_j(\tau_j - t)\mathrm{d}t \right) + h_i \\
&- \left(b + \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} f_i(t + \tau_i) \prod_{j \neq i} F_j(t + \tau_j)\mathrm{d}t \right) = 0,\ i = 1, 2, .., n
\end{aligned}
$$

Konačno stacionarne tačke su rešenja sledećeg sistema nelinearnih jednačina:

$$
\begin{aligned}
h_i &= \left(-h + \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} f_i(\tau_i - t) \prod_{j \neq i} F_j(\tau_j - t)\mathrm{d}t \right) \\
&+ \left(b + \sum_{i=1}^{n} h_i\right) \left(\int_0^{+\infty} f_i(t + \tau_i) \prod_{j \neq i} F_j(t + \tau_j)\mathrm{d}t \right),\ i = 1, 2, .., n
\end{aligned} \tag{3.67}
$$

Dobijeni sistem jednačina se može rešiti numeričkim metodama i neka su $\tau_1^*, \tau_2^*, \ldots, \tau_n^*$, ove vrednosti. Na ovaj način su određeni optimalni vremenski trenuci $t_i^* = t_0^* - \tau_i$, $i = 1, \ldots, n$, kada treba započeti sklapanje i-tog podsklopa tako da su očekivani ukupni troskovi minimalni.

Primer 3.4.1. Primetimo sledeće, neka je $n = 1$ tada se stacionarna tačka, odnosno optimalno rešenje τ_1^* dobija kao rešenje jednačine:

$$(-h + h_1)\int_0^{+\infty} f_1(\tau_1 - t)\mathrm{d}t + (b + h_1)\int_0^{+\infty} f_1(t + \tau_1)\mathrm{d}t = h_1 \tag{3.68}$$

kako je:

$$\int_0^{+\infty} f(\tau - t)\mathrm{d}t = \int_{-\infty}^{\tau} f(u)\mathrm{d}u = \int_{-\infty}^{\tau} f(u)\mathrm{d}u = F(\tau)$$

odnosno:

$$\int_0^{+\infty} f(t + \tau)\mathrm{d}t = \int_{\tau}^{+\infty} f(u)\mathrm{d}u = 1 - F(\tau)$$

gornja jednačina postaje:

$$(-h + h_1)F_1(\tau_1) + (b + h_1)(1 - F_1(\tau_1)) = h_1 \tag{3.69}$$

$$-hF_1(\tau_1) + b + h_1 - bF_1(\tau_1) = h_1 \tag{3.70}$$

$$F_1(\tau_1) = \frac{b}{b + h} \tag{3.71}$$

i njeno rešenje je:

$$\tau_1^* = F_1^{-1}\left(\frac{b}{b + h}\right) \tag{3.72}$$

što je rešenje klasičnog newsboy problema na nivou podsklopa. Ovo je očekivan rezultat jer su pretpostavke da će se dinamika sklapanja podsklopa preslikati na završnu fazu pa su svi troškovi preuzeti sa finalnog sklopa. Dakle predloženi model je uopštenje newsboy modela za slučaj da postoji jedna linija sklapanja rezervnog dela.

3.4.1 Stablo sklapanja sa više nivoa II

Neka je proces sklapanja rezervnog dela opisan stablom visine veće od dva. Na opisani način mi možemo odrediti kritična vremena $t_1^*, \ldots, t_n^*$ na nivou neposrednih podsklopova finalnog sklopa. Uočimo podstablo visine dva čiji je koren jedan od neposrednih podsklopova finalnog sklopa, neka je to i-ti ($1 \le i \le n$) podsklop. Formirajmo funkciju ukupnih troškova (3.54) koji odgovaraju uočenom podstablu sklapanja:

$$\begin{aligned} C_i \quad = \quad & \left(\sum_j (h_i)_j - \hat{h}_i\right)((T_i)_z - t_i^*) + \sum_j (h_i)_j\,(t_i^* - (T_i)_j) \\ & +(\hat{b}_i + \hat{h}_i)\,((T_i)_z - t_i^*)^+ \end{aligned} \tag{3.73}$$

gde su:

- t_i^* - utvrđeni rok za simultanu raspoloživost podsklopova i-tog podsklopa
- $(T_i)_j$ - trenutak raspoloživosti pojedinih podsklopova i-tog podsklopa
- $(T_i)_z$ - simultana raspoloživost podsklopova i-tog podsklopa
- $(h_i)_j$ - jedinični troškovi skladištenja pojedinih podsklopova i-tog podsklopa
- $\hat{h}_i$ - grupni troškovi skladištenja i-tog podsklopa
- $\hat{b}_i$ - grupni troškovi kašnjenja i-tog podsklopa

Da bismo navedni postupak sproveli na ovom stablu sklapanja neophodno je proceniti jedinične troškove grupnog skladištenja $\hat{h}_i$, odnosno jedinične troškove kašnjenja $\hat{b}_i$ i-tog ($1 \leq i \leq n$) podsklopa koji je u korenu ovog podstabla. Procenu jediničnih troskova kašnjenja i-tog podsklopa smo već ustanovili (3.45):

$$\hat{b}_i = \hat{b}\left(\prod_{j\neq i} F_{L_j - L_i}(t_i^* - t_j^*)\right)$$

gde su sa $\hat{b} = b$ označeni jedinični troškovi kašnjenja nadsklopa koji je u ovom slučaju finalni sklop. Na sličan način kao što je to izvedeno sa jediničnim troškovima kašnjenja $\hat{b}_i$, možemo proceniti i jedinične troškove grupnog skladištenja $\hat{h}_i$ za podsklopove i-tog ($1 \leq i \leq n$) podsklopa. Dakle jedno rešenje je da se izvrši raspodela troskova grupnog skladištenja $\hat{h} = h$ podsklopova finalnog sklopa, na troškove grupnog skladištenja za podsklopove njegovih neposrednih podsklopova. Analizirajmo troškove skladištenja:

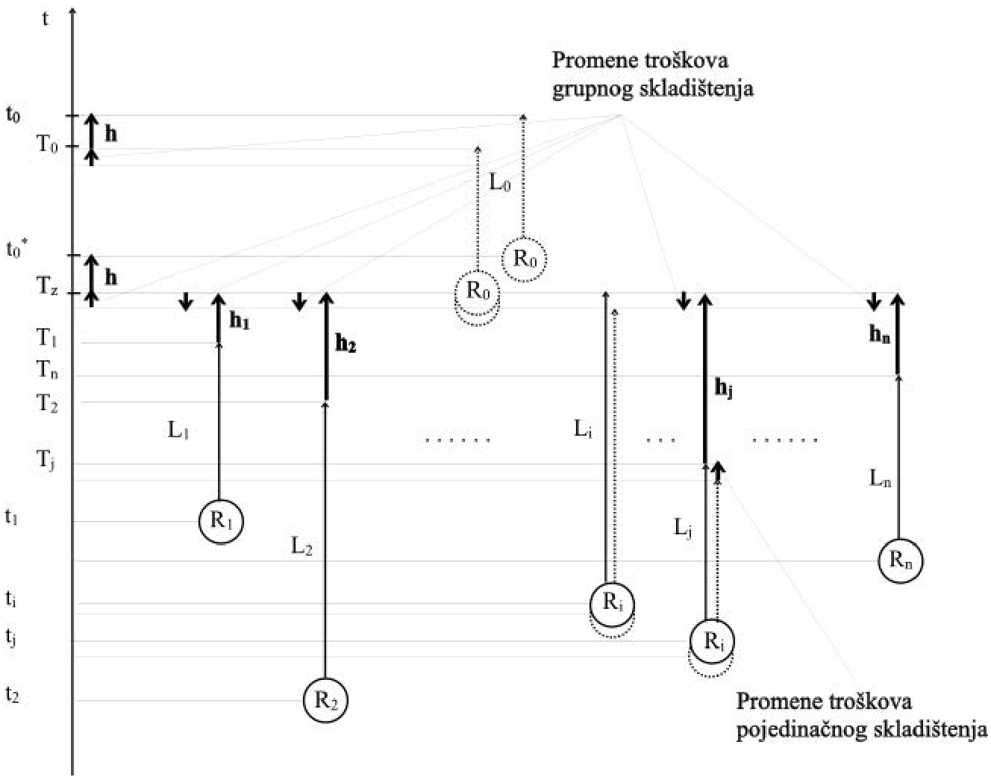
$$H = \hat{h}(t_0^* - T_z)^+ + \sum_{i=1}^{n} h_i(T_z - T_i) \tag{3.74}$$

Svaki od podsklopova ima svoj udeo u ukupnim troškovima skladištenja koji pored troškova pojedinačnog skladištenja $H_i = h_i(T_z - T_i)$ uključuje i troškove vezane za grupno skladištenje $\hat{H} = \hat{h}(t_0^* - T_z)^+$. Ukupnost ovih troškova čini troškove grupnog skladištenja $\hat{h}_i$ podsklopova i-tog podsklopa. Neka se sklapanje i-tog ($1 \leq i \leq n$) podsklopa pokrene u nekom prethodnom trenutku $t_i - \Delta t_i$, gde je Δt_i proizvoljno mali vremenski period. Pretpostavimo $T_z \leq t_0^*$ (u suprotnom nema troškova skladištenja niti ikakve promene za dovoljno malo Δt_i) (3.11):
U situaciji kao na slici (3.11) promena troškova skladištenja je:

$$\Delta_i H = \Delta t_i \begin{cases} \hat{h} - \sum\limits_{j\neq i} h_j; & T_i = T_z \\ h_i; & T_i < T_z \end{cases} \tag{3.75}$$

Očekivani jedinični troškovi skladištenja $E\left(\dfrac{\Delta_i H}{\Delta t_i}\right)$ su:

$$E\left(\frac{\Delta_i H}{\Delta t_i}\right) = \left(\hat{h} - \sum_{j\neq i} h_j\right) P(T_i = T_z) + h_i P(T_i < T_z) \tag{3.76}$$



Slika 3.11: Promena troškova skladištenja

prvi sabirak odgovara učešću u jediničnim troškovima grupnog skladištenja $\hat{h}$, a drugi odgovara jediničnim troškovima pojedinačnog skladištenja i-tog podsklopa. Dakle opravdano je pretpostaviti:

$$\begin{aligned}\hat{h}_i &= \left(\hat{h} - \sum_{j\neq i} h_j\right) P(T_i = T_z) + h_i P(T_i < T_z) \\ &= \left(\hat{h} - \sum_{j} h_j\right) P(T_i = T_z) + h_i\end{aligned}$$

Primer 3.4.2. Primetimo sledeće, u slučaju da nadsklop ima tačno jedan podsklop, tj. $n = 1$, kako je $T_1 = T_z$ biće:

$$\hat{h}_1 = \left(\hat{h} - h_1\right) \cdot 1 + h_1 = \hat{h} \tag{3.77}$$

odnosno u slučaju jedne linije sklapanja troškovi skladištenja se spuštaju niz lanac sklapanja. Slično, ako pretpostavimo $\hat{h} = \sum_i h_i$, odnosno ako pretpostavimo da se podsklopovi pojedinačno skladište do predviđenog roka t_0^* (3.3), imaćemo:

$$\hat{h}_i = \left(\hat{h} - \sum_{j} h_j\right) P(T_i = T_z) + h_i = 0 \cdot P(T_i = T_z) + h_i = h_i \tag{3.78}$$

odnosno grupni troškovi skladištenja za podsklopove i-tog podsklopa su jedinični troškovi skladištenja i-tog podsklopa. Pretpostavimo dalje, da je sklapanje svakog od podsklopova završeno ranije odnosno neka je završeno u trenutku $T_i - \Delta t$, u tom slučaju će sklapanje nadsklopa početi u vremenskom trenutku $T_z - \Delta t$, što će proizvesti promenu troškova skladištenja nadsklopa za iznos $\hat{h}\Delta t$. Pokazaćemo da su ovi troškovi obuhvaćeni modelom na nivou podsklopova:

$$\begin{aligned}
\sum_i \Delta_i H &= \sum_i \left(\left(\hat{h} - \sum_j h_j \right) P(T_i = T_z) + h_i \right) \Delta t \\
&= \left(\left(\hat{h} - \sum_j h_j \right) \sum_i P(T_i = T_z) + \sum_i h_i \right) \Delta t \\
&= \left(\left(\hat{h} - \sum_j h_j \right) \cdot 1 + \sum_i h_i \right) \Delta t \\
&= \hat{h}\Delta t
\end{aligned}$$

Drugim rečima predloženi model lokalno tretira troškove koji se mogu pojaviti u narednoj fazi procesa sklapanja, a koji u toj fazi ne mogu biti tretirani. Na ovaj način smo pokazali da predloženi model na adekvatan način pokriva navedene granične slučajeve.

Analogno kako je to urađeno za jedinične troškove kašnjenja na nivou podsklopova, jedinični troškovi grupnog skladištenja $\hat{h}_i$ podsklopova i-tog posklopa su:

$$\hat{h}_i = \left(\hat{h} - \sum_j h_j \right) \prod_{j \neq i} F_{L_j - L_i}(t_i^* - t_j^*) + h_i \tag{3.79}$$

Neka su poznati grupni troškovi skladištenja odnosno troškovi kašnjenja svakog pojedinačnog podsklopa, sada mozemo formulisati proceduru proračuna vremena na svakom od podsklopova datog rezervnog dela. Odgovarajuća funkcija troškova za podsklopove i-tog podsklopa je:

$$\begin{aligned}
C_i &= \left(\sum_j (h_i)_j - \hat{h}_i \right) ((T_i)_z - t_i^*) + \sum_j (h_i)_j (t_i^* - (T_i)_j) \\
&\quad + (\hat{b}_i + \hat{h}_i)((T_i)_z - t_i^*)^+
\end{aligned} \tag{3.80}$$

dok su očekivani ukupni troškovi $Z_i = E(C_i)$ na i-tom podsklopu:

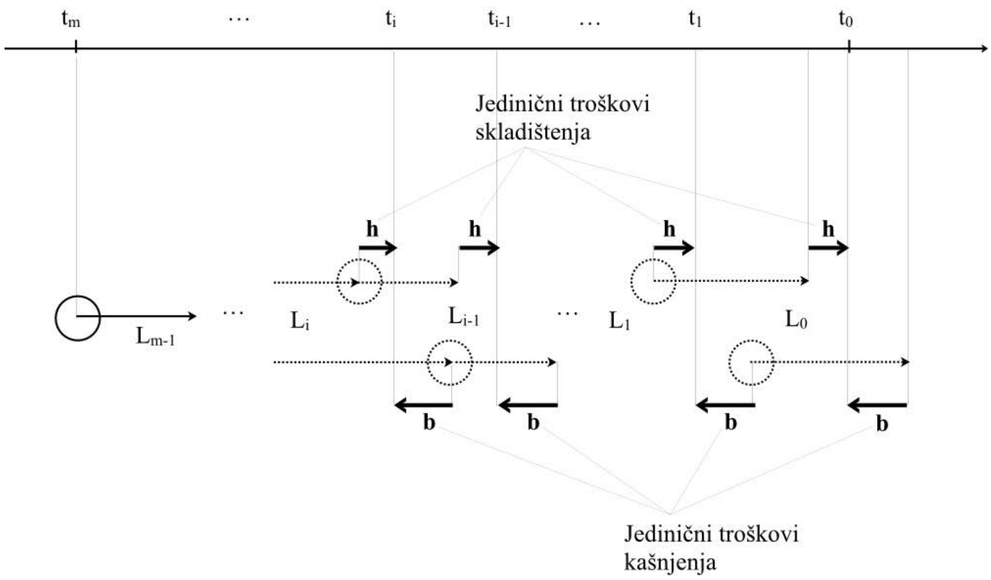
$$\begin{aligned}
Z_i &= \left(\sum_j (h_i)_j - \hat{h}_i \right) E\left((T_i)_z - t_i^*\right) + \sum_j (h_i)_j (t_i^* - E(T_i)_j) \\
&\quad + (\hat{b}_i + \hat{h}_i) E(((T_i)_z - t_i^*)^+)
\end{aligned} \tag{3.81}$$

Da bi predloženi iterativni postupak konvergirao, neophodno je pokazati da je Z_i konveksna funkcija [3], odnosno dovoljno je dokazati da je $\sum_j (h_i)_j \geq \hat{h}_i$. Ispunjenost navedenog uslova direktno sledi iz pretpostavke da troškovi skladištenja nadsklopa nisu veći od troškova skladištenja posklopova $\hat{h} = h \leq \sum_i h_i$, odnosno $h_i \leq \sum_j (h_i)_j$:

$$\begin{aligned}
\hat{h}_i &= \left(\hat{h} - \sum_j h_j\right) P(T_i = T_z) + h_i \\
&\leq \left(\sum_i h_j - \sum_j h_j\right) P(T_i = T_z) + h_i \\
&= h_i \leq \sum_j (h_i)_j
\end{aligned}$$

Na ovaj način možemo produžiti rešenje problema proračuna vremena na procese sklapanja rezervnog dela koji se opisuju stablom proizvoljne visine. Procedura kojom bi se rešavao ovaj problem je identična navedenoj proceduri za proračun vremena kada se podsklopovi skladište do utvrđenog roka (3.3.1), s tom razlikom što se pored izračunavanja jediničnih troškova kašnjenja podsklopa izračunavaju i jedinični grupni troskovi skladištenja podsklopa jednačinama (3.79). Jasno optimalna rešenja se dobijaju rešavanjem sistema jednačina (3.67).

Primer 3.4.3. Razmotrimo problem proračuna vremena u slučaju da postoji samo jedna linija sklapanja, odnosno svako podstablo sklapanja ima tačno jednu granu, u predloženom modelu $n = 1$. Ovaj process se opisuje stablom koje je lanac. Neka je reč o lancu dužine $m > 1$ (3.12)



Slika 3.12: Jedna linija sklapanja

Primetili smo da se u slučaju jedne linije sklapanja troškovi sa finalnog sklopa prenose na podsklopove, odnosno $\hat{h}_i = \hat{h} = h$ i $\hat{b}_i = \hat{b} = b$, $i = 0, 1, 2, \ldots, n-1$. U pojedinim fazama sklapanja $i = 0, 1, \ldots, n-1$ ukupni troškovi su:

$$C_i = h(t_i - T_i)^+ + b(T_i - t_0^*)^+$$

a ovo je klasičan newsboy model , čije je rešenje $t_{i+1} = t_i^* = t_i - F_i^{-1}\left(\frac{b}{b+h}\right)$. Konačno rešenje je:

$$\begin{aligned} t_m &= t_{m-1} - F_{m-1}^{-1}\left(\frac{b}{b+h}\right) = t_{m-2} - F_{m-2}^{-1}\left(\frac{b}{b+h}\right) - F_{m-1}^{-1}\left(\frac{b}{b+h}\right) = \dots \\ &= t_0 - \sum_{i=0}^{m-1} F_i^{-1}\left(\frac{b}{b+h}\right) \end{aligned}$$

Glava 4

Sastavnice

4.1 Uvodna razmatranja

Vazduhoplovi su savremena saobraćajna sredstva vrhunske tehnologije sa velikim brojem komponenti i kompleksnom strukturom. Pojedine komponente vazduhoplova se koriste na različit način, trpe različita opterećenja i održavaju se po programu koji je njima najbolje prilagođen. Jedan od načina savladavanja složenosti je takozvana hijerarhijska dekompozicija. Složeni koncept se na jednom nivou apstrakcije posmatra kao jedinstvena celina. Na nižem nivou apstrakcije se posmatra kao koncept koji se sastoji od delova (komponenti). Uzastopnom primenom opisanog postupka dobija se takozvana hijerarhijska *sastavnica*. Dakle jedan od bitnih koncepata u sistemu održavanja vazduhoplova jeste sastavnica. Sastavnica je najprostije rečeno [4], lista komponenti koji su sastavni delovi posmatranog objekta. Pri tome se vodi računa o vezama između sastavnih komponenti, pa je u osnovi sastavnice hijerarhijska struktura podataka. Bez gubitka opštosti, u daljem tekstu ćemo pod sastavnicom podrazumevati fizičku strukturu posmatranog objekta, pa ćemo koristiti termine sklop, podsklop, nadsklop koji istovremeno označavaju i uzajamni odnos komponenata hijerarhije. Dakle podrazumevaćemo sastavnice sklopova koje se dobijaju hijerarhijskom dekompozicijom vazduhoplova, nad kojima se izvode aktivnosti održavanja. Dalje ćemo podrazumevati da su aktivnosti održavanja zapravo aktivnosti ugradnje podsklopova u nadređeni sklop. U praksi se koriste dve vrste sastavnica:

– *modularna*

– *hijerarhijska*

Modularna sastavnica daje strukturu i veze elemanata samo u okviru jednog modula - sklopa. Pri tome nije važna struktura samih podsklopova. Hijerarhijska sastavnica daje kompletnu strukturu određenog sklopa uključujući i hijerarhiju strukture svih podsklopva. Bez obzira na vrstu i klasu sastavnica, projektovanje i funkcionalna upotreba sastavnica zasniva se na hijerarhijskoj strukturi podataka odnosno na strukturi podataka tipa *stablo*. Svaka sastavnica se može bijektivno preslikati na stablo. Čvor stabla može imati *podređene* i *nadređene* čvorove. Elementi sklopa, u inženjerskom smislu odgovaraju podređenim čvorovima. Elementi se ugrađuju u sklop - koji je predstavljen nadređenim čvorom.

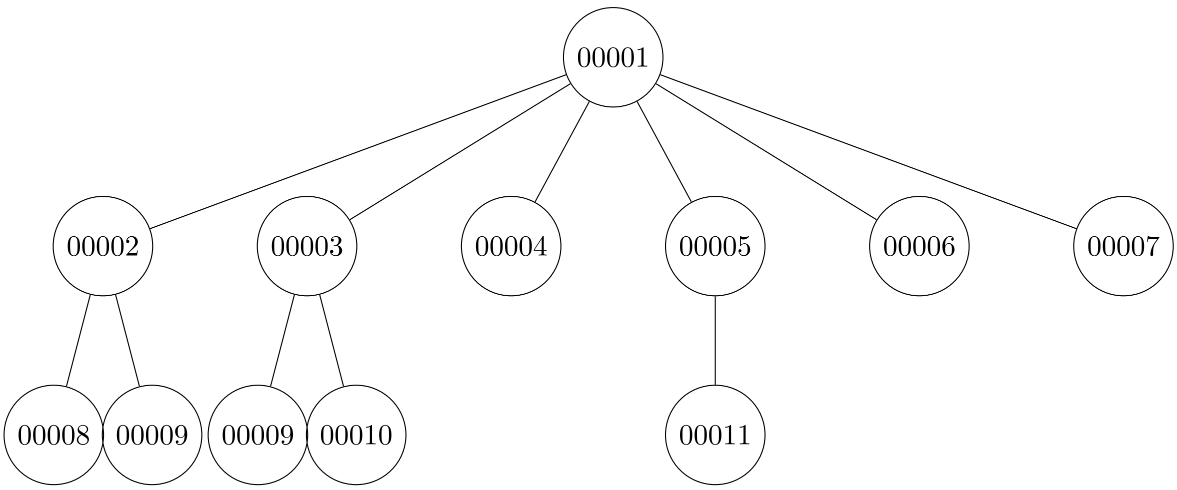
- *Koren stabla* ima samo podređene čvorove i on odgovara složenom sklopu
- *Listovi* nemaju podređene čvorove i odgovaraju materijalima ili složenim sklopovima koji nisu uključeni u sistem održavanja

U nastavku glave koristićemo stvarnu sastavnicu posebnog podsklopa kojim se održava konstantna temperatura hidraulične tečnosti u vazduhoplovima. Ovaj podsklop ima sledeće elemente:

Tabela 4.1: Spisak delova grejnog tela - radijatora hidraulike.

ident	**naziv**
00001	grejno telo
00002	posuda gornja I
00003	posuda gornja II
00004	posuda donja
00005	blok saća
00006	nosač natpisne ploče
00007	natpisna ploča
00008	lim posude gornje
00009	cev
00010	lim posude donje
00011	cevni zid

Hijerarhijska sastavnica sklopa datog u tabeli (4.1), dobijena na osnovu tehničkog crteža ima sledeći oblik (4.1):



Slika 4.1: Sastavnica radijatora hidraulike preuzeta iz proizvodnje.

Baza podataka koja je u osnovi informacionog sistema za podršku održavanja vazduhoplova između ostalog uključuje tabelu sklopova, podsklopova elemenata i materijala koja zajedno sa tabelom skladišta čini osnovu skladišnog poslovanja. Ne ulazeći u problematiku skladišta, izložićemo strukturu pomenutih tabela:

$$\begin{aligned}
sifra &::= \langle \text{CHAR}, 3, \text{FORMAT :999} \rangle \\
naziv &::= \langle \text{CHAR}, 40 \rangle \\
skladiste &::= \langle sifra, naziv, * \rangle \\
skladista &:= \{ skladiste : \text{PRIMARY KEY IS } sifra \}
\end{aligned} \tag{4.1}$$

Odgovarajuća tabela sklopova, podsklopova, elemenata i materijala će biti tabela artikala i imaće sledeću strukturu:

$$\begin{aligned}
ident &::= \langle \text{CHAR}, 5, \text{FORMAT :99999}, \text{AUTONUMBER} \rangle \\
skladiste &::= \text{skladista}.sifra \\
naziv &::= \langle \text{CHAR}, 40 \rangle \\
sifra &::= \langle \text{CHAR}, 12 \rangle \\
artikal &::= \langle ident, skladiste, naziv, sifra, * \rangle \\
artikli &:= \{ artikal : \text{PRIMARY KEY IS } ident \}
\end{aligned} \tag{4.2}$$

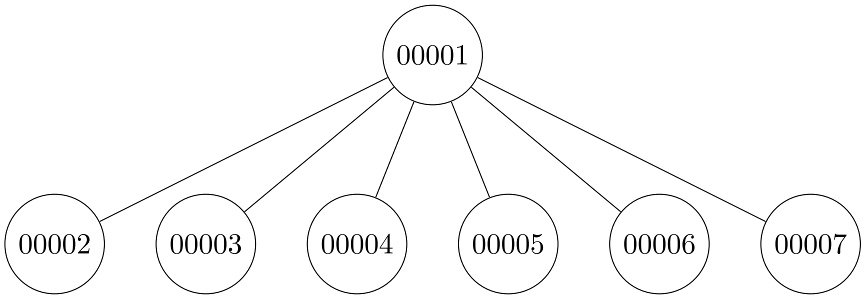
Posebno komponenetama podsklopa (4.1) odgovara niz zapisa tabele artikli (4.2) sledećeg oblika:

$$P_{00001} = \left\{ \begin{array}{ll}
\langle 00001, ., \text{grejno telo}, . \rangle, & \langle 00002, ., \text{posuda gornja I}, . \rangle, \\
\langle 00003, ., \text{posuda gornja II}, . \rangle, & \langle 00004, ., \text{posuda donja}, . \rangle, \\
\langle 00005, ., \text{blok saća}, . \rangle, & \langle 00006, ., \text{nosač natpisne ploče}, . \rangle, \\
\langle 00007, ., \text{natpisna ploča}, . \rangle, & \langle 00008, ., \text{lim posude gornje}, . \rangle, \\
\langle 00009, ., \text{cev}, . \rangle, & \langle 00010, ., \text{lim posude donje}, . \rangle, \\
\langle 00011, ., \text{cevni zid}, . \rangle &
\end{array} \right\} \tag{4.3}$$

Sastavnicu (4.1) ćemo dobiti tek nakon što skup komponenata uočenog podsklopa (4.1) uredimo na odgovarajući način. Predmet daljeg istraživanja će biti će biti modeliranje struktura podataka koje će podržati hijerarhijsko uređenje skupa podataka, a samim tim obezbediti podršku konceptu sastavnica. U nastavku ove glave podrazumevaćemo postojanje tabele artikli sa datom strukturom (4.2).

4.2 Modularna sastavnica

Modularna sastavnica daje veze elemanata samo u okviru jednog modula, između jednog nadređenog i više podređenih elemenata. Posmatrajmo sastavnicu sa slike (4.2), koja je dobijena iz hijerarhijske sastavnice (4.1).



Slika 4.2: Modularna sastavnica radijatora datog tabelom (4.1)

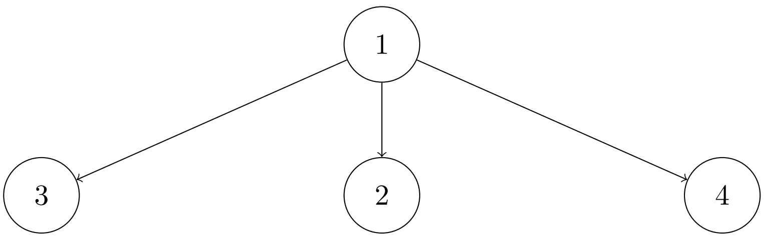
Modularna sastavnica korena stabla najčešće definiše način sklapanja složenog sklopa, što se može pronaći u tehničkoj dokumentaciji. Nadređeni element, u ovom slučaju sam koren 00001, ima skup podređenih sklopova:

$$\{00002, 00003, 00004, 00005, 00006, 00007\}$$

Modularna sastavnica sa slike (4.2), može se modelirati strukturom podataka:

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident \\
ident_pod &::= \text{artikli}.ident \\
grana &::= \langle ident_nad, ident_pod \rangle \\
podstablo &:= \{grana : \text{PRIMARY KEY IS } ident_nad + ident_pod\}
\end{aligned} \tag{4.4}$$

Struktura *grane* u tabeli (4.4), može sadržati i detalje kojima se bliže određuje priroda veze između nadređenih i podređenih, što će kasnije i biti slučaj. Priroda veza će biti ustanovljena analizom modularne sastavnice sa sledeće slike:



Slika 4.3: Modularna sastavnica

Na slici (4.3), prikazan je neki sklop 1 koji se sastoji od tri podsklopa 3, 2, 4. Sklop 1 naziva se *nadređeni* dok se podsklopovi 3, 2, 4, nazivaju *podređeni*. Svako stablo se sastoji od *čvorova* i *grana*. Čvorovima odgovaraju sklopovi i podsklopovi, dok grane opisuju njihovu međusobnu povezanost. Čvorovi su povezani granama i na slici (4.3), se može uočiti da su grane obrnuto orijentisane. Početak grane je na sklopu, dok je kraj grane na odgovarajućem podsklopu. Razlog ovakvog povezivanja je posledica same prirode inženjerske analize problema, jer se uvek posmatra sklop i analizira od čega se on sastoji, a ne obrnuto. Veze podsklopova sa odgovarajućim sklopom, odnosno grane stabla između ostalog mogu uključivati:

– redosled ugradnje pojedinih podsklopova u procesu sklapanja finalnog sklopa

– količine pojedinih podsklopova koje su neophodne za sklapanje finalnog sklopa

– vremena neophodna za ugradnju pojedinih podsklopova u procesu sklapanja finalnog sklopa

Redosled ugradnje podsklopova je obično dat tehničkim crtežom i pri tome uređenje podsklopova na samom crtežu daje nedvosmislenu informaciju o redosledu ugradnje. Pored uređenja podsklopova tehnički crtež može sadržati i informacije o neophodnim količinama pojedinih podsklopova.

4.2.1 Redosled ugradnje

Redosled ugradnje podsklopova se obično zadaje grafičkim prikazom sklopa i njegovih podsklopova. Analizirajući čvorove sa slike (4.3), sledi da se prvo ugrađuje podsklop 3 u sklop 1, zatim podsklop 2 i na kraju se podsklop 4 ugrađuje u sklop 1. Na ovaj način je definisano uređenje podsklopova odnosno na skupu čvorova je definisana relacija parcijalnog uređenja (redosled ugradnje). Iz prethodnog sledi da se skup čvorova stabla višestruko uređuje. Pored uređenja nadređeni-podređeni sklop imamo i uređenje tipa ugrađuje se pre-posle sklopa. Pre svega označimo sa $\mathbb{A}$ skup čvorova koje razmatramo. U slučaju slike (4.3), skup $\mathbb{A}$ jednak je:

$$\mathbb{A} = \{1, 2, 4, 3\} \tag{4.5}$$

Na skupu $\mathbb{A}$ uvedimo relaciju:

$$\mathbb{GBOM} \subseteq \mathbb{A}^2 \tag{4.6}$$

koja je opisnog tipa[1]. Ova relacija je sa računarskog gledišta jako važna, jer se na osnovu nje definiše struktura podataka kojom se modelira stablo sa slike (4.3). Struktura podataka ima sličan oblik kao (4.4):

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident \\
ident_pod &::= \text{artikli}.ident \\
grana &::= \langle ident_nad, ident_pod \rangle \\
GBOM &:= \{grana : \text{PRIMARY KEY IS } ident_nad + ident_pod\}
\end{aligned} \tag{4.7}$$

[1] Ekspertsko znanje inženjera koji određuju koji je nadređeni, a koji je podređeni čvor i ne postoji analitički-matematički izraz za ovu relaciju.

Ako se (4.6) primeni na (4.5) da bi se opisala slika (4.3), dobija se:

$$\mathbb{GBOM}(\mathbb{A}) = \{\langle 1,3\rangle, \langle 1,2\rangle, \langle 1,4\rangle\} \tag{4.8}$$

Dobijeni skup grana nije konzistentan u odnosu na redosled ugradnje podsklopova u sklop, jer:

$$\begin{aligned}\mathbb{GBOM}(\mathbb{A}) &= \{\langle 1,3\rangle, \langle 1,2\rangle, \langle 1,4\rangle\}\\ &= \{\langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle\}\\ &\vdots\end{aligned}$$

Struktura podataka (4.4) u literaturi se naziva *Generic Bill of Material* - generička sastavnica i označava sa *GBOM*. Ova struktura podataka je osnov za stvaranje svih drugih sastavnica, jer se njenom modifikacijom - dodefinisanjem i uvođenjem novih operacija jednostavno dobijaju složeniji modeli drugih sastavnica. Generička sastavnica prvenstveno treba da omogući prikazivanje više verzija sklopova, a zatim i više verzija proizvoda. Prethodni primer, pokazuje osnovni nedostatak generičke sastavnice u primeni, ne uključuje uređenje podsklopova. Ovaj nedostatk generičke sastavnice se rešava proširivanjem odgovarajuće strukture podataka *GBOM* na sledeći način:

$$\begin{aligned}
&ident_nad ::= \text{artikli}.ident\\
&ident_pod ::= \text{artikli}.ident\\
&redosled \;\; ::= \langle \text{NUMBER}, 5\rangle\\
&grana \;\;\;\; ::= \langle ident_nad, ident_pod, redosled\rangle\\
&EBOM \;\;\;\; := \{grana : \text{PRIMARY KEY IS } ident_nad + ident_pod\}
\end{aligned} \tag{4.9}$$

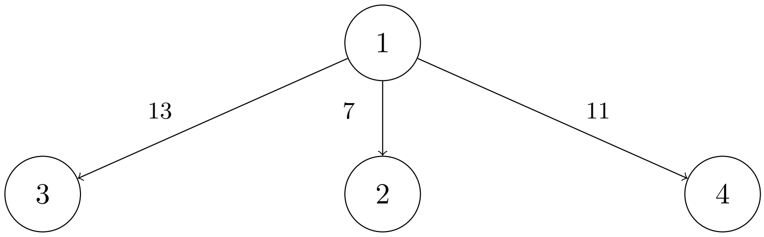
Primenom *EBOM* na (4.5) dobija se:

$$\mathbb{EBOM}(\mathbb{A}) = \{\langle 1,3,1\rangle, \langle 1,2,2\rangle, \langle 1,4,3\rangle\} \tag{4.10}$$

što je potpuno, u smislu prenosa informacija, preslikalo stablo sa slike (4.3). Struktura podataka *EBOM* se u literaturi naziva *Engineering Bill of Material* - inženjerska sastavnica, i predstavlja prvi primer izmene generičke sastavnice. Izmena je, samo na prvi pogled jednostavna, ali ona donosi mogućnost potpunog preslikavanja dvonivovskog stabla kojim je predstavljen odnos sklop-podsklop, odnosno prirodno definisan redosled ugradnje podsklopova da bi se dobio sklop.

4.2.2 Neophodne količine

Sastavnica data slikom (4.3), u stvarnoj upotrebi ima sledeći oblik:
Veličine na granama su neophodne količine odgovarajućih podsklopova za dobijanje jednog sklopa. Ne ulazeći u vrstu jedinice mere inženjerska sastavnica data sa (4.9), jednostavnom



Slika 4.4: Potpuna modularna sastavnica

izmenom na nivou definicije strukture podataka, podržava beleženje količina i ima sledeći oblik:

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident \\
ident_pod &::= \text{artikli}.ident \\
redosled &::= \langle \text{NUMBER}, 5 \rangle \\
kolicina &::= \langle \text{NUMBER}, 13.4 \rangle \\
grana &::= \langle ident_nad, ident_pod, redosled, kolicina \rangle \\
EBOM-T &:= \{grana : \text{PRIMARY KEY IS } ident_nad + ident_pod\}
\end{aligned} \tag{4.11}$$

Konačno, primenom *EBOM-T* na (4.5), stablo dato slikom (4.4), preslikava se u skup:

$$\mathbb{EBOM} - \mathbb{T}\,(\mathbb{A}) = \{\langle 1,3,1,13 \rangle, \langle 1,2,2,7 \rangle, \langle 1,4,3,11 \rangle\} \tag{4.12}$$

Navedene GBOM, EBOM i EBOM-T sastavnice se razlikuju u odnosu na kvalitet podataka - informacija kojim su snabdevene. Kvalitet informacija određuje njihovu upotrebljivost i oblast primene. U teorijskim analizama jednostavnije je raditi sa GBOM jer se postiže maksimalna usredsređenost na problem koji se rešava. Sadržajno kvalitetnija EBOM omogućava povezivanje teorijskih dostignuća sa stvarnim problemima. Pojedini autori zato strukturu (4.11) opisuju na sledeći način:

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident \\
ident_pod &::= \text{artikli}.ident \\
redosled &::= \langle \text{NUMBER}, 5 \rangle \\
lista &::= \langle * \rangle \\
grana &::= \langle ident_nad, ident_pod, redosled, lista \rangle \\
EBOM &:= \{grana : \text{PRIMARY KEY IS } ident_nad + ident_pod\}
\end{aligned} \tag{4.13}$$

pri čemu se podrazumeva da će stvarni problemi biti na odgovarajući način rešavani. Ne smanjujući opštost, u nastavku pod pojmom inženjerske sastavnice - EBOM podrazumevaćemo strukturu (4.13).

4.2.3 Osobine modularne sastavnice

Posle analiza datih u prethodnim delovima ovog teksta možemo zaključiti da modularna sastavnica ima sledeće osobine:

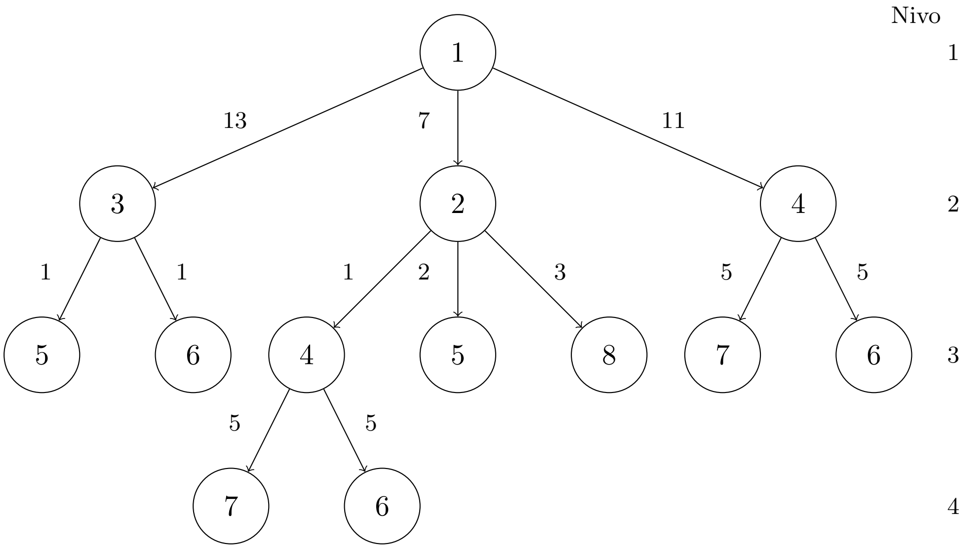
– poseduje samo jedan nadređeni čvor, koji odgovara složenom sklopu

– poseduje jedan ili više podređenih čvorova, kojima odgovaraju neposredni podsklopovi sklopa u nadređenom čoru

Iz prethodnog sledi da je modularna sastavnica *dvonivovska sastavnica* modelirana granama stabla na dva uzastopna nivoa, pa se često u literaturi umesto termina modularna sastavnica koristi termin dvonivovska sastavnica. Samo stablo je opisano strukturom podataka (4.13). Potpuno je jasno da se nadređeni čvora ne sme naći u spisku podređenih čvorova. Ovo ograničenje nije direktno podržano navedenom strukturom podataka (4.13), već se mora programski kontrolisati. Takođe, postavlja se pitanje može li jedan isti sklop imati dve različite modularne sastavnice? Ovo značajno pitanje za inženjere se rešava na poseban način.

4.3 Hijerarhijska sastavnica

Na slici (4.1) data je jedna stvarna hijerarhijska sastavnica. U literaturi se skoro isključivo razmatra prva hijerarhijska sastavnica i najčešće se koristi za dobijanje spiska neophodnih količina materijala ili podsklopova za dobijanje gotovog proizvoda. U suštini je to izveštaj iz baze podatka. Da bi dobili pomenuti izveštaj pored podataka organizovanih u tabelama baze podataka neophodno je razviti i algoritme koji će na odgovarajući način obraditi postojeće podatke. Pre svega analizirajmo sledeću sastavnicu:



Slika 4.5: Hijerarhijska sastavnica

Očigledno hijerarhihjska sastavnica ima strukturu stabla i prirodno je da se koristi stablo kao struktura za modeliranje hijerarhijske sastavnice. Može se primetiti da je i nivo u hijerarhiji komponenta hijerarhijske sastavnice. Ovo je posledica kako organizacije tehničke dokumetacije sa jedne strane, tako i činjenice da je hijerarhijska sastavnica kompozicija modularnih sastavnica. Tradicionalno inženjerski pristup, u formiranju hijerarhijske sastavnice, podrazumevao je:

- izostavljanje grafičke predstave sastavnice, [2]
- korišćenje tehničke dokumentacije uređene prema hijerarhijskoj sastavnici, da bi se odredile količine potrebnog materijala.

Računari su doneli promenu u načinu razmišljanja inženjera i načinu korišćenja hijerahijske sastavnice. U potpunosti je promenjen tradicionalni inženjerski pristup, počev od vizuelizacije proizvoda, do izvođenja različitih proračuna. Personalizacija proizvoda, velika produktivnost i veliki broj verzija proizvoda se može podržati odgovarajućom upotrebom hijerarhijske sastavnice.

[2] Ovo je razumljivo, prirodno je da se za složene proizvode i ne može nacrtati sastavnica

4.3.1 Osnovni pojmovi

Stablo sa slike (4.5) sastavljeno je od više modularnih sastavnica, koje se pojavljuju na različitim pozicijama u stablu. Čvorovi skupa:

$$\mathbb{B} = \{1, 2, 3, 4, 5, 6, 7\} \tag{4.14}$$

prirodno se razvrstavaju po nivoima ugradnje, gde se na nivou 0 nalazi koren. Koren stabla je čvor koji ima samo izlazne grane. Na slici (4.5) je to čvor 1 i on predstavlja gotov proizvod koji se sastoji od podsklopova.

$$\text{root}(\mathbb{B}) = \langle 1 \rangle \tag{4.15}$$

Na ostalim nivoima se nalaze podsklopovi koji su uređeni po nivoima tako da je jednostavno planirati proizvodnju. Čvorovi sa slike (4.5) razvrstavaju se na sledeći način:

$$\begin{aligned}
\text{level}(\mathbb{B}, 0) &= \langle 1 \rangle \\
\text{level}(\mathbb{B}, 1) &= \langle 3, 2, 4 \rangle \\
\text{level}(\mathbb{B}, 2) &= \langle 5, 6, 4, 5, 8, 7, 6 \rangle \\
\text{level}(\mathbb{B}, 3) &= \langle 7, 6 \rangle
\end{aligned} \tag{4.16}$$

U stablu hijerarhijske sastavnice postoje listovi koji odgovaraju neophodnim materijalima ili posebnim sklopovima koji iz nekih razloga nisu podvrgnuti strukturnoj dekompoziciji. Listovi stabla su čvorovi koji imaju samo ulazne grane. Hijerarhijska sastavnica sa slike (4.5) ima sledeće listove:

$$\text{leaf}\,(\mathbb{B}) = \langle 5, 6, 7, 8 \rangle \tag{4.17}$$

Na ovaj način za dati skup čvorova:

$$\mathbb{NODE} := \{\text{spisak čvorova}\} \tag{4.18}$$

definišu se sledeće funkcije:

$$\begin{aligned}
\text{root}(\mathbb{NODE}) &:= \{\text{čvor}\} \\
\text{level}(\mathbb{NODE}, \text{nivo}) &:= \{*\} \\
\text{leaf}(\mathbb{NODE}) &:= \{*\}
\end{aligned} \tag{4.19}$$

Dalje primetimo da se hijerarhijska sastavnica opisuje konačnim stablom. Sve putanje od korena do listova stabla se sastoje od konačnog broja grana. Ove činjenice su posebno važne u programiranju i projektovanju hijerarhijskih sastavnica. Hijerarhijska sastavnica se posebno koristi kao podrška rešaanju sledećih problema:

– pronađi sve puteve od listova do korena i izračunaj potrebne količine

– pronađi sve korene podstabala u kojima list može da učestvuje

i oni se uspešno rešavaju uz pomoć računara. U zavisnosti od prirode samog proizoda čija se struktura opisuje hijerarhijskom sastavnicom razlikovaćemo dva tipa hijerarhijskih sastavnica koji će dalje određivati način njenog formiranja:

– *statička*

– *dinamička*

Priroda gotovog proizvoda čija je struktura opisana hijerarhijskom sastavnicom, određuje i prirodu hijerarhijske sastavnice [20]. Ako se posmatra slika (4.5), kojom je data jednostavna hijerarhijska sastavnica, nameće se pitanje kako se ona formira?

– Od korena ka listovima - *odozgo na dole*

– Od listova ka korenu - *odozdo na gore*

Analizom koja sledi možemo zaključiti sledeće. Ukoliko se radi o malom broju proizvoda, odnosno o proizvodu koji se praktično ne menja odgovarajuća hijerarhijska sastavnica je statička i formira se od korena ka listovima odnosno koristi se pristup *odozgo na dole*. U slučaju velikog broja proizvoda, odnosno ukoliko se prizvod često menja odgovarajuća hijerarhijska sastavnica je dinamička i formira se od listova ka korenu, odnosno koristi se pristup *odozdo na gore*.

4.3.2 O broju proizvoda

Pod brojem proizvoda podrazumevaćemo broj verzija istog proizvoda. Proizvodi u kojima je ugrađena vrhunska tehnologija, tehnički su izuzetno složeni i spadaju u proizvode koji imaju statičku sastavnicu. Tipičan primer su avioni i bilo šta što je povezano sa vojnom industrijom. Statička sastavnica, bez obzira kako je napravljena, je nepromenljiva u bilo kom svom delu, ili ako se menja, struktura joj ostaje nepromenjena. Sa druge strane proizvodi metalske industrije, poput automobila, mogu se tretirati na oba načina i kao proizvodi sa malim brojem verzija i kao proizvodi sa velikim brojem verzija. Veliki broj verzija imaju proizvodi metalske industrije koji imaju karakteristiku proizvoda masovne potrošnje. Proizvođači su prinuđeni da izbacuju "nove proizvode", i to ne retko ozbiljno menja osnovnu verziju proizvoda. Pokazalo se da ja tada pogodno imati sastavnicu za osnovni prozvod, verziju 0, a potom neprekidnim izmenama dobijati "nove" proizvode. Neophodno je analizirati kako broj proizvoda utiče na projektovanje sastavnica i na njihovu prirodu.

Proizvodi sa malim brojem verzija

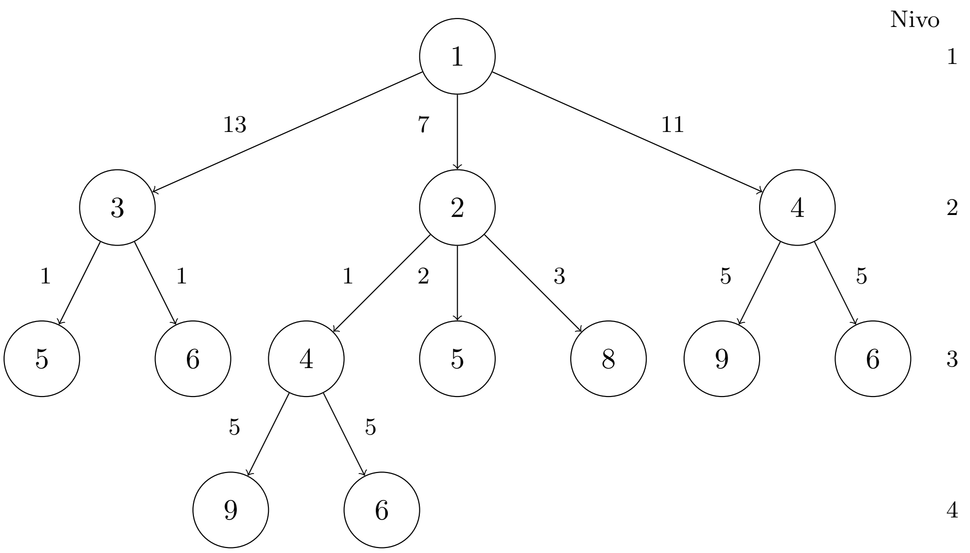
Hijerarhijska sastavnica za proizvode sa malim brojem verzija najčešće se unosi kao jedinstveni skup podataka i nad njima se tako i radi. Da bi ilustrovali ovu činjenicu, skup podataka kojim se modelira hijerarhijska sastavnica sa slike (4.5) ima sledeći izgled:

$$
\begin{aligned}
ident\_nad &::= \text{artikli}.ident\\
ident\_pod &::= \text{artikli}.ident\\
nivo &::= \langle \text{NUMBER}, 5\rangle\\
redosled &::= \langle \text{NUMBER}, 5\rangle\\
kolicina &::= \langle \text{NUMBER}, 13.4\rangle\\
grana &::= \langle ident\_nad, ident\_pod, nivo, redosled, kolicina\rangle\\
EBOM-H &:= \{grana\}
\end{aligned}
\tag{4.20}
$$

Struktura podataka data sa (4.20) može jednostavno da prati slaganje tehničke dokumentacije, u kom god obliku da je. Statičnost strukture je očigledna, jer je jednostavnije iskopirati celokupnu sastavnicu za novi proizod nego dodati, oduzeti ili izmeniti određenu komponentu u sastavnici.

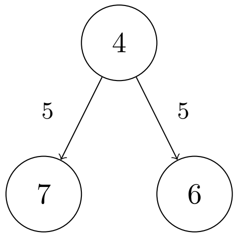
Proizvodi sa velikim brojem verzija

Proizvodi sa velikim brojem verzija imaju jedinstvenu hijerarhijsku sastavnicu, ona se često naziva GBOM bez opasnosti mešanja pojmova. Jednostavno se pretpostavlja da postoji sastavnica koja opisuje više proizvoda iste klase. Verzije proizvoda se dobijaju malim izmenama postojeće hijerarhijske sastavnice. Često se kaže da je polazna hijerarhijska sastavnica, sastavnica verzije 0 ili sastavnica prototipa. Da bi ovo objasnili navedimo jedan primer koji će karakterisati proizvode sa velikim brojem verzija. Iskoristimo zato hijerarhijsku sastavnicu sa slike (4.5) koja će nam poslužiti kao sastavnica verzije 0.

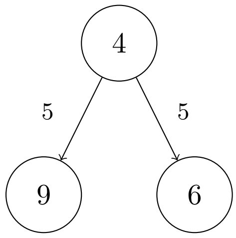


Slika 4.6: Hijerarhijska sastavnica novog proizvoda.

Razlika između sastavnica sa slike (4.5) i (4.6) je u modularnoj sastavnici:
koja je zamenjena sa:
Ovo je dovoljno da se dobije nova verzija istog proizvoda. Dobijena hijerarhijska sastavnica

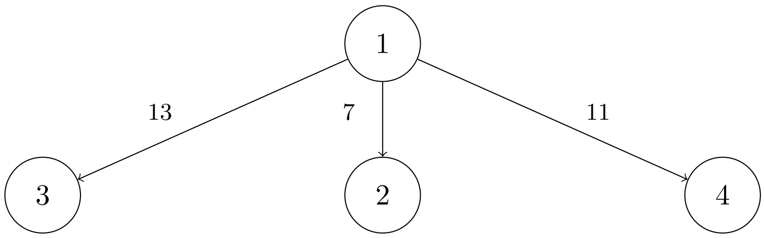


Slika 4.7: Modularna sastavnica $\{\langle 4,7,1,5\rangle, \langle 4,6,2,5\rangle\}$.



Slika 4.8: Modularna sastavnica $\{\langle 4,9,1,5\rangle, \langle 4,6,2,5\rangle\}$.

sa slike (4.6) je istovetna hijerarhijskoj sastavnici (4.5) uz izmenu određene modularne sastavnice. Pokazalo se da je najjednostavnije uočiti zajedničke modularne sastavnice od kojih su obe sastavljene, i posebno istaći razlike. Dakle hijerarhijska sastavnica sa slike (4.6) i hijerarhijska sastavnica sa slike (4.5) sastoje se od sledećih zajedničkih modularnih sastavnica:



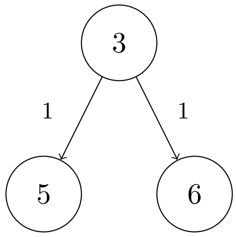
Slika 4.9: Modularna sastavnica $\{\langle 1,3,1,13\rangle, \langle 1,2,2,7\rangle, \langle 1,4,3,11\rangle\}$.

dok se razlikuju u modularnim sastavnicama:

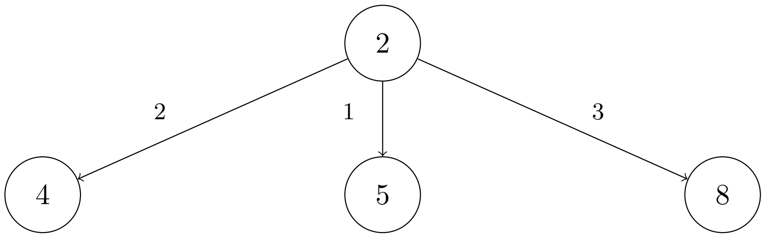
Priroda hijerarhijske sastavnice

Ako se analizira prethodna sekcija jednostavno se može uočiti da je hijerarhijska sastavnica sa slike (4.5) jednaka:

$$
\begin{aligned}
\mathbb{EBOM} - \mathbb{H}\,(\text{Slika }(4.5)) \quad := \quad & \{ \\
& \{\langle 1,3,1,13\rangle, \langle 1,2,2,7\rangle, \langle 1,4,3,11\rangle\}, \\
& \{\langle 3,5,1,1\rangle, \langle 3,6,2,1\rangle\}, \\
& \{\langle 2,4,1,1\rangle, \langle 2,5,2,2\rangle, \langle 2,8,3,3\rangle\}, \\
& \{\langle 4,7,1,5\rangle, \langle 4,6,2,5\rangle\} \\
& \}
\end{aligned}
$$



Slika 4.10: Modularna sastavnica $\{\langle 3,5,1,1\rangle, \langle 3,6,2,1\rangle\}$.



Slika 4.11: Modularna sastavnica $\{\langle 2,4,1,1\rangle, \langle 2,5,2,2\rangle, \langle 2,8,3,3\rangle\}$.

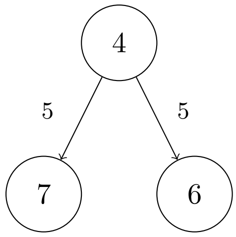
dok je hijerarhijska sastavnica sa slike (4.6) jednaka:

$$\begin{aligned}
\mathbb{EBOM} - \mathbb{H}\,(\text{Slika } (4.6)) \quad := \quad & \{ \\
& \{\langle 1,3,1,13\rangle, \langle 1,2,2,7\rangle, \langle 1,4,3,11\rangle\}, \\
& \{\langle 3,5,1,1\rangle, \langle 3,6,2,1\rangle\}, \\
& \{\langle 2,4,1,1\rangle, \langle 2,5,2,2\rangle, \langle 2,8,3,3\rangle\}, \\
& \{\langle 4,9,1,5\rangle, \langle 4,6,2,5\rangle\} \\
& \}
\end{aligned}$$

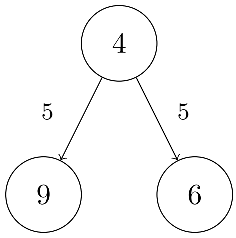
odakle sledi da je:

$$\begin{aligned}
\mathbb{EBOM} - \mathbb{H}\,(\text{Slika } (4.5)) \quad \cap \quad & \mathbb{EBOM} - \mathbb{H}\,(\text{Slika } (4.6)) \\
:= \quad & \{ \\
& \{\langle 1,3,1,13\rangle, \langle 1,2,2,7\rangle, \langle 1,4,3,11\rangle\}, \\
& \{\langle 3,5,1,1\rangle, \langle 3,6,2,1\rangle\}, \\
& \{\langle 2,4,1,1\rangle, \langle 2,5,2,2\rangle, \langle 2,8,3,3\rangle\}, \\
& \{\langle 4,6,2,5\rangle\} \\
& \}
\end{aligned}$$

Prirodno je dakle klasu proizvoda opisati minimalnim skupom modularnih sastavnica koje su elementi hijerarhijske sastavnice sakog proizvoda iz date klase proizoda. Modularne sastavnice su dakle gradivni blokovi za hijerarhijsku sastavnicu. Dinamika hijerarhijskih sastavnica se ostvaruje posebnim izborom modularnih sastavnica. Takođe, ukoliko su formirane modularne sastavnice jednostavno je formirati konkretnu hijerarhijsku sastavnicu, pri tome ni veliki



Slika 4.12: Modularna sastavnica, hijerarhijske sastavnice sa slike (4.5)



Slika 4.13: Modularna sastavnica hijerarhijske sastavnice sa slike (4.6)

broj proizvoda ne predstavlja problem. Pre svega neophodno je proširiti EBOM strukturu podataka (4.13) na sledeći način:

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident\\
verzija &::= \langle \text{CHAR}, 3\rangle\\
ident_pod &::= \text{artikli}.ident\\
redosled &::= \langle \text{NUMBER}, 5\rangle\\
lista &::= \langle * \rangle\\
grana &::= \langle ident_nad, verzija, ident_pod, redosled, lista\rangle\\
EBOM - H &:= \{grana\}
\end{aligned} \tag{4.21}$$

Atribut *verzija* može imati posebnu vrednost *null* koja označava pripadnost verziji 0 datog proizvoda. Tradicionlano, vrednost ***null*** specificira situaciju ne postojanja vrednosti. Na ovaj način se uočena problematika izneta u prethodnom poglavlju, vezana za sastavnice proizvoda sa više verzija (4.5) i (4.6) jednostavno rešava podatkom verzija. Da bi to pokazali pogledajmo kako sada izgleda hijerarhijska sastavnica sa slike (4.5):

$$\begin{aligned}
\mathbb{EBOM}\,(Slika(4.5)) \;\; := \;\; &\{\\
&\{\langle 1, , 3, 1, 13\rangle, \langle 1, , 2, 2, 7\rangle, \langle 1, , 4, 3, 11\rangle\},\\
&\{\langle 3, , 5, 1, 1\rangle, \langle 3, , 6, 2, 1\rangle\},\\
&\{\langle 2, , 4, 1, 1\rangle, \langle 2, , 5, 2, 2\rangle, \langle 2, , 8, 3, 3\rangle\},\\
&\{\langle 4, , 7, 1, 5\rangle, \langle 4, , 6, 2, 5\rangle\}\\
&\}
\end{aligned}$$

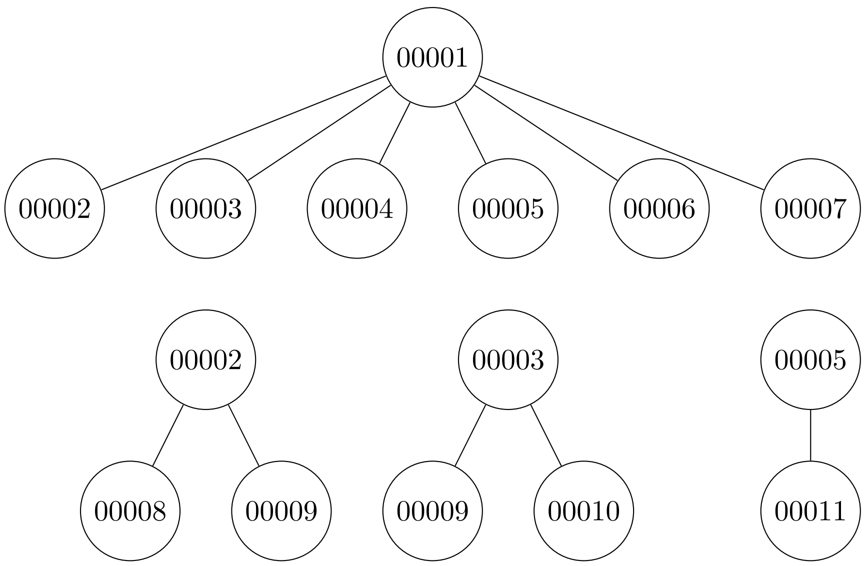
i ona je očigledno hijerarhijska sastavnica verzije 0, dok hijerarhijska sastavnica sa slike (4.6) ima oblik:

$$
\begin{aligned}
\mathbb{EBOM}\,(Slika(4.6)) \quad := \quad & \{ \\
& \{\langle 1, , 3, 1, 13\rangle, \langle 1, , 2, 2, 7\rangle, \langle 1, , 4, 3, 11\rangle\}, \\
& \{\langle 3, , 5, 1, 1\rangle, \langle 3, , 6, 2, 1\rangle\}, \\
& \{\langle 2, , 4, 1, 1\rangle, \langle 2, , 5, 2, 2\rangle, \langle 2, , 8, 3, 3\rangle\}, \\
& \{\langle 4, 1, 9, 1, 5\rangle, \langle 4, , 6, 2, 5\rangle\} \\
& \}
\end{aligned}
$$

jer su to u suštini dva različita proizvoda, proizvod 1 verzije 0 i proizvod 1 verzije 1.

4.3.3 Generisanje hijerarhijske sastavnice

Hijerarhijska sastavnica predstavljena na slici (4.1) daje kompletnu strukturu razrađenu po hijerarhiji nivoa ugradnje podsklopova u konačni sklop. Primetimo da jedan podsklop može ulaziti više puta u strukturu konačnog sklopa na istim ili različitim nivoima. U suštini hijerahijska sastavnica na slici (4.1) sastavljena je od dvonivovskih - modularnih sastavnica. Jednu njenu modularnu sastavnicu smo već pokazali na slici (4.2). Na sledećoj slici se nalaze sve modularne sastavnice koje su neophodne i nalaze se u tabeli (4.1):



Slika 4.14: Modularne sastavnica podsklopova

Podsklop 9 se dva puta ugradjuje prema slici (4.1), ali je važno uočiti da se kao skup podataka pojavljuje samo jednom. U stvarnosti se ove modularne sastavnice zadaju tabelom:

Tabela 4.2: Sve modularne sastavnice radijatora

ident_nad	**ident_pod**	**red**
00001	00002	1
00001	00003	2
00001	00004	3
00001	00005	4
00001	00006	5
00001	00007	6
00002	00008	1
00002	00009	2
00003	00009	1
00003	00010	2
00005	00011	1

Formiranje hijerarhijske sastavnice se sastoji u povezivanju odgovarajućih modularnih stavnica, uz neprekidnu kontrolu šta su listovi, nadređeni i podređeni elementi. Da bi tabelu (4.2) preslikali u tabelu baze podataka, neophodno je promeniti strukturu podataka (4.4). Prva izmena se sastoji u uvođenju redosleda kojim se uređuju podređeni elementi (4.22).

$$\begin{aligned}
ident_nad &::= \text{artikli}.ident \\
ident_pod &::= \text{artikli}.ident \\
redosled &::= \langle \text{NUMERIC}, 5 \rangle \\
grana &::= \langle ident_nad, ident_pod, redosled \rangle \\
podstablo &:= \{grana : \text{PRIMARY KEY IS } nadredjeni + podredjeni\}
\end{aligned} \tag{4.22}$$

Na ovaj način se modelira redosled ugradnje podsklopova u nadređeni sklop. Navedimo primer kojim se definišu modularne sastavnice sa slike (4.14):

$$\begin{aligned}
podstablo = \{ \\
&\langle 00001, 00002, 1 \rangle, \langle 00001, 00003, 2 \rangle, \\
&\langle 00001, 00004, 3 \rangle, \langle 00001, 00005, 4 \rangle, \\
&\langle 00001, 00006, 5 \rangle, \langle 00001, 00007, 6 \rangle, \\
&\langle 00002, 00008, 1 \rangle, \langle 00002, 00009, 2 \rangle, \\
&\langle 00003, 00009, 1 \rangle, \langle 00003, 00010, 2 \rangle, \\
&\langle 00005, 00011, 1 \rangle \\
&\}
\end{aligned} \tag{4.23}$$

Druga karakteristika hijerarhijske sastavnice je što se broj nivoa - "dubina" hijerarhijske sastavnice u strukturi podataka ne može elementarno uvesti. Broj nivoa, odnosno dubina se određuje u toku formiranja hijerarhijske sastavnice.

Opšti algoritam formiranja hijerarhijske sastavnice

Algoritmi za formiranje hijerarhijske sastavnice mogu biti:

– rekurzivni

– nerekurzivni

pri čemu je važno istaći da su ovi drugi izvedeni iz prvih, tako što je rekurzija na neki od standardnih načina eliminisana, na primer simulacijom rekurzije korišćenjem steka [22] [23]. Ulazna veličina je koren podstabla koji odgovara sklopu za koji treba generisati hijerarhijsku sastavnicu. Prikazani algoritam je generičkog tipa i genereše GBOM sastavnicu [19].

Algoritam 4.1 Rekurzivno generisanje GBOM-a

```
void napravi_gbom( c_ident, n_nivo )
{
    local sledeci;
    if ( !dbseek( podstablo, c_ident ) ) {
        dbappend( gbom , n_nivo, c_ident, "00000" );
        return;
    }
    else {
        while( !dbeof( podstablo ) && podstablo.ident_nad == c_ident ) {
            sledeci = dbnext( podstablo );
            dbappend( gbom , n_nivo, c_ident, podstablo.ident_pod );
            napravi_gbom( podstablo.ident_pod, n_nivo+1 );
            dbgoto( podstablo, sledeci )
        };
    };
    return;
}
```

Rekurzivna procedura **napravi_gbom** selektuje čvorove stabla hijerarhijske sastavnice po takozvanim "krajnje levim granama". Rezulat se smešta u tabelu čija je struktura:

$$\begin{aligned} nivo \quad &::= \langle \mathrm{NUMERIC}, 5 \rangle \\ ident\_nad &::= \mathrm{artikli}.ident \\ ident\_pod &::= \mathrm{artikli}.ident \\ grana \quad &::= \langle nivo, ident\_nad, ident\_pod \rangle \\ gbom \quad &:= \{grana\} \end{aligned} \tag{4.24}$$

Navedena tabela se koristi kao spremište za ispis rezultata algoritma. Na primer, ako se algoritam primeni na skup modularnih sastavnica (4.23), sadržaj pomenute tabele će biti:

$$\begin{aligned} gbom \quad = \quad \{ & \\ & \langle 1, 00001, 00002 \rangle \\ & \langle 2, 00002, 00008 \rangle, \langle 3, 00008, 00000 \rangle, \\ & \langle 2, 00002, 00009 \rangle, \langle 3, 00009, 00000 \rangle, \\ & \langle 1, 00001, 00003 \rangle, \langle 2, 00003, 00009 \rangle, \\ & \langle 3, 00009, 00000 \rangle, \langle 2, 00003, 00010 \rangle, \\ & \langle 3, 00010, 00000 \rangle, \langle 1, 00001, 00004 \rangle, \\ & \langle 2, 00004, 00000 \rangle, \langle 1, 00001, 00005 \rangle, \\ & \langle 2, 00005, 00011 \rangle, \langle 3, 00011, 00000 \rangle, \\ & \langle 1, 00001, 00006 \rangle, \langle 2, 00006, 00000 \rangle, \\ & \langle 1, 00001, 00007 \rangle, \langle 2, 00007, 00000 \rangle \\ & \} \end{aligned}$$

Glava 5

Sistem održavanja vazduhoplova

5.1 Uvodna razmatranja

Podaci o sastavnici proizvoda i tehnologiji izrade predstavljaju osnovu za upravljanje sistemom održavanja [16]. Njihova tačnost, dobra priprema i mogućnost upravljanja nastalim promenama su neophodni za efikasno funkcionisanje ostalih sistema, pa samim tim i sistema održavanja. Najjednostavniji način za početno planiranje održavanja sastoji se u preuzimanju sastavnice iz tehničke pripreme i iz proizvodnje. Sledeći korak je prilagoditi ih zahtevima sistema za održavanje. Najčešće aktivnosti u sistemu za održavanje vazduhoplova su zapravo aktivnosti sklapanja (pripreme) rezervnog dela i njegova ugradnja u objekat koji je predmet održavanja (vazduhoplov), pa je jedan od glavnih izvora podataka, na osnovu kojih se projektuje i realizuje sistem za upraljanje održavanjem, konstruktivno-tehnološka dokumentacija. Pored tehnoloških postupaka (pregleda operacija, operacionih listi) i konstrukcionih crteža, sastavnica proizvoda je posebno značajna. Sastavnica gotovih proizvoda pored strukture samog proizvoda daje i postupak ugradnje komponenti u gotov proizvod. Samim tim, ona je osnova za definisanje postupka održavanja određenog proizvoda. Utvrdjeni redosled ugradnje podsklopova čini osnovu za planiranje održavanja, odnosno redosled sklapanja (pripreme) pojedinih komponenti u samom ciklusu održavanja. Sa druge strane, tehnologija daje spisak operacija sa tačno definisanim redosledom i standardnim vremenima po operacijama, što čini osnov za planiranje kapaciteta, pa se ovo može iskoristiti kako za uređivanje niza aktivnosti održavanja tako i za procenu vremena trajanja pojedinih aktivnosti održavanja. Na ovaj način se može efikasno planirati skup aktivnosti održavanja, odnosno može se govoriti o optimalnom planiranju održavanja [18] [12].

U prethodnoj Glavi je predložena fleksibilna struktura sastavnice, dakle postoji osnov za njeno sadržajno raširenje. Sledeći korak je modeliranje aktivnosti sklapanja odnosno ugradnje podsklopova u nadređeni sklop, rezultat pomenutog modeliranja će biti dodatni sadržaj takozvane druge hijerarhijske sastavnice. Sam postupak ugradnje (tehnologija ugradnje) se uobičajenom tehnikom dekompozicije, raščlanjuje na niz elementarnih zahvata koji se zatim modeliraju u strukturi druge hijerarhijske sastavnice.

5.2 Projektovanje algoritma za generisanje druge sastavnice

Druga hijerarhijska sastavnica je sastavnica kojom se definiše tehnologija održavanja proizvoda [21]. Da bi pojednostavili projektovanje podrazumeva se da postoje odgovarajuće strukture podataka u kojima se beleže informacije o prolazu u određenom zahvatu određene operacije. Između ostalog podrazumevaćemo postojanje evidencije zahvata, odnosno tabele sa sledećom strukturom:

$$\begin{aligned}
ident &::= \langle \mathrm{CHAR}, 5, \mathrm{FORMAT} : 99999, \mathrm{AUTONUMBER} \rangle \\
sifra &::= \langle \mathrm{CHAR}, 15 \rangle \\
naziv &::= \langle \mathrm{CHAR}, 40 \rangle \\
zahvat &::= \langle ident, sifra, naziv, * \rangle \\
operacije &:= \{ zahvat : \text{PRIMARY KEY IS } ident \}
\end{aligned} \tag{5.1}$$

U zavisnosti od strukture grane u modularnoj (dvonivovskoj) sastavnici koja definiše kontekst u kome se zahvat izvodi, razlikuju se četiri tipa zahvata u održavanju. Ova klasifikacija je izvedena prema tome šta su krajevi grane kojoj je zahvat pridružen:

- zahvat ima nadređeni i nema podređeni sklop
- zahvat ima nadređeni i podređeni sklop i pri tome su to isti sklopovi
- zahvat ima nadređeni i podređeni sklop i to su različiti sklopovi
- zahvat ima podređeni i nema nadređeni sklop

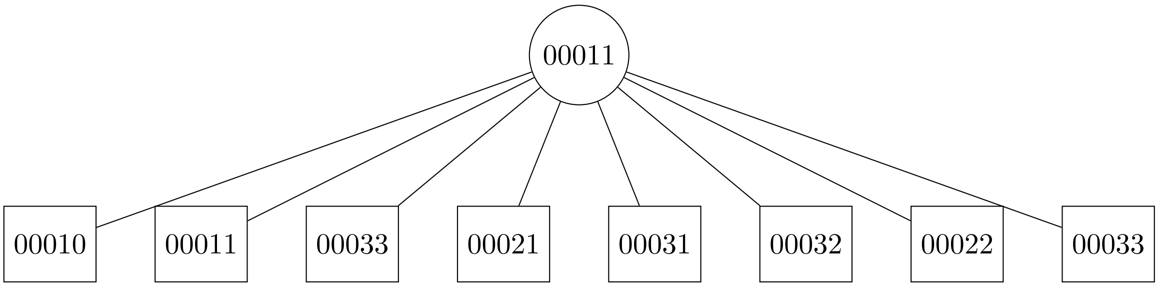
Zahvati se vezuju za tehnološki postupak i imaju svoj redosled i ostale atribute. Redosled je bitan jer u suštini on sa vrstom zahvata i određuje tehnološki postupak. U nastavku ove glave daćemo prikaz sva četri tipa zahvata i njihove veze sa modularnim sastavnicama. Pre svega definisaćemo modularne sastavnice koje u granama nose informacije o operacijama - zahvatima. Tabela *tehnologija* u kojoj se beleže ove modularne sastavnice, ima sledeću strukturu:

$$\begin{aligned}
ident_nad &::= \mathrm{artikli}.ident \\
ident_pod &::= \mathrm{artikli}.ident \\
ident &::= \mathrm{operacije}.ident \\
redosled &::= \langle \mathrm{NUMERIC}, 5 \rangle \\
zahvat &::= \langle ident_nad, ident_pod, ident, redosled, * \rangle \\
tehnologija &:= \{ zahvat : \text{PRIMARY KEY IS } ident_nad + ident_pod + redosled \}
\end{aligned} \tag{5.2}$$

Pokazalo se da je vrlo važno analizirati klase zahvata jer direktno utiču na način konstruisanja algoritma formiranja druge sastavnice. U nastavku ove glave daju se predstavnici pojedinih tipova zahvata i odgovarajući primeri.

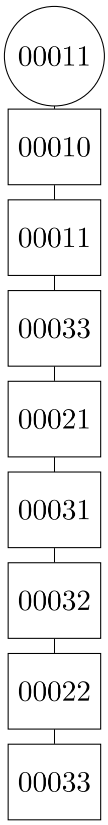
5.2.1 Zahvat ima nadređeni i nema podređeni sklop

Na sledećoj slici prikazan je skup zahvata koji imaju nadređeni sklop i pri tome nemaju podređene sklopove.



Slika 5.1: Skup zahvata sa nadređenim sklopom i bez podređenih sklopova

Na slici (5.1) očigledna je veza sa modularnom sastavnicom, ali nije istaknut redosled zahvata koji je izuzetno bitan jer taj redosled čini tehnologiju. Na sledećoj slici se posebno ističe redosled zahvata:



Slika 5.2: Redosled zahvata sa nadređenim sklopom i bez podređenih sklopova

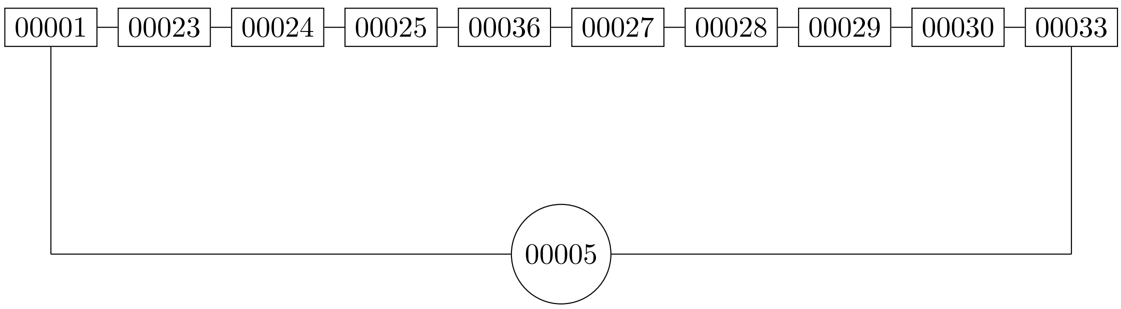
U slučaju zahvata sa slike (5.2), podskup $P_{nad_1 \to pod_0}$ tabele (5.2) je:

$$
\begin{aligned}
P_{nad_1 \to pod_0} \quad = \quad & \{ \\
& \langle 00011, 00000, 00010, 1 \rangle, \langle 00011, 00000, 00011, 2 \rangle, \\
& \langle 00011, 00000, 00033, 3 \rangle, \langle 00011, 00000, 00021, 4 \rangle, \\
& \langle 00011, 00000, 00031, 5 \rangle, \langle 00011, 00000, 00032, 6 \rangle, \\
& \langle 00011, 00000, 00022, 7 \rangle, \langle 00011, 00000, 00033, 8 \rangle \\
& \}
\end{aligned}
$$

gde je nepostojanje sklopa označeno identifikacionom šifrom 00000.

5.2.2 Zahvat ima nadređeni i podređeni sklop i pri tome su to isti sklopovi

Postoje sklopovi u održavanju koji zahtevaju niz pripremnih radnji pre nego što se započne njihova ugradnja odnosno apliciranje u sklop. Jedan dobar primer je avio motor koji zahteva niz zahvata nad samim motorom pre nego što dođe do prvih zahvata njegove ugradnje. Na slici koja sledi je prikazan takav sklop koji zahteva izvršavanje niza zahvata u definisanom redosledu da bi se dalje ugradio u sklop.



Slika 5.3: Skup zahvata sa nadređenim sklopom koji je istovremeno i podređeni sklop

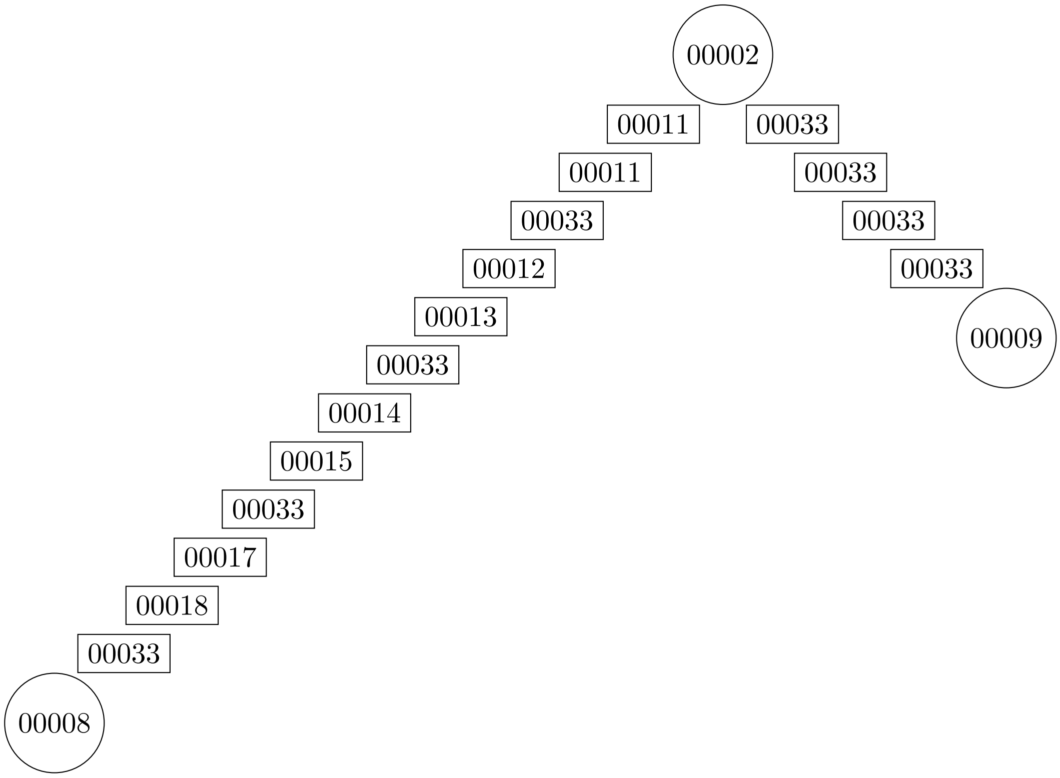
U slučaju zahvata sa slike (5.3), podskup $P_{nad_1 \to pod_1}$ tabele (5.2) je:

$$
\begin{aligned}
P_{nad_1 \to pod_1} \quad = \quad & \{ \\
& \langle 00005, 00005, 00001, 1 \rangle, \langle 00005, 00005, 00023, 2 \rangle, \\
& \langle 00005, 00005, 00024, 3 \rangle, \langle 00005, 00005, 00025, 4 \rangle, \\
& \langle 00005, 00005, 00026, 5 \rangle, \langle 00005, 00005, 00027, 6 \rangle, \\
& \langle 00005, 00005, 00028, 7 \rangle, \langle 00005, 00005, 00029, 8 \rangle, \\
& \langle 00005, 00005, 00030, 9 \rangle, \langle 00005, 00005, 00033, 10 \rangle \\
& \}
\end{aligned}
$$

Ovakvi zahvati narušavaju strukturu stabla (pojavljuju se ciklusi) kojim je sastavnica definisana. Algoritam za generisanje druge hijerarhijske sastavnice mora na odgovarajući način tretirati ovu situaciju.

5.2.3 Zahvat ima nadređeni i podređeni sklop i to su različiti sklopovi

Pokazalo se u različitim fazama održavanja da pokušaj ugradnje jednog podsklopa izaziva niz provera i aktivnosti - zahvata na različitim podsistemima - podsklopovima. Svi ovi zahvati se posebno kontrolišu jer obezbeđuju dobro funkcionisanje sklopa. Na sledećoj slici je prikazan primer takvog zahvata:



Slika 5.4: Skup zahvata sa nadređenim sklopom i podređenim sklopovima

U slučaju zahvata sa slike (5.4), podskup $P_{nad_1 \to pod_2}$ tabele (5.2) je:

$$
\begin{aligned}
P_{nad_1 \to pod_2} = \{ & \\
& \langle 00002, 00008, 00010, 1\rangle, \langle 00002, 00008, 00011, 2\rangle, \\
& \langle 00002, 00008, 00033, 3\rangle, \langle 00002, 00008, 00012, 4\rangle, \\
& \langle 00002, 00008, 00013, 5\rangle, \langle 00002, 00008, 00033, 6\rangle, \\
& \langle 00002, 00008, 00014, 7\rangle, \langle 00002, 00008, 00015, 8\rangle, \\
& \langle 00002, 00008, 00016, 9\rangle, \langle 00002, 00008, 00033, 10\rangle, \\
& \langle 00002, 00008, 00017, 11\rangle, \langle 00002, 00008, 00018, 12\rangle, \\
& \langle 00002, 00008, 00033, 13\rangle, \\
& \langle 00002, 00009, 00019, 1\rangle, \langle 00002, 00009, 00033, 2\rangle, \\
& \langle 00002, 00009, 00020, 3\rangle, \langle 00002, 00009, 00033, 4\rangle \\
& \}
\end{aligned}
$$

Ovaj slučaj je posebno složen i predstavlja problem u projektovanju tehnologije održavanja.

5.2.4 Zahvat ima podređeni i nema nadređeni sklop

Zahvat nad korenom hijerarhije je najčešće ovog tipa. Ovaj tip zahvata nije posebno zanimljiv za analizu.

5.2.5 Algoritam druge hijerarhijske sastavnice

Rekurzija se prirodno nameće jer je to u osnovi i strukture podataka (5.2) i osnovnog algoritma za formiranje hijerarhijske sastavnice. Algoritam za generisanje druge hijerarhijske sastanice se može dobiti razradom osnovnog algoritma (4.1) koja na odgovarajući način tretira situacije opisane u prethodnoj analizi zahvata (5.1).

Algoritam 5.1 Rekurzivno generisanje druge sastavnice

```
void napravi_tbom( c_ident, n_nivo )
{
  local sledeci, c_ident_pod;

  if ( !dbseek( tehnologija, c_ident ) ) {
dbappend( tbom , n_nivo, c_ident, "00000",
                             tehnologija.ident, tehnologija.red );
    return;
  }
  while( !dbeof( tehnologija ) && tehnologija.ident_nad == c_ident )
  {
    c_ident_pod = tehnologija.ident_pod;
    if ( c_ident_pod == c_ident )
    {
      while( !dbeof( tehnologija ) && tehnologija.ident_nad == c_ident
                                   && tehnologija.ident_pod == c_ident )
      {
        dbappend( tbom , n_nivo, c_ident, c_ident,
                                 tehnologija.ident, tehnologija.redosled );
        dbskip( tbom );
      }
    }
    if ( c_ident_pod == "00000" )
    {
      while( !dbeof( tehnologija ) && tehnologija.ident_nad == c_ident_pom
                                   && tehnologija.ident_pod == "00000" )
      {
        dbappend( tbom , n_nivo, c_ident, "00000",
                                 tehnologija.ident, tehnologija.redosled );
        dbskip( tbom );
      }
    }
    sledeci = dbnext( tehnologija );
    dbappend( tbom , n_nivo, c_ident, tehnologija.ident_pod,
                             tehnologija.ident, tehnologija.redosled );
    napravi_tbom( tehnologija.ident, n_nivo+1 );
    dbgoto( tehnologija, sledeci );
  }
  return;
}
```

Rezulat rada ovog algoritma se smešta u tabelu koja ima sledeću strukturu:

$$
\begin{aligned}
nivo &::= \langle \mathrm{NUMERIC}, 5\rangle \\
ident\_nad &::= \mathrm{artikli}.ident \\
ident\_pod &::= \mathrm{artikli}.ident \\
ident &::= \mathrm{operacije}.ident \\
redosled &::= \langle \mathrm{NUMERIC}, 5\rangle \\
zahvat &::= \langle nivo, ident\_nad, ident\_pod, ident, redosled\rangle \\
TBOM &:= \{zahvat\}
\end{aligned} \tag{5.3}
$$

U Dodatku B dat je primer obrade podataka opisanim algoritmom, ulazni podaci su dati u (2.1), a rezltat obrade je dat u (2.2)

5.3 Proračun vremena

5.3.1 Osnovne pretpostavke

Neophodno je pre svega proširiti strukturu tabele *operacije* (5.1) tako da uključuje podatak o vremenu trajanja operacije [17]. To se može izvesti na sledeći način (5.4):

$$
\begin{aligned}
ident &::= \langle \mathrm{CHAR}, 5, \mathrm{FORMAT} : 99999, \mathrm{AUTONUMBER} \rangle \\
sifra &::= \langle \mathrm{CHAR}, 15 \rangle \\
naziv &::= \langle \mathrm{CHAR}, 40 \rangle \\
vreme &::= \langle \mathrm{NUMERIC}, 10.4 \rangle \\
zahvat &::= \langle ident, sifra, naziv, opis, vreme, * \rangle \\
operacije &:= \{ zahvat : \mathrm{PRIMARY\ KEY\ IS}\ ident \}
\end{aligned} \tag{5.4}
$$

Slično proširimo i strukturu tabele *tehnologija* (5.2) i imaćemo (5.5):

$$
\begin{aligned}
ident\_nad &::= \mathrm{artikli}.ident \\
ident\_pod &::= \mathrm{artikli}.ident \\
ident &::= \mathrm{operacije}.ident \\
redosled &::= \langle \mathrm{NUMERIC}, 5 \rangle \\
napomena &::= \langle \mathrm{CHAR}, 40 \rangle \\
vremeprip &::= \langle \mathrm{NUMERIC}, 10.4 \rangle \\
vremeizr &::= \langle \mathrm{NUMERIC}, 10.4 \rangle \\
zahvat &::= \langle ident\_nad, ident\_pod, ident, redosled, napomena, vremeprip, vremeizr, * \rangle \\
tehnologija &:= \{ zahvat : \mathrm{PRIMARY\ KEY\ IS}\ ident\_nad + ident\_pod + redosled \}
\end{aligned} \tag{5.5}
$$

gde je:

vremeprip pomoćno mašinsko vreme

vremeizr tehnološko mašinsko vreme

Ova vremena su detaljno objašnjena u Dodatku C. U nastavku pretpostavimo da u tabeli *tehnologija* postoje sledeći zapisi (5.1):
Zapisi (5.1) su karakteristični za sklopove koji nemaju podređene, ali se nad njima moraju izvesti određene operacije. Zapisi (5.2) su karakteristični za sklopove koji se nalaze visoko u sastavnicama i nad njima se moraju izvesti određene operacije tako da su u pogledu zahvata sami sebi podsklopovi.
Konačno zapisi (5.3) opisuju nizove zahvata koji se odnose na sklop koji ima dva podređena sklopa.

5.3.2 Određivanje maksimalnog vremena

Prvi korak je određivanje maksimalnog vremena trajanja pojedine operacije:

$$maxvreme = vremeprip + vremeizr$$

Maksimalna vremena pojedinih operacija se mogu dobiti jednostavnim upitom, u slučaju operacija datih u (5.1) rezultat je sledeći (5.4):

ident_n	**ident_p**	**operacija**	**redosled**	**vremeprip**	**vremeizr**	**maxvr**
00011	-	00010	1	20	0.20	20.20
00011	-	00011	2	20	0.20	20.20
00011	-	00033	3	-	-	0
00011	-	00021	4	30	0.60	30.60
00011	-	00031	5	-	-	0
00011	-	00032	6	30	0.05	30.05
00011	-	00022	7	30	1.20	31.20
00011	-	00033	8	-	-	0

Tabela 5.4: Maksimalno vreme operacije

5.3.3 Određivanje vremena kompletiranja

Neka je maksimalno vreme operacije određeno na prethodno opisani način, vreme kompletiranja niza operacija se može odrediti na sledeći način:

$$vremenorm = \sum maxvreme$$

U slučaju (5.4), nakon grupisanja na nivou sklopa imaćemo (5.5):

Tabela 5.5: Vreme kompletiranja

ident_n	**ident_p**	**vremenorm**
00011	-	132.25

što je dovoljno za dobijanje neophodnog izveštaja. Dakle možemo pretpostaviti postojanje niza zapisa oblika (5.6):

Tabela 5.6: Vreme kompletiranja

ident_n	ident_p	vremenorm
00001	00001	256.00
00002	00002	47.60
00002	00008	252.20
00002	00009	61.00
00003	00003	47.60
00003	00009	71.20
00003	00010	242.00
00004	-	170.10
00005	00005	316.50
00011	-	132.25

5.3.4 Algoritam proračuna vremena

Iskoristimo prvu hijerarhijsku sastavnicu datu u (4.25), odnosno pretpostavimo postojanje zapisa (5.7):

Tabela 5.7: GBOM

nivo	ident_n	ident_p
1	00001	00002
2	00002	00008
3	00008	-
2	00002	00009
3	00009	-
1	00001	00003
2	00003	00009
3	00009	-
2	00003	00010
3	00010	-
1	00001	00004
2	00004	-
1	00001	00005
2	00005	00011
3	00011	-
1	00001	00006
2	00006	-
1	00001	00007
2	00007	-

Na osnovu rezultata iz (5.6) proračun vremena se može dobiti na sledeći način:

Prvi korak Terminalnim cvorovima se dodeljuju odgovarajuća vremena koja se preuzimaju iz (5.6), rezultat ove obrade je (5.8):

Tabela 5.8: GBOM-nivo 3

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	00002	
2	00002	00008	
3	00008	-	0.00
2	00002	00009	
3	00009	-	0.00
1	00001	00003	
2	00003	00009	
3	00009	-	0.00
2	00003	00010	
3	00010	-	0.00
1	00001	00004	
2	00004	-	170.10
1	00001	00005	
2	00005	00011	
3	00011	-	132.25
1	00001	00006	
2	00006	-	0.00
1	00001	00007	
2	00007	-	0.00

Drugi korak Iterativno se vremena izračunavaju na višim nivoima, tako što se svakoj grani na datom nivou pridruži zbir vremena sa same grane iz (5.6) i vremena pridruženog podsklopu na datoj grani. U navedenom primeru dobićemo (5.9):

Tabela 5.9: GBOM-nivo 2a

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	00002	
2	00002	00008	252.20+0.00
2	00002	00009	61.00+0.00
1	00001	00003	
2	00003	00009	71.20+0.00
2	00003	00010	242.00+0.00
1	00001	00004	
2	00004	-	170.10
1	00001	00005	
2	00005	00011	0.00+132.25
1	00001	00006	
2	00006	-	0.00
1	00001	00007	
2	00007	-	0.00

Zatim se određuje vreme koje odgovara nadređenom sklopu na datom nivou i ono je jednako maksimalnom od svih vremena koja odgovaraju granama u kojima je dati sklop nadređeni, odnosno (5.10):

Tabela 5.10: GBOM-nivo 2b

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	00002	
2	00002	-	$\max\{252.20, 61.00\}$
1	00001	00003	
2	00003	-	$\max\{71.20, 242.00\}$
1	00001	00004	
2	00004	-	$\max\{170.10\}$
1	00001	00005	
2	00005	-	$\max\{132.25\}$
1	00001	00006	
2	00006	-	$\max\{0.00\}$
1	00001	00007	
2	00007	-	$\max\{0.00\}$

Jedan korak iteracije se završava tako što se odgovarajuće vreme koje odgovara sklopu na datom nivou uvećava za vreme koje odgovara operaciji u kojoj je dati sklop istovremeno i nadređeni i podređeni, a sadržano je u (5.6). U datom primeru imaćemo

(5.11):

Tabela 5.11: GBOM-nivo 2c

nivo	ident_n	ident_p	vreme
1	00001	00002	
2	00002	-	252.20+47.60
1	00001	00003	
2	00003	-	242.00+47.60
1	00001	00004	
2	00004	-	170.10+ 0.00
1	00001	00005	
2	00005	-	132.25+316.50
1	00001	00006	
2	00006	-	0.00+ 0.00
1	00001	00007	
2	00007	-	0.00+ 0.00

Dakle posle jedne iteracije dobićemo raspodelu vremena na drugom nivou (5.12):

Tabela 5.12: GBOM-nivo 2d

nivo	ident_n	ident_p	vreme
1	00001	00002	
2	00002	-	299.80
1	00001	00003	
2	00003	-	289.60
1	00001	00004	
2	00004	-	170.10
1	00001	00005	
2	00005	-	448.75
1	00001	00006	
2	00006	-	0.00
1	00001	00007	
2	00007	-	0.00

Sledeća iteracija daje proračun vremena na prvom nivou, odnosno maksimalno vreme sklapanja finalnog sklopa 00001. Tok obrade dat je sledećim tabelama (5.13), (5.14) i (5.15):

Tabela 5.13: GBOM-nivo 1a

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	00002	299.80+0.00
1	00001	00003	289.60+0.00
1	00001	00004	170.10+0.00
1	00001	00005	448.75+0.00
1	00001	00006	0.00+0.00
1	00001	00007	0.00+0.00

Tabela 5.14: GBOM-nivo 1b

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	-	$\max\{299.80, 289.60, 170.10, 448.75, 0.00, 0.00\}$

Tabela 5.15: GBOM-nivo 1c

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	-	448.75+256.00

Konačno dobijamo (5.16):

Tabela 5.16: GBOM-nivo 1d

nivo	**ident_n**	**ident_p**	**vreme**
1	00001	-	704.75

Tabela 5.1: Operacije nad sklopom I

ident_n	**ident_p**	**operacija**	**redosled**	**napomena**	**vremeprip**	**vremeizr**
00011	-	00010	1	Sečenje	20	0.20
00011	-	00011	2	Pomoćnik pri sečenju	20	0.20
00011	-	00033	3	Kontrola sečenja	-	-
00011	-	00021	4	Isecanje uglova	30	0.60
00011	-	00031	5	Reglaža alata	-	-
00011	-	00032	6	Probijanje	30	0.05
00011	-	00022	7	Savijanje	30	1.20
00011	-	00033	8	Kontrola izrade komada	-	-

Tabela 5.2: Operacije nad sklopom II

ident_n	**ident_p**	**operacija**	**redosled**	**napomena**	**vremeprip**	**vremeizr**
00005	00005	00001	1	Hemijska obrada	30	1
00005	00005	00023	2	Izrada lamela	30	3.5
00005	00005	00024	3	Izrada cevčica	90	3
00005	00005	00025	4	Sastavljanje saća	30	18
00005	00005	00026	5	Pečenje saća	-	-
00005	00005	00027	6	Ugradnja cevnih zidova	30	8
00005	00005	00028	7	Kapilarno lemljenje cevnih zidova	30	5
00005	00005	00029	8	Pranje bloka saća	-	-
00005	00005	00030	9	Ispitivanje bloka saća	30	8
00005	00005	00033	10	Kontrola u izradi bloka saća	-	-

Tabela 5.3: Operacije nad sklopom III

ident_n	**ident_p**	**operacija**	**redosled**	**napomena**	**vremeprip**	**vremeizr**
00002	00008	00010	1	Sečenje	20	0.20
00002	00008	00011	2	Pomoćnik pri sečenju	20	0.20
00002	00008	00033	3	Kontrola sečenja	-	-
00002	00008	00012	4	Isecanje	30	0.80
00002	00008	00013	5	Opsecanje	30	3
00002	00008	00033	6	Kontrola opsecanja	-	-
00002	00008	00014	7	Savijanje	30	1.5
00002	00008	00015	8	Savijanje i doterivanje	30	4
00002	00008	00016	9	Zavarivanje	15	4
00002	00008	00033	10	Kontrola doterivanja i zavarivanja	-	-
00002	00008	00017	11	Frezovanje	30	1.50
00002	00008	00018	12	Pertlovanje	30	2
00002	00008	00033	13	Kontrola izrade komada	-	-
00002	00009	00019	1	Sečenje cevi	30	0.50
00002	00009	00033	2	Kontrola sečenja	-	-
00002	00009	00020	3	Siknovanje	30	0.50
00002	00009	00033	4	Kontrola izrade komada	-	-

Glava 6

Zaključak

Analizom kako praktičnih rešenja, tako i teorijskih istraživanja u sistemu održavanja vazduhoplova, može se zaključiti da ima prostora za značajnija unapređenja u ovoj oblasti. Savremeni sistemi održavanja obuhvataju i upravljanje održavanjem, preciznije zasnovani su na automatizovanoj podršci održavanju primenom računara (*Computer Integrated Maintenance Management*). Samim tim jedan od važnih zahteva je obezbediti model podataka koji je dovoljno bogat da bi mogao, na relativno lak način prihvatiti i u sebe uključiti različite koncepte realnog sistema zajedno sa svojim atributima i međusobnim vezama. Hijerarhijske strukture podataka (strukture podataka tipa stablo) su posebno interesantne u sistemu održavanja vazduhoplova i one su predmet istraživanja u Glavi 2. Rekurzivnost koja je u osnovi hijerarhijske strukture podataka odnosno stabla koji je reprezent ovakve strukture podataka, može biti problem u efikasnom upravljanju sistemom održavanja vazduhoplova. U Glavi 2 su date nove karakterizacije stabla, koje daju osnov za nerekurzivne modele hijerarhijske strukture podataka:

– *model sa izlistanim putanjama*

– *model sa ugnježdenim skupovima*

– *model sa ugnježdenim intervalima*

Teoreme (2.3.2) i (2.3.3) odnosno (2.3.6) i (2.3.7) daju vezu između proizvoljne hijerarhijske strukture i hijerarhijske strukture uređene relacijom podskup odnosno nadskup. Drugim rečima svako stablo je izomorfno stablu koje je uređeno relacijom podskup odnosno nadskup. Pogodnim indeksiranjem čorova stabla u skupu prirodnih brojeva, relacijama podskup, odnosno nadskup će odgovarati uobičajeno leksikografsko uređenje nizova prirodnih brojeva (*model sa izlistanim putanjama*), odnosno uobičajeno uređenje intervala prirodnih brojeva (*model sa ugnježdenim skupovima*). Na ovaj način se pojedine operacije nad hijerarhijskom strukturom podataka mogu efikasno implementirati. Slabosti pomenuta dva modela se mogu otkloniti, ako se čvorovi stabla indeksiraju u skupu intervala realnih brojeva. Posebno *model sa ugnježdenim intervalima* otklanja neke od uočenih slabosti. U ovoj klasi modela analizirana su dva tipa:

– *binarni model sa ugnježdenim intervalima*

– *ternarni model sa ugnježdenim intervalima*

Analizom slabosti binarnog modela sa ugnježdenim intervalima, koji ne podržava direktno operaciju umetanja čvora, predložen je ternarni model sa ugnježdenim intervalima koji pomenutu operaciju direktno podržava.

U kontekstu ograničenih resursa u sistemu za održavanje vazduhoplova posebno su interesantne optimalne strategije održavanja. Bitan segment sistema za upravljanje održavanjem je planiranje održavanja. Jedan od osnovnih problema između ostalog je neizvesnost događaja koji značajno utiču na samo planiranje. Reč je o događajima koji su po svojoj prirodi slučajni. Ovi slučajni događaji su pored ostalog vezani za:

– vremenski trenutak otkaza rezervnog dela

– vremenski trenutak zahteva za nekim delom

– vreme isporuke, odnosno vreme potrebno za sklapanje (pripremu) rezervnog dela

Planiranje održavanja u kontekstu slučajnih faktora je predmet istraživanja u Glavi 3. Posebno analizira se planiranje održavanja u slučaju kada skup aktivnosti ima hijerarhijsku strukturu i pri tome je trajanje pojedinih aktivnosti iz datog skupa aktivnosti slučajna veličina. Pod pretpostavkom da aktivnost na višem nivou hijerarhije ne može započeti pre nego što su završene sve aktivnosti na nižem nivou hijerarhije i da su poznati gubici koji nastaju u slučaju da se sa nekim aktivnostima kasni, odnosno da su poznati gubici u slučaju da su neke aktivnosti preuranjene, potrebno je formirati optimalnu strategiju aktiviranja aktivnosti iz datog skupa. Bez gubitka opštosti pretpostavljeno je da su aktivnosti iz skupa aktivnosti zapravo aktivnosti sklapanja (pripreme) rezervnog dela. Pod ovim pretpostavkama funkcija koju treba optimizoati je funkcija očekivanih troškova i ona uključuje troškove:

– skladištenja u slučaju da je aktivnost sklapanja završena pre utvrđenog roka

– kašnjenja u slučaju da je aktivnost sklapanja završena nakon utrđenog roka

U prvom delu Glave 3 problem planiranja aktivnosti u slučaju jedne linije sklapanja je rešen svođenjem na poznati *newsboy problem.* Zatim je dato rešenje problema višestruke neodređenosti na jednoj grani hijerarhije, kada su i trajanje aktivnosti sklapanja i utvrđeni rok slučajne veličine. Što se tiče problema sklapanja (pripreme) rezervnog dela u opštem slučaju, kada se radi o hijerarhiji sa proizvoljnim brojem nivoa, on je isuviše komplikovan da bi se postavila funkcija očekianih troškova. U slučaju hijerarhije koja se opisuje stablom visine jedan, predložena su dva modela:

– simultani zahtev I

– simultani zahtev II

Za svaki od modela je formirana funkcija očekivanih troškova i dokazana njena konveksnost (Teorema (3.3.1) i Teorema (3.4.1)) čime se određivanje optimalnog rešenja svodi na određivanje lokalnog minimuma funkcije očekianih troškova. Dalje je pokazano kako se rešenje problema na stablu sklapanja rezervnog dela visine dva, može produžiti na stablo sklapanja proizvoljne visine. Ovo produženje se svodi na iterativnu primenu rešenja na stablu sklapanja visine dva. Na kraju su date procedure koje (obradom odgovarajuće hijerarhijske strukture podataka) daju optimalna vremena aktiviranja pojedinih aktivnosti sklapanja. Jasno je da se navedeni rezultati mogu primeniti na planiranje proizvoljnog skupa aktivnosti koji ima hijerarhijsku strukturu.

Vazduhoplovi su savremena saobraćajna sredstva vrhunske tehnologije sa velikim brojem komponenti i kompleksnom strukturom. Pojedine komponente vazduhoplova se koriste na različit način, trpe različita opterećenja i održavaju se po programu koji je njima najbolje prilagođen. Jedan od načina savladavanja složenosti je takozvana hijerarhijska dekompozicija. Složeni koncept se na jednom nivou apstrakcije posmatra kao jedinstvena celina. Na nižem nivou apstrakcije se posmatra kao koncept koji se sastoji od delova (komponenti). Uzastopnom primenom opisanog postupka dobija se takozvana hijerarhijska *sastavnica*. Dakle jedan od bitnih koncepata u sistemu održavanja vazduhoplova jeste sastavnica. Sastavnica je najprostije rečeno, lista komponenti koji su sastavni delovi posmatranog objekta. Pri tome se vodi računa o vezama između sastavnih komponenti, pa je u osnovi sastavnice hijerarhijska struktura podataka. Sastavnica kao hijerarhijska struktura podataka u sistemu održavanja vazduhoplova je predmet istraživanja u Glavi 4. U Glavi 2 su analizirani neki modeli hijerarhijske strukture podataka. U analizi modela sva pažnja je bila skoncentrisana na jedan aspekt modela, konkretno analizirana je efikasnost obrade hijerarhijske strukture podataka u datom modelu. Drugi problem je obezbediti kvalitetan sadržaj hijerarhijske strukture podataka, odnosno u sistemu za upravljanje održavanjem vazduhoplova treba obezbediti kvalitetnu sastavnicu. Sastavnice predstavljaju posebnu vrstu stabla i u suštini su hijerarhijske strukture podataka koje su osnov za modeliranje problema iz prethodnih poglavlja, samim tim su posebno interesantne u sistemu za održavanje vazduhoplova. U Glavi 4 daje se metodički pristup rešavanju problema generisanja sadržajno kvalitetne sastavnice. Pre svega neophodno je sistematizovati podatke koji bi bili podrška kreiranju sastavnice. Pomenuta sistematizacija podataka je izvedena navođenjem opisa više struktura podataka, čije se postojanje dalje podrazumeva jer obezbeđuju sadržaj kojim će se sastavnice snabdeti. Dalje se daje opis strukture sastavnice koji određuje sadržaj same sastavnice. Pri tome se razlikuju dva tipa sastavnica:

– *modularna*

– *hijerarhijska*

i oba tipa su detaljno obrađeni. Predložena struktura sastavnica je fleksibilna i ta fleksibilnost se ogleda u tome što se može elementarno proširiti i na taj način obezbediti dodatni sadržaj. U slučaju modularnih sastavnica dat je primer kako se može modelirati uređenje potomaka čvora stabla, koje se dalje u sistemu održavanja vazduhoplova može interpretirati kao redosled ugradnje delova ili još više kao redosled izvođenja aktivnosti održavanja. Slično dat je primer kako se može modelirati višestruka podređenost pojedinih čvorova stabla koja se dalje u sistemu održavanja može interpretirati kao neophodna količina pojedinih podsklopova za sklapanje nadređenog sklopa. Kako su modularne sastanice gradivni blokovi hijerarhijske sastavnice, fleksibilnost strukture modularne sastavnice se prenosi na hijerarhijsku sastavnicu. Na kraju Glave 4 uvodi se pojam generičke sastavnice, a zatim se daje odgovarajuća procedura za generisanje hijerarhijske sastavnice.

Imajući za osnov fleksibilnu strukturu sastavnice, u Glavi 5 se uvodi pojam druge hijerarhijske sastavnice. Druga hijerarhijska sastavnica je sadržajno raširenje hijerarhijske sastavnice uvedene u Glavi 4. Posebno druga hijerarhijska sastavnica obezbeđuje sadržaj kojim se opisuje postupak simultane ugradnje podsklopova u nadređeni sklop o čemu je bilo reči u Glavi 3, s tom razlikom što je u Glavi 3 ova problematika istraživana sa aspekta optimizacije troškova. Sam postupak ugradnje se uobičajenom tehnikom dekompozicije, raščlanjuje na niz zahvata koji se zatim modeliraju u strukturi druge hijerarhijske sastavnice. Zahvati bliže određuju prirodu veze između nadređenog i podređenog sklopa. Nakon što je izvedena studija slučaja koji se koriste u praksi, prepoznata su četri tipa zahvata prema tome šta su krajevi veze između nadređenog i podređenog sklopa:

– *zahvat ima nadređeni i nema podređeni sklop*

– *zahvat ima nadređeni i podređeni sklop i pri tome su to isti sklopovi*

– *zahvat ima nadređeni i podređeni sklop i to su različiti sklopovi*

– *zahvat ima podređeni i nema nadređeni sklop*

Zatim je data odgovarajuća procedura za generisanje druge hijerarhijske sastavnice, koja na odgovarajući način tretira prepoznate tipove zahvata. Na kraju je dat algoritam za proračun neophodnih vremena u postupku simaultane ugradnje na više nivoa.

U daljem radu svakako bi se išlo na softversku implementaciju predloženih modela stabla. Ocigledno je da se predloženim modelima hijerarhijska struktura podataka može implementirati linearnom strukturom podataka kojima su pridružena odgoarajuća obeležja. Komplikovan hijerarhijski odnos između elemenata strukture se prenosi na elementaran odnos između obeležja elemenata linearne strukture. Što se tiče održavanja vazduhoplova potrebno je testirati predložene modele planiranja održavanja. Pre svega neophodno je obezbediti odgovarajuće raspodele vremena trajanja pojedinih aktivnosti održavanja, a to se može izvesti nekim od standardnih metoda statističke obrade podataka.

Literatura

[1] I. Arandjelović, Z. Mitrović, and V. Stojanović. *Verovatnoća i Statistika*. Zavod za udžbenike, Beograd, 2011.

[2] Sven Axsater. Planning order releases for an assembly system with random operation times. *OR Spectrum*, Vol. 27(2-3):459–470, 2005.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 7th edition, 2009.

[4] Franjo Cecelja. *Manufacturing Information and Data SystemsM*. Penton Press, London, 2002.

[5] Joe Celko. *Joe Celko's Trees and hierarchies in SQL for smarties*. Morgan-Kaufmann, 2004.

[6] Fangruo Chen. Market segmentation, advanced demand information, and supply chain performance. *Manufacturing & Service Operations Management*, Vol. 3(1):53–67, 2001.

[7] Vadim Tropashko Oracle Corp. Nested intervals tree encoding in sql. *SIGMOD Record*, Vol. 34(2):47–52, 2005.

[8] Mohsen Elhafsi. Optimal leadtimes planning in serial production systems with earliness and tardiness costs. *IIE Transactions*, Vol. 34(3):233–243, 2002.

[9] W. Hopp and M. Spearman. Setting safety leadtimes for purchased components in assembly systems. *IIE Transactions*, Vol. 25(2):1–11, 1993.

[10] F. Karaesmen, J.A. Buzacott, and Y. Dallery. Integrating advance order information in make-to-stock production systems. *IIE Transactions*, Vol. 34(8):649–662, 2002.

[11] Donald E. Knuth. *The Art of Computer Programming*, volume 1. Addison-Wesley, Reading, 1969.

[12] Č. Mitrović, D. Bekrić, N. Petrović, and S. Radojević. Relevantnost ljudskih faktora u vazduhoplovstvu. *XXXII Naučno - stručni skup Održavanje mašina i opreme*, pages 209–220, 2007.

[13] I. Moerdijk and J. Oosten. *Sets, Models and Proofs*. Utrecht University, 2006.

[14] Pavle Mogin. *Strukture podataka i organizacija datoteka*. Računarski fakultet, Beograd, 2008.

[15] John Moubray. *Reliability Centred Maintenance RCM.* Butterworth-Heinnemann Ltd., Oxford, 2nd edition, 1997.

[16] S. Radojević, Č. Mitrović, and M. Aranđelović. Priprema sastavnica u održavanju. *XXXIII Naučno – stručni skup Održavanje mašina i opreme*, 2008.

[17] S. Radojević, Č. Mitrović, M. Aranđelović, and N. Dondur. Maksimalno vreme održavanja aviona dobijeno sastavnicom. *XXXIII Naučno – stručni skup Održavanje mašina i opreme*, 2008.

[18] S. Radojević, Č. Mitrović, M. Lečić, and N. Dondur. Edukacija korisnika za korišćenje ebom case alata. *33. JUPITER konferencija, 26. CIM u strategiji tehnološkog razvoja industrije prerade metala, Zbornik radova*, Vol. 1.

[19] S. Radojević and Z.A. Veljković. Rekurzivni algoritam za stvaranje prve hijerarhijske sastavnice. *28. JUPITER konferencija, 30. simpozijum Upravljanje proizvodnjom u industriji prerade metala, Zbornik radova*, Vol. 4.

[20] S. Radojević and Z.A. Veljković. Osnovne pretpostavke za stvaranje meta-jezika formiranja sastavnica. *SIE 98*, pages 231–234, 1998.

[21] S. Radojević, Z.A. Veljković, and B. Milinković Katalinić. Tehnološka sastavnica u planiranju proizvodnje. *Drugi skup privrednika i naučnika, Beograd*, 2004.

[22] Slobodan Radojević. Formiranje sastavnica gotovih proizvoda simuliranom rekurzijom i pokazivačima u relacionim bazama podataka. *Zbornik radova Mašinskog Fakulteta*, pages 40–43, 1991.

[23] Slobodan Radojević. Algoritam za formiranje sastavnica gotovih proizvoda u relacioniim bazama podataka. *Tehnika – Organizacija*, pages 46–50, 1994.

[24] Boško Rašuo. *Vazduhoplovnotehnicko obezbedenje.* Vojna akademija, Beograd, 2003.

[25] Walter Rudin. *Principles of mathematical analysis.* McGraw-Hill, Inc., 3rd edition, 1976.

[26] J.S. Song, C.A. Yano, and P. Lerssrisuriya. Contract assembly: Dealing with combined supply lead time and demand quantity uncertainty. *Manufacturing & Service Operations Management 2*, Vol. 2(3):287–296, 2000.

[27] D. Stevanović, M. Ćirić, S. Simić, and V. Baltić. *Discrete Mathematics – Fundamentals of Combinatorics and Graph Theory.* Mathematical Society of Serbia, Beograd, 2007.

[28] William J. Stevenson. *Operations Management.* McGraw-Hill/Irwin, 10th edition, 2009.

[29] O. Tang and R.W. Grubbstrom. The detailed coordination problem in a two-level assembly system with stochastic lead times. *International Journal of Production Economics*, Vol. 81-82(1):415–429, 2003.

[30] Candace A. Yano. Stochastic leadtimes in two-level assembly systems. *IIE Transactions*, Vol. 19(4):371–378, 1987.

Prilog A

Newsboy problem

Problem je sledeći, potrebno je oceniti broj primeraka dnevnih novina koje treba naručiti imajući u vidu da je potražnja slučajna promenljiva i da na kraju dana neprodate novine gube vrednost . Neka je nabavna cena 10din., a prodajna 20din. i neka se za povraćaj neprodatog primerka novina dobija 5din. Očigledno je da je rešenje po kome je broj primeraka koje treba naručiti jednak srednjoj vrednosti broja prodatih primeraka na dnevnom nivou neodgovarajuće. Ovaj zaključak se može izvesti iz sledeće analize:

– višak primerka novina koji je neporodat, proizvodi gubitak od

$$10\text{din} - 5\text{din} = 5\text{din}$$

– manjak primerka novina koji je mogao biti prodat, proizvodi gubitak od

$$20\text{din} - 10\text{din} = 10\text{din}$$

Dakle gubitak je veći ako se ne naruči dovoljan broj primeraka nego ako se naruči više primeraka nego što je neophodno. U slučaju da se naručuje srednja vrednost broja dnevno prodatih primeraka, sa istom frekvencijom će se dešavati situacije kada nema dovoljno primeraka i situacije kada ima višak primeraka. Heuristička analiza nalaže da frekvenciju situacije kada nema dovoljno primeraka treba smanjiti u korist ove druge, odnosno treba naručivati više od srednje vrednosti broja dnevno prodatih primeraka. Ovo je primer newsboy problema u kome se dati proizvod naručuje na početku izvesnog perioda i može se koristiti samo u toku tog perioda, a njegovim rešavanjem se dobija optimalni nivo zaliha. Pretpostavimo da su poznate sledeće cene koštanja $c_o, c_u > 0$:

– c_o : cena koštanja po jedinici proizvoda u slucaju pozitivnog nivoa zaliha na kraju perioda u kome je proizvod upotrebljiv (*overage cost*)

– c_u : cena koštanja nezadovoljenog zahteva, odnosno cena koštanja po jedinici proizvoda u slucaju negativnog nivoa zaliha na kraju perioda u kome je proizvod upotrebljiv (*underage cost*)

Dalje ćemo pretpostaviti da je broj zahteva D u toku uočenog perioda slučajna promenljiva. Upravljajući parameter q je broj jedinica proizvoda koji se naručuje na početku perioda. Cilj je odrediti vrednost parametra q koji minimizira očekivane ukupne troškove na kraju perioda. Definišimo sa $C(q, D)$ funkciju ukupnih troškova koji su posledica pozitivnog odnosno negativnog nivoa zaliha na kraju perioda:

$$C(q, D) = c_o \cdot \max\{q - D, 0\} + c_u \cdot \max\{D - q, 0\} \tag{A.1}$$

Ako uvedemo sledeće oznake:

$$x^+ = \max\{x, 0\} \tag{A.2}$$
$$x^- = \max\{-x, 0\} = (-x)^+ \tag{A.3}$$

imaćemo:

$$C(q, D) = c_o \cdot (q - D)^+ + c_u \cdot (D - q)^+ \tag{A.4}$$

Kako važi:

$$\begin{aligned} x^+ &= \max\{x, 0\} = x + \max\{x - x, 0 - x\} = x + \max\{0, -x\} \\ &= x + x^- = x + (-x)^+ \end{aligned} \tag{A.5}$$

dobijamo:

$$\begin{aligned} C(q, D) &= c_o \cdot \big((q - D) + (D - q)^+\big) + c_u \cdot (D - q)^+ \\ &= c_o \cdot (q - D) + (c_u + c_o) \cdot (D - q)^+ \end{aligned} \tag{A.6}$$

pa su očekivani troškovi jednaki:

$$Z(q) = E(C(q, D)) = c_o \cdot (q - E(D)) + (c_u + c_o) \cdot E((D - q)^+) \tag{A.7}$$

Teorema A.1. *Neka je broj zahteva D u toku uočenog perioda diskretna slučajna promenljiva sa funkcijom raspodele $F(d) = P(D < d)$, tada je optimalno rešenje problema (A.7) dato sa:*

$$F(q^*) \leq \frac{c_u}{c_u + c_o} \leq F(q^* + 1) \tag{A.8}$$

Dokaz. *Neka je broj zahteva D u toku uočenog perioda diskretna slučajna promenljiva sa funkcijom raspodele $F(d) = P(D) < d)$, tada je funkcija očekivanih ukupnih troškova:*

$$
\begin{aligned}
Z(q) &= c_o \cdot (q - E(D) + (c_u + c_o) \cdot \sum_{i \geq 1} P((D-q)^+ \geq i) \\
&= c_o \cdot (q - E(D) \ + (c_u + c_o) \cdot \sum_{i \geq 1} P(D \geq q + i) \\
&= c_o \cdot (q - E(D) + (c_u + c_o) \cdot \sum_{j \geq q+1} P(D \geq j) \\
&= c_o \cdot (q - E(D) + (c_u + c_o) \cdot (E(D) - \sum_{j=1}^{q} P(D \geq j)) \\
&= c_o q + c_u E(D) - (c_u + c_o) \cdot \sum_{j=1}^{q} (1 - F(j)) \\
&= c_o q + c_u E(D) - (c_u + c_o) q + (c_u + c_o) \cdot \sum_{j=1}^{q} F(j) \\
&= c_u (E(D) - q) + (c_u + c_o) \cdot \sum_{j=1}^{q} F(j)
\end{aligned}
$$

a njen priraštaj je:

$$
\begin{aligned}
\Delta Z(q) &= Z(q+1) - Z(q) \\
&= c_u \cdot (E(D) - (q+1)) + (c_u + c_o) \cdot \sum_{j=1}^{q+1} F(j) \\
&\quad - \left(c_u \cdot (E(D) - q) + (c_u + c_o) \cdot \sum_{j=1}^{q} F(j) \right) \\
&= -c_u + (c_u + c_o) \cdot F(q+1)
\end{aligned}
$$

q^* *je optimalno rešenje ako je:* $\Delta Z(q^* - 1) \leq 0$ *i* $\Delta Z(q^*) \geq 0$, *odnosno:*

$$
\begin{aligned}
-c_u + (c_u + c_o) F(q^*) &\leq 0 \\
-c_u + (c_u + c_o) F(q^* + 1) &\geq 0
\end{aligned}
$$

pa je optimalno rešenje dato sa:

$$
F(q^*) \leq \frac{c_u}{c_u + c_o} \leq F(q^* + 1)
$$

◇

Teorema A.2. *Neka je* D *neprekidna slučajna promenljiva sa gustinom raspodele* f *i funkcijom raspodele* F, *tada je optimalno rešenje problema (A.7) dato sa:*

$$
q^* = F^{-1} \left(\frac{c_u}{c_{u+c_o}} \right) \tag{A.9}
$$

Dokaz. *Neka je broj zahteva D u toku uočenog perioda neprekidna slučajna promenljiva sa gustinom raspodele f i funkcijom raspodele F, tada su očekivani ukupni troškovi:*

$$
\begin{aligned}
Z(q) &= c_o \cdot (q - E(D)) + (c_u + c_o) \cdot \int_0^{\infty} P\left(D - q \geq x\right) \mathrm{d}x \\
&= c_o \cdot (q - E(D)) + (c_u + c_o) \cdot \int_0^{\infty} (1 - P(D < x + q)) \mathrm{d}x \\
&= c_o \cdot (q - E(D)) + (c_u + c_o) \cdot \int_0^{\infty} (1 - F(x + q)) \mathrm{d}x \\
&= c_o \cdot (q - E(D)) + (c_u + c_o) \cdot \int_q^{\infty} (1 - F(x)) \mathrm{d}x
\end{aligned}
$$

Stacionarna tačka funkcije očekivanih troškova je rešenje sledeće jednačine:

$$
\begin{aligned}
\frac{\mathrm{d}Z(q)}{\mathrm{d}q} &= 0 \\
c_o - (c_u + c_o) \cdot \int_q^{\infty} f(x) \mathrm{d}x &= 0 \\
c_o - (c_u + c_o)(1 - F(q)) &= 0 \\
-c_u + (c_u + c_o) F(q) &= 0 \\
F(q) &= \frac{c_u}{c_u + c_o}
\end{aligned}
$$

Kako je:

$$
\frac{\mathrm{d}^2 Z(q)}{\mathrm{d}q^2} = (c_u + c_o) f(q) \geq 0
$$

funkcija očekivanih ukupnih troškova je konveksna. Neka je pri tome F strogo rastuća funkcija, tada je jedinstveno optimalno rešenje dato sa:

$$
q^* = F^{-1}\left(\frac{c_u}{c_{u + c_o}}\right)
$$

◇

Newsboy model se može koristiti u svim situacijama kada treba izvršiti procenu neke slučajne veličine, a ta procena je rezultat kompromisa između gubitaka u slučaju da je vrednost slučajne promenljive precenjena i gubitaka koji su posledica potcenjene vrednosti slučajne promenljive.

Prilog B

HSII

2.1 HSII1

$$
\begin{aligned}
P = \{ & \\
& \langle 00001, 00001, 00001, 1\rangle, \\
& \langle 00001, 00001, 00002, 2\rangle, \\
& \langle 00001, 00001, 00003, 3\rangle, \\
& \langle 00001, 00001, 00004, 4\rangle, \\
& \langle 00001, 00001, 00005, 5\rangle, \\
& \langle 00001, 00001, 00006, 6\rangle, \\
& \langle 00001, 00001, 00007, 7\rangle, \\
& \langle 00001, 00001, 00008, 8\rangle, \\
& \langle 00001, 00001, 00033, 9\rangle, \\
& \langle 00002, 00002, 00001, 1\rangle, \\
& \langle 00002, 00002, 00009, 2\rangle, \\
& \langle 00002, 00002, 00033, 3\rangle, \\
& \langle 00002, 00008, 00010, 1\rangle, \\
& \langle 00002, 00008, 00011, 2\rangle, \\
& \langle 00002, 00008, 00033, 3\rangle, \\
& \langle 00002, 00008, 00012, 4\rangle, \\
& \langle 00002, 00008, 00013, 5\rangle, \\
& \langle 00002, 00008, 00033, 6\rangle, \\
& \langle 00002, 00008, 00014, 7\rangle, \\
& \langle 00002, 00008, 00015, 8\rangle, \\
& \langle 00002, 00008, 00016, 9\rangle, \\
& \langle 00002, 00008, 00033, 10\rangle,
\end{aligned}
$$

$\langle 00002, 00008, 00017, 11\rangle$,

$\langle 00002, 00008, 00018, 12\rangle$,

$\langle 00002, 00008, 00033, 13\rangle$,

$\langle 00002, 00009, 00019, 1\rangle$,

$\langle 00002, 00009, 00033, 2\rangle$,

$\langle 00002, 00009, 00020, 3\rangle$,

$\langle 00002, 00009, 00033, 4\rangle$,

$\langle 00003, 00003, 00001, 1\rangle$,

$\langle 00003, 00003, 00009, 2\rangle$,

$\langle 00003, 00003, 00033, 3\rangle$,

$\langle 00003, 00009, 00019, 1\rangle$,

$\langle 00003, 00009, 00033, 2\rangle$,

$\langle 00003, 00009, 00020, 3\rangle$,

$\langle 00003, 00009, 00033, 4\rangle$,

$\langle 00003, 00010, 00010, 1\rangle$,

$\langle 00003, 00010, 00011, 2\rangle$,

$\langle 00003, 00010, 00033, 3\rangle$,

$\langle 00003, 00010, 00012, 4\rangle$,

$\langle 00003, 00010, 00013, 5\rangle$,

$\langle 00003, 00010, 00033, 6\rangle$,

$\langle 00003, 00010, 00014, 7\rangle$,

$\langle 00003, 00010, 00015, 8\rangle$,

$\langle 00003, 00010, 00016, 9\rangle$,

$\langle 00003, 00010, 00033, 10\rangle$,

$\langle 00003, 00010, 00017, 11\rangle$,

$\langle 00003, 00010, 00018, 12\rangle$,

$\langle 00003, 00010, 00033, 13\rangle$,

$\langle 00004, 00000, 00010, 1\rangle$,

$\langle 00004, 00000, 00011, 2\rangle$,

$\langle 00004, 00000, 00033, 3\rangle$,

$\langle 00004, 00000, 00021, 4\rangle$,

$\langle 00004, 00000, 00022, 5\rangle$,

$\langle 00004, 00000, 00015, 6\rangle$,

$\langle 00004, 00000, 00016, 7\rangle$,

$\langle 00004, 00000, 00033, 8\rangle$,

$\langle 00005, 00005, 00001, 1\rangle$,

$$
\begin{aligned}
&\langle 00005, 00005, 00023, 2\rangle,\\
&\langle 00005, 00005, 00024, 3\rangle,\\
&\langle 00005, 00005, 00025, 4\rangle,\\
&\langle 00005, 00005, 00026, 5\rangle,\\
&\langle 00005, 00005, 00027, 6\rangle,\\
&\langle 00005, 00005, 00028, 7\rangle,\\
&\langle 00005, 00005, 00029, 8\rangle,\\
&\langle 00005, 00005, 00030, 9\rangle,\\
&\langle 00005, 00005, 00033, 10\rangle,\\
&\langle 00011, 00000, 00010, 1\rangle,\\
&\langle 00011, 00000, 00011, 2\rangle,\\
&\langle 00011, 00000, 00033, 3\rangle,\\
&\langle 00011, 00000, 00021, 4\rangle,\\
&\langle 00011, 00000, 00031, 5\rangle,\\
&\langle 00011, 00000, 00032, 6\rangle,\\
&\langle 00011, 00000, 00022, 7\rangle,\\
&\langle 00011, 00000, 00033, 8\rangle\\
&\}
\end{aligned}
$$

2.2 HSII2

$$
\begin{aligned}
P \;=\; &\{\\
&\langle 0, 00001, 00001, 00001, 1\rangle,\\
&\langle 0, 00001, 00001, 00002, 2\rangle,\\
&\langle 0, 00001, 00001, 00003, 3\rangle,\\
&\langle 0, 00001, 00001, 00004, 4\rangle,\\
&\langle 0, 00001, 00001, 00005, 5\rangle,\\
&\langle 0, 00001, 00001, 00007, 7\rangle,\\
&\langle 0, 00001, 00001, 00008, 8\rangle,\\
&\langle 0, 00001, 00001, 00033, 9\rangle,\\
&\langle 1, 00002, 00002, 00001, 1\rangle,\\
&\langle 1, 00002, 00002, 00009, 2\rangle,\\
&\langle 1, 00002, 00002, 00033, 3\rangle,\\
&\langle 2, 00002, 00008, 00010, 1\rangle,
\end{aligned}
$$

$$\langle 2, 00002, 00008, 00011, 2\rangle,$$
$$\langle 2, 00002, 00008, 00033, 3\rangle,$$
$$\langle 2, 00002, 00008, 00012, 4\rangle,$$
$$\langle 2, 00002, 00008, 00013, 5\rangle,$$
$$\langle 2, 00002, 00008, 00033, 6\rangle,$$
$$\langle 2, 00002, 00008, 00014, 7\rangle,$$
$$\langle 2, 00002, 00008, 00015, 8\rangle,$$
$$\langle 2, 00002, 00008, 00016, 9\rangle,$$
$$\langle 2, 00002, 00008, 00033, 10\rangle,$$
$$\langle 2, 00002, 00008, 00017, 11\rangle,$$
$$\langle 2, 00002, 00008, 00018, 12\rangle,$$
$$\langle 2, 00002, 00008, 00033, 13\rangle,$$
$$\langle 2, 00002, 00009, 00019, 1\rangle,$$
$$\langle 2, 00002, 00009, 00033, 2\rangle,$$
$$\langle 2, 00002, 00009, 00020, 3\rangle,$$
$$\langle 2, 00002, 00009, 00033, 4\rangle,$$
$$\langle 1, 00003, 00003, 00001, 1\rangle,$$
$$\langle 1, 00003, 00003, 00009, 2\rangle,$$
$$\langle 1, 00003, 00003, 00033, 3\rangle,$$
$$\langle 2, 00003, 00009, 00019, 1\rangle,$$
$$\langle 2, 00003, 00009, 00033, 2\rangle,$$
$$\langle 2, 00003, 00009, 00020, 3\rangle,$$
$$\langle 2, 00003, 00009, 00033, 4\rangle,$$
$$\langle 2, 00003, 00010, 00010, 1\rangle,$$
$$\langle 2, 00003, 00010, 00011, 2\rangle,$$
$$\langle 2, 00003, 00010, 00033, 3\rangle,$$
$$\langle 2, 00003, 00010, 00012, 4\rangle,$$
$$\langle 2, 00003, 00010, 00013, 5\rangle,$$
$$\langle 2, 00003, 00010, 00033, 6\rangle,$$
$$\langle 2, 00003, 00010, 00014, 7\rangle,$$
$$\langle 2, 00003, 00010, 00015, 8\rangle,$$
$$\langle 2, 00003, 00010, 00016, 9\rangle,$$
$$\langle 2, 00003, 00010, 00033, 10\rangle,$$
$$\langle 2, 00003, 00010, 00017, 11\rangle,$$
$$\langle 2, 00003, 00010, 00018, 12\rangle,$$
$$\langle 2, 00003, 00010, 00033, 13\rangle,$$

$\langle 1, 00004, 00000, 00010, 1\rangle,$

$\langle 1, 00004, 00000, 00011, 2\rangle,$

$\langle 1, 00004, 00000, 00033, 3\rangle,$

$\langle 1, 00004, 00000, 00021, 4\rangle,$

$\langle 1, 00004, 00000, 00022, 5\rangle,$

$\langle 1, 00004, 00000, 00015, 6\rangle,$

$\langle 1, 00004, 00000, 00016, 7\rangle,$

$\langle 1, 00004, 00000, 00033, 8\rangle,$

$\langle 1, 00005, 00005, 00001, 1\rangle,$

$\langle 1, 00005, 00005, 00023, 2\rangle,$

$\langle 1, 00005, 00005, 00024, 3\rangle,$

$\langle 1, 00005, 00005, 00025, 4\rangle,$

$\langle 1, 00005, 00005, 00026, 5\rangle,$

$\langle 1, 00005, 00005, 00027, 6\rangle,$

$\langle 1, 00005, 00005, 00028, 7\rangle,$

$\langle 1, 00005, 00005, 00029, 8\rangle,$

$\langle 1, 00005, 00005, 00030, 9\rangle,$

$\langle 1, 00005, 00005, 00033, 10\rangle,$

$\langle 3, 00011, 00000, 00010, 1\rangle,$

$\langle 3, 00011, 00000, 00011, 2\rangle,$

$\langle 3, 00011, 00000, 00033, 3\rangle,$

$\langle 3, 00011, 00000, 00021, 4\rangle,$

$\langle 3, 00011, 00000, 00031, 5\rangle,$

$\langle 3, 00011, 00000, 00032, 6\rangle,$

$\langle 3, 00011, 00000, 00022, 7\rangle,$

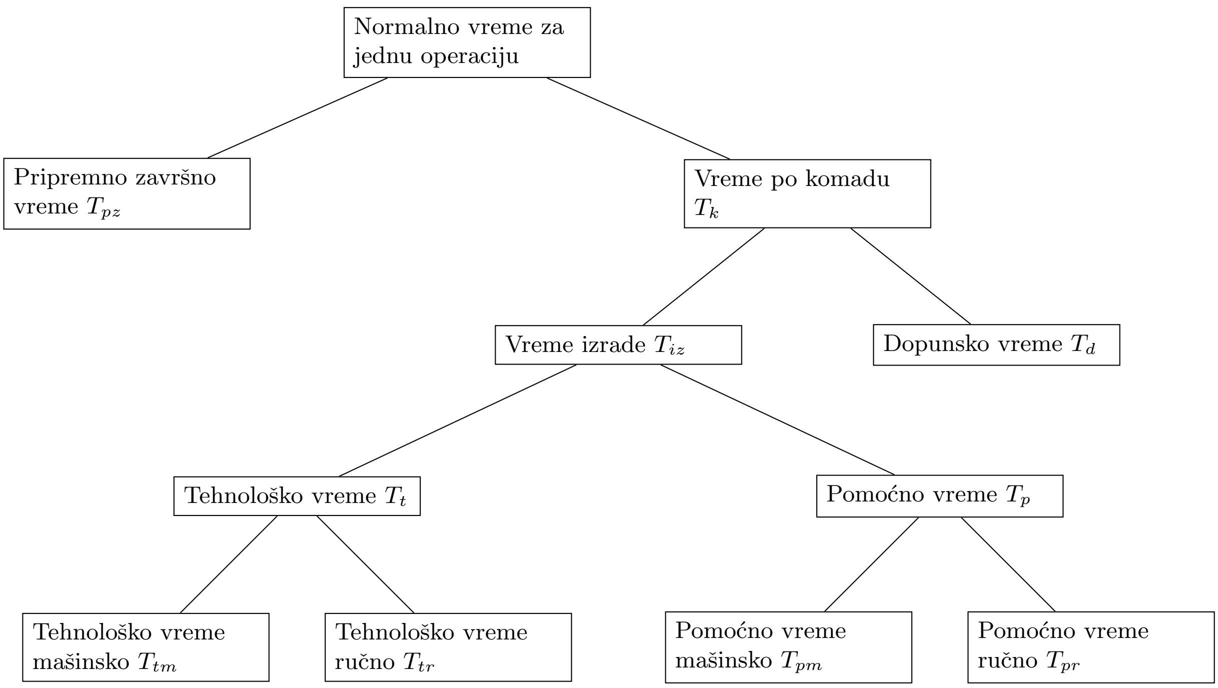
$\langle 3, 00011, 00000, 00033, 8\rangle$

$\}$

Prilog C

Klasifikacija vremena u tehnologiji

Načelno objašnjenje postupka dobijanja normalnog vremena, zahteva raščlanjavanje opracije na zahvate i njihovu klasifikaciju. Karakter zahvata nije uvek isti, nisu sve radnje usmerene na promenu oblika radnog predmeta ili izmene strukture i slično, ali su one isto tako sastavni deo zadate operacije ili preduslov za njeno odvijanje. Ovo je posebno izraženo u postupcima održavanja jer niz zahvata ima posebnu klasifikaciju.



Slika C.1: Komponentna vremena

Na slici (C.1), dat je grafički prikaz strukture komponentnih vremena potrebnih za izvršenje jedne operacije. Količina normalnog vremena izrade izračunava se po sledećem obrascu:

$$T = \frac{T_{pz}}{N} + T_{iz} + T_d \tag{C.1}$$

gde je:

T_{pz} ukupno potrebno pripremno završno vreme za izradu izvesnog broja komada, a obuhvata:

- prijem naloga za rad sa dokumentacijom i proučavanje zadataka
- pripremu alata
- pripremu ostalih elemenata potrebnih za rad
- predaju gotovih komada kontroli
- raspremanje radnog mesta posle završenog rada

N broj istih komada koji se izrađuju jedan za drugim bez prekida - broj komada u seriji

T_{iz} vreme predviđeno za radnje u vezi sa promenom oblika radnog predmeta i računa se po formuli (C.2)

T_d dopunsko vreme predviđeno za:

- odmor radnika
- fiziološke potrebe
- opsluživanje mašine, koje je predviđeno da obavlja sam radnik

Vreme T_{iz} je količina vremena vezana za promenu radnog predmeta i računa se:

$$T_{iz} = T_t + T_p \tag{C.2}$$

gde je:

T_t tehnološko vreme izrade predviđeno za promenu oblika radnog predmeta, izmenu strukture materijala, postizanje novih osobina spajanjem više delova u celinu, montaža i računa se po formuli (C.3)

T_p pomoćno vreme predviđeno za:

- uzimanje komada koje treba obrađivati
- osposobljavanje alata za početak obrade
- uključivanje radnog dela mašine
- kontrolu u toku rada
- skidanje i odlaganje komada po završenoj obradi

i računa se po formuli (C.4)

Vreme montaže se računa po formuli:

$$T_t = T_{tm} + T_{tr} + T_{tk} \tag{C.3}$$

gde je:

T_{tm} tehnološko vreme izrade predviđeno za mašinski rad u vremenskim jedinicama

T_{tr} tehnološko vreme izrade predviđeno za ručni rad u vremenskim jedinicama

T_{tk} tehnološko vreme izrade kombinovano, mašinski rad i ručni rad u vremenskim jedinicama

Pomoćno vreme se računa po formuli:

$$T_p = T_{pm} + T_{pr} + T_{pk} \tag{C.4}$$

gde je:

T_{pm} pomoćno vreme izrade predviđeno za mašinski rad u vremenskim jedinicama

T_{pr} pomoćno vreme izrade predviđeno za ručni rad u vremenskim jedinicama

T_{pk} pomoćno vreme izrade kombinovano, mašinski rad i ručni rad u vremenskim jedinicama

Proračunom se dobijaju vremena T_{tm} i T_{pm}, dok se ostala vremena dobijaju snimanjem proizvodnje ili na osnovu standardizovanih podataka dobijenih višestrukim ispitivanjima u laboratorijama. Obrazac za normalno vreme izrade u razvijenom obliku je:

$$T = \frac{T_{pz}}{N} + T_{tm} + T_{tr} + T_{tk} + T_{pm} + T_{pr} + T_{pk} + T_d \tag{C.5}$$

i sve može biti izraženo u različitim vremenskim jedinicama.

Biografija

Goran Lazović, diplomirani matematičar rođen je 28.01.1964. godine u Prištini. Završio je osnovnu 1978 i srednju školu 1982. godine u Smederevu. Iste godine upisao je Matematički fakultet u Beogradu, odsek za Računarstvo i informatiku. Diplomirao je 01.07.1988. godine sa srednjom ocenom 9.42. Poslediplomske studije je upisao školske 1989/90. godine na istom smeru Matematičkog fakulteta u Beogradu. Magistarsku tezu pod nazivom "Prilog fazi-modalnim logičkim strukturama i njihovim softverskim implementacijama" odbranio je 05.10.1995. godine na Matematičkom fakultetu u Beogradu. Na Mašinskom fakultetu u Beogradu radi od 01.09.1990. godine u zvanju asistenta pripravnika. U zvanje asistenta izabran je 1996. godine i reizabran 2000, 2004, 2008. i 2011. godine. Od dolaska na Mašinski fakultet do danas, Goran Lazović, diplomirani matematičar, uspešno je držao vežbe iz sledećih predmeta: Matematika 1, Matematika 2, Matematika 3, Programiranje, Nacrtna geometrija, Računarski alati, Numeričke metode, Bioautomatika, C/C++, Objektno orijentisana paradigma, Projektovanje baza podataka, Projektovanje inženjerskog softvera na Mašinskom fakultetu u Beogradu, odnosno Matematika 1, Matematika 2, Matematika 3 i Matematika 4 na Vojno tehničkoj akademiji. Govori engleski jezik i služi se ruskim jezikom. Otac je petoro dece Lazara (1996), Nikole (1997), Ksenije (2000), Anastasije (2002) i Sofije (2006).

Прилог 1.

Изјава о ауторству

Потписани-а Горан М. Лазовић

број индекса

Изјављујем

да је докторска дисертација под насловом

Пројектовање и испитивање структура база података у управљању одржавањем ваздухопловних система

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 18.05.2012.

Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Горан М. Лазовић

Број индекса

Студијски програм

Наслов рада Пројектовање и испитивање структура база података у управљању одржавањем ваздухопловних система

Ментор Др Слободан Радојевић

Потписани/а Горан М. Лазовић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 18.05.2012.

Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Пројектовање и испитивање структура база података у

управљању одржавањем ваздухопловних система

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 18.05.2012.