

KEMIJSKI INFORMACIJSKI SISTEMI II ALGORITMI ZA OBRAVNAVO IN OBDELAVO KEMIJSKO-STRUKTURNIH INFORMACIJ

UDK: 681.3 : 54

B. DŽONOVA-JERMAN-BLAŽIČ
N. TRINAJSTIĆ*INSTITUT JOŽEF STEFAN, LJUBLJANA
*INSTITUT RUDJER BOŠKOVIĆ, ZAGREB

Vsebina: V prispevku smo podali kratek pregled algoritmov za obravnavo in obdelavo računalniško zapisanih kemijskih struktur. Poskusili smo oceniti njihovo učinkovitost in uporabnost z ozirom na funkcije v okviru kemijsko-informacijskega sistema.

CHEMICAL INFORMATION SYSTEMS II: ALGORITHMS FOR HANDLING AND PROCESSING CHEMICAL INFORMATIONS.

Abstract: The paper discusses the computer-based algorithms that support the handling and processing of information about chemical substances. Some assessments are made about the efficiency and usefulness of the algorithms that perform interconversion, registration, structure and substructure searching, according to the functions they realize in the chemical information systems.

1. UVOD

Vsak informacijski sistem definirajo štiri funkcionalne enote(1):

- zbiranje in shranjevanje informacij tekom določenega časovnega intervala,
- tehnike in metode vnašanja novih informacij ter proizvodnje o zahtevanih informacijah, ali z drugimi besedami priprava odgovorov na vprašanja uporabnikov,
- skupina ljudi, ki vodi in oblikuje informacijski sistem. Skupina izloča uporabne informacije, dopolnjuje podatkovno bazo, ažurira podatke v podatkovni bazi, pripravlja vprašanja, posreduje odgovore uporabnikom ter implementira in dopolnjuje tehnike in metode za izvajanje različnih funkcij sistema,
- skupina uporabnikov sistema, ki ovrednoti sistem glede na svoje potrebe in kriterije.

Kemijski informacijski sistemi (KIS v nadaljevanju) imajo vse značilnosti splošnih informacijskih sistemov, so pa glede na svojstvenosti metod obdelave in posredovanja informacij dobili posebno mesto v okviru splošnih informacijskih sistemov (2). Najbolj pogoste zahteve, oziroma vprašanja uporabnikov KIS najdemo v naslednji skupini vprašanj:

- ali je določena spojina že navedena v literaturi (kdaj in od koga),
- ali obstajajo spojine, ki so podobne po strukturi spojnini definirani v vprašanju,
- katere so lastnosti te spojine (fizikalne, biološke, kemijske ipd.),
- kakšne so in katere so skupne lastnosti spojin iz navedene skupine spojin,
- kako pripravimo oziroma kako sintetiziramo navedeno spojino ali serijo spojin,
- kdaj je spojina prvič sintetizirana, kateri so postopki sinteze ipd.,
- katere spojine imajo navedene lastnosti.

Vsa navedena vprašanja zahtevajo, ali identifikacijo spojin v podatkovni datoteki ali proizvodnje o lastnosti spojin. Vprašanja v katerih je struktura spojin natančno definirana in, ki zahtevajo identifikacijo spojin v sistemu imenujemo strukturno-definirana vprašanja.

Vprašanja v katerih so definirane fizikalno-kemijske lastnosti spojine in, ki kot odgovor zahtevajo ime spojine ali skupine spojin, ki kažejo določene lastnosti imenujemo vprašanja za iskanje po vsebini datoteke (3). Ta vprašanja se definirajo s pomočjo deskriptorjev (4). Deskriptorji so besede ali kode s katerim so opisane lastnosti spojin ali kakšen drug pojem kot je: število in identiteta atomov v spojnini, število in identiteta kemijskih zvez v spojnini, molekulska teža, število in identiteta obrobov, opis okolja določene podstrukture v spojnini, molekulska povezanost, molekulska geometrija, molekularni model v prostoru ipd. (4). Na sl. 1 smo pokazali kot ilustracijo elemente, ki definirajo podatkovno bazo "Toxicology Data Bank" (3) realizirane v okviru programa Library's Toxicology Information Program v ZDA.

Večina obstoječih KIS je načrtovana tako da omogoča dostop do informacij na oba omenjena načina: glede na strukturo spojine in glede na vsebino datoteke. Za te namene v okviru KIS najbolj pogosto se kreirajo invertirane datoteke, ki omogočajo hiter dostop do informacij z različnimi ključi oziroma deskriptorji (5), (6). Poleg tega zaradi fleksibilnosti in vse večjih potreb uporabnikov, je v velikih KIS omogočeno zapisovanje strukture spojine na več načinov. Primerno temu, so implementirane tehnike, ki omogočajo transformacije med različnimi zapisi, ali drugače povedano interkonverzijo med zapisi. Algoritmi, ki omogočajo avtomatsko interkonverzijo med različnimi predstavitvami spojin realizirajo nekatere funkcije v KIS, od katerih navajamo najbolj pomembne:

- obdelava strukturnih diagramov ter priprava strukturnih formul v obliko sprejemljivo za publiciranje,
- prikazovanje strukturnih diagramov na video-terminalih, na podlagi zapisov z linearnimi notacijami,
- transformacija strukturnih diagramov iz video-terminalov v zapise z obliko tabele povezanosti,
- izmenjava podatkov med različnimi bazami podatkov.

V nadaljevanju bomo na kratko obdelali ter analizirali skupino najbolj pomembnih algoritmov, na katerih sloni realizacija vseh osnovnih funkcij KIS.

Toxicology Data Bank

Podatkovni elementi

1. Identifikacija substance
 - a) kemijsko ime
 - b) registrska številka službe CAS
 - c) sinonimi
 - d) molekulska formula
 - e) molekulska teža
 - f) zapis v Wiswesserjevi notaciji
2. Razvrstitev substance
 - a) kemijski razred
 - b) najbolj pogosta uporaba
3. Kemijsko/fizikalne lastnosti
 - a) temperatura topljenja
 - b) temperatura vrenja
 - c) gostota/specifična teža
 - d) barva/oblika
 - e) stabilnost/življenjska doba molekulskih obel
 - f) spektroskopski in drugi podatki
 - g) raztopljivost
4. Toksikološki učinki: eksperimentalne študije:
 - a) na živalih
 - b) z ljudmi
5. Toksikološke vrednosti
 - a) minimalna strupena količina
 - b) maksimalna dovoljena dnevna količina
 - c) LD-vrednosti
6. Laboratorijske metode in sinteze
7. Interakcije v bioloških sistemih
8. Farmakologija
 - a) metabolizem
 - b) absorpcija, distribucija, izločanje
9. Farmakoterapija
10. Ukrepi v primerih zastrupitve oziroma eksplozije
11. Informacije proizvajalcev
12. Metode prevoza
13. Podatki v zvezi z okoljem
 - a) meje eksplozivnosti
 - b) možnosti vžiga
 - c) meje strupenosti
 - d) meje radioaktivnosti
 - e) meje onesnaženja
 - f) meje izpostavljanja
 - g) prog mejnih vrednosti
 - h) kopičenje, razgrajevanje in obstojnost v okolju.

Slika 1.

2. REGISTRACIJA SPOJINE

Registracija spojine je algoritemski postopek, ki omogoča sprejemanje, povezovanje in ureditev vseh informacij v KIS, ki se nanašajo na določeno spojino. Postopek mora najprej ugotoviti ali se v datoteki nahaja snov, ki je po strukturi ekvivalentna kandidatu za vpis v podatkovni datoteki. V primeru da takšne spojine ni, zapis nove spojine ter ostale informacije se uvrstijo na ustrezno mesto glede na lastnosti in konfiguracijo spojine.

Postopek za registracijo spojin je ozko vezan in omejen z osnovnim sistemom za predstavitev spojin. Najbolj pomembna faza je primerjava strukture kandidata s strukturami spojin v osnovni datoteki sistema. Uporabljene metode so zelo različne in prilagojene uporabljeni notaciji za predstavitev kemijskih struktur. Za vse metode pa velja naslednje:

- kandidat za registracijo je zmeraj zapisan na enoličen in nedvoumen način,
- osnovna datoteka je urejena tako da omogoča urejanje spojin v skupine s skupnim strukturnim značajem,
- poleg osnovnega zapisa, omogočeno je vnašanje dodatnih parametrov, (to je najbolj pogosto molekulska teža ali

molekulska formula), zaradi kontroliranja napak.

Če je datoteka organizirana tako da so spojine grupirane v skupine, najprej najdemo odgovarjajočo skupino in zatem začnemo s primerjavo med spojinami. Učinkovitost postopka je odvisna od velikosti skupin oziroma od izbire parametrov, ki ločijo posamezne skupine. Največji vpliv na čas za registracijo spojine ima uporabljena metoda za zapisovanje spojin. Linearne notacije omogočajo dokaj hitro preiskovanje podatkovnih datotek zaradi enoličnosti, nedvoumnosti ter kompaktnosti zapisov. Registracija spojine zapisane v Wiswesserjevi notaciji se izvaja v okviru sistema CROSSBOW (7).

V sistemih, kjer je osnovni zapis spojin v obliki tabel povezanosti so nujno potrebne tehnike za generiranje enoličnih in nedvoumnih zapisov iz poljubno podanih tabel povezanosti. Ti zapisi so znani pod imenom "kanonične tabele povezanosti" (8), (9). Problem generiranja kanonične tabele povezanosti se sestoji v izbiri invariantnega oštevilčenja atomov. Problem invariantnega oštevilčenja atomov in zapisa v tabeli povezanosti je ekvivalenten problemu izomorfizma grafov. Molekulske strukturne formule lahko opisujemo kot grafe, ki imajo vozlišča s semantično vsebino. Če sta enolična in nedvoumna zapisa dveh grafov G_1 in G_2 enaka, oziroma še sta njihovi kodi enaki, potem sta G_1 in G_2 izomorfna in sam postopek kodiranja je izomorfizem (10). Najbolj preprost postopek za zapis spojin v kanonični obliki je generiranje vseh $n!$ možnih tabel povezanosti, oziroma vseh možnih oštevilčenj atomov v molekulskem grafu in leksikografski ureditvi $n!$ tabel. Kanonična oblika tabele bi bila tista, ki bi imela najnižjo leksikografsko ureditev. Ta način izbire kanonične oblike je izredno zamuden in je primeren le če imamo spojine z zelo majhnim številom atomov. V primeru da je število atomov 20 hipotetičen računalnik, ki lahko generira eno matriko in to matriko primerja z drugo v eni mikrosekundi bi porabil več kot 75000 let (11) za izpeljavo $20!$ operacij.

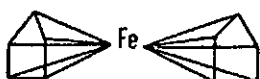
V preteklosti je bilo več poizkusov za izpeljavo (8), (12), (13a, 13b), matematične funkcije, ki bi omogočala hitro identifikacijo izomorfizma grafov. Do danes ni take funkcije, ki bi to opravila v polinomskem času za poljubni graf.

Na splošno problem izomorfizma grafov se nahaja v skupini NP-popolnih problemov. NP-popolni problemi so dobro raziskani problemi imenovani teški problemi. V te probleme štejemo: problem trgovskega potnika s področja operacijskih raziskav, problem tautologije iz propozicijskega računa ter druge podobne kombinatorične probleme. NP-popolni problemi imajo to lastnost da če je en problem iz skupine NP-popolnih problemov rešljiv z algoritmom, ki ima polinomski čas, potem so vsi ostali problemi tudi rešljivi v polinomskem času. Seveda za vsak problem je treba dokazati da res pripada skupini NP-popolnih problemov.

Pri reševanju problemov iz skupine NP-popolnih problemov, so zelo pogosto bili uporabljeni hevristični algoritmi, posebej takrat ko je rešitev problema bila povezana z dejansko realizacijo kakšnega sistema za obdelavo podatkov. Tako so Ungar (14) ter drugi avtorji (15), (16), (17) poizkusili zmanjšati časovna kompleksnost algoritmov za ugotavljanje izomorfizma grafov s pomočjo hevrističnih pravil. Uspeh je bil dosežen le pri načrtovanju algoritmov za ugotavljanje izomorfizma ravninskih grafov (18), (19), (20). Predlagani algoritmi imajo polinomske čase. Ti algoritmi so neuporabni za splošne grafe.

Dosedanje izkušnje so pokazale da problem izomorfizma grafov nasplošno, ni mogoče rešiti z dobrimi algoritmi, zato se je treba pri konkretnih problemih zadovoljiti s hevrističnimi algoritmi, ki dajejo dobre rešitve v večini primerov. Algoritem te vrste, ki se je v praksi pokazal kot zelo učinkovit, so razvili in implementirali v KIS službe CAS (5). Algoritem za kanonično oštevilčenje atomov, omejuje generiranje vseh možnih tabel povezanosti, tako da predčasno uredi atome in shrani rezultate začetnih poizkusov

oštevilčenja. Algoritem je implementiran na IBM 370/168 v obliki programa za registracijo spojine. Program obdela približno 13000 spojin tedensko, poprečen čas obdelave je 1000 struktur na minuto CPU(5). Zaradi nekatere spojine s simetrijo v strukturi, ki zahtevajo velika števila iteracij, implementacija algoritma predvideva prekinitve obdelave, če je čas porabljen za eno strukturo večji od 3 sec. Med 677000 struktur obdelanih tekom 1975 leta obdelava 990 je zahtevala več kot 3 sec. Zgled spojine te vrste je ferocene (sl. 2). Za te primere se uporablja posebna tehnika oštevil-



Slika 2.

čenja, ki je znana kot tehnika za sortiranje in registriranje izomerov. Da bi postopek pojasnili, smo v nadaljevanju ponazorili algoritem za hitro generiranje kompaktne kanonične tabele povezanosti. Delovanje algoritma smo ilustrirali z zgledom preproste spojine (slika 3):

Algoritem za generiranje kanonične tabele povezanosti v sistemu službe CAS:

N - množica poljubno oštevilčenih atomov

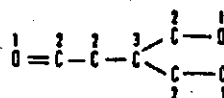
$n \in N$, n je atom iz N z zaporedno številko n

- določi vrednost povezanosti l_n vsem elementom iz N : l_n^i je enako številu neogljikovih atomov vezanih na atom n (i za ta korak je enako 1),
- določi spremenljivko k^i , ki je enako številu različnih vrednosti l_n^i ,
- $i \rightarrow i+1$, določi novo vrednost povezanosti l ,
 $l_n^i = \sum_{j=1}^{k^{i-1}} l_n^{j-1}$ (l_n^i je enaka vsoti l vseh r atomov vezanih na atom n)
- določi spremenljivko k^i ,
- če je $k^i < k^{i+1}$, nadaljuj, drugače pojdi na 7,
- pojdi na 3,
- vrednost povezanosti elementov iz N je enaka l_n^i ,
atom št. 1 ($n=1$) je atom z največjo vrednostjo l_n^i ,
atom št. 2, 3, ..., itd. so atomi vezani na atom št. 1 z opadajočimi vrednostmi l od 2 naprej. V primeru da dva atoma imata enako vrednost l potem manjšo zaporedno številko dobi atom z nižjo leksikografsko vrednostjo. Če ta razločitev ni možna (enake vrednosti l ter atome istega elementa), potem oštevilči ta par atomov poljubno in označi da so ti atomi poljubno oštevilčeni,
- neoštevilčene atome vezane na atom št. 2 oštevilči glede naopadajoče vrednosti l . Če srečaš med sosedi atoma št. 2 atome, ki se ne dajo oštevilčiti po danih pravilih, potem te atome oštevilči poljubno, ter označi da so ti atomi poljubno oštevilčeni,
- oštevilči vse atome iz N z enako proceduro in zgradi tabelo povezanosti,
- vrni se k atomom, ki je bil poljubno oštevilčen in ki ima največjo zaporedno številko. Če poleg tega atoma ni drugih atomov, ki so bili poljubno oštevilčeni končaj. Generirana matrika je enolična. Končaj postopek. Če to ni res nadaljuj,
- zamenjaj zaporedne številke atomom, ki so bili poljubno oštevilčeni. Izbrši oznake o poljubnem oštevilčenju,
- konstruiraj tabelo povezanosti,
- primerjaj to tabelo s predhodno,
- če nova tabela povezanosti ima nižjo leksikografsko vrednost od predhodne, potem zamenjaj stara z novo, staro zbrši in pojdi na 11.

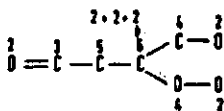
Čas potreben za generiranje kanonične tabele povezanosti je odvisen od števila prisotnih atomov v spojin ter od števila atomov z enako vrednostjo povezanosti l . Za te atome je treba poizkusiti vse možne kombinacije oštevilčenja, kar vpliva na to da je postopek zamuden. Opisani algoritem ni splošna rešitev problema izomorfizma grafov v okviru KIS, je pa relativno dobra rešitev za grafe, ki imajo najbolj pogosto zasedenost vozlišč od 1 do 4.

3. POIZVEDOVANJE O SPOJINAH, KI VSEBUJEJO DOLOČENI STRUKTURNI FRAGMENT

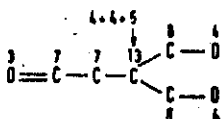
Postopek registracije spojine je oblika iskanja informacij o popolni strukturi spojine, ali z drugimi besedami to je postopek iskanja grafa, ki je izomorfen molekelskemu grafu spojine definirane v vprašanju uporabnika. Druga zelo pomembna zahteva uporabnikov v okviru KIS je iskanje strukturnih fragmentov oziroma iskanje spojin v datoteki sistema, ki vsebujejo strukturni fragment definiran v vprašanju uporabnika. Povedano z besedami iz teorije grafov, to iskanje je ekvivalentno posplošitvi problema izomorfizma grafov v problem izomorfizma podgrafov. Če imamo dva grafa $G_1=(V_1,E_1)$ in $G_2=(V_2,E_2)$, potem G_1 je izomorfen podgrafu grafa G_2 , če in samo če je V_1 podmnožica V_2 in E_1 podmnožica E_2 . Problem izomorfizma podgrafov se sestoji v določanju ali je graf G_1 izomorfen podgrafu grafa G_2 .



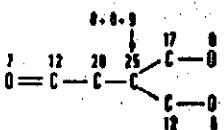
Prvi poizkus
vrednost $l = 1, 2, 3$
število različnih $k = 3$



Drugi poizkus
vrednost $l = 2, 3, 4, 5, 6$
število različnih $k = 5$



Tretji poizkus
vrednost $l = 3, 4, 7, 8, 9, 13$
število različnih $k = 6$



Četrti poizkus
vrednost $l = 7, 8, 12, 17, 20, 25$
število različnih $k = 6$

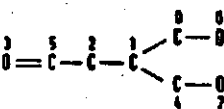


TABELA POVEZANOSTI št. 1									
atom št.	1	2	3	4	5	6	7	8	
povezave	1	1	1	2	3	4	5		
element	C	C	C	C	C	C	O	O	
zveza	S	S	S	S	S	S	S	D	

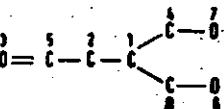
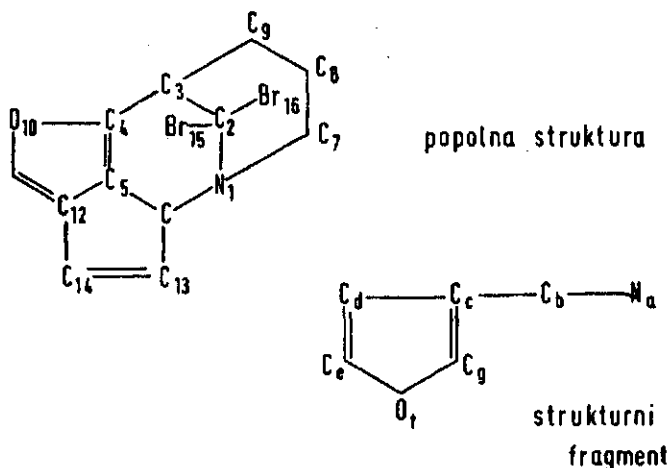


TABELA POVEZANOSTI št. 2									
atom št.	1	2	3	4	5	6	7	8	
povezave	1	1	1	1	2	3	4	5	
element	C	C	C	C	C	C	C	O	

Slika 3.

Problem ugotavljanja izomorfizma podgrafov je bolj težaven od ugotavljanja izomorfizma grafov. V okviru KIS so ta problem reševali na več različnih načinov. Uspeh posameznih metod je bil odvisen od sistema za zapisovanje in shranjevanje kemijskih spojin ter od zahtevane natančnosti poizvedovanja. Določeni problemi so še naprej nezadostno obdelani in čakajo na boljše rešitve (6). Nekatere prednosti linearnih notacij, kanoničnih tabel povezanosti ter nomenklature zapisov, ki so jih imeli pri registriranju spojin zaradi enoličnosti in nedvoumnosti zapisov, v tem iskanju se popolnoma izgubijo. Tako, pri kanoničnih tabelah se lahko zgodi, da so atomi v podstrukturi oštevilčeni različno od atomov v popolni strukturi ker ti atomi nimajo iste atome za sosedo. (V nadaljevanju bomo pod "popolno strukturo" razumeli strukturo formulo spojin zapisane v enem od možnih računalniških zapisov v datotekah KIS in pod podstrukturo, strukturo formulo nepopolne strukture definirane v vprašanju uporabnika). Kot ilustracijo smo na sliki 4 pokazali eno strukturo in odgovarjajočo podstrukturo. Pri nomenklature ali notacijskih zapisih, je podstruktura lahko zapisana s simboli, ki se razlikujejo od simbolov uporabljenih za zapis popolne strukture, spet zaradi različnega okolja v katerem se podstruktura nahaja v različnih spojinah.

Iskanje po podstrukturah se lahko odvija na več nivojev natančnosti. Najnižji nivo natančnosti je iskanje v datoteki kjer imamo zapis spojin s fragmentacijskimi kodami ali če iščemo s pomočjo mask. Maske (5) se najbolj pogosto uporabljajo pri preiskovanju velikih bank podatkov, in so podobne fragmentacijskim kodam, le da predstavljajo računalniško generirane podatke v katerih je prisotnost oziroma odsotnost kemijskih značilnosti (elemente, povezave, obročje ipd.) označena z enicami ali ničlami, v nekaterih primerih tudi s števili. Za ilustracijo smo na sl. 5 ponazorili en zapis spojin s fragmentacijsko kodo. V primeru sistema s fragmentacijskimi kodami, v vprašanju uporabnika se zahtevani fragment zapiše s kodo, ki je bila uporabljena pri ustvarjanju datoteke. Odgovor vsebuje seznam molekul v katerih se zahtevani fragment nahaja.

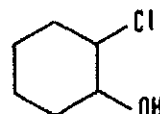


Slika 4.

V sistemu GREMAS (21) se datoteka fragmentov generira avtomatsko iz tabele povezanosti ter ostalih topoloških deskriptorjev. Avtomatsko generiranje fragmentacijskih kod omogoča nemoteno spreminjanje uporabljenih kod, oziroma dodajanje novih, brisanje starih ipd. Podobno generiranje datotek s fragmenti je realizirano v KIS National Institute of Health in Environmental Protection Agency v ZDA ali skrajšano NIH-EPA (6). V tem KIS, se za vsako poizvedovanje po podstrukturah generirajo invertirane datoteke iz osnovne datoteke. Pri tem obstajajo dve možnosti: generiranje datoteke kjer so spojine zapisane kot niz fragmentov in ge-

neriranje datoteke s podatki o obročih. Datoteka fragmentov, ki poleg fragmentov vsebuje še podatek o registrski številki, se generira z obdelavo vsakega atoma in vsake povezave posamezne spojine. Pri tem se določajo naslednji podatki: dimenzije fragmenta (število atomov v fragmentu), semantika centralnega atoma (centralni atom je atom, ki povezuje fragment z ostalo strukturo spojine), semantiko prvega sosedo (prvi sosed je atom vezan na centralnega, ki se najmanj pogosto sreča v datoteki, ponavadi je različen od C, N, O, če takega atoma ni potem je to C atom z dvojno ali trojno povezavo ipd.), semantiko drugega, tretjega in četrtega sosedo.

Koda	Pomen
2/12	obroč
4/1	šestčlanski benzenov obroč
17/1	klor (en atom)
18/1	ena OH skupina

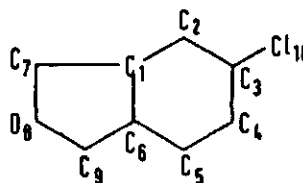


Slika 5.

Generiranje se konča ko je celotna struktura zapisana v obliki fragmentov. Datoteka s podatki o obročih ima enako strukturo, le da tukaj nastopajo podatki o obročih. Za vsakega obročka se generira zgoščena koda (hash koda) iz tabele povezanosti ter podatki o hetero atomih, njihovi poziciji, o substituentih ipd.). Poleg tega se generirajo še podatki o možnih kombinacijah med obročji (2,3,4 skupaj), kjer se ostali obroči obravnavajo kot substituenti. Za ilustracijo smo na sliki 6 pokazali iskanje podstrukture sestavljene iz petčlanskega heteroatomskega obročka in iz šestčlanskega obročka s substituentom pri tretjem atomu. Pogoji iskanja so bili naslednji:

zahteva št. 1: kombinacija obročev v popolni strukturi naj bo enaka (ni dovoljeno več kot dva obročka),
zahteva št. 2: le vozlišče 8 je lahko različno od C.
zahteva št. 3: v šestčlanskem obročju je prisoten najmanj en substituent in naj bo vezan na atomu št. 3.
Pri iskanju je bila uporabljena podatkovna baza NIH-EPA Mass Spectral Search System (22), (23), ki vsebuje okrog 30000 različnih spojin in njihove masne spektre. V prvi invertirani datoteki s podatki o obročih je bilo najdeno 18 spojin, s po enim ali dveh obročki. V drugi datoteki s podatki o fragmentih je bilo najdeno 180 struktur, ki vsebujejo fragment s substituentom iz slike 6. Presek množic spojin iz

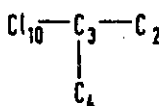
a) definicija podstrukture



večkratne povezave ter H atomi niso določeni

številk označujejo oštevilčenje atomov v tabeli povezanosti

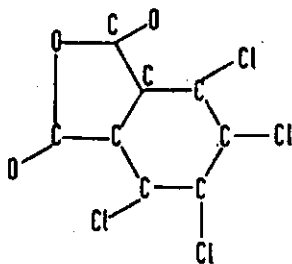
b) definicija zahtevanega fragmenta



Slika 6.

obeh datotek je dalo le eno spojino pokazano na sliki 7. Seveda, večkrat se zgodi, da presek dveh datotek da več kot eno spojino, v tem primeru se uporablja natančnejši način iskanja, znan kot "iskanje atom za atomom" (27).

najdena struktura



molekulska formula : C6Cl6

registrska številka CAS: 112 080

Slika 7.

Pri vseh kompleksnih KIS, se natančno poizvedovanje po podstrukturah izvaja z iskanjem atom za atomom. To iskanje je precej zamudno in zaradi tega se povsod pred tem izvaja predhodno iskanje po fragmentih (tako kot je bilo navedeno v primeru zgoraj), po nomenklaturnih zapisih, ali po linearnih notacijah. V okviru sistema CAS je predhodno iskanje izvedeno s pomočjo nomenklaturnih zapisov. Pri tem so bili uporabljeni najnovejši dosežki s področja preiskovanja tekstov in računalniško čitljivih datotek z obliko teksta (24).

V sistemu z Dyson-IUPAC-ovo notacijo predhodno iskanje se izvaja s pomočjo datotek s permutiranimi indeksi (25). Wiswesserjeva notacija je bila uporabljena za isti namen v sistemu "Institut-a for Scientific Information" (26). Uspeh iskanja je zelo odvisen tudi od uporabnika oziroma od sposobnosti da se pravilno opiše in kodira pričakovano okolje podstrukture. V koliko je to bolj uspešno v toliko je manjše število spojin za natančnejše preiskovanje.

Natančno poizvedovanje o tem ali se določena podstruktura nahaja v nekaterih strukturah je možno le s primerjanjem atomov iz podstrukture z atomi strukture in z primerjanjem povezav iz podstrukture s povezavami iz strukture. Primerjanje je možno le če so strukture zapisane v topološki obliki, oziroma če zapis spojin izhaja iz njihovih grafov. V dosednji tehnologiji KIS srečamo dva algoritma za primerjavo podstrukture s strukturo implementiranih v najrazličnejših enačicah. To so: iterativno tehnika primerjanja atom z atomom (27) in tehnika postopnega izločanja množic (16). Oba algoritma v svoji prvotni obliki, so bila načrtovana za primerjanje popolnih struktur, oziroma za ugotavljanje izomorfizma dveh grafov. Univerzalnost postopkov je omogočila uporabo tudi pri določanju izomorfizma podgrafov.

3.1 Iterativno iskanje in primerjanje atom z atomom

Iterativno iskanje se sestoji v primerjanju atomov iz podstrukture z atomi iz strukture do popolnega ujemanja ali neujemanja. Da bi skrajšali čas primerjanja in hitreje izločili neprimerne strukture, se primerjanje začne z atomom iz podstrukture, ki se najmanj pogosto sreča v datoteki. Vse strukture, ki ta atom ne vsebujejo odpadejo že na začetku. Po prvi uspešni primerjavi dveh atomov, se iskanje nadaljuje na enak način. Za naslednji atom se vzame atom vezan na predhodnega in ki se najmanj pogosto sreča v datoteki. Če do ujemanja pride, se primerjanje nadaljuje z naslednjim so-

sedom, ki se izbira po enakih kriterijih. V primeru da pride do neujemanja, postopek se vrne v točki zadnjega ujemanja in se primerjanje nadaljuje z drugim atomom. Prehoda pot primerjanja se sproti zaznamuje zaradi vračanja v primerih neujemanja. Postopek je iterativen in teče do ugotovitve popolnega ujemanja med podstrukturami in kakšnim delom strukture. V primeru da postopek prehodi celotno strukturo in do ujemanja ne pride, postopek konča, kar pomeni da podstruktura ni vsebovana v strukturi.

3.2. Postopno izločanje množic atomov

Postopek postopnega izločanja množic je popolnoma različen od postopka primerjanja atom z atomom in je bolj računalniško pobarvan. Avtor algoritma je Sussenguth (16), posamezne spremembe s ciljem izboljšanja algoritma so predlagali Ming in Tauber (29).

Postopek je izpeljan na podlagi naslednjih trditvev:

- če sta grafa G in G^* izomorfna, potem podmnožice vozlišč grafa G , ki se razlikujejo med seboj glede na nekatere lastnosti vozlišč, so ekvivalentne podmnožicam grafa G^* ,
- če podmnožice vozlišč grafov G in G^* z enakimi lastnostmi nimajo enako število elementov potem G in G^* nista izomorfna.

V primeru izomorfizma med grafom G in podgrafom grafa G^* so pogoji oslabiljeni in glasijo: odgovarjajoča vozlišča grafa G so vsebovana v množice vozlišč grafa G^* z enakimi lastnostmi. Zapisano z matematičnim jezikom te dve trditvi dobita naslednjo obliko:

a) graf G je izomorfen grafu G^*

vrednost vozlišča

$$(x: \text{vrednost}(x)=v) = (x^*: \text{vrednost}(x^*)=v)$$

vrednost povezave

$$(x: \text{vrednost}((x,y))=b) = (x^*: \text{vrednost}((x^*,y^*))=b)$$

valenca

$$(x: \text{valenca}(x)=d) = (x^*: \text{valenca}(x^*)=d)$$

stopnja zasedenosti

$$(x: \text{stopnja}(x)=d) = (x^*: \text{stopnja}(x^*)=d)$$

povezanost

$$A = A^* \rightarrow \bigwedge A = \bigwedge^* A^*$$

$\bigwedge$ - množica vozlišč vezana na vozlišče x

b) graf G je izomorfen podgrafu grafa G^*

vrednost vozlišča

$$(x: \text{vrednost}(x)=v) \subseteq (x^*: \text{vrednost}(x^*)=v)$$

vrednost povezave

$$(x: \text{vrednost}((x,y))=b) \subseteq (x^*: \text{vrednost}((x^*,y^*))=b)$$

valenca

$$(x: \text{valenca}(x)=d) \subseteq (x^*: \text{valenca}(x^*)=d)$$

stopnja zasedenosti

$$(x: \text{stopnja}(x)=d) \subseteq (x^*: \text{stopnja}(x^*)=d)$$

povezanost

$$A \subseteq A^* \rightarrow \bigwedge A \subseteq \bigwedge^* A^*$$

Postopek dela v dveh fazah: generiranje podmnožic in izločevanje podmnožic. Generiranje podmnožic smo ilustrirali na zgledu ponazorjenim na sliki 8. Množice se med seboj razlikujejo po vsebini elementov (C, O, N, Br), vrednost povezave (dvojna, enojna), valence atomov v strukturi brez H atomov (1, 2, 3), (stopnja zasedenosti), številu povezav v najmanjšem ciklu kateremu vozlišče pripada (povezanost). Izločevanje množic, omogoča zmanjševanje števila vozlišč, ki so možni ekvivalentni vozlišču x . Idealen primer je, če je to število enako 1, potem je $x = x^*$. Če sta Si in Si^* par množic z elementi z enakimi lastnostmi in če velja $Si \subseteq Si^*$, potem lahko sklepamo naslednje: če je $x \in Si$, potem je tudi $x^* \in Si^*$. Za vse i v katerih je x vsebovan v Si in x^* je vsebovan v Si^* , In od tukaj sledi naslednja relacija:

$$x^* \in \bigcap Si^*$$

$$i \in I$$

1

i. I, le če je $x \in Si$.

Če sta Sj in Sj^* množici z enakim številom vozlišč, $Sj = Sj^*$,

Množice podstrukture	Množice strukture
vrednost vozlišč: C (b,c,d,g)	(2,3,4,5,6,7,8,9,11,12,13,14)
O (f)	(10)
N (a)	(1)
Br $\emptyset$	(15,16)
vrednost povezav: enojna	
(a,b,c,d,e,f,g)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
dvojna	
(c,d,e,g)	(4,5,7,8,11,12,13,14)
stopnja zasedenosti: 1	
(a)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)
2	
(b,d,c,f,g)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14)
3	
(c)	(1,2,3,4,5,6,12)
povezanost: 5	
(c,d,e,f,g)	(4,5,6,10,11,12,13,14)
(a,b)	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16)

Slika 8.

potem velja naslednje $\bar{S}_j \subseteq \bar{S}_j^*$. Vozlišče, ki ne pripada S_i , tj., ki pripada $\bar{S}_j$ nima ekvivalenta v S_j^* (ker je $\bar{S}_j = S_j^*$ in je ekvivalenca vozlišč ena proti ena), kar pomeni da so ekvivalenti vozlišč iz $\bar{S}_j$ vsebovani v S_j^* . Z drugimi besedami, če je $S_j = S_j^*$ in če $x \in S_j$, potem $x^* \in S_j$. Z uporabo množic, ki imajo to lastnost dobimo naslednjo relacijo:

$$x^* \in S_i \cap S_j^* \iff x \in S_i \cap S_j \quad 2$$

i-1, če je $x \in S_i$ in $j \in J$, če je $S_j = S_j^*$ in $x \in S_j$. Z operacijo presek iz relacije 2 v veliki meri zmanjšamo število elementov v množicah S in S^* . V primeru da za vsako vozlišče iz S najdemo ekvivalent v S^* , smo problem rešili in postopek lahko zaključimo. Če se to ne zgodi, potem je treba prvo fazo - generiranje novih parov množic z elementi z enakimi lastnostmi ponoviti in zatem tudi drugo fazo. Če so v prvi iteraciji bili uporabljeni množici, ki so se razlikovali med seboj po vsebini elementov, v drugi iteraciji bodo uporabljeni množici, ki se razlikujejo v vrednosti povezav. Zgled generiranja množic in izvajanja operacije presek je ponazorjen na sliki 9. Postopek iterativno teče do

Množice podstrukture	Množice strukture
(1,2,4) S	(b,d,e,f) S*
1	1
(1,3) S	(a,d) S*
2	2
(2,4,5) S	(b,c,d,e,f) S*
3	3
(3,4,5) S	(a,b,c,f) S*
4	4

Rezultati po operaciji presek		
Množice strukture	Množice podstrukture	uporabljene množice
(1)	(d)	$S \cap S$
		1 2
(2)	(b,e,f)	$S \cap S \cap S$
		1 2 3
(3)	(a)	$S \cap S$
		1 4
(4)	(b,f)	$S \cap S \cap S \cap S$
		1 2 3 4
(5)	(b,c,f)	$S \cap S \cap S$
		2 3 4

Slika 9.

do izpolnitve enega od naslednjih pogojev:

- za vse atome iz G so določeni ekvivalenti v G^* ,
- podmnožica S vsebuje več elementov od odgovarjajoče množice S^* (v tem primeru G ne more biti izomorfen s podgrafom grafa G in izomorfizem ni možen),
- ni pogojev za generiranje novih podmnožic (vse lastnosti po katerih se atomi v grafu G razlikujejo med seboj so izčrpani in rezultat je nedoločen). To pomeni da za nekatere atome ekvivalent ni določen ali da za nekatere atome ni pokazano da ekvivalenti ne obstajajo.

Nedoločen rezultat dobimo v dveh primerih: med G in G^* obstaja večkratni izomorfizem (ni metode, ki bi izbrala enega od izomorfizmov) in uporabljene lastnosti elementov niso zadosti močne, da bi razločile posamezne atome. V tem primeru se ekvivalenti določajo naključno, vsaka naključna izbira je potrjena potrditvi. Če je potrditev negativna potem izberemo naključno nov par atomov in postopek ponovimo dokler ne dobimo pravilno izbire.

Oba algoritma, primerjanje atom z atomom (11) in tehnika postopnega izločanja množic (16) imata veliko časovno kompleksnost (28), kar pomeni da čas potreben za ugotavljanje izomorfizma narašča izredno hitro s številom atomov v grafu. Poleg tega velik vpliv na časovno kompleksnost ima tudi struktura grafa in način kako so podatki zapisani v računalniku. Časovna kompleksnost algoritma za primerjanje atom z atomom je ocenjena na $O((d-1)^n)$, kjer je d maksimalna valenca atoma v strukturi in n število atomov v strukturi. Bolj uspešni algoritmi za ugotavljanje izomorfizma grafov uporabljajo različne kombinacije obeh algoritmov (10), (14), (30), (31).

4. INTERKONVERZIJA MED ZAPISI

Algoritmi, ki omogočajo avtomatsko interkonverzijo med različnimi zapisi spojin omogočajo realizacijo nekaterih pomembnih funkcij vsakega KIS na najbolj enostaven način. Poleg osrednjega pomena, ki ga algoritmi za interkonverzijo imajo v okviru interne uporabe posameznega KIS, ti algoritmi omogočajo tudi izmenjavo informacij med različnimi podatkovnimi bazami.

Algoritme za interkonverzijo lahko grupiramo v tri skupine:

- algoritmi za interkonverzijo iz linearnih notacij, nomenklature zapisov, koordinatnih zapisov in zapisov v obliki strukturnih diagramov v tabele povezanosti,
- algoritmi za interkonverzijo med različnimi tabelami povezanosti,

c) algoritmi za interkonverzijo iz tabel povezanosti v linearne notacije, nomenklaturne zapise, koordinatne zapise in predstavitev v obliki strukturnih diagramov.

V prvi skupini algoritmov so najbolj zanimivi algoritmi, ki omogočajo generiranje zapisov v obliki tabel povezanosti iz strukturnega diagrama predstavljenega na grafičnem zaslonu. Koncepte teh algoritmov zasledimo pri Corey in Wipke-u (32), realizacijo algoritmov za namene KIS pri Feldmanu v okviru NIH-EPA (33), (34), ter v sistemu GREMAS (35), (36). Podobno funkcijo opravljajo algoritmi za generiranje tabel povezanosti iz koordinatnih zapisov kemijskih struktur, ki se v računalniku vnašajo s pomočjo pisalnega stroja za kemijske formule (37). Konverzija linearnih notacij in nomenklaturnih zapisov v tabele povezanosti zasledimo pri vseh večjih KIS. Tako danes obstajajo programi za konverzijo iz nomenklaturnih zapisov (38), Wiswesserjeve notacije (39), Hayward-ove (40), in IUPAC-ove linearne notacije (25) v tabele povezanosti. Fragmentacijske kode ne predstavljajo popoln in enoličen zapis struktur, zato njihova konverzija v druge zapise ni mogoča.

Druge skupine algoritmov se uporabljajo za generiranje kompaktnih tabel povezanosti iz redundantnih in obratno (41), (42) ter pri konverziji tabel povezanosti iz različnih KIS. Tako danes obstajajo programi za konverzijo tabele povezanosti iz sistema CROSSBOW v tabele povezanosti sistema CAS (43). Algoritmi za konverzijo iz ene tabele povezanosti v drugo, se uporabljajo tudi pri konverziji zapisov iz ene linearne notacije v drugo. Konverzija zapisa iz linearne notacije v tabele povezanosti je veliko bolj enostavna od obratne konverzije, zaradi zapletenih sintaksnih in semantičnih pravil notacije (6). Veliko bolj zapletena je konverzija iz ene linearne notacije v drugo. Zato se ta postopek opravlja preko transformacij v tabelah povezanosti in potem iz tabele v linearno notacijo.

Tretja skupina algoritmov realizira konverzijo zapisov iz tabele povezanosti v kakšen drug nedvoumen in enoličen zapis. Tabela povezanosti je najmanj strukturirana predstavitev kemijske spojine in ne vsebuje elemente s kemijsko značilnostjo razen seznama atomov in kemijskih zvez. Obratna konverzija iz bolj strukturiranih predstavitev v obliki tabel povezanosti zahteva le interpretacijo simbolov sintakse. Konverzija tabele povezanosti v kakšno linearno notacijo je možna le s pomočjo kompleksnih algoritmov za analizo strukture in za ustvarjanje linearnega zapisa upoštevajoči zapletena pravila notacije. Eden od bolj dognanih algoritmov za konverzijo tabele povezanosti v Wiswesserjevo notacijo (44), (45), je omogočil razvoj algoritmov za urejanje Wiswesserjevih notacij za kemijske sisteme s kompleksnimi obroči. Podoben temu algoritmu je algoritem Farrella (46) ter algoritmi v okviru sistema DARC (47).

Iz te skupine algoritmov, so zelo zanimivi algoritmi za generiranje strukturnih diagramov iz linearnih notacij. Večina teh algoritmov so bili načrtovani za prikazovanje diagramov preko vrstičnega tiskalnika in tehnične zmogljivosti teh aparatov so omejevale kompleksnost prikazanih diagramov (48), (49). Dognani sistemi za prikazovanje strukturnih diagramov so realizirani v okviru službe CAS (50). Poleg podatkov v obliki tabele povezanosti, sistem za generiranje strukturnih diagramov uporablja datoteko s koordinatnimi podatki o posameznih obročih, ter o strukturah sestavljenih iz več obročev. Podatki za to datoteko so računski pripravljeni. Datoteka vsebuje več kot 15000 najrazličnejših oblik obročev in sistemov obročev, ki se nahajajo v $3,5 \times 10^6$ različnih spojin zapisanih v osrednji datoteki sistema službe CAS (CAS Chemical Registry System). Koordinatni zapis obročev iz datoteke obročev omogoča hitro generiranje strukturnega diagrama (izpis je na elektronskem risalniku ali na fotostavnim strojem), ker so vsi zapletljivi okrog določanja koordinatov obročev na ta način onemogočeni. V bistvu datoteka s podatki o koordinatih obročev precej poenostavi postopek, čigava osrednja naloga je da ugotovi medsebojno povezavo obročev, verig, linearnih aciklič-

nih nizov s končnim atomom ali brez ter substituentov v obročih. Potem se šele začne sestavljanje koordinatov strukture ali direktno (za aciklični del) ali iz podatkov iz datoteke z obroči. Opisani sistem je bil izpopolnjen s programom, ki direktno iz grafičnega zaslona sprejema strukturne diagrame, generira tabele povezanosti ter strukturni diagram nariše na risalniku ali ga posreduje fotostavnim strojem (51).

5. SKLEPNE BESEDE IN BODOČE USMERITVE

S pregledom algoritmov za obravnavo in obdelavo kemijskih struktur v okviru različnih KIS smo poizkusili oceniti njihovo učinkovitost in uporabnost z ozirom na funkcije, ki jih opravljajo. Lahko rečemo, da je za večino problemov v zvezi z obdelavo računalniško zapisanih kemijskih struktur, najdena učinkovita in praktična rešitev. O tem priča tudi veliko število zelo učinkovitih KIS, ki posredujejo uporabnikom po celem svetu široko paleto najrazličnejših podatkov. Tako kot pri predstavitvi kemijskih struktur (2), tako tudi za algoritme za obravnavo in obdelavo kemijskih informacij lahko ugotovimo da so se prilagajali sistemom s katerim so spojine zapisane, uporabljeni materialni opremljeni, zahtevam naročnikov ter ostalim posebnosti v zvezi s posedovanjem KIS. Od tukaj tudi izhaja raznovrstnost načrtovanih postopkov ter uporabljenih tehnik. Pri predstavitvi algoritmov smo več prostora uporabili za predstavitev algoritmov bolj splošnega značaja in širše uporabe. To so algoritmi za ugotavljanje izomorfizma grafov in izomorfizma podgrafov v okviru KIS. Za oba algoritma velja da problema ne rešujeta v polinomskem času. Zaradi pomembnosti problemov in potrebi po praktični rešitvi v okviru KIS, so se pri iskanju rešitev zatekli k različnim heurističnim algoritmom, ali k metodam, ki občutno zmanjšajo število spojin za preiskovanje. Implementirani heuristični algoritmi za ugotavljanje izomorfizma grafov, ki predstavljajo kemijske strukture, so dali zelo dobre rezultate v praksi. Seveda problemi iz splošne teorije grafov še naprej privlačijo pozornost znanstvenikov, ki si prizadevajo izboljšati časovno kompleksnost omenjenih algoritmov (10).

Algoritmi za obravnavo in obdelavo strukturnih kemijskih informacij, poleg v sistemih za shranjevanje in iskanje informacij se uporabljajo tudi v drugih področjih kemije, kjer se računalnik uporablja kot zelo koristen pripomoček (52). Tako na primer izjemne dosežke zasledimo pri uporabi računalnikov pri iskanju korelacij biološke aktivnosti spojine/struktura spojine (53), (54). Podobne analize in uporabo velikih podatkovnih baz, zasledimo pri študiju reaktantov in produktov v sintetskih reakcijah s ciljem izločanja parcialnih struktur značilnih za določene sintetske reakcije (54), (55), ter pri matematski sintezi in analizi struktur molekul in njihovih homologov (56). Algoritme za obdelavo kemijsko-strukturnih informacij zasledimo tudi pri programih za geometrijsko modeliranje molekul, oziroma za generiranje tridimenzionalnih deskriptorjev kemijskih spojin (57), (58), ter v programih za načrtovanje sintetskih poti (59).

6. REFERENCE:

1. C.M. Bowman, The development of Chemical Information Systems, v knjigi Chemical Information Systems, ed. Ash & Hyde, J. Wiley & Sons, New York, N.Y. (1974)
2. B. Džonova-Jerman-Blažič, Kem. Ind., 28 (1979), 67
3. M.A. Oxman, H. Kissman, J. Burnside, J. Edge, C. Haberman, A. Wyres, J. Chem. Inf. Comp. Sci., 16 (1976) 19
4. A.J. Stuper, W.E. Burgger, P.C. Jurs, Computer-Assisted Studies of Chemical Structures and Biological Functions, J. Wiley & Sons, New York, N.Y. (1979), pogl. 3.
5. L.J. O'Korn, Algorithms in the Computer Handling of Chemical Information, v knjigi Algorithms for Chemical Computation, ed. E. Christoffersen, ACS Symposium Series No. 46, Washington (1977), 122

6. J. Feldman, G.W.A. Milne, S.R. Heller, A. Fein, J.A. Muller, B. Koch, *Jour. Chem. Inf. Comp. Sci.*, **17** (1977) 173
7. D. Schmidt, L. Druffel, *Jour. ACM*, **23** (1976) 433
8. J. Turner, *SIAM & Appl. Math.*, **16** (1968), 520
9. C. Jochum, J. Gasteiger, *Jour. Chem. Inf. Com. Sci.*, **17** (1977), 2
10. R.E. Tarjan, *Graph Algorithms in Chemical Computation*, v knjigi *Algorithms for Chemical Computation*, ed. E. Christoffersen, ACS Symposium Series No. **46**, Washington (1977), 1
11. A. Bertziss, *J. ACM*, **20** (1973), 365
12. M. Karp, *Tech. Rep.*, **3** Comp. Sci. Dep. Univ. of California, (april 1972)
13. a) M. Randić, *J. Chem. Inf. Comp. Sci.*, **15** (1975), 105
b) M. Randić, *J. Chem. Phys.*, **60** (1974), 3920
14. S. Unger, *Comm. ACM*, **7** (1964), 26
15. G. Saucier, *Rev. Française Inform. Rech. Oper.*, **5** (1971), 39
16. E. Sussenguth Jr., *J. Chem. Doc.*, **5** (1965), 36
17. J. Steen, *Jour. Chem. Doc.*, **3** (1969), 51
18. J. Hopcroft, E. Tarjan, v knjigi *Complexity of Computer Computation*, ed. R.E. Miller & J.W. Thatcher, Plenum Press, New York, N.Y. (1972), 143
19. J. Hopcroft, J. Wong, *Proc. 6th Annual ACM Symp.*, Seattle, Washington; (april 1974), 172
20. L. Weinberg, *IEEE Trans. on Circuit Theory*, CT-13, (1966), 142
21. R. Fugmon, *The IDC System*, v knjigi *Chemical Information Systems*, ed. Ash & Hyde, J. Wiley & Sons, New York, N.Y., (1975)
22. S.R. Heller, H.M. Fales, G.W. Milne, *Org. Mass. Spectr.*, **7** (1973), 107
23. S.R. Heller, G.W. Milne, R.J. Feldman, R.S. Heller, *Jour. Chem. Inf. Comp. Sci.*, **16** (1976), 176
24. W. Fisonick, L.D. Mitchell, J.A. Scott, C.G. Wonder Stouw, *Jour. Chem. Inf. Comp. Sci.*, **15** (1975), 73
25. G.M. Dyson, *The Dyson-IUPAC Notation*, v knjigi *Chemical Information Systems*, ed. Ash & Hyde, J. Wiley & Sons, New York, N.Y., 1975
26. C.E. Granito, E. Garfield, *Naturwissenschaften*, **60** (1973), 189
27. L.C. Ray, R.A. Kirsh, *Science*, **126** (1957), 814
28. M.O. Robin, *Comm. ACM*, **20** (1977), 625
29. T.K. Ming, S.J. Tauber, *J. Chem. Doc.*, **11** (1971), 47
30. D.G. Corneil, C.C. Gotlieb, *Jour. ACM*, **17** (1970), 51
31. Y. Shah, G. David, M. McCarthy, *Trans. on Systems Man and Cybernetics*, SMC-4 (1974), 313
32. E.J. Corey, W.T. Wipke, *Science*, **166** (1969), 179
33. R.J. Feldman, S.R. Heller, *Jour. Chem. Doc.*, **12** (1973), 48
34. R.J. Feldman, *Interactive Graphic Chemical Structure Searching*, v knjigi *Computer Representation and Manipulation of Chemical Information*, ed. Wipke, Heller, Feldman, Hyde, J. Wiley & Sons, New York, N.Y., (1974)
35. E. Meyer, *Topological Search of Compounds in Large Files*, v knjigi *Computer Representation and Manipulation of Chemical Information*, ed. Wipke, Heller, Feldman, Hyde, J. Wiley & Sons, New York, N.Y., 1974
36. E. Ziegler, K. Boll, *Anal. Chem. Acta*, *Computer Technique and Optimization*, **103** (1978) 237
37. A. Zamora, D.L. Dayton, *Jour. Chem. Inf. Comp. Sci.*, **8** (1976), 74
38. E. Wander Stouw, P.M. Elliot, A.C. Isenberg, *Jour. Chem. Doc.*, **14** (1979), 185
39. E. Hyde, L. Matthews, L.H. Thompson, J.W. Wiswesser, *Jour. Chem. Doc.*, **7** (1967), 200
40. S.J. Tauber, S.J. Fraction, H.W. Hayward, v knjigi *Chemical Structures as Information: Representation Transformations and Calculations*, Spartan Books, Washington D.C., 1965
41. J.E. Ash, *Connection Tables and Their Role in a System*, v knjigi *Chemical Information Systems*, J. Wiley & Sons, New York, N.Y., 1974
42. D. Lefkowitz, *Jour. Chem. Doc.*, **1** (1967), 186
43. L.H. Campey, E. Hyde, H. Jackson, *Chem. Br.*, **6** (1970), 427
44. C. Ebe, T. Tommy, A. Zamora, *Jour. Chem. Inf. Comp. Sci.*, **16** (1976), 36
45. C.M. Bowman, F.A. Lander, N.W. Lee, M.H. Reslock, *Jour. Chem. Doc.*, **8** (1968), 133
46. C.D. Farrell, A.R. Chauvenet, D.A. Conner, *Jour. Chem. Doc.*, **11** (1971), 52
47. J.E. Dubois, *DARC System in Chemistry*, v knjigi *Computer Representation and Manipulation of Chemical Information*, ed. Wipke, Heller, Feldman, Hyde, J. Wiley & Sons, New York, N.Y., 1974
48. R. Rogers, M.A.T. "CROSSBOW", *Proc. 158th National Meeting of ACS*, New York, N.Y., 1969
49. B.L. Zimmerman, *Computer Generated Chemical Structural Formulas with Standard Ring Orientation*, doktorska disertacija University of Pennsylvania, Philadelphia, 1971
50. P.G. Dittmar, J. Mockus, *Jour. Chem. Inf. Comp. Sci.*, **17** (1977), 186
51. J. Blake, N. Farmer, R. Haines, *Jour. Chem. Inf. Comp. Sci.*, **17** (1977), 223
52. H. Skolnik, *Jour. Chem. Inf. Comp. Sci.*, **17** (1977), 234
53. A. Stuper, W. Brugger, P.C. Jurs, *Computer-assisted Studies of Chemical Structures and Biological Function*, J. Wiley & Sons, New York, N.Y., 1979, poglavja 5,6,7
54. P.C. Jurs, T.L. Isenhour, *Chemical Application of Pattern Recognition*, J. Wiley & Sons, New York, N.Y., 1975
55. J. Walls, *Chemical Reaction Indexing v knjigi Chemical Information Systems*, J. Wiley & Sons, New York, ed. Ash & Hyde, N.Y., 1975
56. V.V. Serov, M.E. Elyashberg, L.A. Gribov, *J. Mol. Str.*, **31** (1976), 381
57. G.R. Marshall, H.E. Bossard, R.A. Ellis, *Computer Handling of Chemical Structures: Application in Crystallography, Conformational Analysis and Drug Design*, v knjigi *Computer Representation and Manipulation of Chemical Structures*, ed Wipke, Heller, Feldman, Hyde, J. Wiley & Sons, New York, N.Y., 1975
58. A. Stuper, W. Brugger, P.C. Jurs, *Computer-Assisted Studies of Chemical Structures and Biological Function*, J. Wiley & Sons, New York, N.Y., 1979, poglavje 8
59. M. Bersohn, M. Esack, *Chemical Reviews*, **76** (1976), 269