“ALQORITMLƏR NƏZƏRIYYƏSI” FƏNNININ MÜHAZIRƏLƏRI

Mühazirə 1. Alqoritm anlayışı, onun xassələri və tərifinin dəqiqləşməsi zərurəti

Alqoritm dedikdə müəyyən bir məsələnin həlli üçün dəqiq və aydın əməliyyatlar ardıcıllığı başa düşülür. Başqa sözlə, alqoritm ilkin verilənlər üzərində elə əməliyyatlar ardıcıllığıdır ki, onları yerinə yetirdikdə baxılan məsələnin həlli alınır və s. Qeyd edək ki, bunlar alqoritmin intuitiv anlayışıdır. Çünki alqoritmin dəqiq riyazi tərifi yoxdur. Lakin biz hər zaman alqoritmdən istifadə edirik. Məsələn: ədədlərin vurulması alqoritmi, kvadrat tənliyin həlli alqoritmi və s. Bu vaxta qədər bizə məlum olan alqoritmlərə əsasən onun bəzi xassələrini yaza bilərik.

1. Diskretlik xassəsi. Hər bir alqoritm müəyyən diskret addımdan ibarətdir. Hər bir addımdakı əməliyyatlar diskret taktlarla yerinə yetirilir.

2. Determiniklik (müəyyənlik) xassəsi. Bu xassəyə görə hər bir alqoritmdəki əməliyyatlar dəqiq və aydın olmalıdır.

3. Kütləvilik xassəsi. Bu xassəyə görə müəyyən bir sinifdən olan məsələnin həll alqoritmi həmin sinifdən olan istənilən məsələni həll etməlidir.

4. Sonluluq xassəsi. Alqoritmin sona qədər yerinə yetirilməsi üçün lazım gələn əməliyyatların maksimal sayı sonlu və real olmalıdır.

5. Nəticəvilik xassəsi. Hər bir alqoritm yerinə yetirildikdən sonra hökmən müəyyən nəticə hasil olmalıdır.

Əgər hər hansı alqoritm bu xassələrdən heç olmazsa birini ödəmirsə o alqoritm deyil. Ümumiyyətlə biz hər hansı bir məsələni həll etdikdən sonra onun həll alqoritmi haqqında nəticə çıxara bilərik. Lakin hər hansı problemi həll edə bilmədikdə onun həll alqoritminin olub olmamağı haqqında hökm çıxara bilmirik. Bunun səbəbi odur ki, alqoritmin dəqiq riyazi tərifini bilmirik. əgər bu tərifi bilsəydik onda həmin problemin bu tərifi ödəyib ödəmədiyini yoxlayardıq. Əgər ödənilərdisə onda həmin problemin həll alqoritmi olar, əks halda isə belə bir alqoritm olmazdı. Beləliklə alqoritmə dəqiq riyazi tərif vermək zərurəti meydana çıxır.

Mühazirə 2. Hesablanan və qismən rekursiv funksiyalar. Klini-Çerç tezisi və onun əhəmiyyəti:

Alqoritmik problemlərdə ümumiliyi pozmadan həmişə arqumentləri mənfi olmayan tam qiymətlər alan funksiyanın mənfi olmayan tam qiymətlərinin tapılmasından söhbət gedir. Ümumiliyi pozmadan bunu həmişə qəbul etmək olar. Qiyməti müəyyən bir alqoritm vasitəsilə tapılan funksiyalara hesablanan funksiyalar deyilir. Bu tərif intuitivdir. Dəqiq riyazi deyil. Çünki burada alqoritm anlayışından istifadə olunur. Arqumentlərinin heç də hamısında təyin olunmayan funksiyalar, yəni arqumentlərinin müəyyən hissəsində təyin olunmuş funksiyalara qismən funksiyalar deyilir. Qiyməti hər hansı alqoritmin tətbiqi ilə tapıla bilən qismən funksiyalara hesablanan qismən funksiyalar deyilir. Bu vaxta qədər məlum olan bütün hesablanan qismən funksiyalar məlum olmuşdur ki, qismən rekursiv funksiyalardır. Rekursiv funksiyaların isə ciddi riyazi tərifi var. Bundan sonra biz funksiya dedikdə ilkin verilənlər üzərində müəyyən əməllər ardıcıllığı, yığımı başa düşəcəyik. Deməli, bizə məlum olan alqoritmin hər birinə biz müəyyən funksiya kimi baxa bilərik. Klini belə bir tezisi irəli sürmüşdür ki, alqoritmik hesablanan qismən funksiyalar sinfi ilə qismən rekursiv funksiyalar sinfi üst üstə düşür. Qeyd edək ki, hesablanan qismən funksiyaların tərifi intuitiv olduğu halda qismən rekursiv funksiyaların tərifi dəqiq riyazi şəkildə verilir. Kliniden bir qədər əvvəl Çerç belə bir tezis vermişdir ki, hər yerdə təyin olunmuş hesablanan qismən funksiyalar sinfi ilə qismən rekursiv funksiyalar sinfi eynidir. Bu iki tezis birləşdirilib Klini-Çerç tezisi adı ilə verilir. Tezisin əhəmiyyəti ondan ibarətdir ki, hər hansı problemi əks etdirən funksiyaların rekursiv funksiyalar tərifini ödədiyini bilsək onun həll alqoritminin olduğu birbaşa aydındır. Lakin o, problemi əks etdirən funksiya rekursiv funksiya tərifini ödəmədikdə deyirik ki, Klini-Çerç tezisinə görə həmin məsələnin həll alqoritmi yoxdur. Beləliklə, rekursiv funksiya anlayışını verməliyik. Onun tərifini və xassələrini göstərməklə, alqoritmin tərifini dəqiqləşdirmək olar. Alqoritmin tərifini dəqiqləşdirmək üçün yuxarıdakı birinci yanaşmadır.

Mühazirə 3. Alqoritmin tərifinin dəqiqləşməsi üçün yanaşmalar:

İkinci yanaşma Türing-Post maşını nəzəriyyəsidir. Türing-Post bir birindən asılı olmayaraq elə nəzəri hesablama aparatı yaratmışlar ki, orada funksiyaların qiymətlərini hesablamaq mümkündür. Məlum olmuşdur ki, bu vaxta kimi məlum alqoritmlərin hamısının Türing-Post maşınında yerinə yetirmək mümkündür. Beləliklə, hər hansı problemi Türing-Post maşınında həll etmək olursa deməli onun həll alqoritmi var. Bu da bir daha Klini-Çerç tezisinə inam yaradır.

Alqoritmin tərifinin dəqiqləşməsi üçün üçüncü yanaşma "Normal-Markov alqoritmi" nəzəriyyəsidir. Bu nəzəriyyəyə görə baxılan problemin ilkin verilənləri sözlər şəklində yazılır. Söz dedikdə müəyyən sonlu simvollar yığımı başa düşülür. Bu nəzəriyyəyə görə hər bir əməliyyat sözlər üzərində əməliyyata gətirilir. Yenə də bu vaxta qədər məlum alqoritmlərin hamısı Normal-Markov alqoritmi vasitəsilə yerinə yetirilə bilir. Deməli, Normal-Markov alqoritmi nəzəriyyəsi də alqoritmin tərifinin dəqiqləşməsidir və Normal-Markov alqoritmi ilə yerinə yetirilən hesablanan funksiya rekursiv funksiyadır. Beləliklə, bu yanaşmaların üçünün də müəyyən mənada ekvivalent olduğu aydın olur. Ona görə də biz rekursiv funksiyalar haqqında bəzi anlayışları və tərifləri öyrənəcəyik.

Mühazirə 4.Qismən funksiya anlayışı, Ən sadə qismən funksiyalar və superpozisiya (əvəzləmə) operatoru.

Tutaq ki, bizə ixtiyari təbiətli $X=\{x_1,x_2,...\}$ və $Y=\{y_1,y_2,...\}$ çoxluqları verilib. Əgər X çoxluğunun müəyyən elementlərinə yeganə qayda ilə Y çoxluğunun müəyyən elementləri uyğun qoyularsa, onda deyirlər ki, X çoxluğundan Y çoxluğuna təsir edən qismən funksiya verilmişdir. Başqa sözlə bu uyğunluğa qismən funksiya deyilir. Qeyd edək ki, bu vaxta qədər bildiyimiz funksiya tərifi qismən funksiyanın xüsusi halıdır. Biz qismən funksiyanı $y=f(x)$ ilə işarə edəcəyik. X çoxluğunun Y çoxluğunda uyğun elementləri olan yığımına biryerli funksiyanın müəyyənlik oblastı deyilir. Əgər funksiyanın müəyyənlik oblastı bütün X çoxluğu ilə üst-üstə düşərsə, onda bu funksiyaya hər yerdə təyin olunmuş funksiya deyilir. Y çoxluğunun elə elementlərinə qismən funksiyanın qiymətlər oblastı deyilir ki, həmin elementlərə X çoxluğunda uyğun gələn elementlər olsun. $y=f(x)$ funksiyasında x-ə sərbəst dəyişən, y-ə isə asılı dəyişən deyilir. Əgər X çoxluğunun x_i elementinə Y çoxluğunun uyğun y_i qiyməti uyğun qoyularsa və tərsinə onda deyəcəyik ki, $y=f(x)$ funksiyası qarşılıqlı birqiymətli funksiyadır. Başqa sözlə $x_i \neq x_j \Leftrightarrow y_i \neq y_j$ $i \neq j$ onda deyirlər ki, $y=f(x)$ funksiyası qarşılıqlı birqiymətlidir. İki g və f funksiyaları o zaman bərabər hesab olunur ki, onların müəyyənlik oblastları üst-üstə düşsün və həmin müəyyənlik oblastındakı uyğun $f(x_i)$ və $g(x_i)$ qiymətləri də bərabər olsun yəni, $f(x_i)=g(x_i)$

Tutaq ki, $X_i, X_2,...,X_n$ çoxluqlarının dekart hasili $X_1 \times X_2 \times ... \times X_n$-dir. Onda $X_i, X_2,...,X_n$ çoxluğundan müəyyən Y çoxluğuna təsir edən qismən funksiyaya n yerli qismən funksiya deyəcəyik. Natural ədədlər çoxluğunu N ilə işarə edək. $\underbrace{N \times N \times ... \times N}_{k}$ çoxluğundan N çoxluğuna təsir edən qismən funksiyaya k yerli ədədi qismən funksiya deyəcəyik.

Mühazirə 4. Ən sadə qismən funksiyalar və superpozisiya (əvəzləmə) operatoru

Sonralar bizə lazım olan aşağıdakı üç ən sadə funksiyanı yazaq.

1. $S(x) = x+1$ izləmə funksiyası.

2. $C_a^n(x_1, x_2,...,x_n) = a$ n yerli sabit funksiya.

3. $I_m^n(x_1, x_2,..., x_n) = x_m$, $(m \le n,\ n = 1,2,3...)$ eynilik funksiyası.

Funksiyalar üzərindəki çevrilmələrə operator deyilir. Rekursiv funksiyaların tərifinin verilməsi və qurulması üçün bizə lazım gələn bəzi əsas operatorları öyrənək.

1. Superpozisiya (əvəzləmə) operatoru.

Tutaq ki, bizə müəyyən A çoxluğundan B çoxluğuna təsir edən n ədəd m yerli $f_1(x_1, x_2,..., x_m)$, $f_2(x_1, x_2,..., x_m)$, ... , $f_n(x_1, x_2,..., x_m)$ funksiyaları verilmişdir. Fərz edək ki, B çoxluğundan C çoxluğuna təsir edən n yerli müəyyən $f(*,*,...,*)$ funksiyası verilmişdir.

Aşağıdakı kimi m yerli g funksiyasını quraq.

$$g_1(x_1, x_2,..., x_m) = f(g(x_1, x_2,..., x_m), f_2(x_1, x_2,..., x_m),..., f_n(x_1, x_2,..., x_m)).$$

m yerli g funksiyasını almaq üçün $f_{1,} f_2,..., f_n$ funksiyaları üzərindəki f çevirməsinə superpozisiya və yaxud əvəzləmə operatoru deyilir. Bu operatoru S^{n+1} ilə işarə edək. Superpozisiya operatorunu tətbiq edərkən verilmiş $f_{1,} f_2,..., f_n$ funksiyalarının hamısı eyni m yerli olmalıdır. Əgər bunlardan hər hansı birində dəyişənlərin sayı digərlərindən az olarsa onda fiktiv dəyişənlər qəbul etməklə hamısını eyni yerli etmək olar. Məsələn: Fiktiv dəyişən götürməklə $\varphi(x_1, x_2)$ funksiyasını 3 dəyişənli etmək olar. Belə ki,

$$\varphi(x_1, x_2) = \varphi(I_1^3(x_1, x_2, x_3), I_2^3(x_1, x_2, x_3)) = \xi(x_1, x_2, x_3).$$

Mühazirə 5. Primitiv rekursiya operatoru və ona aid misallar.

Tutaq ki, n yerli müəyyən $g(*,*,...,*)$ və $n+2$ yerli $h=(*,*,...,*)$ ədədi qismən funksiyaları verilmişdir. $n+1$ yerli f funksiyasını aşağıdakı kimi quraq

$$\begin{aligned} & f(x_1, x_2, ..., x_n, 0) = g(x_1, x_2, ..., x_n), \\ & f(x_1, x_2, ..., x_n, y+1) = h(x_1, x_2, ..., x_n, y, f(x_1, x_2, ..., x_n, y)). \end{aligned} \qquad (1)$$

Verilmiş g və h ədədi qismən funksiyalarına görə (1) münasibəti vasitəsilə $n+1$ yerli funksiyasının qurulma prosesinə primitiv rekursiya operatoru deyilir. (1) prosesini bir qədər açıq yaza bilərik.

$$f(x_1, x_2, ..., x_n, 0) = g(x_1, x_2, ..., x_n)$$

$$f(x_1, x_2, ..., x_n, 1) = h(x_1, x_2, ..., x_n, 0, g(x_1, x_2, ..., x_n))$$

$$f(x_1, x_2, ..., x_n, 2) = h(x_1, x_2, ..., x_n, 1, f(x_1, x_2, ..., x_n, 1))$$

- -

$$f(x_1, x_2, ..., x_n, k+1) = h(x_1, x_2, ..., x_n, k, f(x_1, x_2, ..., x_n, k)).$$

Primitiv rekursiya operatorunu $f = R(g, h)$ işarə edilir. Əgər verilmiş g və h funksiyaları hər yerdə təyin olunmuş funksiyalardırsa onda primitiv rekursiya operatoru vasitəsilə alınan f funksiyası da hər yerdə təyin olunmuşdur.

Çünki f funksiyası g və h funksiyaları vasitəsilə təyin olunur. g və h funksiyaları hər yerdə təyin olunduğundan f funksiyası da hər yerdə təyin olunacaq:

$g(x_1,...,x_n)$ hər yerdə təyin olunub.

$f(x_1,...,x_n,1) = h(x_1,...,x_n,0, g(x_1,...,x_n))$ g və h hər yerdə təyin olunduğundan f də hər yerdə təyin olunub.

$f(x_1,...,x_n,2) = h(x_1,...,x_n,1, g(x_1,...,x_n))$ olduğundan bu da hər yerdə təyin olunub və s.

(1) primitiv rekursiya operatoru ilə qurulan f funksiyası müəyyən $k-1$ üçün təyin olunubsa, yəni, $f(x_1, x_2, ..., x_n, k-1)$ varsa lakin, $f(x_1, x_2, ..., x_n, k)$ təyin olunmayıbsa, onda ixtiyari $p \geq k+1$ qiymətlərində də $f(x_1, x_2, ..., x_n, p)$ təyin olunmayacaq.

Beləliklə, əgər g və h funksiyaları hər yerdə təyin olunubsa, onda (1) prosesi vasitəsilə yeganə f funksiyası tapılar. Aydındır ki, primitiv rekursiya operatoru vasitəsilə tapılan f funksiyası $x_1, x_2, ..., x_n$ və y dəyişənlərinin ədədi qismən funksiyasıdır.

Misal:

1. $g = 0$, $h(x, y) = 2x + y$ funksiyasına primitiv rekursiya operatorunu tətbiq edək.

$f(0) = g = 0$

$f(1) = h(0, f(0)) = h(0,0) = 2.0 + 0 = 0$,

$f(2) = h(1, f(1)) = h(1,0) = 2.1 + 0 = 2$,

$f(3) = h(2, f(2)) = h(2,2) = 2.2 + 2.1$,

$f(4) = h(3, f(3)) = h(3,6) = 2.3 + 2.2 + 2.1$,

$f(5) = h(4, f(4)) = 2.4 + 2.3 + 2.2 + 2.1$,

$$f(x+1) = h(x, f(x)) = 2.x + 2.(x-1) + ... + 2.1 = 2(1 + 2 + ... + x) = 2\frac{1+x}{2}x$$

$$f(x+1) = x(x+1)$$

Qeyd edək ki, 1-ci misalda $h = 2x + y$ funksiyasında toplananların yerini dəyişsək funksiya dəyişmədiyi halda ona tətbiq olunan primitiv rekursiya operatoru tamam başqa cür funksiyanı verir.

Misal 2.

$g = 0$, $h(x, y) = x + 2y$ funksiyalarına primitiv rekursiya operatorunu tətbiq edək.

$f(x+1) = ?$

$f(0) = g = 0$,

$f(1) = h(0, f(0)) = h(0,0) = 0 + 2.0 = 0$,

$f(2) = h(1, f(1)) = h(1,0) = 1 + 2.0 = 1$,

$f(3) = h(2, f(2)) = h(2,1) = 2 + 2.1 = 4$,

$f(4) = h(3, f(3)) = h(3,2 + 2.1) = 3 + 2.2 + 2^2.1$

$f(5) = h(4, f(4)) = h(4,3 + 2.2 + 2^2.1) = 4 + 3.2 + 2.2^2 + 1.2^3$

$f(6) = h(5, f(5)) = h(5,4 + 3.2 + 2.2^2 + 1.2^3) = 5 + 4.2 + 3.2^2 + 2.2^3 + 1.2^4$

$f(x+1) = x + 2^1(x-1) + 2^2(x-2) + 2^3(x-3) + ... + 2^{x-1}(x-(x-1)) =$

$$= \sum_{k=0}^{x-1} 2^k (x-k) = ... = 2^{x+1} - x - 2$$

Beləliklə alırıq ki, $f(x+1) = 2^{x+1} - x - 2$.

Misal 3.

$g = 0$, $h(x, y) = 2x - y$ funksiyalarına primitiv rekursiya operatorunu tətbiq edək.

$f(x+1) = ?$

Misal 4.

$g = 0$, $h(x, y) = x - 2y$

$f(0) = 0$

$f(1) = h(0, f(0)) = 0$

$f(2) = h(1, f(1)) = h(1,0) = 1 - 2.0 = 1$

$f(3) = h(2, f(2)) = h(2,1) = 2 - 2 = 0$

$f(4) = h(3, f(3)) = h(3,0) = 3 - 2.0 = 3$

$f(5) = h(4, f(4)) = h(4,3) = 4 - 3.2 = -2$

$f(5)$ mənfi olduğundan bu funksiyaya primitiv rekursiya operatorunu tətbiq etmək olmaz.

Mühazirə 6. Minimallaşma operatoru və ona aid misallar

Tutaq ki, n yerli $f(x_1, x_2, ..., x_n)$ ədədi qismən funksiyası verilmişdir və arqumentlərinin $x_1, x_2, ..., x_n$ verilmiş qiymətlərində funksiyanın qiymətini hesablamaq mümkündür. Onda $x_1, x_2, ..., x_{n-1}$ arqumentlərini qeyd edib aşağıdakı tənliyi quraq.

$$f(x_1, x_2, ..., x_{n-1}, y) = x_n \qquad (1)$$

(1) tənliyində y-in yerinə 0,1,2,... qiymətlərini yazmaqla həmin tənliyin ödənilməsini yoxlaya bilərik. y dəyişəninin (1) tənliyinin ödəyən qiymətlərindən ən kiçiyinin tapılması prosesinə minimallaşma operatoru deyilir. y-in həmin ən kiçik qiymətini a ilə işarə edək.

$a = \min\{\, y \geq 0 \ \text{ tam} \ / \ f(x_1, x_2, ..., x_{n-1}, y) = x_n \,\}$

Minimallaşma operatorunu M_f ilə işarə edək

$M_f = \mu_y(f(x_1, x_2, ..., x_{n-1}, y) = x_n)$

Birbaşa aydındır ki, minimallaşma operatoru $x_1, x_2, ..., x_n$ dəyişənlərin qismən funksiyasıdır.

Minimallaşma operatoru aşağıdakı hallarda təyin olunmayıb.

1. $f(x_1, x_2, ..., x_{n-1}, 0)$ təyin olunmayıb

2. $f(x_1, x_2, ..., x_{n-1}, y)$ funksiyası $y = 0,1,2,...,k$ qiymətlərində təyin olunub lakin, x_n-ə bərabər deyil. Eyni zamanda $y = k+1$ olduqda isə təyin olunmayıb.

3. $f(x_1, x_2, ..., x_{n-1}, y)$ funksiyası y-in mənfi olmayan bütün tam qiymətlərində təyin olunub lakin, x_n-ə bərabər olmur.

Göründüyü kimi minimallaşma operatoru ümumiyyətlə hər yerdə təyin olunmayıb. Beləliklə, hər bir n yerli f funksiyasına görə müəyyən

$$a = \mu_y(f(x_1, x_2, ..., x_{n-1}, y) = x_n)$$

kiçik ədədi tapıla bilərsə onda deyəcəyik ki, minimallaşma operatorunun nəticəsi var. Əks halda minimallaşma operatoru nəticə vermir.

Misallara baxaq.

1. $f(y) = 2^y$ funksiyasına minimallaşma operatorunu tətbiq edək.

$f(x) = M_f = \mu_y(2^y = x) = \log_2 x$ burada $x = 2^k$, $k = 0,1,...$

Beləliklə, $f(y) = 2^y$ funksiyasına minimallaşma operatorunu tətbiq etdikdə $f(x) = \log_2 x$ funksiyasını alırıq. Deməli, bir yerli funksiyaya minimallaşma operatorunun tətbiqi onun tərs funksiyasını verir.

2. $f(x, y) = y - x$ funksiyasına minimallaşma operatorunu tətbiq edək

$M(f(x,z)) = \mu_y(y - x = z)$.

Bu tənlik isə $y = 0$, $x > 0$ təyin olunmayıb. Digər tərəfdən $y - x = z$ tənliyindən $y = x + z$ həllini alırıq. Bu onu göstərir ki, minimallaşma operatoru (1) tənliyinin hər hansı həllini yox onun ən kiçik natural həllini verməlidir.

Mühazirə 7.Primitiv rekursiv funksiya.

Müxtəlif pilləli primitiv rekursiv funksiyalar:

$S(x)$, C_0^1 və I_m^n funksiyalarından heç olmasa birinə və ya bu funksiyalara superpozisiya, yaxud primitiv rekursiya operatorlarının sonlu sayda tətbiqi nəticəsində alınan funksiyalara primitiv rekursiv funksiya deyilir. Bu tərifdən görünür ki, primitiv rekursiv funksiyalar hər yerdə təyin olunmuş funksiyalardır.

Misal:

Göstərək ki, $C_a^n(x_1, x_2, ..., x_m) = a$ sabit funksiyası primitiv rekursiv funksiyadır.

$I_1^n(x_1, x_2, ..., x_n) = x_1$,

$C_0^1(I_1^n(x_1, x_2, ..., x_n)) = C_0^1(x_1) = 0$,

$S(C_0^1(I_1^n(x_1, x_2, ..., x_n)) = S(0) = 1$,

$S(S(C_0^1(I_1^n(x_1, x_2, ..., x_n)))) = S(1) = 2$,

$\underbrace{S(S(S...S}_{a}(C_0^1(I_1^n(x_1, x_2, ..., x_n)))...)) = a$.

a- natural ədədini almaq üçün S, C_0^1 və I_1^n funksiyalarına superpozisiya operatorunu tətbiq etdik. Ona görə də $c_a^n(x_1, x_2, ..., x_n) = a$ funksiyası primitiv rekursiv funksiyadır.

$P_0(x, y) = x + y$, $P_1(x, y) = xy$, $P_2(x, y) = x^y$ funksiyalarına uyğun olaraq, sıfırıncı, birinci və ikinci pilləli funksiyalar deyilir.

1) Sıfırıncı pilləli $P_0(x, y) = x + y$ funksiyası primitiv rekursiv funksiyadır.

Çünki,

$$\begin{cases} P_0(x,0) = x = I_1^1(x) \\ P_0(x, y+1) = S(P_0(x, y)) = h(x, y, P_0(x, y)) \end{cases}$$

$P_0(x,1) = S(P_0(x,0)) = S(x) = x + 1$

$P_0(x,2) = S(P_0(x,1)) = S(x+1) = x + 2$

- -

$P_0(x, k+1) = S(P_0(x, k)) = S(x+k) = x + k + 1$

2) $P_1(x,y)=xy$ funksiyası primitiv rekursiv funksiyadır. Bu funksiyanı qurmaq üçün aşağıdakı kimi primitiv rekursiya operatorunu yazaq.

$$\begin{cases} P_1(x,0)=0=g(x) \\ P_1(x,y+1)=P_0(x,P_1(x,y))=h(x,y,P_1(x,y)) \end{cases} \qquad (1)$$

Göstərək ki, (1) münasibətində yazılmış primitiv rekursiya operatoru birinci pilləli funksiyanı verir.

$$P_1(x,1)=P_0(x,P_1(x,0))=P_0(x,0)=x=1.x\,,$$

$$P_1(x,2)=P_0(x,P_1(x,1))=P_0(x,x)=x+x=2.x\,,$$

$$P_1(x,3)=P_0(x,P_1(x,2))=P_0(x,2x)=x+2x=3.x\,,$$

- -

$$P_1(x,k+1)=P_0(x,P_1(x,k))=P_0(x,kx)=x+kx=(k+1).x$$

Beləliklə, (1) münasibətindəki primitiv funksiya operatoru birinci pilləli funksiyanı verir.

3) $P_2(x,y)=x^y$ funksiyası da primitiv rekursiv funksiyadır. Bunun üçün aşağıdakı kimi primitiv rekursiya operatoru yazaq.

$$\begin{cases} P_2(x,0)=1=S(0)=g(x) \\ P_2(x,y+1)=P_1(x,P_2(x,y)) \end{cases} \qquad (2)$$

Doğrudan da,

$$P_2(x,1)=P_1(x,P_2(x,0))=P_1(x,1)=x.1=x^1$$

$$P_2(x,2)=P_1(x,P_2(x,1))=P_1(x,x)=x.x=x^2$$

$$P_2(x,3)=P_1(x,P_2(x,2))=P_1(x,x^2)=x.x^2=x^3$$

- -

$$P_2(x,k+1)=P_1(x,P_2(x,k))=P_1(x,x^k)=x.x^k=x^{k+1}$$

Aşağıdakı primitiv rekursiya operatoru vasitəsilə istənilən n-ci pilləli $P_n(x,y)$ funksiyasını qurmaq olar.

$$\begin{cases} P_n(x,0)=1=g(x) \\ P_n(x,y+1)=P_{n-1}(x,P_n(x,y)) \end{cases} \qquad (3)$$

(3) münasibətindən istifadə edərək 3-cü pilləli funksiyanı quraq.

$$P_n(x,0)=1=S(0)=g(x)$$

$$P_n(x, y+1) = P_{n-1}(x, P_n(x, y))$$

$$P_3(x,1) = P_2(x, P_3(x,0)) = P_2(x,1) = x.1 = x^1$$

$$P_3(x,2) = P_2(x, P_3(x,1)) = P_2(x,x) = x^x$$

$$P_3(x,3) = P_2(x, P_3(x,2)) = P_2(x, x^x) = x^{x^x}$$

- -

$$P_3(x,k) = P_2(x, P_3(x,k-1)) = \left. x^{x^{x^{\cdot^{\cdot^{\cdot^{x}}}}}} \right\} k$$

Göstərin ki, aşağıdakı funksiyalar primitiv rekursiv funksiyalardır.

1) $f(x,y) = x \underline{\ \ \dot{}\ \ } y = \begin{cases} x-y, & x \ge y \\ 0, & x < y \end{cases}$, bu kəsik fərq funksiyasıdır.

2) $f(x,y) = |x-y| = \begin{cases} x-y, & x \ge y \\ y-x, & x < y \end{cases}$, bu mütləq qiymət funksiyasıdır.

3) $Sg(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \end{cases}$ - işarə funksiyası.

4) $\overline{S}g(x) = \begin{cases} 1, & x = 0 \\ 0, & x > 0 \end{cases}$ -əks işarə funksiyası.

Mühazirə 8.Qismən rekursiv funksiya anlayışı

S, C_0^1 və I_m^n funksiyalarından hər hansı birinə və ya bu funksiyalara sonlu sayda superpozisiya, primitiv rekursiya, yaxud minimallaşma operatorlarının tətbiqi nəticəsində alınan funksiyalara qismən rekursiv funksiyalar deyilir. Qeyd edək ki, qismən rekursiv funksiyalar sinfi primitiv rekursiv funksiyalar sinfindən daha genişdir. Çünki, hər bir primitiv rekursiv funksiya eyni zamanda həm də qismən rekursiv funksiyadır. Çünki, primitiv rekursiv funksiya S, C_0^1 və I_m^n funksiyalarından hər hansı birinə və ya bu funksiyalara sonlu sayda yalnız superpozisiya və ya primitiv rekursiya operatorunu tətbiq etməklə alınır. Qismən rekursiv funksiyalar hər yerdə təyin olunmaya da bilər, amma primitiv rekursiv funksiyalar hər yerdə təyin olunub.

Qismən rekursiv funksiyalar sinfində hər yerdə təyin olunmayanı da ola bilər. Məsələn,

$$f(x,y) = \mu_z(x + y + z = 1), \quad x > 0, \quad y > 0$$

Bu funksiya hər yerdə təyin olunmayıb lakin, qismən rekursiv funksiyadır. Çünki sıfırıncı pilləli $x + y$ funksiyasına minimallaşma operatoru tətbiq olunub. Qismən rekursiv funksiyalara misallar göstərək.

1. Göstərək ki, $f(x,y) = x - y$ funksiyası qismən rekursiv funksiyadır. Bu funksiyanı aşağıdakı kimi ifadə edə bilərik.

$$x - y = \mu_z(y + z = x)$$

Buradan görünür ki, $y + z$ funksiyasına minimallaşma operatorunu tətbiq etdikdə $x - y$ alınır. Yəni, $x - y$ funksiyası $y + z$ primitiv rekursiv funksiyasından minimallaşma operatorunun vasitəsilə alınır. Deməli, $x - y$ funksiyası qismən rekursiv funksiyadır.

2. Göstərək ki, $f(x,y) = \dfrac{x}{y}$ funksiyası qismən rekursiv funksiyadır. Bu funksiyanı aşağıdakı kimi ifadə edək.

$$\frac{x}{y} = \mu_z(y\,z = x)$$

Əvvəl göstərmişdik ki, $y\,z$ birinci pilləli funksiya, primitiv rekursiv funksiyadır. Yəni, onu primitiv rekursiv operatoru vasitəsilə qurmuşuq. İndi isə həmin birinci pilləli $y.z$ funksiyası minimallaşma operatorunun tətbiqindən sonra $\frac{x}{y}$-ə çevrilir. Deməli, $\frac{x}{y}$ funksiyası qismən rekursiv funksiyadır.

3. Tutaq ki, müəyyən $f(x)$ funksiyası primitiv rekursiv funksiyadır və $a_1, a_2, ..., a_n$ müəyyən natural ədədlərdir. Onda

$$g(x) = f(x) - \overline{S}g|x - a_1| - \overline{S}g|x - a_2| - ... - \overline{S}g|x - a_n|$$

funksiyası qismən rekursiv funksiya olar. Çünki, $x \neq a_i$, $i = \overline{1,n}$ qiymətlərində $g(x)$ funksiyası hər yerdə təyin olunub. $x = a_i$, $i = \overline{1,n}$ qiymətlərində isə $g(x)$ funksiyası ola bilər ki, hər yerdə təyin olunmayıb.

Beləliklə, hər bir $f(x)$ qismən rekursiv funksiyası üçün ixtiyari $x = a$ natural qiymətində elə hesablama prosesi var ki, həmin proses $f(x)$ funksiyasını $f(a)$ ədədinə çevirir. Belə hesablama prosesi o zaman nəticə vermir ki, $f(x)$ funksiyası heç də hər yerdə təyin olunmayıb. Ona görə də hər yerdə təyin olunmuş rekursiv funksiya anlayışı vermək lazım gəlir. Bu məqsədlə aşağıdakı kimi zəif minimallaşma operatoru anlayışını verək.

Mühazirə 9.Zəif minimallaşma operatoru, Ümumi rekursiv funksiyalar:

Aşağıdakı kimi zəif minimallaşma operatoru təyin edək:

$$M_f^z = \begin{cases} M_f & \text{əgər } M_f \text{ hər yerdə təyin olunubsa} \\ yoxdur, & \text{yəni təyin olunmayib.} \end{cases}$$

Əgər M_f müəyyən yerdə təyin olunmayıbsa, yəni heç də hər yerdə təyin olunmayıbsa, Onda M_f^z -hər yerdə təyin olunub. Yazılışından göründüyü kimi zəif minimallaşma operatoru adi minimallaşma operatoru ilə o zaman eyni nəticə verir ki, adi minimallaşma operatoru hər yerdə təyin olunsun. Lakin adi minimallaşma operatoru heç olmasa bir nöqtədə təyin olunmayıbsa, zəif minimallaşma operatoru bütün yerlərdə təyin olunub. Ona görə də buna zəif minimallaşma operatoru deyilir. Lakin zəif minimallaşma operatoru hər yerdə təyin olunduğundan o, adi minimallaşma operatorundan daha güclüdür.

Tərif: S, C_0^1 və I_m^n funksiyalarından hər hansı birinə və ya bu funksiyalara super pozisiya, primitiv rekursiya yaxud, zəif minimallaşma operatorlarının sonlu sayda tətbiqi nəticəsində alınan funksiyalara ümumi rekursiv funksiyalar deyilir.

Super pozisiya, primitiv rekursiya və ya zəif minimallaşma operatorlarını hər yerdə təyin olunmuş funksiyalara tətbiq etdikdə yenə də hər yerdə təyin olunmuş funksiya alınır. Çünki minimallaşma operatoru hər yerdə təyin olunubsa, onun nəticəsi zəif minimallaşma operatoru ilə eynidir. Digər tərəfdən, M_f təyin olunmadıqda M_f^z təyin olunub. Deməli ümumi rekursiv funksiyalar hər yerdə təyin olunmuş funksiyalardır. Digər tərəfdən əgər qismən rekursiv funksiyalar hər yerdə təyin olunubsa, o həm də ümumi rekursiv funksiyalardır. Əksinə, hər bir ümumi rekursiv funksiya hər yerdə təyin olunmuş qismən rekursiv funksiyalardır. Beləliklə aşağıdakı hökmü alırıq.

Teorem. Qismən funksiyanın ümumi rekursiv funksiya olması üçün zəruri və kafi şərt onun hər yerdə təyin olunmuş qismən rekursiv funksiya olmasıdır.

Beləliklə, yekun olaraq deyə bilərik ki, hər hansı bir prosesi əks etdirən funksiya ümumi rekursiv funksiya tərifini ödəyirsə onda onun hesablama alqoritmi var. Rekursiv funksiyanın tərifini isə biz ciddi riyazi şəkildə vermişik. Deməli, alqoritmin tərifi xeyli dərəcədə dəqiqləşmiş oldu.

Mühazirə 10.Kütləvi məsələ. Tacir və çanta məsələləri.

Biz yuxarıda öyrəndik ki, hər hansı bir problemi (prosesi) əks etdirən funksiya rekursiv funksiya olarsa onda həmin problemin həll alqoritmi var. Əks halda Klini-Çerç tezisinə görə o problemin həll alqoritmi yoxdur. Qeyd edək ki, müəyyən problemin həll alqoritminin olduğunu göstərmək hələ onu həll etmək demək deyil. Burada ən birinci sual olaraq həmin alqoritmin hansı müddətdə yerinə yetirilməsi çıxır. Ona görə də məsələlər siniflərə bölünür. – "asan həll olunanlar", "çətin həll olunanlar". Bu iki anlayışı dəqiqləşdirmək üçün hər şeydən əvvəl bizə lazımdır ki, məsələ nədir?, mürəkkəblik nədir?, suallarına cavab verək.

Kütləvi məsələ dedikdə, ümumiyyətlə elə bir ümumi sual başa düşülür ki, ona cavab vermək lazımdır. Bu kütləvi məsələdir. Hər bir kütləvi məsələ adətən çox sadə parametrlərdən asılı olur. Həmin parametrlərə konkret qiymət verməklə Alınan məsələyə fərdi məsələ deyilir. Biz bundan sonra kütləvi məsələni Π, fərdi məsələni isə I ilə işarə edəcəyik. Məsələni yazmaq üçün aşağıdakı formadan istifadə edəcəyik.

1. Həmin məsələnin parametrlərinin siyahısı.
2. Sualın elə şəkildə tərtib olunması ki, o sualın cavabı məsələnin həlli olsun və ya, xassələrin elə ifadə olunması ki, oradan məsələnin həlli alınsın. Bu kütləvi məsələnin yazılış formasıdır.

Sonralar müraciət edəcəyimiz aşağıdakı iki kütləvi məsələni yazaq.

1. Tacir məsələsi.

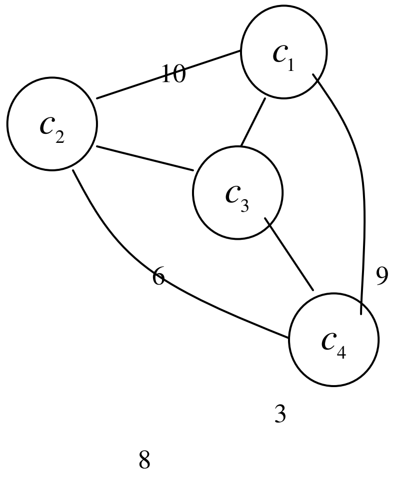
Verilmiş şəhərlər və onlar arasındakı məsafəyə görə hər hansı bir şəhərdən başlayıb hər bir şəhərdə yalnız bir dəfə olmaqla başlanğıc şəhərə qayıdan ən kiçik uzunluqlu marşrutun tapılması məsələsi tacir məsələsi adlanır.

1) n dənə $c_1, c_2, .., c_n$ şəhərləri, onların istənilən ikisi arasındakı $d(c_i, c_j)$ $i \neq j$ məsafəsi verilib.

2)Elə $P(1), P(2), ..., P(n)$ ardıcıllığı tapın ki, $\sum_{i=1}^{n-1} d(c_{p(i)}, c_{p(i+1)}) + d(c_{p(n)}, c_{p(1)})$ cəmi minimal olsun.

Qeyd edək ki, yuxarıda yazılan kütləvi tacir məsələsidir.

Bir fərdi tacir məsələsi yazaq.



Tutaq ki, c_1, c_2, c_3, c_4 şəhərləri, $d(c_1, c_2) = 10$, $d(c_1, c_3) = 5$, $d(c_1, c_4) = 9$, $d(c_2 c_3) = 6$, $d(c_2, c_4) = 8$, $d(c_3, c_4) = 3$ məsafələri verilib. Baxdığımız fərdi tacir məsələsinin həlli (c_1, c_2, c_4, c_3) marşrutudur və onun uzunluğu 26-dır.

2. Çanta məsələsi.

Turist səfərə hazırlaşarkən qiymətləri $c_1, c_2, .., c_n$ və çəkiləri uyğun olaraq $a_1, a_2, .., a_n$ vahid olan predmetlərdən elələrini öz çantasına yığmalıdır ki, onların ümumi çəkisi turistin götürə biləcəyi b çəkisindən çox olmasın, eyni zamanda ümumi maksimum qiymətə malik olsun. Bunu məsələ formasında təsvir edək.

1) $c_1, c_2, .., c_n$, $a_1, a_2, .., a_n$ və b müsbət ədədləri verilib.

2) Elə ikilik $X = (x_1, x_2, ..., x_n)$ vektoru tapmalı ki, $\sum_{j=1}^{n} a_j x_j \le b$ şərtini ödəməklə $\sum_{j=1}^{n} c_j x_j$ - maksimal qiymət alsın.

Fərdi çanta məsələsi yazaq.

Tutaq ki, $n = 7$, $c_1 = 10$, $c_2 = 8$, $c_3 = 15$, $c_4 = 5$, $c_5 = 9$, $c_6 = 15$, $c_7 = 3$, $a_1 = 4$, $a_2 = 2$, $a_3 = 3$, $a_4 = 1$, $a_5 = 5$, $a_6 = 2$, $a_7 = 3$, $b = 8$

$f = 10x_1 + 8x_2 + 15x_3 + 5x_4 + 9x_5 + 15x_6 + 3x_7 \to \max$ olsun.

$4x_1 + 2x_2 + 3x_3 + 1x_4 + 5x_5 + 2x_6 + 3x_7 \le 8$ ödənilsin.

$$x_j \in \{0,1\},\ x_j = 1 \vee 0,\quad j = \overline{1,m}.$$

Bu məsələnin optimal həlli $x = (0111010)$. $f_{\max} = 43$.

Mühazirə 11. Məsələnin öıçüsü. Polinomial və eksponensial alqoritmlər. Onların zamana və sürətə görə müqayisəsi.

Hər bir məsələni təsvir etmək üçün onun parametrlərini müəyyən "kodlaşma sxemində" yazma lazımdır. Həmin kodlaşma sxemi ümumiyyətlə yeganə qaydada təyin olunmur. Hər bir məsələ öz ölçüsü ilə xarakterizə olunur. Ölçü dedikdə hələlik məsələni xarakterizə edən əsas kəmiyyət nəzərdə tutulur. Məsələn, çanta məsələsində məchulların sayı olan n .

Seçilmiş kodlaşma sxeminə görə məsələni təsvir edərkən alınan simvolların maksimal sayına məsələnin ölçüsü deyilir.

Yuxarıda yazdığımız tacir məsələsi üçün aşağıdakı kodlaşma sxemi verək.

$K = \{c,[,],/,0,1,2,3,4,5,6,7,8,9\}$

Onda baxdığımız məsələnin giriş uzunluğu aşağıdakı yazılışdakı simvolların sayıdır.

$c[1]$, $c[2]$, $c[3]$, $c[4]$, /10, 5, 9 / 6, 8 // 3

Yəni bu məsələnin giriş uzunluğu 34-dür. Məsələnin ölçüsü dedikdə həmin məsələni təsvir edərkən lazım gələn giriş uzunluqlarının ən böyüyü götürülür. Bundan başqa bizə sonralar məsələnin "zaman mürəkkəbliyi" anlayışı da lazımdır. Zaman mürəkkəbliyi dedikdə, məsələnin ölçüsündən asılı elə funksiya başa düşülür ki, həmin funksiya məsələnin həllinə lazım gələn əməliyyatların ən pis halda maksimal sayı olsun. Qısa desək, məsələnin zaman mürəkkəbliyi həmin məsələnin həllinə sərf olunan əməliyyatların (hesabi, məntiqi və s.) maksimal sayıdır.

Əgər $\exists c > 0$ varsa ki, ixtiyari n natural ədədi üçün $|f(n)| \le c|g(n)|$ şərti ödənilirsə onda deyəcəyik ki, $f(n)$ funksiyası $g(n)$ tərtibdəndir. Yəni, $f(n) = 0(g(n))$.

Əgər $f(n)$ funksiyası müəyyən bir məsələnin həll alqoritminin zaman mürəkkəbliyini göstərirsə və həmin n ölçüsündən asılı olan elə bir $P(n)$ polinomu varsa ki, $f(n) = 0(P(n))$ şərti ödənilsin, onda deyirlər ki, baxılan alqoritm polinomial zaman mürəkkəbliyinə malikdir. Buna biz qısa olaraq, polinomial alqoritm deyəcəyik. Başqa sözlə polinomial alqoritmlərin zaman mürəkkəbliyi, yəni əməliyyatların maksimal sayı müəyyən $P(n)$ polinomundan çox deyil. Əgər zaman mürəkkəbliyini göstərən $f(n)$ funksiyası $f(n) = 0(P(n))$ şərtini ödəmirsə, həmin alqoritmə eksponensial alqoritm deyəcəyik. Polinomial zaman mürəkkəbliyinə malik alqoritmlərə uyğun məsələlər, "asan həll olunan", eksponensial zaman mürəkkəblikli alqoritmlərə uyğun məsələlər isə "çətin həll olunan" məsələlər adlanır. Asan həll olunan məsələlər sinfini P, çətin həll olunan məsələləri isə NP - tam ilə işarə edəcəyik.

Polinomial və eksponensial alqoritmi müqayisə etmək üçün aşağıdakı cədvəli qururuq.

Fərz edək ki, müəyyən maşının sürəti saniyədə 10^6 əməliyyatdır. Polinomial və exsponensial alqoritmləri zamana görə müqayisə etmək üçün aşağıdakı cədvəli dolduraq.

n / $f(n)$	10	20	30	40	50	60
n	0.00001 san	0.00002 san	0.00003 san	0.00004 san	0.00005 san	0.00006 san
n^2	0.0001 san	0.0004 san	0.0009 san	0.0016 san	0.0025 san	0.0036 san
n^3	0.001 san	0.008 san	0.027 san	0.064 san	0.125 san	0.216 san
n^5	0.1 san	3.2 san	24.3 san	1.7 dəq	5.2 dəq	13 dəq
2^n	0.001 san	1 san	17.9 san	12.7 gün	367 il	366 əsr

Çünki,

1) $\frac{2^{40}}{10^6} = \frac{(2^{10})}{10^6} = \frac{10^{12}}{10^6} = 10^6 \, san$

2) $\frac{10^6}{60} = \frac{10^5}{6} = \text{dəq.} = \frac{10^5}{6.60} = \frac{10^4}{36} \, saat = \frac{10^4}{36.24} = \frac{10000}{800} = \frac{100}{8} = 12.7 \, gün$ və s.

Cədvəldən görünür ki, polinomial zaman mürəkkəbli alqoritmlərə uyğun məsələlərin ölçüsünün artması məsələnin həll zamanının yerinə yetirilən olaraq artırır. Lakin, exsponensial zaman mürəkkəblikli alqoritmlərin tələb etdiyi zaman məsələnin ölçüsünün artması ilə eksponensial olaraq artır. Yəni, həmin məsələləri real zaman müddətində həll etmək mümkün deyil. İqtisadiyyatda və ya texnikada elə məsələlər var ki, (tacir, çanta və s.) onların həlli üçün hələlik polinomial mürəkkəbliyə malik alqoritmlər tapılmamışdır. Lakin bu məsələlərin həlli üçün effektiv riyazi üsullar tapılmamasına baxmayaraq bəlkə yeni daha sürətli kompüterlər yaradılsa bu məsələlər həll oluna bilər? Bu məqsədlə aşağıdakı cədvəli dolduraq.

Komputerlərin sürətinin artırılmasının məsələnin ölçüsünün artmasına təsiri.

	Müasir kompüter	100 dəfə sürətli kompüter	1000 dəfə sürətli kompüter
n	N	100 N	1000 N
n^2	N_1	10 N_1	31.6 N_1
$n^{3.}$	N_2	4.64 N_2	10 N_2
2^n	N_3	$N_3 + 6.64$	$N_3 + 9.87$

Mühazirə 12.Təqribi (lokal) alqoritmlər. Çanta və Tacir məsələləri üçün təqribi alqoritmlər:

Cədvəl bir daha göstərdi ki, əgər gələcəkdə indikindən 100, 1000 dəfə sürətli kompüterlər yaradılsa həmin kompüterlər vasitəsilə polinomial alqoritmlərin ölçüsü indikindən dəfələrlə çox olan məsələlər həll edər. Lakin gələcəkdə indikindən 1000 dəfə sürətli kompüter yaradılsa da onların həll etdiyi məsələnin ölçüsü müasir kompüterin həll etdiyi məsələnin ölçüsündən bir neçə vahid çox olar. Deməli, bəzi məsələlərin effektiv həll üsullarının tapılmaması və ya bu məsələlərin gələcəkdə daha sürətli kompüterlərlə həll oluna bilməməsi onu göstərir ki, qarşıya çıxan məsələlər içərisindən asanları olduğu kimi çətinləri də olmalıdır. Ona görə də məsələləri siniflərə bölmək zəruriəti meydana çıxır. Zaman mürəkkəbliyi polinomial olmayan məsələlərin optimal həllinin tapılması ciddi elmi və texniki problem olduğuna görə və eyni zamanda iqtisadiyyatın tələbatını ödəmək üçün həmin məsələlərin təqribi həll alqoritmləri işlənib hazırlanır. Belə təqribi həll alqoritmləri məsələnin mahiyyətinə uyğun olaraq hazırlanır.

Belə alqoritmlərin yaradılmasında əsas məqsəd sürətlə işləyən və optimal həldən ciddi fərqlənməyən həll verə bilən üsulların işlənməsidir. Aşağıdakı kimi çanta məsələsi yazıb onun təqribi həll alqoritmini yazin.

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n \to \max \qquad (1)$$

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \le b \qquad (2)$$

$$x_j = 1 \vee 0,\ j \equiv \overline{1,n} \qquad (3)$$

Burada ümumiliyi pozmadan qəbul edirik ki,

$$\frac{c_1}{a_1} \ge \frac{c_2}{a_2} \ge \ldots \ge \frac{c_n}{a_n}$$

şərtləri ödənir. Məsələn,

$$7x_1 + 10x_2 + 15x_3 + 6x_4 + 8x_5 + 12x_6 + 9x_7 + 3x_8 \to \max$$

$$2x_1 + 3x_2 + 4x_3 + 2x_4 + 5x_5 + 6x_6 + 3x_7 + 2x_8 \le 10$$

Bu məsələyə aşağıdakı iqtisadi interpretasiya verək:

Tutaq ki, n dənə obyektdən hər birini istifadə etmək üçün ya seçə bilərik ya da seçmərik. Əgər j-cu obyekt seçilərsə onda a_j vahid xərc çəkilməlidir və bu zaman c_j vahid gəlir əldə olunmalıdır. Bu lahiyənin həyata keçirilməsi üçün b miqdarda vəsait ayrılıb. Məsələ belə qoyulub ki, elə obyektlər seçilsin ki, onlara çəkilən ümumi xərc ayrılmış b vəsaitindən çox olmasın və bu zaman əldə olunan gəlir maksimal olsun. Bu məsələnin həll alqoritmini aşağıdakı mülahizə əsasında qururuq.

Tutaq ki, j_* nömrəli obyekti seçirik. a_{j*} qədər xərc çəkməli, c_{j*} qədər gəlir əldə olunmalıdır.

Aydındır ki, bu zaman vahid xərcə düşən gəlir $\frac{c_{j*}}{a_{j*}}$ olar.

Buradan görünür ki, elə j_* nömrəli obyektiv seçmək lazımdır ki,

$$\frac{c_{j*}}{a_{j*}} = \max_j \frac{c_j}{a_j}$$

Şərti ödənsin. Bizim məsələ də $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$ şərti ödəndiyindən məsələnin təqribi həllini aşağıdakı kimi yaza bilərik.

$$x_j = \begin{cases} 1 \quad \text{əgər} \quad \mathrm{a_j} \leq \mathrm{b} - \sum_{\mathrm{i=1}}^{\mathrm{j-1}} \mathrm{a_i} x_{\mathrm{i}} \\ 0, \text{əks halda } \mathrm{j} = 1,2,\ldots,\mathrm{n} \end{cases}$$

Misal:

$$7x_1 + 10x_2 + 15x_3 + 6x_4 + 8x_5 + 12x_6 + 9x_7 + 2x_8 \to \max$$
$$2x_1 + 3x_2 + 5x_3 + 2x_4 + 4x_5 + 7x_6 + 6x_7 + 2x_8 \leq 9$$
$$x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 1, x_5 = 0, x_6 = 0, x_7 = 0, x_8 = 1$$
$$x = (11010001)$$
$$f(x) = 25$$

(1) – (3) - də (3) – ü $0 \leq x_j \leq 1$ ilə əvəz etdikdə alınan kəsilməz məsələnin $\bar{X} = (\bar{x}_1, \ldots, \bar{x}_n)$ optimal həlli aşağıdakı düsturla tapılır.

$$\bar{x}_j = \begin{cases} 1 \quad \text{əgər} \quad a_j \le b - \sum_{i=1}^{j-1} a_i x_i \\ \left(b - \sum_{i=1}^{j-1} a_i \overline{x_i} \right) \Big/ a_j \quad \text{əgər } a_j > b - \sum_{i=1}^{j-1} a_i \overline{x_i} \quad k := j; \\ 0, \quad j = k+1, k+2, ..., n \quad j = \overline{1,n}. \end{cases}$$

Tapılan həlli yuxarıda nəzərə alsaq, optimal həlli

$$\overline{f} = \sum_{j=1}^{n} c_j \overline{x_j}$$

$$f^* \le f^{\max} \le \overline{f}$$

alırıq.

$$\overline{x} = (1,1,\frac{5}{4},0,0...0) \qquad \overline{f} = 7 + 10 + 15\frac{4}{5} = 29 \qquad f^* = 25 \qquad 25 \le f^{\max} \le 29$$

Tacir məsələsinin həlli üçün təqribi alqoritmi yazaq. $\omega = \{1,2,...,n\}$ şəhərlər çoxluğu, n - şəhərlərin sayıdır. Hər hansı $j_* \in \omega$ qeyd edək. (Adətən $j_* = 1$ qəbul edilir)

$$\min_{j \in \omega} d(c_{j*}...c_j) = d(c_{j*}...c_{j_1}), \qquad \omega := \omega \setminus \{j_1\}.$$

$$\min_{j \in \omega} d(c_{j_1}...c_j) = d(c_{j_1}...c_{j_2}), \qquad \omega := \omega \setminus \{j_2\}.$$

$$\min_{j \in \omega} d(c_{j_2}...c_j) = d(c_{j_2}...c_{j_3}), \quad ....$$

Bu proses $\omega \ne 0$ olana kimi davam edir.

Nəticədə aldığımız $(j_*, j_1, ..., j_{n-1})$ ardıcıllığı tacir məsələsinin təqribi həlli olar. Biz çanta və tacir məsələləri üçün təqribi həll alqoritmləri yazdıq. Bu alqoritmlər hər addımda ən yaxşı qərar qəbul etmək prinsipinə əsaslanmışdır. Belə alqoritmlərə <u>lokal</u> alqoritmlər də deyilir.

Mühazirə 13. Tanınma məsələsi, onun standart yazılış forması, misallar.

Məsələləri siniflərə bölmək üçün həmin məsələləri tanınma məsələləri kimi yazmaq lazımdır. Ümumiyyətlə " NP - tam "- lıq nəzəriyyəsi tanınma məsələləri üzərində qurulur. Tanınma məsələsi dedikdə isə cavabı yalnız "Hə" ya da " Yox " olan məsələlər başa düşülür. Hər bir Π tanınma məsələsi 2 çoxluqla təyin olunur.

1) Bütün mümkün olan fərdi D_Π məsələlər çoxluğu.

2) Cavabı yalnız "Hə" olan Y_Π fərdi məsələlər çoxluğu. $Y_\Pi \subset D_\Pi$

Biz bundan sonra məsələni təsvir etməyin standart formasından istifadə edəcəyik. Bu forma 2 bəndlə təyin olunur.

I. Bənd: Birinci bənddə məsələnin şərti müxtəlif komponentlər termini ilə verilməlidir. Bu bəndi Şərt: kimi işarə edəcəyik.

II. Bəndi Sual kimi işarə edəcəyik. Sual bəndində şərtin yazıldığı komponentlər termininə görə elə formada sual tərtib olunur ki, bu sual yalnız və yalnız 2 cavabdan birinə malik olmalıdır. "Hə" , ya "Yox". Hər bir fərdi məsələ D_Π çoxluğuna yalnız və yalnız o zaman daxildir ki, məsələnin standart formada təsvirində parametrlərə konkret qiymətlər qoyula bilsin. Fərdi məsələ o zaman Y_Π çoxluğuna daxildir ki, standart formada onun yazılışında cavab yalnız "Hə" olsun. Bir neçə misal göstərək.

I. Alt qrafik izomorfluğu – tanınma məsələsi.

(Təpələr kimi təsvir olunmuş nöqtələr və bu nöqtələri birləşdirən tillər adlanan əyrilərdən ibarət tərtibat qraf adlanır). Qrafı $G(V,E)$ ilə işarə edirlər. Burada V - təpələr çoxluğu, E - tillər çoxluğudur.

1). Şərt: $G_1 = G_1(V_1, E_1)$ və $G_2 = G_2(V_2, E_2)$ qrafları verilmişdir.

2). Sual: G_1 qrafı G_2 qrafına izomorf olan altqraf saxlayırmı? O zaman deyirlər ki, G_1 qrafı, G_2 qrafı ilə izomorfdur aşağıdakı 2 şərt ödənilsin.

1. $\exists\ V^1 \in V_1,\ \ E^1 \in E_1$ təpələr və tillər çoxluğu var ki,

$$|V^1| = |V_2|\ ,\ |E^1| = |E_2|$$

olsun.

2.Elə qarşılıqlı birqiymətli $f : V_2 \to V^1$ funksiyası olsun ki, $(u,v) \in E_2$ olduqda $(f(u), f(v)) \in E^1$ olsun.

II. Tacir tanıma məsələsi

1). Şərt: $c_1, c_2, .., c_n$ şəhərləri, onların istənilən ikisi arasındakı $d(c_i, c_j)$ $(i \neq j)$ məsafələri və müəyyən müsbət B ədədi verilmişdir.

2). Sual: Elə $P(1), P(2), ..., P(n)$ ardıcıllığı varmı ki,

$$\sum_{i=1}^{n-1} d(c_{p(i)}, c_{p(i+1)}) + d(c_{p(n)}, c_{p(i)}) \leq B$$

Qeyd edək ki, tacir tanınma məsələsi ilə tacir məsələsi polinomiyal ekvivalentdir. Çünki B ədədini ən azı bir vahid azalda - azalda sonlu real addımdan sonra ən kiçik maşurutu tapa bilərik.

III. Çanta- tanınma məsələsi.

1). Şərt: $c_1, c_2, .., c_n$, $a_1, a_2, .., a_n$, b və M müsbət tam ədədləri verilmişdir.

2). Sual: Elə ikilik $X = (\mathrm{x}_1, x_2, ..., x_n)$ vektoru varmı ki,

$$\begin{cases} \sum_{j=1}^{n} c_j x_j \geq M \\ \sum_{j=i}^{n} a_j x_j \leq b \end{cases}$$

şərtləri ödənsin.

Mühazirə 14. Əlifba, söz və dil. Kodlaşma sxemi. $L(\Pi, e)$ -dili

NP-tamlıq nəzəriyyəsinin tanınma məsələləri üzərində qurulması onunla əlaqədardır ki, tanınma məsələləri onları təsvir etməyin əlverişli formal ekvivalentinə malikdir.

Ümumiyyətlə hesablama nəzəriyyəsi üsullarını öyrənməyin belə formal ekvivalenti dil adlanır. $\sum$ (siqma) ilə müəyyən sonlu simvol çoxluğunu işarə edək. Bu çoxluğa əlifba deyəcəyik. $\sum$ əlifbasının simvollarından düzəlmiş bütün mümkün sonlu uzunluqlu zəncirlər çoxluğunu $\sum^*$ -lə göstərək. $\sum^*$ sözlər çoxluğunun istənilən alt çoxluğuna dil deyilir.

Məsələn: Əlifba olaraq $\sum = \{0,1\}$ götürək.

$$\sum^* = \{\phi, (0), (1), (01), (111), (011), (110)\}$$

$$L = \{(0), (011), (111)\}$$

Hər bir tanınma məsələsi ilə dil arasında uyğunluq müəyyən bir kodlaşma sxemi vasitəsilə həyata keçirilir. Hər bir Π tanınma məsələsinin e kodlaşma sxemində yazılışı həmin məsələnin sözlər çoxluğunu 3 sinfə bölür:

I. Π tanınma məsələsinin kodları olmayan sözlər çoxluğudur.

II. Π tanınma məsələsinin kodları olan elə sözlər çoxluğudur ki, onlar “yox” cavabına gətirir.

III. Π tanınma məsələsinin kodu olan elə sözlərdir ki, məsələ bu sözlərlə ifadə olunduqda “Hə” cavabı alınır.

Bu 3 sinif sözlər çoxluğundan düzələn dilə $L(\Pi, e)$ dili deyilir.

$$L(\Pi, e) = \left\{ x \in \sum^* \;\middle|\; \begin{array}{l} \sum^* \; - \text{əlifbasinin sozler coxlugudur;} \\ x \in D_\Pi;\, e,\ \Pi \text{ meselesinin kodudur; } I \in Y_\Pi \end{array} \right\}$$

Biz o zaman hər hansı Π' məsələsinin məna kəsb etdiyini deyəcəyik ki, həmin məsələ müəyyən e kodlaşma sxemində $L(\Pi', e)$ dilinin sözlərində yazıla bilsin.

Məsələni $L(\Pi', e)$ dilində yazmaq işin bir tərəfidir. Digər tərəfdən, məsələnin zaman mürəkkəbliyini müəyyənləşdirmək üçün bu məsələnin kodunun uzunluğunu göstərən hər hansı funksiya daxil etmək lazımdır. Bu funksiya hər bir Π məsələsini müsbət ədədlər çoxluğuna inikas etdirir.

Bu funksiya $Lenght:\ [I] \to z^{+}$ kimi yazılır və uzunluq funksiyası adlanır. Hər bir I fərdi məsələsi üçün yazılmış uzunluq funksiyası həmin məsələnin kodunun uzunluğuna polinomial ekvivalent olmalıdır. Burada polinomial ekvivalentlik dedikdə aşağıdakını başa düşürük.

Seçilmiş e kodlaşma sxemində yazılan hər bir I tanınma məsələsi üçün elə P və P' polinomları olmalıdır ki, I məsələsinin x kodu

$$Lenght\,[I] \le P(|x|) \text{ və}$$

$$|x| \le P'(lenght\,[I])$$

şərtlərini ödəsin. Uzunluq funksiyasına misallar göstərək.

1. Fərdi tacir məsələsinin uzunluq funksiyası aşağıdakı kimidir:

$$lenght\,[I] = n + \log_2 B + \log_2\{\max_{i,j} d(c_i, e_j)\}$$

2. Çanta tacir məsələsinin uzunluq funksiyası aşağıdakı kimidir:

$$lenght\,[I] = n + [\log_2 B] + [\log_2 M] + \left[\log_2\{\max_{j} d(c_i, a_j)\}\right].$$

Mühazirə 15. Düşünülmüş və standart kodlaşma sxemi. Düzgün qurulmuş söz.

Qeyd edək ki, hər bir fərdi tanınma məsələsini müəyyən dildə yazmaq üçün hökmən müəyyən e kodlaşma sxemi seçmək lazımdır. Hər kəsin seçdiyi kodlaşma sxeminin həmişə müəyyən üstünlükləri və çatışmazlıqları olur. Ona görə də "düşünülmüş kodlaşma sxemindən" istifadə olunur. Hər bir düşünülmüş kodlaşma sxemi hökmən aşağıdakı 2 xassəyə malik olmalıdır.

1. Yığcamlıq xassəsi.
2. Əks kodlaşmaya malik olmaq.

Əgər 1-ci xassə pozularsa, onda hər hansı çətin həll oluna bilən məsələ süni olaraq asan həll olunana çevrilər.

2-ci xassəyə görə hər bir məsələnin kodu polinomial mürəkkəbliklə ölçülən zaman müddətində açıla bilməlidir. Başqa sözlə, az bir müddətdə verilən kodda nə yazıldığını aydınlaşdırmaq mümkün olmalıdır.

Qeyd edək ki, hər bir düşünülmüş kodlaşma sxemi də hamı tərəfindən bir mənalı qarşılanmır. Bu məqsədlə standart kodlaşma sxemindən istifadə olunur.

Hər bir standart kodlaşma sxemi $\Psi = \{0, 1, -, [,], (,), , \}$ əlifbasında yazılmış kodu "düzgün qurulmuş söz"-ə çevirir. Düzgün qurulmuş söz isə rekursiv olaraq aşağıdakı kimi təyin olunur.

1) Hər bir k tam ədədinin 0 və 1 – lərdən ibarət söz kimi baxılan ikilik yazılışı həmin k tam ədədini təmsil edən düzgün qurulmuş sözdür.
2) Əgər müəyyən x sözü hər hansı k tam ədədinin düzgün qurulmuş sözüdürsə, onda ad, şərh kimi baxılan $[x]$ də düzgün qurulmuş sözdür.
3) Əgər $x_1, x_2, ..., x_n$ yazılışları müəyyən $X_1, X_2, ..., X_n$ obyektlərinin uyğun düzgün qurulmuş sözlərdirsə, onda $(x_1, x_2, ..., x_n)$ yığımı da $< X_1, X_2, ..., X_n >$ ardıcıllığının (çoxluğunun) düzgün qurulmuş sözü olar.

Beləliklə, məsələləri təsvir etmək üçün yuxarıdakı Ψ əlifbasına əsasən standart kodlaşma sxemindən istifadə olunur.

Düzgün qurulmuş sözlərə misallar göstərək.

1. Hər bir $r = \frac{p}{q}$ rasional ədədi müəyyən (x, y) cütü kimi düzgün qurulmuş sözə kodlaşır. Burada ƏBOB (p, q) = ± 1. x - p tam ədədinin, y - q tam ədədinin düzgün qurulmuş sözüdür. Məsələn:

$$\frac{3}{5} = (011, 101)$$

2. Sonlu f funksiyası $(U_1, U_2, ..., U_n) \to V$, $((x_1, y_1), (x_2, y_2), ..., (x_n, y_n))$ şəklində düzgün qurulmuş sözə kodlaşır.

Burada x_i $(i = \overline{1, n})$, $U_i (i = \overline{1, n})$ obyektlərinin, y_i isə $f(U_i) \in V$ obyektlərinin düzgün qurulmuş sözüdür. Məsələn:

x	1	2	3
y	3	-2	4

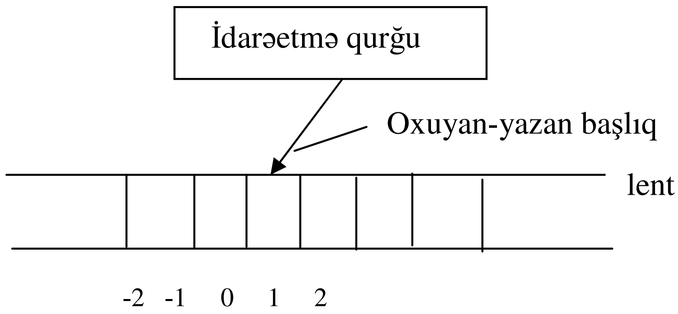
((001, 011), (010, - 010), (011,100))

3. Hər bir $G = G(V, E)$ qrafı müəyyən (x, y) cütü kimi düzgün qurulmuş sözə kodlaşdırılır. Burada x - V təpələr çoxluğunun, y isə E tillər çoxluğunun düzgün qurulmuş sözüdür. Aydındır ki, y cütlər şəklindədir.

Mühazirə 16. Birlentli Determinik Turing maşını (D. T. M.), onun işsxemi, misal

Biz məsələni düzgün qurulmuş söz şəklində standart formada yaza bilərik. Həmin məsələnin həllini tapmaq üçün aydındır ki, bizə müəyyən hesablama aparatının modeli lazımdır. Belə nəzəri model olaraq birlentli D. T. M.-i öyrənək. Determinik Turing maşını əsas 3 elementdən ibarətdir.

1. İdarəetmə qurğusu
2. Oxuyan yazan başlıq
3. Lent



İdarəetmə qurğusu – sonlu vəziyyətlər çoxluğuna malik olmalıdır.

Oxuyan yazan başlıq – üzərində dayandığı xanadakı simvolu oxuya bilər, yaxud onu silib yerinə yenisini də yaza bilər.

Lent – hər iki tərəfə sonsuz uzanmaqla tam ədədlərlə nömrələnmiş xanalardan ibarətdir və hər bir xanaya yalnız bir simvol ($0,1,b$) yazıla bilər.

D. T. M. özünün M proqramı vasitəsilə işləyir. (Bundan sonra biz M D. T. M. proqramı deyəcəyik.) Hər bir belə M proqramı 3 hissədən ibarətdir.

1. Sonlu Γ simvollar çoxluğu. Buna Lent əlifbası da deyirlər. $\Sigma(\Sigma \subset \Gamma)$ giriş simvolları çoxluğu və $b(b \subset \Gamma \setminus \Sigma)$ boş simvollar.

2. Sonlu Q vəziyyətlər çoxluğu. Burada başlanğıc q_0 vəziyyəti və yekun q_y, q_N vəziyyətləri ayrıca qeyd olunur.

3. δ keçid funksiyası $\delta : (Q \setminus \{q_y, q_N\}) \text{ x } \Gamma \to Q \text{ x } \Gamma \text{ x } \{-1:1\}$

Məsələn: $\delta(q, S) = (q', s-1)$.

Aydındır ki, δ funksiyası iki yerli, üç qiymətli funksiyadır. M proqramı verildikdə hər bir D. T. M. aşağıdakı qaydada işləyir.

Əvvəlcə baxılan məsələnin x sözü lentin $1,2,...,|x|$ xanalarına hər birinə bir simvol olmaqla yazılır. Bu zaman maşının vəziyyəti başlanğıc q_0 vəziyyətində olur, oxuyan-yazan başlıq isə bir nömrəli xananın üzərində dayanır.

D. T. M.-in iş sxeminin bir addımını izah edək.

Tutaq ki, maşın $q \in Q$ müəyyən vəziyyətindədir. Əgər $q = q_y$ və ya $q = q_N$ olarsa, maşının işi dayanır. Birinci "Hə" cavabı ilə, ikinci "Yox" cavabı ilə. Əks halda, yəni cari q vəziyyəti üçün $q \in Q \setminus \{q_y, q_N\}$. Bu zaman oxuyan-yazan başlıq üzərində dayandığı xanadakı S simvolunu oxuyur. Beləliklə, δ keçid funksiyasının q və S arqumentləri məlum olur. Həmin funksiyanın quruluşuna görə tutaq ki,

$$\delta(q, S) = (q', S', \Delta)$$

Burada $\Delta = -1$ və ya $\Delta = 1$ olmalıdır. Bu zaman cari vəziyyət q' olur, oxuyan-yazan başlıq üzərində dayandığı xanadakı simvolu silib onun yerinə S' simvolu yazır və $\Delta = -1$ olsa bir xana sola, $\Delta = 1$ bir xana sağa çəkilir. Yenə də əgər $q' = q_y$ və ya $q' = q_N$ olsa hesablama prosesi dayanır. Əks halda proses yuxarıdakı qaydada davam edir.

D. T. M.-də həll etmək üçün aşağıdakı misala baxaq:

(Verilmiş natural ədədin dördə bölünməsi məsələsi).

1. Şərt: N natural ədədinin x ikilik yazılışı verilmişdir.

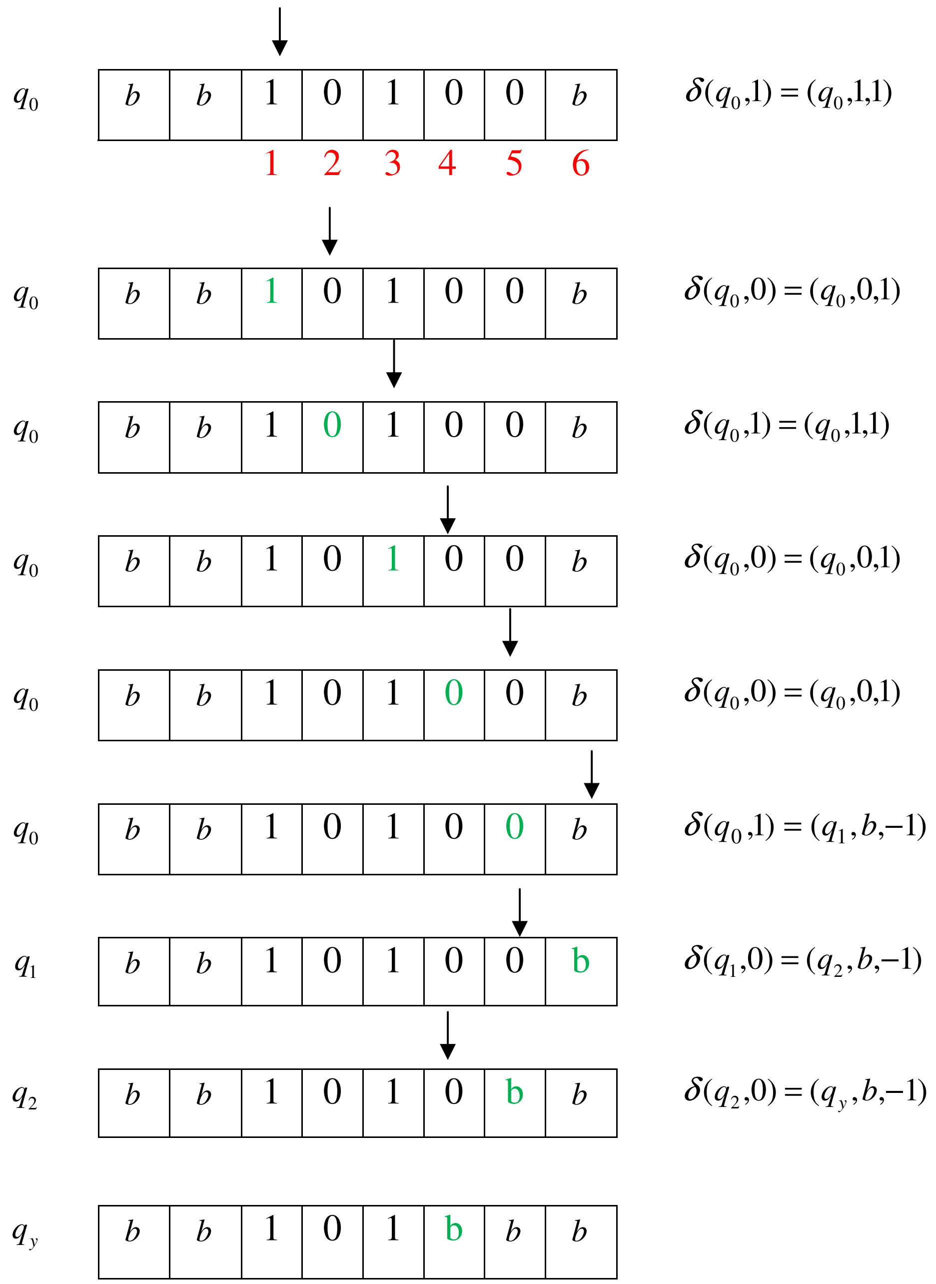
2. Sual: x ədədinin sonuncu 2 rəqəmi sıfırdırmı?

(Ədədin ikilik yazılışında son iki rəqəmi sıfıra bərabər olarsa həmin ədəd dördə bölünür və tərsinə).

$\Gamma = \{0,1,b\}$, $\Sigma = \{0,1\}$, $x = (10100)$

$Q = \{q_0, q_1, q_2, q_3, q_y, q_N\}$ $\delta(q, S)$ funksiyasını aşağıdakı cədvəllə təyin edək:

	0	1	b
q_0	$(q_0,0,1)$	$(q_0,1,1)$	$(q_1,b,-1)$
q_1	$(q_2,b,-1)$	$(q_3,b,-1)$	$(q_N,b,-1)$
q_2	$(q_y,b,-1)$	$(q_N,b,-1)$	$(q_N,b,-1)$
q_3	$(q_N,b,-1)$	$(q_N,b,-1)$	$(q_N,b,-1)$



Beləliklə, yazdığımız M proqramması 8 addımdan sonra "Hə" cavabında dayandı.

Mühazirə 17. L_M -dili, $T_M(n)$ -zaman mürəkkəbliyi və P -sinfi.

Əgər $\sum$ simvollarından düzəlmiş x sözünü M D.T.M proqrammasının girişinə verdikdə hesablama prosesi q_y yəni "Hə" cavabında başa çatarsa onda deyirlər ki, M D.T.M proqrqmması x sözünü qəbul edir. M D.T.M proqrammasının qəbul etdiyi x sözlərindən düzəlmiş dilə L_M dili deyilir. Yəni,

$$L_M = \left\{ x \in \Sigma \;\middle|\; \begin{array}{l} \text{M.D.T.M.} - \text{proqramı } x \text{ sűzunu;} \\ \text{qəbul edir.} \end{array} \right\}$$

Verilmiş x sözünü D.T.M-in girişinə verdikdə onun dayandığı vəziyyətə qədər olan addımların sayına M D.T.M proqramının x sözü üçün tələb etdiyi zaman deyilir. M D.T.M proqrammasının zaman mürəkkəbliyi isə $T_M(n)$ ilə işarə olunur və

$$T_M(n) = \max\left\{ m \;\middle|\; \begin{array}{l} \exists x \in \Sigma \text{ sozu var ki, } |x| = n \text{ və} \\ \text{M.D.T.M. proqraminin onu qəbul} \\ \text{etməyə tələb etdiyi zaman } m - \text{dir.} \end{array} \right\}$$

Əgər $\forall_n$ natural ədədi üçün $\exists_{p(n)}$ polinomu varsa ki, $T_M(n) \le p(n)$ şərti ödənilsin. Onda deyirlər ki, baxılan M D.T.M proqramı polinomiyal zaman mürəkkəbliyinə malikdir.

Beləliklə biz vacib sinif olan P sinifinə tərif verə bilərik.

$$P = \left\{ L \;\middle|\; \begin{array}{l} \text{polinomial zaman murəkkəbliyi} \\ \text{elə M.D.T.M. proqramı var ki, onun} \\ \text{tanıtanı } L_M \text{ dilleri ğzğn } L = L_M \end{array} \right\}$$

Göründüyü kimi P dillər sinfi dedikdə polinomyial zaman müddətində "Hə" cavabına gətirən sözlərdən düzəlmiş dil başa düşülür. Biz o zaman deyəcəyik ki, müəyyən kodlaşma sxemində yazılmış Π məsələsi P sinifinə daxildir., onun yazıldığı $L(\Pi, e)$ dili üçün $L(\Pi, e) \in P$ olsun.

Mühazirə 18. Qeyri-Determinik alqoritm.
Qeyri-Determinik Turinq Maşını.

İkinci vacib sinif olan NP sinfinə tərif vermək üçün qeyri- determinik hesablama və qeyri-determinik alqoritm anlayışından istifadə etməliyik. Hər bir qeyri-determinik hesablama iki mərhələdən ibarətdir. 1-ci mərhələdə baxılan məsələnin həlli ola bilən müəyyən struktura intuitiv olaraq təxmin olunur. 2-ci mərhələdə isə təxmin olunmuş strukturanın doğrudan da baxılan məsələnin həlli olub-olmadığı yoxlanılır. Əgər yoxlama mərhələsinin cavabı "Hə" olarsa, onda həll prosesi başa çatır, əks halda 1-ci mərhələyə qayıdılır və başqa bir struktura təxmin olunur. Aydındır ki, bu proses ən pis halda bütün variantlara baxıldıqdan sonra başa çata bilər. Qeyri-determinik hesablamanı yerinə yetirən alqoritm qeyri-determinik alqoritm adlanır. Qeyri-determinik alqoritmin iş sxemi tacir məsələsi üzərində izah edək. N şəhərləri və onların ixtiyari 2-si arasındakı məsafələri və B müsbət ədədi verilmişdir. Uzunluğu 1-ci mərhələdə hər hansı şəhərlər ardıcıllığı təxmin olunur. 2-ci mərhələdə təxmin olunmuş strukturun doğrudan da B-dən böyük olub-olmaması yoxlanılır. Əgər yox mərhələsində bu ardıcıllığın uzunluğu doğrudan da B –dən böyük olmazsa, onda həll prosesi başa çatır. Əks halda 1-ci mərhələyə qayıdılır və başqa 1 ardıcıllıq təxmin olunur. Bu proses bütün variantlara baxıldıqdan sonra başa çata bilər. Qeyri-determinik alqoritmdə hər hansı I fərdi məsələsi o zaman həll olunur ki, həmin məsələ üçün müəyyən S strukturunu tapıb (I,S) cütlüyünü alqoritmin girişinə verdikdə "Hə" cavabı alınsın. Başqa sözlə desək, biz o zaman deyəcəyik ki, $I \in D_n$ fərdi məsələsi qeyri-determinik alqoritm vasitəsilə həll olunur. Aşağıdakı 2 şərt ödənilsin

1. Əgər $I \in Y_n$ olarsa, onda elə bir S strukturası var ki, onu təxmin edib (I,S) cütünü qeyri-determinik alqoritmə verdikdə onu yoxlama mərhələsi "Hə" cavabında qurtarsın.

2. Əgər $I \notin Y_n$ deyilsə, onda elə S strukturası yoxdur ki, onu təxmin edib (I,S) cütünü qeyri-determinik alqoritmə verdikdə "Hə" cavabı alınsın.

Qeyri-determinik alqoritmin iki vacib xüsusiyyətini qeyd etmək lazımdır.

1. Qeyri-determinik alqoritmin yoxlama mərhələsi determinik, yəni polinomial müddətdə baş verir. Başqa sözlə desək, biz dedikdə ki, qeyri-determinik alqoritm polinomial zaman mürəkkəbliyinə malikdir, onu başa düşürük ki, hər hansı $I \subset D_n$ fərdi məsələsi üçün yoxlama mərhələsinin zaman mürəkkəbliyi $P\,(length\,[i])$ polinomu ilə məhdud olsun. Digər tərəfdən məsələnin həlli ola biləcək S strukturasını, birinci mərhələdə necə tapılması, bizi maraqlandırmır. Beləliklə, biz polinomial zaman mürəkkəblikli qeyri-determinik alqoritm dedikdə yalnız yoxlama mərhələsinin polinomial müddətdə aparılmasını başa düşəcəyik, Yəni baxılan məsələnin polinomial müddətdə həll olunması burada hökm olunmur.

İkinci vacib xassəsi isə aşağıdakılardan ibarətdir. 2 məsələyə baxaq.

I məsələ

1 şərt. Müəyyən I məsələsi verilmişdir.

2 sual. Doğrudurmudur ki, I məsələsi üçün x strukturası ödənilir?

II məsələ.

1 şərt. I fərdi məsələsi verilmişdir.

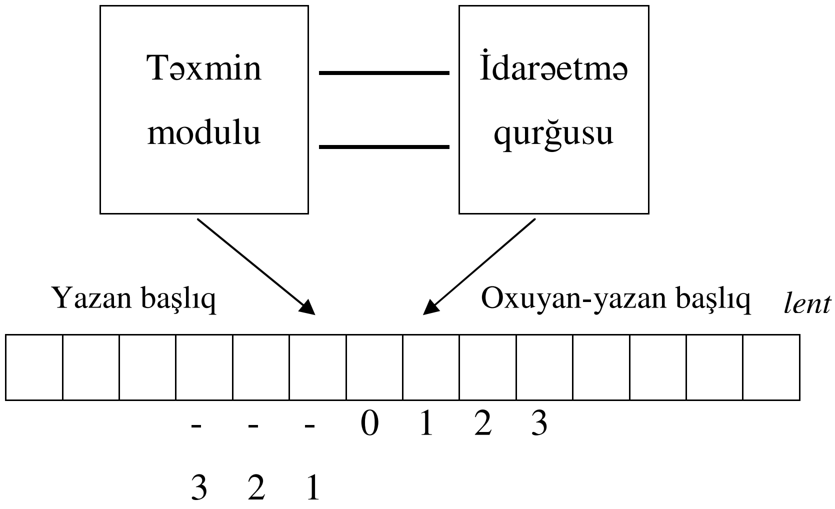
2 sual. Doğrudurmudur ki, I məsələsi üçün x strukturası ödənilirmi?

Birinci məsələyə düz, ikinciyə isə tərs məsələ deyilir.

Determinik alqoritm ilə düz və tərs məsələlər eyni zaman mürəkkəbliklə həll edilir. Sadəcə olaraq ikinci məsələdə q_y əvəzinə q_N yazılmalıdır. Lakin qeyri-determinik alqoritm düz məsələni polinomial müddətdə həll edə bilsə də, tərs məsələni yalnız bütün variantlara baxdıqdan sonra həll edə bilər.

Qeyri-determinik alqoritmin formal nəzəri ekvivalenti qeyri-determinik turinq maşınıdır. Bunun əsas hissələri aşağıdakılardır.

1. Təxmin modulu
2. Təxmin modulunun yazan başlığı.
3. İdarəetmə qurğusu.
4. İdarəetmə qurğusunu oxuyan-yazan başlığı.
5. Lent.



Q. D. T. M. özünün xüsusi M Q.D.T.M proqramı ilə işləyir. Hər bir belə proqramın 3 əsas elementi var.

1. Sonlu Γ lent əlifbası adlanan simvollar çoxluğu, Σ giriş simvollar çoxluğu, $b \in \Gamma \setminus \sum$ boş simvollar çoxluğu.

2. Sonlu Q vəziyyətlər çoxluğu. Burada başlanğıc q_0, və yekun q_y, q_N vəziyyətləri qeyd olunur.

3. $\delta : (Q \setminus \{q_y, q_n\}) \times \Gamma \to Q \times \Gamma \times \{-1,1\}$-keçid funksiyası.

Q. D.T.M. aşağıdakı qaydada işləyir. Əvvəlcə x sözü lentin $1,2,...,x-ci$ xanalarına hər birinə yalnız bir simvol olmaqla yazılır. Bu zaman maşın q_0 vəziyyətində olur. Təxmin modulunun yazan başlığı -1 nömrəli xananın, idarəetmə qurğusunun oxuyan yazan başlığı isə 1 nömrəli xananın üzərində dayanır. Maşın işə başlarkən idarəetmə qurğusu passiv vəziyyətdə olur. Təxmin modulu Γ əlifbasındakı istənilən simvolu öz başlığına verir. Bu iş tamamilə sərbəst baş verir. Yazan başlığın vəzifəsi ona verilmiş simvolu üzərində dayandığı xanaya yazmaqdır və ya yazmamaqdır. Əgər bu başlıq xanaya simvol yazırsa bir xana sola çəkilir. Bu zaman təxmin modulu yeni simvolu ona sərbəst olaraq verir. Bu proses o vaxta qədər davam edir ki, yazan başlıq ona verilmiş simvolu xanaya yazmır. Bu andan etibarən təxmin modulu passiv vəziyyətə keçir və idarəetmə qurğusu işə başlayır. Bundan sonrakı hesablamalar tamamilə determinik alqoritmdə olan kimidir. Əgər cari vəziyyət q_y olarsa onda bütün hesablama prosesi başa çatır. Əks halda yəni cari vəziyyət q_N olarsa onda idarəetmə qurğusu passiv vəziyyətə keçir, təxmin modulu yenidən işə başlayır. Bununla da qeyri-determinik turinq maşınının bir addımı başa çatır.

Mühazirə 19. Qeyri determinik alqoritm üçün L_M -dili, $T_M(n)$ -zaman mürəkkəbliyi və NP -sinfi.

Göründüyü kimi bir x sözünü girişə verdikdə buna uyğun çoxlu sayda hesablamalar mümkündür. Əgər x sözünü girişə verdikdə aparılmış hesablamalardan heç olmazsa biri q_y vəziyyətinə gətirib çıxarırsa buna “qəbul olunan hesablama” deyilir. Biz o zaman deyəcəyik ki, M Q.D.T.M. proqramı x sözünü tanıyır, həmin sözü girişə verdikdə aparılan hesablamalardan heç olmazsa biri q_y vəziyyətində qurtarsın. Yəni, bu hesablamalardan heç olmazsa biri qəbul olunan olsun. M Q.D.T.M. proqramının tanıdığı L_M dili isə həmin proqramın tanıdığı sözlərdən təşkil olunub. Yəni,

$$L_M = \left\{ x \in \sum \middle| \begin{array}{l} \text{M D.T.M proqramı} \\ \text{x - i taniyir} \end{array} \right\}$$

Hər hansı x sözünü girişə verdikdə M Q.D.T.M. proqramının bu sözü tanımağa lazım gələn zaman, qəbul olunan hesablamalardakı addımlardan ən kiçiyidir. M Q.D.T.M. proqramının $T_M(n): z^+ \to z^+$ zaman mürəkkəbliyi isə aşağıdakı kimi təyin olunur.

$$T_M(n) = \max\left\{ 1; m \middle| \begin{array}{l} \text{uzunlugu n olan elə x } (|\text{x}| = n) \text{sozu} \\ \text{var ki, M Q.D.T.M. proqraminin bu} \\ \text{sozu tanimaga sərf etdiyi zaman } m \text{ - dir.} \end{array} \right\}.$$

Burada 1 ona görə yazılır ki, qeyri-determinik alqoritmdə ola bilsin ki, idarə etmə qurğusu heç işləməsin.

Əgər $\forall n > 0$ tam ədədi üçün elə $p(n)$ polinomu varsa ki, $T_M(n) \le p(n)$ şərti ödənilsin, onda deyirlər ki, M Q.D.T.M proqramı polinomiyal zaman mürəkkəbliyinə malikdir. Bir daha qeyd edək ki, burada polinomiyal müddətdə işləmə $T_M(n)$ ifadəsindən göründüyü kimi yalnız yoxlama mərhələsinin polinomiyal müddətdə yerinə yetirilməsidir. Beləliklə, biz 2-ci vacib sinif olan NP dillər sinfinə tərif verə bilərik.

$$NP = \left\{ L \;\middle|\; \begin{array}{l} \text{polinomial zaman mürəkkəbliyinə} \\ \text{malik elə } M \ Q.D.T.M. \text{ proqramı var ki,} \\ \text{onun tanidigi } L_M \text{ dili ucun } L = L_M \end{array} \right\}.$$

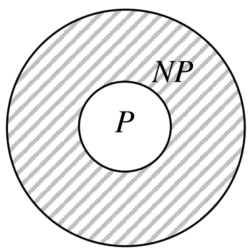
Əgər müəyyən $\Pi \in D_n$ tanınma məsələsinin müəyyən e kodlaşma sxemində yazıldığı $L(\Pi, e)$ dili üçün $L(\Pi, e) \subset NP$ olarsa, onda deyəcəyik ki, $\Pi \in NP$, yəni Π məsələsi NP sinfindədir.

Mühazirə 20. P və NP sinifləri arasında əlaqə.
Dillərin və məsələlərin bir-birinə polinomial çevrilməsi.

Hər şeydən əvvəl isbat edək ki:

Lemma: $P \subseteq NP$ yəni determinik alqoritm vasitəsilə polinomial müddətdə həll olunan məsələlər qeyri-determinik alqoritm vasitəsilə də polinomial müddətdə həll oluna bilər.

İsbatı: Göstərək ki, $\Pi \in P$ tanınma məsələsi üçün $\Pi \in NP$ olar. Onu polinomial müddətdə həll edən alqoritmi A ilə işarə edək. Həmin alqoritmi Q. D.T.M.-in yoxlama məsələsinə qoyub girişə baxılan Π məsələsini versək, bu məsələ orada həll olunar. Lemma isbat olundu. P və NP siniflərini aşağıdakı bu əlaqəni aşağıdakı dairədə göstərə bilərik.



Bu vaxta qədər $P = NP$ və ya $P \neq NP$ münasibətlərinin heç biri isbat olunmayıb. Başqa sözlə $NP \setminus P = \varnothing$ və ya $NP \setminus P \neq \varnothing$ münasibətləri isbat olunmayıb. Ancaq ehtimal edirlər ki, $NP \setminus P$ boş deyil. Aşağıdakı teoremi isbat edək:

<u>Teorem</u>: Əgər hər hansı $P \in NP$ olarsa, onda elə $p(n)$ polinomu var ki, Π məsələsi zaman mürəkkəbliyi $0(2^{P(n)})$ olan determinik alqoritmlə həll oluna bilər.

İsbatı: Tutaq ki, Π məsələsi NP sinfindədir. Həmin məsələni həll edən qeyri-determinik alqoritmi A ilə işarə edək. Həmin alqoritmin həyata keçirilməsini təmin edən polinom $q(n)$ olsun. Tutaq ki, Γ giriş əlifbasının simvollarının sayı $|(\Gamma)| = k$.

Onda qeyri-determinik alqoritmə verilmiş hər bir giriş, $q(n)$ addım müddətində yoxlanılır. A alqoritmi qeyri-determinik olduğundan bütün girişə veriləcək sözlərin maksimal sayı $k^{q(n)}$ olacaq. Bu sözlərin hər biri yoxlama mərhələsindən keçə bildiyindən buradakı əməliyyatların sayı $q(n)\,k^{q(n)}$ olur. Qeyd edək ki, müəyyən kodlaşma sxemi götürməklə (məsələn 0 və 1-lərdən ibarət) $k=2$ qəbul etmək olar. $q(n)$ əvəzinə isə elə $p(n)$ polinomu götürmək olar ki, bu hər bir $q(n)$-i əvəz etsin. Onda $q(n)\,k^{q(n)}$ əvəzinə $0(2^{P(n)})$ yaza bilərik.

Teorem isbat olundu.

Hələlik $P=NP$ və ya $P\neq NP$ münasibətlərinin heç biri isbat olunmadığına görə $NP\setminus P$ çoxluğunun boş olması və yaxud onun boş olmaması isbat olunmalıdır. Əgər qəbul etsək ki, $NP\setminus P\neq$ onda P sinfinə daxil olan h$\varnothing$ bir məsələ asan həll olunan, $NP\setminus P$ sinfinə daxil olan məsələlər isə çətin həll olunan məsələlər olar. Belə hökmlər isbat olunmadığından daha sadə teoremlər isbat etmək lazımdır. Yəni, $P\neq NP$ olarsa, onda elə Π məsələsi var ki, $\Pi\in NP\setminus P$. Bu tipli teoremlərin isbat olunması üçün dillərin və ya məsələlərin birinin digərinə polinomial çevrilməsi anlayışından istifadə olunur.

Tərif: Əgər elə $f:\Sigma_1^*\to\Sigma_2^*$ funksiyası varsa ki, o aşağıdakı 2 şərti ödəsin, onda deyəcəyik ki, L_1 dili $(L_1\subseteq\Sigma_1^*)$, L_2 dilinə $(L_2\subseteq\Sigma_2^*)$ polinomial çevrilir.

1. f funksiyasını polinomial zaman mürəkkəblikli M D.T.M. proqramı vasitəsilə hesablamaq olar. Yəni elə M D.T.M. var ki, o polinomial zaman mürəkkəbliyinə malikdir və f funksiyasını hesablaya bilir.

2. $\forall x\in\Sigma_1^*$ sözü üçün x sözü yalnız və yalnız o zaman L_1-ə daxil olsun ki, $f(x)$ sözü L_2-yə daxil olsun. Yəni $\forall x\in\Sigma_1^*$ üçün $x\in L_1\Leftrightarrow f(x)\in L_2$.

L_1 dilinin L_2 dilinə polinomial çevrilməsini $L_1\infty L_2$ kimi işarə edirlər. Əgər $\Pi_1\in D_{\Pi_1}$, və $\Pi_2\in D_{\Pi_2}$ məsələləri üçün $L_1(\Pi_1,e_1)\infty L_2(\Pi_2,e_2)$ olarsa, onda deyirlər ki, Π_1 məsələsi Π_2 məsələsinə polinomial çevrilir, yəni $\Pi_1\infty\Pi_2$. Lakin burada e_1,e_2 kodlaşma sxemləri qeyri-müəyyənlik yaratdığına görə bir məsələnin digərinə polinomial çevrilməsinə aşağıdakı kimi tərif verək.

Əgər elə $f : D_{\Pi_1} \rightarrow D_{\Pi_2}$ funksiyası varsa ki, o, aşağıdakı iki şərti ödəsin, onda deyəcəyik ki, Π_1 məsələsi Π_2 məsələsinə polinomial çevrilir və bunu $\Pi_1 \infty \Pi_2$ kimi işarə edəcəyik.

1. f funksiyasını hesablaya bilən polinomial zaman mürəkkəblikli determinik alqoritm var.

2. $\forall I \in \mathrm{D}_{\Pi}$ fərdi tanınma məsələsi üçün $I \in \mathrm{Y}_{\Pi_1} \Leftrightarrow f(I) \in \mathrm{Y}_{\Pi_2}$.

Yəni hər bir $I \in \mathrm{D}_{\Pi}$ məsələsi üçün I məsələsi yalnız və yalnız o zaman Y_{Π_1}-ə daxil olar ki, $f(I)$ məsələsi Y_{Π_2}-yə daxil olsun.

Mühazirə 21. Polinomial çevrilməyə aid əsas lemmalar. Hamilton dövrü məsələsinin Tacir məsələsinə polinomial çevrilməsi.

Lemma 1: Tutaq ki, L_1 dili L_2 dilinə polinomial çevrilir və L_2 dili P sinfindəndir. Onda L_1 dili də P sinfində olar. Yəni $(L_1 \infty L_2) \wedge (L_2 \in P) \Rightarrow L_1 \in P$.

Lemma 2: $(L_1 \infty L_2) \wedge (L_1 \notin P) \Rightarrow L_2 \notin P$. Yəni, əgər L_1 dili L_2 dilinə çevrilir və L_1 dili P sinfində deyilsə, onda L_2 dili də P sinfində deyil.

Lemma 3: Əgər Π_1 məsələsi Π_2 məsələsinə polinomial çevrilərsə və Π_2 məsələsi P sinfindədirsə onda Π_1 məsələsi də P sinfindədir. Yəni $(\Pi_1 \infty \Pi_2) \wedge (\Pi_2 \in P) \Rightarrow \Pi_1 \in P$.

Lemma 4. $(\Pi_1 \infty \Pi_2) \wedge (\Pi_1 \notin P) \Rightarrow \Pi_2 \notin P$. Yəni əgər Π_1 məsələsi Π_2 məsələsinə polinomial çevrilir və Π_1 məsələsi P sinfində deyilsə, onda Π_2 məsələsi də P sinfində deyil.

Lemma 1-i isbat edək. $L_1 \infty L_2$ olduğundan L_1 və L_2 dillərinin əlifbasını uyğun olaraq Σ_1^*, Σ_2^* ilə işarə edək və bu çevrilməni həyata keçirən funksiya f olsun. M_f isə f funksiyasını hesablayan D. T. M.-in proqramı olsun. M_2 ilə L_2 dilini tanıyan D. T. M. proqramını işarə edək. Hər hansı ixtiyari $x \in \Sigma_1^*$ sözünü götürək. Bu sözə f funksiyasını tətbiq etməklə $f(x) \in \Sigma_2^*$ alarıq. $f(x)$ sözünə M_2 proqramını tətbiq etsək alarıq ki, $f(x) \in L_2$.

Polinomial çevrilmənin tərifinə görə $f(x) \in L_2 \Rightarrow x \in L_1$ olar. Çünki, $f(x) \in L_2$ P sinfindədir.

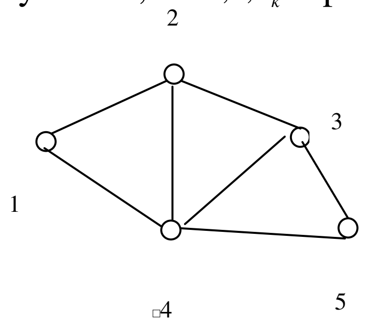
Aydındır ki, M_f və M_2 polinomial mürəkkəbliyə malik olduğundan bunların kombinasiyası da polinomial olar. Daha doğrusu bunların kombinasiyasından düzələn proqram polinomial olar.

Onun mürəkkəbliyi isə $O(P_f(|x|) + P_2(f_f(|x|))$ tərtibdən olar.

Burada P_f və P_2 uyğun olaraq Σ_1^*, Σ_2^* olan sözlərin hesablanması üçün polinomlardır.

Lemma 1 isbat olundu.

Polinomial çevrilməyə bir misal göstərək. Tutaq ki, sonlu rabitəli $G(V,E)$ qrafı verilmişdir. Əgər elə $v_1, v_2,.., v_k \in V$ təpələri varsa ki, $\{v_i, v_{i+1}\} \in E$, $1 = \overline{1, k-1}$ və $\{v_k, v_1\} \in E$ olsun, onda deyirlər ki, $v_1, v_2, .., v_k$ təpələri sadə dövr əmələ gətirir.



Əgər sadə dövr qrafın bütün təpələrindən yalnız bir dəfə keçərsə onda buna Hamilton dövrü deyilir, uyğun qrafa isə Hamilton qraf deyilir. Qrafda Hamilton dövrünün tapılması məsələsinə Hamilton məsələsi deyilir.

Göstərək ki, hamilton dövrü məsələsi Tacir məsələsinə polinomial çevrilir.Yəni, $HD \infty TM$.

Bunun üçün Polinomial çevrilməni həyata keçirən funksiyanı f ilə işarə edək və quraq. Həmin f funksiyası G qrafının V təpələr sayından n-i şəhərlərin n sayına bərabər olur. Bundan əlavə f -funksiyası hər bir $d(c_i, c_j)$ şəhərlər arasındakı məsafələri aşağıdakı kimi təyin edir:

$$d(c_i, c_{i+1}) = \begin{Bmatrix} 1, \text{əgər } (v_i, v_{i+1}) \in \text{E} \\ 2, \text{əgər } (v_i, v_{i+1}) \in \text{E} \end{Bmatrix}$$

Aydındır ki, belə f funksiyası polinomial hesablanır. Doğrudan da, ən çoxu $\frac{n(n-1)}{2}$ sayda yoxlama keçirilir. Polinomial çevrilmənin ikinci bəndini göstərmək üçün

$$\sum_{i=1}^{n-1} d(c_i, c_{i+1}) + d(c_n, c_1) \le n \qquad (1)$$

$v_1, v_2,.., v_n$ Hamilton dövrü olduğundan uyğun $c_1, c_2,.., c_n$ şəhərləri dövr təşkil edər. f funksiyası xassəsinə görə hər bir $d(c_i, c_{i+1}) = 1$ olar. (1) münasibətinin sol tərəfində n dənə toplanan var və hər biri 1-ə bərabərdir.

$HD \propto TM$ münasibətinin doğruluğunu tamamilə isbat etmək üçün göstərək ki, (1) münasibəti ödənilirsə, onda $v_1, v_2, .., v_n$ Hamilton dövrü təşkil edir. (1) münasibətinin sol tərəfində n dənə toplanan var və onların cəmi n-ə bərabərdir. Hər birisi ya 1 ya da 2 ola bilər. Onlardan hər hansı biri 2-yə bərabər olsa, başqa biri toplanan 0 olmalıdır. Bu isə f funksiyasının şərtinə ziddir. f funksiyasının təyininə görə şəhərlərlə tillər arasında qarşılıqlı birqiymətli uyğunluq var, yəni V tilləri Hamilton dövrü təşkil edir.

Mühazirə 22.Tranzitivlik lemması.

$NP-$ tam sinif və bura daxil olma kriteriyası.

Aşağıdakı tranzitivlik lemmalarını isbat edək.

Lemma 5: Əgər $L_1 \infty L_2$ və $L_2 \infty L_3$ olarsa, onda $L_1 \infty L_3$ olar.

Lemma 6: $\Pi_1 \infty \Pi_2$ və $\Pi_2 \infty \Pi_3$ olarsa, onda $\Pi_1 \infty \Pi_3$ olar.

Lemma 5-nın isbatı: Tutaq ki, Σ_1^*, Σ_2^* və Σ_3^* uyğun olaraq L_1, L_2, L_3 dillərinin sözlər çoxluğudur. Bundan əlavə fərz edək ki, $f_1(x)$ funksiyası $L_1 \infty L_2$ çevrilməsini həyata keçirən funksiyadır. $f_2(x)$ isə $L_2 \infty L_3$ çevirməsini həyata keçirən funksiyadır. Aşağıdakı kimi funksiyaya baxaq:

$$f(x) = f_2(f_1(x)).$$

Lemmanın şərtinə görə $\forall x \in \Sigma_1^*$ üçün $f_1(x) \in L_2$ olar. Digər tərəfdən $L_2 \infty L_3$ çevirməsini həyata keçirən f_2 olduğundan $f_2(f_1(x)) \in L_3$. Yəni $f(x) \in L_3$ olar. $f(x) = f_2(f_1(x))$ funksiyasının polinomial müddətdə hesablanması isə f_1-in və f_2-nin polinomial hesablanmasından alınır. Çünki, P_1 və P_2 uyğun olaraq f_1 və f_2-nin polinomları olsa, onda $f(x)$-i məhdud edən funksiya

$$O(P(|x|)) + P_2(|x|))$$

olar. Lemma 6 analoji olaraq, isbat olunur. Tranzitivlik lemması məsələləri çətinliyə görə nizamlamağa imkan verir. Belə ki, $L_1 \infty L_2 \infty ... \infty L_k \infty L_{k+1} \infty ... \infty L_n$ münasibətindən və lemma 1-dən alırıq ki, əgər $L_k \in P$ olarsa, onda $L_1, L_2, ..., L_{k-1} \in P$ olar. Əksinə əgər L_k çətin həll olunan məsələ olsa, onda $L_{k+1}, ..., L_n$ də çətin həll olunur. Beləliklə, biz III çox mühüm olan NP-tam sinfinə tərif verə bilərik. Əvvəlcə qeyd edək ki, $L_1 \infty L_2$ və $L_2 \infty L_1$ şərtlərini ödəyən L_1 və L_2 dillərinə polinomial ekvivalent dillər deyilir. Eyni ilə $\Pi_1 \infty \Pi_2$ və $\Pi_2 \infty \Pi_1$ olsa, Π_1 və Π_2 polinomial ekvivalent məsələ adlanır.

Tərif: Tutaq ki, müəyyən $L \in NP$ və $\forall\ L' \in NP$ üçün $L' \infty L$ münasibəti ödənir. Onda L dilinə NP-tam dili deyilir.

Tərif: Əgər $\Pi \in NP$ və $\forall\ \Pi' \in NP$ məsələsi üçün $\Pi' \infty \Pi$ olarsa, onda deyirlər ki, P məsələsi NP-tam məsələlər sinfindəndir.

Bu təriflərdən göründüyü kimi əgər NP-tam sinfində olan heç olmasa bircə məsələ polinomial həll olunursa, onda deyirlər ki, NP sinfindəki bütün məsələlər polinomial həll olunur. Əksinə, əgər NP sinfində heç olmasa bircə çətin həll olunan məsələ varsa, onda NP-tam sinfindəki bütün məsələlər çətin həll olunandır. Beləliklə, biz göstərdik ki, əgər $NP \setminus P \neq$ olarsa, onda $\exists \Pi$ ∅malıdır ki, $\Pi \in NP \setminus P$.

Aşağıdakı lemmanı isbat edək.

Lemma 7. Tutaq ki, $L_1 \in NP$ və $L_2 \in NP$. Bundan əlavə fərz edək ki, $L_1 \infty L_2$. Əgər $L_1 \in NP$ tam olsa, onda $L_2 \in NP$ tam olar.

İsbatı: Şərtə görə $L_1 \in NP$ tam sinfindədir. Yəni, $\forall\ L' \in NP$ üçün $L_1' \infty L_1$ və lemmanın şərtinə görə $L_1 \infty L_2$. Tranzitivlik lemmasına görə alırıq ki, $\forall\ L' \in NP$ üçün $L_1' \infty L_2$. Deməli, $L_2 \in NP$ tamdır. Yəni, lemma isbat olundu. Bu lemma məsələ halında da aşağıdakı kimi yazıla bilər.

Lemma 8. Tutaq ki, $\Pi_1, \Pi_2 \in NP$, $\Pi_1 \infty \Pi_2$ və $\Pi_1 \in NP$-tam. Onda $\Pi_2 \in NP$-tam olar.

Lemma 7 və lemma 8 imkan verir ki, hər hansı Π məsələsinin - NP tam sinfində olması , yaxud olmaması üçün kriteriya yazaq. Bunun üçün aşağıdakı 2 şərt ödənilməlidir.

1). $\Pi \in NP$ olsun.

2). Hər hansı məlum $\Pi' \in NP$ tam üçün $\Pi' \infty \Pi$

Mühazirə 23."Doğruluq" məsələsi və Kuk teoremi

Beləliklə NP- tam sinfində olan hər hansı konkret məsələnin olduğunu göstərmək lazımdır. Belə bir məsələnin NP -tam sinfində olmasını ilk dəfə göstərən Kuk adlı alim olub və məsələ "Doğruluq" məsələsi adlanır. Doğruluq məsələsi aşağıdakı kimi yazılır.

Şərt: $U = (U_{1,}, U_2, ..., U_n)$ Bul dəyişənləri çoxluğu və bu çoxluq üzərində C dizyunksiyalarının konyuksiyası verilmişdir.

Sual: C konyuksiyasının doğruluq qiymətləri varmı?

Doğruluq məsələsini izah edək.

Tutaq ki, $U = (U_{1,}, U_2, ..., U_n)$ Bul dəyişənləri çoxluğu verilib.

Bu çoxluq üzərində $t : U \to (T, F)$ funksiyasına U dəyişənindən asılı Bul funksiyası deyəcəyik. Belə ki, $t(U) = T$ yalnız və yalnız o zaman T olur ki, U doğru olsun. $t(U) = F$ yalnız və yalnız o zaman olur ki, U yalan olsun. U ve $\overline{\mathrm{U}}$ - yə literal deyilir. $u \in U$ literalı yalnız və yalnız o zaman t funksiyasına nəzərən doğru qiymət alır ki, u dəyişəni t-yə nəzərən doğru qiymət alsın. Literal $\bar{u}$ yalnız və yalnız o zaman t funksiyasına nəzərən doğru qiymət alır ki, u dəyişəni "yalan" qiymət alsın və tərsinə.

U və $\overline{U}$ literalından düzəlmiş dizyunksiya o zaman doğru olur ki, bunlardan heç olmazsa biri doğru olsun. $U = (U_{1,}, U_2, ..., U_n)$ çoxluğu üzərində qurulmuş dizyunksiyaların C konyuksiyası yalnız və yalnız o zaman doğru olur ki, onun tərkibindəki, bütün dizyunksiyalar doğru olsun. C konyuksiyasını doğru edən U literallar çoxluğuna onun doğruluq qiymətləri deyilir. C konyuksiyası verildikdə onun doğruluq qiymətləri yığımını tapmaq məsələsinə doğruluq məsələsi deyilir.

Teorem: (Kuk) Doğruluq məsələsi NP-tam məsələdir.