

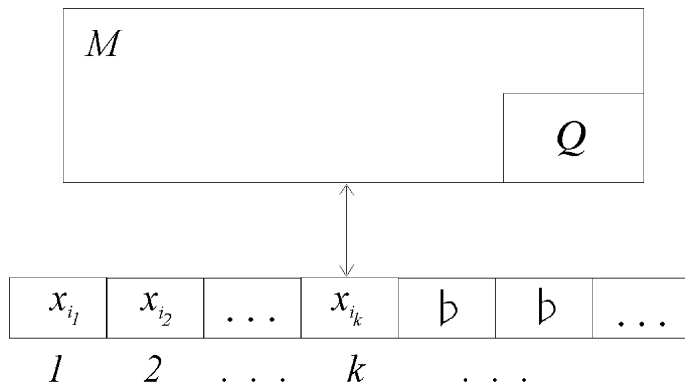
Сложност на алгоритъм. Асимптотично поведение на целочислени функции (O-, Ω -, Θ -, o- и ω -нотация).

Сложност на рекурсивни програми.

Понятието алгоритъм е неформално. Съществуват много опити за неговото формализиране. Между най-известните от тях са: машина на Тюринг; машина с произволен достъп до паметта; език за програмиране; частично-рекурсивните функции, въведени от Ст. Клини; λ -смятането, въведено от А. Чърч; машините на Пост, въведени от Е. Пост, нормалните алгоритми на Марков, въведени от А. А. Марков и др., но **важно е да се отбележи, че в теоретичен план е доказана еквивалентността на всички известни до днес формални модели**. По същество доказването на еквивалентността на произволни два модела се състои в симулиране на изчисления направени в единия модел с изчисления в другия и обратно. Следователно за изследване с математически апарат на понятието алгоритъм е достатъчно да разгледаме **само един негов формален модел – машина на Тюринг**.

Машина на Тюринг

На схемата е представена схематично абстрактна математическа машина. Тя чете входа си от безкрайно в едната посока (без ограничение на общността сме избрали посоката надясно) запомнящо устройство и записва изхода си на същото устройство, наричано за кратко лента. Входно-изходната лента е разделена на изброимо множество еднотипни клетки, във всяка една от които може да бъде записана някоя от буквите на крайна (входно-изходна) азбука. Клетките са номерирани (отляво на дясно) с естествените числа $1, 2, \dots, k$. Четенето от лентата и писането на нея се извършва от четящо-пишещо устройство, наричано глава. Главата е подвижна, във всеки момент се намира върху точно една клетка на лентата и в края на всеки такт може да се премести върху една клетка наляво (ако не е била на най-лявата клетка), една клетка надясно или да остане на място. Означаваме с L , R и S тези възможности за преместване на главата.

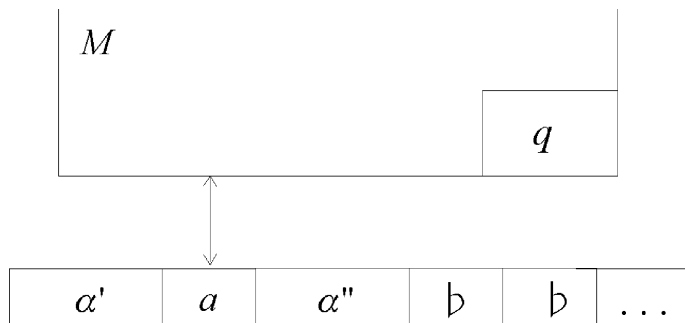


Дефиниция. Петорката $M = \langle Q, X, q_0, \delta, F \rangle$, където Q е крайно множество от състояния, X е крайна входна азбука, $q_0 \in Q$ е начално състояние, F са заключителни състояния, $F \cap Q = \emptyset$, а $\delta : Q \times X \rightarrow (Q \cup F) \times X \times \{L, R, S\}$ е функция на преходите, наричаме машина на Тюринг (MT) с безкрайна в едната посока лента.

Една от буквите на азбуката X има специално предназначение. Ще я наричаме запълваща буква (или бленк) и я бележим с **b**. Ще разглеждаме машини на Тюринг с условието, че преди започване на работата, крайно множество от клетки на лентата не съдържат бленк. При това условие, след отработване на краен брой тактове, клетките на лентата, които не съдържат бленкове, ще продължат да бъдат крайно множество.

Дефиниция. Нека k е най-малкото естествено число такова, че всички клетки на лентата на MT M с номера по-големи или равни на k съдържат бленк. Думата $x_{i1} x_{i2} \dots x_{ik} \in X^*$ такава, че x_{ij} е записана в клетката с номер j , наричаме текуща лентова дума на M .

Дефиниция. Нека MT $M = \langle Q, X, q_0, \delta, F \rangle$ е в състояние $q \in Q$, текущата лентова дума е $\alpha' a \alpha'' \in X^*$, а главата на машината се намира над буквата a (виж схемата). Тройката $(\alpha', q, a \alpha'')$ наричаме конфигурация на MT M . Очевидно, конфигурацията на MT еднозначно определя бъдещото и поведение. Конфигурациите от вида $(\varepsilon, q_0, \alpha)$ наричаме начални конфигурации.



Ще опишем формално работата на M чрез трансформация на конфигурации.

Дефиниция. Релацията $R_{\vdash} \subseteq (X^* \times Q \times X^*) \times (X^* \times Q \times X^*)$, съставена от наредени двойки конфигурации дефинираме така:

- а) $(\alpha' b, q, a \alpha'') \vdash (\alpha', q', b a_0 \alpha'')$, ако $\delta(q, a) = (q', a_0, L)$;
- б) $(\alpha' b, q, a \alpha'') \vdash (\alpha' b a_0, q', \alpha'')$, ако $\delta(q, a) = (q', a_0, R)$;
- в) $(\alpha' b, q, a \alpha'') \vdash (\alpha' b, q', a_0 \alpha'')$, ако $\delta(q, a) = (q', a_0, S)$.

С други думи, ако $\delta(q, a) = (q', a_0, t)$, то в края на такта машината на Тюринг записва a_0 на мястото на a , преминава в състояние q' и премества главата на една клетка в ляво, на една клетка в дясно или я оставя на място, ако $t = L, R$ или S , съответно. Ако $k1$ и $k2$ са две конфигурации на M , то трансформацията $k1 \vdash k2$

наричаме непосредствен преход на κ_1 в κ_2 . Казваме още, че κ_1 преминава непосредствено в κ_2 .

Дефиниция. Нека $R \models \subseteq (X^* \times Q \times X^*) \times (X^* \times Q \times X^*)$ е рефлексивно и транзитивно затваряне на $R_{\perp}$. Трансформацията $\kappa' \models \kappa''$ наричаме преход на κ' в κ'' . Казваме още, че κ' преминава в κ'' . Последователността от непосредствени преминавания $\kappa' \vdash \kappa_1 \vdash \kappa_2 \vdash \dots \vdash \kappa_r \vdash \kappa''$ наричаме изчисление с машината на Тюринг M .

Естествен начин за завършване на едно изчисление на машината на Тюринг M е, тя да достигне някое от заключителните състояния. В такъв случай, казваме че M спира нормално. Машината може да спре аварийно, когато се намира в конфигурация, за която стойността на δ не е дефинирана, или пък ако машината се опитва да премести главата си вляво от най-лявата клетка на лентата.

Възможно е, за някоя лентова дума α , машината на Тюринг M да не спре работа никога. Например, ако машината се намира в състояние q , главата попадне над клетка, съдържаща буквата a и $\delta(q, a) = (q, a, S)$. В такъв случай казваме, че M зацикля.

Дефиниция. Нека M е МТ. Нека с α сме означили произволна лентова дума и $\beta = \beta' \beta''$. Тогава функцията $f_M: X^* \rightarrow X^*$

$$f_M(\alpha) = \begin{cases} \beta & , \quad \text{ако } (\varepsilon, q_0, \alpha) \models (\beta', q, \beta''), q \in F \\ \text{неопределена,} & \text{ако } M \text{ зацикля или спира аварийно} \end{cases}$$

наричаме *изчислима по Тюринг*

Дефиниция. Нека M е МТ. Нека $F = \{qu, qn\}$. Дефинираме език, разпознаван от МТ M като множеството от думи $LM \subseteq X^*$ такова, че

$$\forall \alpha \in LM, (\varepsilon, q_0, \alpha) \models (\alpha', qu, \alpha''), \text{ а } \forall \beta \notin LM, (\varepsilon, q_0, \beta) \models (\beta', qn, \beta'').$$

Състоянието qu наричаме допускащо, а състоянието qn отхвърлящо. Не е трудно да се трансформира една МТ в еквивалентна на нея със само две заключителни състояния едно допускащо и едно отхвърлящо.

Сложност на алгоритми

Основен ресурс, спрямо който ще изследваме качествата на алгоритмите е времето за работа, но успоредно с това ще обърнем внимание и на количеството използвана памет, като дефинираме понятията сложност на алгоритмите, по отношение на необходимите им време и памет.

Формализацията на понятието размер на входните данни за определен алгоритъм е много сложна задача. Затова ще поставим едно важно условие. Ще поискаме кодирането на екземплярите с думи над избрана азбука да е разумно, т.е. всичко необходимо за решаването на съответната задача да се съдържа в думата, представяща екземпляра и да няма излишък, т.е. такава част от входната дума, че ако я премахнем, същността на екземпляра не се променя.

Нека A е множество от думи над азбуката X , т.е. $A \subseteq X^*$. Ще означаваме дължината на думата $\alpha \in X^*$ с $d(\alpha)$, а с $A_n = \{\alpha \mid \alpha \in A, d(\alpha) = n\}$, $n = 0, 1, 2, \dots$ подмножеството от всички думи на A с дължина n .

Дефиниция. Нека $f_M : A \rightarrow X^*$ е тотална изчислима по Тюринг функция. Да означим с $\#M(\alpha)$ броя на стъпките, които M прави, при работа върху лентовата дума α . Функцията

$$t_M(n) = \max_{\alpha \in A_n} \#M(\alpha)$$

наричаме сложност по време на M в най-лошия случай, а функцията

$$\tilde{t}_M(n) = \frac{\sum_{\alpha \in A_n} \#M(\alpha)}{|A_n|}$$

наричаме средна сложност по време на МТ M .

Дефиниция. Нека $f_M : A \rightarrow X^*$ е тотална изчислима по Тюринг функция. Да означим с $\&M(\alpha)$ броя на клетките на лентата, които M използва, при работа върху лентовата дума α . Клетките, в които е разположена самата α винаги включваме в този брой, затова $\&M(\alpha)$ не може да е по-малко от дължината на α . Функцията

$$s_M(n) = \max_{\alpha \in A_n} \&M(\alpha)$$

наричаме сложност по памет на M в най-лошия случай, а функцията

$$\tilde{s}_M(n) = \frac{\sum_{\alpha \in A_n} \&M(\alpha)}{|A_n|}$$

наричаме средна сложност по памет на МТ M .

Тъй като разпознаването на език е частен случай на изчисляване на функция $f_M : A \rightarrow \{\text{true}, \text{false}\}$, горните дефиниции са валидни и в този случай.

Асимптотично поведение на целочислени функции

Дефиниция. Нека A^+ е множеството от целочислени асимптотически положителни функции $A^+ = \{f: \mathbb{N} \rightarrow \mathbb{Z} \mid \exists n_0 \in \mathbb{N} \text{ такава, че за } \forall n \in \mathbb{N} \geq n_0 \text{ следва, че } f(n) > 0\}$.

За улесняване на предстоящото изложение ще приемем за верни следните твърдения:
 $n, n_0, n_1, n_2 \in \mathbb{N}$; $c, c_1, c_2 \in \mathbb{R}$; $f(n), g(n)$ и $h(n) \in A^+$;

Дефиниция. Нека $O(f(n)) \subseteq A^+$ и

$$O(f(n)) = \{g(n) \mid \exists n_0 > 0 \text{ и } \exists c > 0 \text{ такива, че за } \forall n \geq n_0 \text{ } 0 \leq g(n) \leq cf(n)\}.$$

Ще казваме, че " $g(n)$ е O от $f(n)$ ", ако $g(n) \in O(f(n))$ и ще бележим с $g(n) = O(f(n))$ или това са всички функции, които нарастват НЕ по-бързо от f .

Дефиниция. Нека $\Omega(f(n)) \subseteq A^+$ и

$$\Omega(f(n)) = \{g(n) \mid \exists n_0 > 0 \text{ и } \exists c > 0 \text{ такива, че за } \forall n \geq n_0 \text{ } 0 \leq cf(n) \leq g(n)\}.$$

Ще казваме, че " $g(n)$ е Ω от $f(n)$ ", ако $g(n) \in \Omega(f(n))$ и ще бележим с $g(n) = \Omega(f(n))$ или това са всички функции, които нарастват НЕ по-бавно от f .

Дефиниция. Нека $o(f(n)) \subseteq A^+$ и

$$o(f(n)) = \{g(n) \mid \forall c > 0, \exists n_0 > 0 \text{ такива, че за } \forall n \geq n_0 \text{ } 0 \leq g(n) < cf(n)\}.$$

Ще казваме, че " $g(n)$ е o от $f(n)$ ", ако $g(n) \in o(f(n))$ и ще бележим с $g(n) = o(f(n))$ или това са всички функции, които нарастват по-бавно от f .

Дефиниция. Нека $w(f(n)) \subseteq A^+$ и

$$w(f(n)) = \{g(n) \mid \forall c > 0, \exists n_0 > 0 \text{ такива, че за } \forall n \geq n_0 \text{ } 0 \leq cf(n) < g(n)\}.$$

Ще казваме, че " $g(n)$ е w от $f(n)$ ", ако $g(n) \in w(f(n))$ и ще бележим с $g(n) = w(f(n))$ или това са всички функции, които нарастват по-бързо от f .

Дефиниция. Нека $\theta(f(n)) \subseteq A^+$ и

$$\theta(f(n)) = \{g(n) \mid \exists c_1, c_2 > 0, \exists n_0 > 0 \text{ такива, че за } \forall n \geq n_0 \text{ } 0 \leq c_1 f(n) \leq g(n) \leq c_2 f(n)\}.$$

Ще казваме, че " $g(n)$ е θ от $f(n)$ ", ако $g(n) \in \theta(f(n))$ и ще бележим с $g(n) = \theta(f(n))$ или това са всички функции, които нарастват еднакво бързо с f .

$O(), \Omega(), o(), w(), \theta() \subseteq A^+ \times A^+$ са релации над декартовото произведение на целочислените асимптотически функции. Нека разгледаме техните свойства:

1. **Рефлексивност** – непосредствено от дефинициите следва, че $O()$, $\Omega()$ и $\theta()$ са рефлексивни, а $o()$, $w()$ не са:
 $f(n) = O(f(n))$;
 $f(n) = \Omega(f(n))$;
 $f(n) = \theta(f(n))$;
 $f(n) \neq o(f(n))$;
 $f(n) \neq w(f(n))$;
2. **Симетричност** – от дефинициите следва, че само $\theta()$ е симетрична:
 $g(n) = \theta(f(n)) \Leftrightarrow f(n) = \theta(g(n))$;
3. **Транзитивност** – всички релации за транзитивни:
 $f(n) = O(g(n))$ и $g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$
 $f(n) = \Omega(g(n))$ и $g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$
 $f(n) = o(g(n))$ и $g(n) = o(h(n)) \Rightarrow f(n) = o(h(n))$
 $f(n) = w(g(n))$ и $g(n) = w(h(n)) \Rightarrow f(n) = w(h(n))$
 $f(n) = \theta(g(n))$ и $g(n) = \theta(h(n)) \Rightarrow f(n) = \theta(h(n))$
4. **Антисиметричност** – само релациите $o()$, $w()$ са антисиметрични, т.е.
Ако $g(n) = o(f(n)) \Rightarrow f(n) \neq o(g(n))$
Ако $g(n) = w(f(n)) \Rightarrow f(n) \neq w(g(n))$

Лема 1 без доказателство:

Ако $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0$, то $f(n) = o(g(n))$;

Лема 2 без доказателство:

Ако $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = \infty$, то $f(n) = w(g(n))$;

Лема 3 без доказателство:

Ако $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c \neq 0$, то $f(n) = \theta(g(n))$;

Рекурентни отношения

Дефиниция. Нека $f(x)$ е функция, $f(x)$ е мултипликативна, ако

$$f(x * y) = f(x) * f(y)$$

Теорема. Нека е зададено рекурентното отношение от вида

$$\begin{cases} t(n) = at\left(\frac{n}{b}\right) + d(n) \\ t(p) = c, \quad p \leq 1 \end{cases}$$

(задача с размер на входа n се разпада (разбива) до a задачи от същия вид с размер $\frac{n}{b}$), където $d(n)$ е мултипликативна функция зависеща от n , а $a, b \geq 1$ то:

1. Ако $a = d(b)$, то $t(n) = O(\log_b n n^{\log_b a})$;

2.1 Ако $a > d(b)$, то $t(n) = O(n^{\log_b a})$;

2.2 Ако $a < d(b)$, то $t(n) = O(n^{\log_b d(b)})$.

Доказателство:

За решаването на зададеното рекурентно отношение прилагаме стандартна итерационна техника:

$$\begin{aligned} t(n) &= at\left(\frac{n}{b}\right) + d(n) & * a^0 & + \\ t\left(\frac{n}{b}\right) &= at\left(\frac{n}{b^2}\right) + d\left(\frac{n}{b}\right) & * a^1 & + \\ t\left(\frac{n}{b^2}\right) &= at\left(\frac{n}{b^3}\right) + d\left(\frac{n}{b^2}\right) & * a^2 & + \\ &\dots & \dots & \\ t\left(\frac{n}{b^{k-1}}\right) &= at\left(\frac{n}{b^k}\right) + d\left(\frac{n}{b^{k-1}}\right), \frac{n}{b^k} \leq 1 & * a^{k-1} & + \\ t\left(\frac{n}{b^k}\right) &= c & * a^k & \end{aligned}$$

След привеждане получаваме:

$$t(n) = a^0 d(n) + a^1 d\left(\frac{n}{b}\right) + a^2 d\left(\frac{n}{b^2}\right) + \dots + a^{k-1} d\left(\frac{n}{b^{k-1}}\right) + a^k c. \text{ или}$$

$$t(n) = \sum_{i=0}^{k-1} a^i d\left(\frac{n}{b^i}\right) + a^k c$$

От това, че $d(n)$ е мултипликативна, следва

$$t(n) = \sum_{i=0}^{k-1} a^i d\left(\frac{n}{b^i}\right) + a^k c \leq \sum_{i=0}^{k-1} a^i d\left(\frac{b^k}{b^i}\right) + a^k c = ,$$

$\frac{n}{b^k} \leq 1$ или $n \leq b^k$, приемаме най-лошият случай, че $n = b^k$ и $k = \log_b n$, следователно

$$= \sum_{i=0}^{k-1} a^i d(b^{k-i}) + a^k c = \sum_{i=0}^{k-1} a^i d(b)^{k-i} + a^k c$$

1 случай. Ако $a = d(b)$

$$\sum_{i=0}^{k-1} a^i d(b)^{k-i} + a^k c = \sum_{i=0}^{k-1} a^k + a^k c = k a^k + a^k c = (k + c) a^k = (\log_b n + c) a^{\log_b n} \Rightarrow$$

$$t(n) = O(\log_b n a^{\log_b n}) = O(\log_b n n^{\log_b a})$$

2 случай. Ако $a \neq d(b)$

$$\sum_{i=0}^{k-1} a^i d(b)^{k-i} + a^k c = \sum_{i=0}^{k-1} \frac{a^i d(b)^k}{d(b)^i} + a^k c = d(b)^k \sum_{i=0}^{k-1} \left(\frac{a}{d(b)}\right)^i + a^k c =$$

$$= d(b)^k \frac{\left(\frac{a}{d(b)}\right)^k - 1}{\frac{a}{d(b)} - 1} + a^k c = \frac{a^k - d(b)^k}{\frac{a}{d(b)} - 1} + a^k c, \quad \text{имамаме } \frac{1}{\frac{a}{d(b)} - 1} = T \text{ const}$$

$$= T(a^k - d(b)^k) + a^k c = (T + c)a^k - T d(b)^k$$

2.1 случай. Ако $a > d(b) \Rightarrow T > 0, t(n) = O(a^k) = O(a^{\log_b n}) = O(n^{\log_b a})$

2.2 случай. Ако $a < d(b) \Rightarrow T < 0, t(n) = O(d(b)^k) = O(d(b)^{\log_b n}) = O(n^{\log_b d(b)})$