Programiranje 1/I

Milenko Mosurović

Univerzitet Crne Gore

studijska godina 2020/21.

Uvodne napomene o predmetu

- Fond časova 3+2.
- Kolokvijumi (programski jezik C) nose 50 bodova - u dogovoru sa asistentom.
- Završni ispit (teorija) nosi 50 bodova.
- Prelazna ocjena se dobija za 50 ili više bodova. Tačnije: 0-49 ocjena F, 50-59 ocjena E, 60-69 ocjena D, 70-79 ocjena C, 80-89 ocjena B i 90-100 ocjena A.
- Literatura: Skripta u elektronskom obliku može se naći na sajtu www.ucg.ac.me od autora M. Martinovića.

Kratak sadržaj predmeta

- Tjuringova mašina kao model izračunjivosti
  - Primjeri Tjuringovih mašina
  - Tjuringovi diagrami
  - Funkcije izračunljive po Tjuringu
  - Funkcije normalno izračunljive po Tjuringu
  - Primjeri nerešivih skupova
- Model izračunivosti RAM
- Model izračunivosti RASP
- Odnosi naznačenih modela

Tjuringova mašina i izračunljive funkcije

Uvod - Intuitivni pojam algoritma

- Neka je A konačan neprazan skup koji nazivamo azbukom, a njegove elemente slovima azbuke.
- **Def.** Skup svih riječi nad azbukom A, u oznaci $\Omega(A)$, je minimalni skup koji zadovoljava sledeća svojstva: 1) prazna riječ λ pripada $\Omega(A)$ (tj. $\lambda \in \Omega(A)$), 2) Ako je $\omega \in \Omega(A)$ i $a \in A$ onda je i $\omega a \in \Omega(A)$ (tj. nadovezivanjem slova a na već postojeću riječ ω dobijamo novu riječ ωa).
- Elemente skupa $\Omega(A)$ nazivamo riječima nad azbukom A.
- Primjer. Ako je $A = \{x, y\}$ onda je $\Omega(A) = \{\lambda, x, y, xx, xy, yx, yy, xxx, xxy, ...\}$.
- **Def.** Neka je $\Omega(A)$ skup svih riječi nad azbukom A. Za skup L kažemo da je jezik nad azbukom A ako je $L \subset \Omega(A)$.

- Drugim riječima ako od skupa svih riječi izdvojimo neke koje su važeće onda nam te riječi čine jezik. Npr. KOŹE, KIŚELO, MLIJEKO jesu riječi crnogorskog jezika a ZZRVFT nije riječ crnogorskog jezika i ako je to riječ nad crnogorskom azbukom.
- Def. Dužina riječi ω u oznaci $|\omega|$ je broj slova te riječi. Preciznije $|\lambda| = 0$ i $|\omega a| = |\omega| + 1$.
- Ako azbuka ima n elemenata onda će mo je označavati sa A_n a njene elemente sa $a_1, \ldots, a_n$ tj. $A_n = \{a_1, \ldots, a_n\}$.
- Uvodimo još jedan simbol $a_0 \notin A$ koji nazivamo praznim simbolom (blanko).
- Za $n = 1$ koristimo i oznaku $a_1 = |$. Tako za $A_1 = \{|\}$ imamo $\Omega(A_1) = \{\lambda, |, ||, |||, \ldots\}$. Očiti $f(n) = \underbrace{|\ldots|}_{n\text{-puta}}$ je bijekcija iz N_0 u $\Omega(A_1)$.

- Intuitivno algoritam je precizno opisan postupak za rješavanje nekog zadatka (problema).
- Taj postupka treba da je napisan na nekom jeziku. Takođe na nekom jeziku treba da zadamo ulaz i rezultat našeg algoritma.
- Otuda na intuitivni pojam algoritma možemo gledati kao petorku $\mathcal{A} = (\mathcal{P}, E, A, B, n)$, gdje su E, B i A redom ulazna, izlazna i radna azbuka ($E \cup B \subset A$), n je dimenzija ulaza a $\mathcal{P}$ je opis postupka (kako od ulaza doći do rezultata) - nazivaćemo ga pravilo.
- Algoritam ulaz $\omega_1, \ldots, \omega_n$, ($\omega_k \in \Omega(E)$) transvormiše u izlaz $r \in \Omega(B)$ koristeći pravilo $\mathcal{P}$ koje je zapisano u azbuci A.
- Primjer. Euklidski algoritam: za $n \geq m > 0$ $NZD(n, m) = NZD(m, n \bmod m)$; $NZD(n, 0) = n$

Zahtjevi (svojstva) za pravilo $\mathcal{P}$

1. $\mathcal{P}$ sadrži neke "elementarne radnje" - operacije. Operacije iz $\mathcal{P}$ izvršavaju se etapno jedna za drugom i u svakom koraku jednoznačno je određeno koja sledeća operacija treba da se izvrši.
2. Pravilo $\mathcal{P}$ je jednoznačno.
3. Algoritam je deterministički tj. da će pri svakom izvršavanju za isti ulaz prolaziti kroz iste korake i dati isti rezultat.
4. Pravilo $\mathcal{P}$ ne koristi nikakve spoljne informacije osim ulaza.
5. Pravilo $\mathcal{P}$ treba da bude konačno.
6. Broj koraka da se potrebnih da se izvrši $\mathcal{P}$ treba da bude konačan.
7. $\mathcal{P}$ zahtjeva samo konačan "memorijski" prostor.

- Za neke zadatke nepostoji algoritam koji ih rešava takvi zadaci su nerješivi (neodlučivi). Gedel je 1931. godine dokazao nerješivost jednog zadatka u logici.
- Da bi mogli da izvodimo matematičke dokaze (npr. o nerješivosti) moramo imati precizan pojam algoritma.
- Postoji više pristupa za (matematičku) definiciju algoritma. Dokazano je da su sve one međusobno ekvivalentne.
- Čerčova hipoteza. Intuitivni i matematički pojam algoritma su ekvivalentni.
- Ekvivalentnost je u smislu da ako za jedan zadatak postoji algoritam u smislu jedne definicije onda će za taj zadatak postojati algoritam i u smislu druge definicije (tj. dobijamo isti skup "izračunljivih funkcija").

Izračunljive funkcije

- Za f-ju (tj. funkciju) $f : \Omega^n(E) \to \Omega(B)$ čiji je domen skup $\Omega^n(E)$ kažemo da je potpuna ili totalna.
- Za f-ju $f : \mathrm{Dom}_f \to \Omega(B)$ čiji je domen $\mathrm{Dom}_f \subset \Omega^n(E)$ kažemo da je djelimična ili parcijalna.
- Algoritmu $\mathcal{A} = (\mathcal{P}, E, A, B, n)$ pridružimo f-ju $f_{\mathcal{A}}$ na sledeći način. Ako algoritam $\mathcal{A}$ primi $\omega = (\omega_1, \ldots, \omega_n)$ kao ulaz i
  - *i)* dobijemo rezultat $r \in \Omega(B)$ onda je $\omega \in \mathrm{Dom}_f$ i $f_{\mathcal{A}}(\omega) = r$;
  - *ii)* ne dobijemo rezultat onda $\omega \notin \mathrm{Dom}_f$.
- Algoritam $\mathcal{A}$ jednoznačno određuje domen funkcije i samu funkciju $f_{\mathcal{A}}$.
- **Def.** Za djelimičnu f-ju $f : \mathrm{Dom}_f \to \Omega(B)$ $(\mathrm{Dom}_f \subset \Omega^n(E))$ kažemo da je izračunljiva ako postoji algoritam $\mathcal{A} = (\mathcal{P}, E, A, B, n)$ takav da je $f = f_{\mathcal{A}}$. Tada se za algoritam $\mathcal{A}$ kaže da predstavlja računsku proceduru za f-ju f.

- Ne traži se da algoritam bude "konstruisan" već samo da se dokaže da postoji.
- Primjer. Neka je $f(n) = \begin{cases} 0 & , \quad \text{ako je Rimanova hipoteza istinita} \\ 1 & , \quad \text{inače} \end{cases}$
  Da li je f-ja f izračunljiva?
- Rimanova hipoteza. Sve netrivijalne nule Rimanove zeta-funkcije $\zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$ se nalaze na pravoj $Re(z) = \frac{1}{2}$.
- Rimanova hipoteza je važan matematički zadatak koji još uvjek nije riješen .
- Funkcija f je izračunljiva jer algoritam postoji. Međutim za sada ne znamo koji algoritam je tačan onaj koji vraća nulu ili onaj koji vraća jedinicu.

Rješivi skupovi

- Neka su $\mathcal{M}_1$ i $\mathcal{M}_2$ dva skupa i neka je $\mathcal{M}_1 \subset \mathcal{M}_2$. Kažemo da je skup $\mathcal{M}_1$ riješiv u odnosu na $\mathcal{M}_2$ ako postoji algoritam koji može da se primijeni na bilo koji element skupa $\mathcal{M}_2$ i koji nam u tom slučaju daje odgovor da li taj element pripada ili ne pripada skupu $\mathcal{M}_1$.
- Drugim riječima skup je rješiv ako možemo algoritamski da utvrdimo koji elementi mu pripadaju. Međutim bitno je i u odnosu na koji skup posmatramo. Npr. ako je $\mathcal{M}_1 \subset \mathcal{M}_2$ i $\mathcal{M}_1 \subset \mathcal{M}_3$ može biti da je $\mathcal{M}_1$ riješiv u odnosu na $\mathcal{M}_2$ a da nije rješiv u odnosu na $\mathcal{M}_3$.
- Možemo govoriti i o rješivosti predikata - relacija. Kako je relacija poskup Dekartovog proizvoda skupova to se opet svodi na rješivost skupova.

Rješivi skupovi - primjeri

- Da li je skup $\mathcal{M}_1$ riješiv u odnosu na skup $\mathcal{M}_2$ ako je: $\mathcal{M}_2 = \{a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \mid a_1, \ldots, a_n, b \in Z \text{ i } n \geq 1\}$ a $\mathcal{M}_1 = \{a_1x_1 + \cdots + a_nx_n = b \in \mathcal{M}_2 \mid (\exists \alpha_1, \ldots, \alpha_n \in Z)\ a_1\alpha_1 + \cdots + a_n\alpha_n = b\}$?
- Drugim riječima da li iz skupa svih jednačina sa cjelobrojnim kojeficijentima možemo izdvojiti one koje imaju cjelobrojno rješenje?
- Pravilo. Jednačina pripada skupu $\mathcal{M}_1$ akko $NZD(a_1, \ldots, a_n) \mid b$. Otuda je skup $\mathcal{M}_1$ riješiv u odnosu na $\mathcal{M}_2$.
- Primjer. Jednačina $4x_1 + 6x_2 = 34$ pripada skupu $\mathcal{M}_1$ jer $NZD(4, 6) = 2 \mid 34$. Zaista $x_1 = 1$, $x_2 = 5$ je rješenje.
- Primjer. Jednačina $9x_1 + 6x_2 = 34$ ne pripada skupu $\mathcal{M}_1$ jer $NZD(9, 6) = 3 \nmid 34$. Lijeva strana djeljiva sa 3 a desna ne.

- Neka je $p(x_1, \ldots, x_k) = \sum_{\alpha=(j_1,\ldots,j_k)} a_\alpha x_1^{j_1} \cdots x_k^{j_k}$ gdje je $a_\alpha \in Z$, $j_i \in N_0$ i $j_1 + j_2 + \cdots + j_k \leq n$ (polinom n-tog stepena s cjelobrojnim kojeficijentima). Da li jednačina $p(x_1, \ldots, x_k) = 0$ (opšta diofantska jednačina) ima cjelobrojno rješenje? Ovo je poznato kao deseti Hilbertov problem.
- Da li postoji algoritam koji nam može dati odgovor na prethodno pitanje?
- Matijašević 1971. dokazao da je pitanje cjelobrojnih nula opšte diofantske jednačine algoritamski nerješivo i time rješio 10 Hilbertov problem. Dok za slučaj $k = 1$ postoji algoritam.
- Otuda za $\mathcal{M}_2 = \{p(x_1, \ldots, x_k) = 0\}$,
$\mathcal{M}_1 = \{p(x_1, \ldots, x_k) = 0 \mid (\exists \alpha_1, \ldots, \alpha_k \in Z)\ p(\alpha_1, \ldots, \alpha_k) = 0\}$
dobijamo $\mathcal{M}_1$ neriješiv u odnosu na $\mathcal{M}_2$.

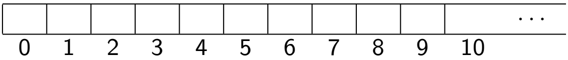
- **Def.** Neka je $|E| \geq 1$, $n \geq 1$ i $S \subset \Omega^n(E)$. Za skup S kaže se da je rješiv u odnosu na skup $\Omega^n(E)$ ako postoji algoritam $\mathcal{A} = (\mathcal{P}, E, A, B, n)$ takav da: kada se algoritam $\mathcal{A}$ primijeni na bilo koju n-torku riječi $\omega \in \Omega^n(E)$ onda se taj algoritam zaustavlja i saopštava kao rezultat riječ a_0 ako je $\omega \in S$ a riječ a_1 ako $\omega \notin S$. Za svaki takav algoritam se kaže da predstavlja rješavajuću proceduru za S u odnosu na $\Omega^n(E)$.
- Svakom skupu S ($S \subset \Omega^n(E)$) možemo pridružiti karakterističnu funkciju u odnosu na skup $\Omega^n(E)$

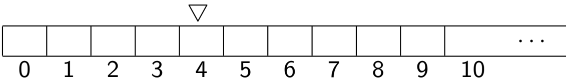
$$\chi(x) = \begin{cases} a_0 & \text{, ako} \quad x \in S \\ a_1 & \text{, ako} \quad x \notin S \end{cases}$$

- **Teorema.** Skup S je rješiv u odnosu na skup $\Omega^n(E)$ akko je karakteristična funkcija χ skupa S u odnosu na skup $\Omega^n(E)$ izračunljiva.
- **Dokaz.** ...

Definicija Tjuringove mašine

- Da bi mogli precizno da definišemo pojam algoritma (pravila $\mathcal{P}$) moramo definisati model izračunljivosti. Jedan od takvih modela je Tjuringova mašina (TM).
- TM treba da ima radnu azbuku. Neka je to A_t.
- Da bi čuvala informacije (zapisane na azbuci A_t) TM ima jednu traku koja je izdjeljena na polja. U svako polje trake može biti upisan jedan od simbola iz $A_t \cup \{a_0\}$. Ako je upisan simbol a_0 kažemo da je polje prazno.
- Traka ima prebrojivo mnogo polja koja su numerisana redom brojevima $0, 1, 2, \cdots$.





- Polje označeno sa 0 je početno. Desno od njega se nalazi polje označeno sa 1. Opšte ako je polje označeno sa $m \in N_0$ desno od njega se nalazi polje koje je označeno sa $m + 1$.
- Za ovakvu traku kažemo da je ograničena sa lijeve strane a sa desne da nije ograničena.
- Da bi mogla da mijenja sadržaj polja TM ima glavu. Glava se u jednom momentu (koraku TM) nalazi tačno iznad jednog polja. To polje se naziva radno polje.
- Glava može da pročita sadržaj radnog polja (tj. jedan od simbola iz $A_t \cup \{a_0\}$), izbriše njegov sadržaj ili u radno polje upiše neki od simbola azbuke A_t.

- Kad kažemo da TM upisuje simbol $a \in A_t$ onda to podrazumjeva da se se predhodni sadržaj radnog polja izbriše i u to polje upiše simbol a. Dok upis simbola a_0 podrazumjeva brisanje sadržaja radnog polja.
- Postoji mehanizam za pomjeranje trake (ili glave). Ovaj mehanizam nam omogučava da glavu TM dovedemo iznad jednog od dva polja susjedna tekućem radnom polju, tako da to polje postane novo radno polje. U tom slučaju ćemo govoriti da se radno polje (glava) pomjerilo za jedno mjesto ulijevo ili udesno.
- TM ima i operacioni uređaj koji može da se nalazi u jednom od diskretnih stanja $q \in Q$, gdje je $Q = \{q_0, \ldots, q_s\}$ konačan neprazan skup (skup stanja).
- Stanje $q_0 \in Q$ nazivamo početno stanje.

- TM ima i program koji se sastoji od niza komandi. Operacioni uređaj u određenim vremenskim intervalima "izvršava" komande programa. Izvršavanje jedne komande nazivamo korakom TM.
- Jedna komanda programa je četvorka (q, a, v, q'), gdje je $q, q' \in Q$, $a \in A_t \cup \{a_0\}$ i $v \in A_t \cup \{a_0, r, l, s\}$. Ovu četvorku (komandu) ćemo jednostavnije zapisivati sa $qavq'$.
- Komanda $qavq'$ će biti izvršena ako je mašina u stanju q i u radnom polju je upisan simbol a. Simbol v označava radnju koja treba da bude izvršena.
  - Ako je $v \in A_t \cup \{a_0\}$ u radno polje biće upisan simbol v;
  - ako je $v = r$ onda se radno polje (glava) pomjera udesno;
  - ako je $v = l$ onda se radno polje (glava) pomjera ulijevo;
  - ako je $v = s$ onda se mašina zaustavlja.

  Nakon izvršene radnje mašina prelazi u stanje q'.

- Da bi program bio jednoznačan treba da imamo tačno jednu komandu za svaki par $(q, a) \in Q \times (A_t \cup \{a_0\})$. Odnosno program treba da ima $(s+1) \cdot (t+1)$ komandi.
- Takav program možemo zadati tablicom koja ima $(s+1) \cdot (t+1)$ redova i 4 kolone tj. u svakom redu je upisana po jedna komanda koja ima 4 znaka.
- Dogovor je da se komande pišu po leksikografskom poretku tj. prvo idu komande koje počinju sa q_0 pa onda one koje počinju sa q_1,... Takođe, od komandi koje opčinju sa q_i prvo se piše komanda kod koje je na drugom mjestu a_0, pa a_1,...
- Tablicom je jednoznačno određena TM. Iz nje možemo saznati i A_t i Q.
- Napomena. U buduće ćemo koristiti i oznaku $\star$ umjesti praznog simbola a_0.

Primjer. Odredi šta radi TM zadata tablicom:

$$\begin{array}{cccc} q_0 & \star & r & q_1 \\ q_0 & | & r & q_1 \\ q_1 & \star & | & q_2 \\ q_1 & | & | & q_2 \\ q_2 & \star & s & q_2 \\ q_2 & | & s & q_2 \end{array} .$$

Rj. Iz tablice vidimo da je $Q = \{q_0, q_1, q_2\}$ a azbuka $A_1 = \{|\}$.

- Ako je u stanju q_0 mašina (bez obzira šta je upisano u radno polje $\star$ ili $|$) pomijeri glavu desno i pređe u stanje q_1.
- Ako je u stanju q_1 mašina upiše $|$ i pređe u stanje q_2.
- Ako je u stanju q_2 mašina se zaustavi. Stanje u kojem se TM nalazi nakon zaustavljanja nije bitno. Pišemo ga (u komandi) samo radi simetričnosti.

- Kada mašina izvrši komandu oblika $qasq'$ onda dolazi do zaustavljanja mašine. U ovom slučaju kažemo da je u pitanju mašinsko zaustavljanje (MZ). TM ima lampu MZ koja se upali kada je došlo do MZ.
- Ako je glava iznad početnog polja trake ($m = 0$) a izvršava komandu oblika $qalq'$ onda ponovo dolazi do zaustavljanja mašine. U tom slučaju se pali lampa PT da ukaže da je došlo do prelaska preko kraja trake.
- Napomena. Matematički (opisanu) TM možemo zadati kao četvorku $T = (A, Q, q_0, \mathcal{P})$, gdje je A konačan neprazan skup - nazivamo ga azbukom TM, Q konačan neprazan skup - nazivamo ga skupom stanja TM, $q_0 \in Q$ - nazivamo ga početnim stanjem i $\mathcal{P}$ (nazivamo ga programom TM) je preslikavanje
$\mathcal{P} : Q \times (A \cup \{a_0\}) \to (A \cup \{a_0, r, l, s\}) \times Q$.

Konfiguracija i pozicija TM

- Zamislimo da je TM izvršila određen broj koraka a onda smo je zaustavili. Želimo da nastavimo izvršavanje na drugoj TM (sa istom tablicom) od koraka u kome smo zaustavili prvu TM. Koje informacije treba da prenesemo?
- Da bi nastavak rada na drugoj TM bio korektan moramo znati: sadržaj trake, poziciju radnog polja i stanje u kome se mašina nalazi. Kažemo da ove tri informacije određuju konfiguraciju mašine.
- Zapis (na traci) u oznaci $\mathcal{J}$ je funkcija $\mathcal{J} : N_0 \to A \cup \{\star\}$ za koje važi da je skup $\{i \in N_0 : \mathcal{J}(i) \neq \star\}$ konačan.
- Intuitivno $\mathcal{J}(i)$ je simbol koji je upisan na i-tom polju trake.
- Tekuća konfiguracija TM je uređena trojka $C = (m, \mathcal{J}, q)$, gdje je $m \geq 0$ redni broj tekućeg radnog polja, $\mathcal{J}$ tekući zapis i $q \in Q$ tekuće stanje.

- Uočimo da je $\mathcal{J}(m)$ sadržaj tekućeg radnog polja.
- Konfiguracija $C_0 = (m, \mathcal{J}, q_0)$, gdje je q_0 početno stanje se naziva početnom konfiguracijom. U tom slučaju zapis $\mathcal{J}$ predstavlja ulaz i kažemo da se TM primijenjuje na zapis $\mathcal{J}$.
- Kažemo da je $C' = (m', \mathcal{J}', q')$ konfiguracija koja slijedi nakon konfiguracije $C = (m, \mathcal{J}, q)$, u oznaci $C \mapsto C'$, ako se dobija iz C primijenom komande $q\ \mathcal{J}(m)\ v\ q'$, gdje je $v \in A \cup \{a_0, r, l\}$ i nije ($v = l$ i $m = 0$). Tačnije ako je $v = r$ onda je $m' = m + 1$, $\mathcal{J}' = \mathcal{J}$; ako je $v = l$ onda je $m' = m - 1$, $\mathcal{J}' = \mathcal{J}$; ako je $v = a \in A \cup \{a_0\}$ onda je $m' = m$, $\mathcal{J}'(m) = a$ i $\mathcal{J}'(k) = \mathcal{J}(k)$ za svako $k \neq m$.
- Izostavili smo slučaj $v = s$ jer se tada mašina zaustavlja i nema sledeće konfiguracije, kao i slučaj PT tj. nije ($v = l$ i $m = 0$). Za ove slučajeve C nazivamo završnom konfiguracijom mašine.

- Niz konfiguracija $\mathcal{C} = C_0, C_1, C_2, \ldots$ za koje je $C_{i-1} \mapsto C_i$, za $i > 0$ nazivamo izračunavanje koje odgovara početnoj konfiguraciji $C_0 = (m, \mathcal{J}, q_0)$.
- C_i je konfiguracija poslije i tog koraka TM.
- Ako je niz (izračunavanje) $\mathcal{C}$ bezkonačan onda mašina radi vječno. Ako je $\mathcal{C}$ konačan niz i C_j poslednji član toga niza (završna konfiguracija) onda kažemo da se TM zaustavlja poslije j koraka.
- Ponekad nas ne interesuje stanje TM otuda ako je $C = (m, \mathcal{J}, q)$ konfiguracija onda je $S = (m, \mathcal{J})$ pozicija.
- Koristimo oznake $b_0 b_1 \ldots b_m^q \ldots$ - za konfiguraciju i $b_0 b_1 \ldots \underset{\uparrow}{b_m} \ldots$ - za poziciju, gdje je $b_i \in A \cup \{a_0\}$ i $q \in Q$.

- Koristimo i oznake kod zapisa, pozicije, konfiguracije:
  - $\sim$ - dio zapisa, od uzastopnih simbola, koji nas ne interesuje,
  - $-$ - jedan simbol koji nas ne interesuje ili ga ne znamo,
  - $]$ - početak trake,
  - $\star \ldots$ - na kraju zapisa označava da su dalje u zapisu (na traci) sve znaci $\star$.
- Neka su S, S' dvije moguće pozicije za TM T. Koristimo oznaku $S \overset{T}{\Rightarrow} S'$ da bi ukazali da ako je S početna pozicija TM T onda se T zaustavi i S' je njena završna pozicija.
- Primijeniti T na zapis poslije (ispred) riječi ω znači da se kao početna pozicija mašine uzima $] \star \omega \underset{\uparrow}{\star} \ldots$ $(] \underset{\uparrow}{\star} \omega \star \ldots)$.
- Primijeniti T na zapis poslije (ispred) n-torke riječi $\omega_1, \ldots, \omega_n$ znači da se kao početna pozicija mašine uzima $] \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots$ $(] \underset{\uparrow}{\star} \omega_1 \star \cdots \star \omega_n \star \ldots)$.

- Primijeniti T na praznu traku znači da se kao početna pozicija mašine uzima $]\underset{\uparrow}{\star}\dots$
- Kažemo da se mašina T zaustavlja poslije (ispred) riječi ω ako završna pozicija mašine T glasi $]\sim\star\omega\underset{\uparrow}{\star}$ $(]\sim\underset{\uparrow}{\star}\omega\star)$.
- Ako je $]\sim\star\omega\underset{\uparrow}{\star}$ završna pozicija tada je razlog zaustavljanja MZ (jer je $m>0$ - u najgorem slučaju ima bar dvije $\star$ jednu u nultom polju a drugu u polju na poziciji 1 koje je radno polje).
- Primjer. Slici odgovara konfiguracija $(4,\mathcal{J},q)$, gdje je $\mathcal{J}(0)=\star$, $\mathcal{J}(1)=|$, $\mathcal{J}(2)=|$, $\mathcal{J}(3)=|$, $\mathcal{J}(4)=|$, $\mathcal{J}(5)=|$, $\mathcal{J}(6)=\star$, $\mathcal{J}(7)=\star$, ... Dok poziciju možemo zapisati sa $]\star||||\underset{\substack{\uparrow\\ q}}{|}\star\dots$

				$\bigtriangledown$							
$\star$	$\vert$	$\vert$	$\vert$	$\vert$	$\vert$	$\star$	$\star$	$\star$	$\star$	$\star$	$\cdots$
0	1	2	3	4	5	6	7	8	9	10	

Funkcije izračunljive po Tjuringu

- Možemo reći da je algoritam TM (ili Tjuringova tablica). Kao i kod intuitivnog pojma algoritma prvo ćemo TM dodjeliti funkciju, a zatim definisati izrčunljivu funkciju po Tjuringu.
- **Def.** Neka je T TM sa radnom azbukom A i neka je dalje $E \cup B \subset A$ i $n \geq 1$. Pridružimo mašini T funkciju $f : \mathrm{Dom}_f \to \Omega(B)$ ($\mathrm{Dom}_f \subset \Omega^n(E)$) na sledeći način: ako se mašina T primijenjena na zapis poslije n-torke $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(E)$, zaustavlja poslije riječi $\omega \in \Omega(B)$ onda je $\hat{\omega} \in \mathrm{Dom}_f$ i $f(\hat{\omega}) = \omega$, u suprotnom $\hat{\omega} \notin \mathrm{Dom}_f$.
- Očito za $\hat{\omega} \in \mathrm{Dom}_f$ važi:
$$] \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots \quad \overset{T}{\Rightarrow} \;] \sim \star f(\omega_1, \ldots, \omega_n) \underset{\uparrow}{\star}.$$

- Ako TM primijenjena na zapis poslije n-torke $\hat{\omega}$ vječno radi ili pređe preko kraja trake onda $\hat{\omega} \notin \mathrm{Dom}_f$.
- **Def.** Djelimična funkcija $f : \mathrm{Dom}_f \to \Omega(B)$ $(\mathrm{Dom}_f \subset \Omega^n(E))$ naziva se izračunljivom po Tjuringu ako postoji TM T čija je radna azbuka A $(E \cup B \subset A)$ tako da se funkcija $f_T : \mathrm{Dom}_{f_T} \to \Omega(B)$ $(\mathrm{Dom}_{f_T} \subset \Omega^n(E))$ pridružena mašini T poklapa sa f. Za svaku takvu mašinu T kažemo da računa f.
- Uočim da ako smo TM pridružili funkciju nad ulaznom azbukom E i $E' \subset E$ neprazan skup onda toj TM možemo pridružiti i funkciju nad ulaznom azbukom E'.
- Umjesto funkcija izarčunljiva po Tjuringu koristi se i termin izračunljiva funkcija. Naime dokazano je da se skupovi izračunlji funkcija za različite modele poklapaju - to je skup parcijalno rekurzivnih funkcija. Zato ne naglašavamo model.

- **Def.** Djelimična funkcija $f : \mathrm{Dom}_f \to \Omega(B)$ $(\mathrm{Dom}_f \subset \Omega^n(E))$ naziva se normalno izračunljivom po Tjuringu ako postoji TM T koja računa f i još ispunjava sledeća dva uslova:
  a) ako $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(E) \backslash \mathrm{Dom}_f$ onda se mašina T poslije primijene na $\hat{\omega}$ nikad ne zaustavlja;
  b) ako je $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \mathrm{Dom}_f$ onda
  $] \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots \overset{T}{\Rightarrow}] \star \omega_1 \star \cdots \star \omega_n \star f(\omega_1, \ldots, \omega_n) \underset{\uparrow}{\star} \ldots .$
- Ako je funkcija normalno izračunljiva po Tjuringu onda je ona i izračunljiva po Tjuringu. Dokaz je očigledan. Kasnije ćemo dokazati i deugi smjer.
- **Def.** Za skup $S \subset \Omega^n(E)$ kažemo da je rješiv (odlučiv) po Tjuringu u odnosu na skup $\Omega^n(E)$ ako postoji TM T sa svojstvom: ako se mašina T primijeni na zapis poslije n-torke $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(E)$ onda je završna pozicija $] \sim \star \underset{\uparrow}{\star}$ ako $\hat{\omega} \in S$, a $] \sim \star a_1 \underset{\uparrow}{\star}$ ako $\hat{\omega} \notin S$. (Tj. ako je karakteristična funkcija skupa izračunljiva).

Elementarne TM i Tjuringovi dijagrami

Za radnu azbuku $A_t = \{a_1, \ldots, a_t\}$ uvodimo $t+3$ elementarne mašine koje označavamo sa r, l i a_i $i \in \{0, 1, \ldots, t\}$. Tablice tih mašina su:

a_i				r				l			
q_0	a_0	a_i	q_1	q_0	a_0	r	q_1	q_0	a_0	l	q_1
	$\ldots$				$\ldots$				$\ldots$		
q_0	a_t	a_i	q_1	q_0	a_t	r	q_1	q_0	a_t	l	q_1
q_1	a_0	s	q_1	q_1	a_0	s	q_1	q_1	a_0	s	q_1
	$\ldots$				$\ldots$				$\ldots$		
q_1	a_t	s	q_1	q_1	a_t	s	q_1	q_1	a_t	s	q_1

- TM a_i upisuje slovo a_i u radno polje i onda se zaustavi na tom polju. Pri tome ne mijenja druga polja zapisa.
- TM r pomjera radno polje za jedno mjesto udesno i onda se zaustavlja. Pri tome ne mijenja zapis.
- TM l ako je pozicija radnog polja $m > 0$ onda ona pomjera radno polje za jedno mjesto ulijevo a zatim se zaustavlja. Pri tome ne mijenja zapis. Ako je pozicija radnog polja $m = 0$ onda TM l se zaustavlja i pali lampu PT (prelazak preko kraja trake). Pri tome ne mijenja poziciju.

- Zadatak. Napisati tablice za sledeće TM.

a) TM T_1 ima radnu azbuku A_t. Ona pomjeri radno polje za tri mjesta udesno, zatim upiše a_0 u tekuće radno polje i zaustavi se.

b) TM T_2 ima radnu azbuku $A_1 = \{|\}$ i

$$] \sim \star\omega \underset{\uparrow}{\star} \ldots \quad \overset{T_2}{\Rightarrow} \;] \sim \star\omega \star \omega \underset{\uparrow}{\star} \ldots;$$

tj. kopira riječ $\omega \in \Omega(A_1)$.

- Napraviti tablicu za TM T_1 je jednostavan zadatak. Tabela je data niže.
- Opišimo ideju za pravljenje tablice za TM T_2. Ova ideja se koristi i u drugim zadacima. Kopiramo slovo po slovo idući s lijeva u desno. Na polje simbola koji se kopira upisujemo $\star$ da bi znali mijesto povratka (tj. vraćamo se do druge $\star$).

Tablica za T_1. Nakon svakog pomjeranja udesno prelazimo u novo stanje tj. stanje nam pamti broj pomjeranja.

$$
\begin{array}{cccc}
q_0 & a_0 & r & q_1 \\
 & \ldots & & \\
q_0 & a_t & r & q_1 \\
q_1 & a_0 & r & q_2 \\
 & \ldots & & \\
q_1 & a_t & r & q_2 \\
q_2 & a_0 & r & q_3 \\
 & \ldots & & \\
q_2 & a_t & r & q_3 \\
q_3 & a_0 & a_0 & q_4 \\
 & \ldots & & \\
q_3 & a_t & a_0 & q_4 \\
q_4 & a_0 & s & q_4 \\
 & \ldots & & \\
q_4 & a_t & s & q_4
\end{array}
$$

Tablica za T_2.

q_0	$\star$	l	q_1	q_4	$\star$	r	q_5	q_8	$\star$	$\vert$	q_2
q_0	$\vert$	l	q_1	q_4	$\vert$	l	q_1	q_8	$\vert$	l	q_8
q_1	$\star$	$\star$	q_2	q_5	$\star$	r	q_6	q_9	$\star$	s	q_9
q_1	$\vert$	l	q_1	q_5	$\vert$	r	q_5	q_9	$\vert$	r	q_9
q_2	$\star$	r	q_3	q_6	$\star$	$\vert$	q_7				
q_2	$\vert$	r	q_3	q_6	$\vert$	r	q_6				
q_3	$\star$	r	q_9	q_7	$\star$	l	q_8				
q_3	$\vert$	$\star$	q_4	q_7	$\vert$	l	q_7				

- Na osnovu tablice teško je ispratiti rješenje zadatka. Da bi razumjeli ideju zapisati konfiguracije mašine za slučajeve $\omega = |||$, $\omega = |$ i $\omega = \lambda$ (tj. prazna riječ).

Posmatrajmo poziciju $] \sim \star \omega_1 \underset{\uparrow}{|} \omega_2 \star \omega_1 \star \ldots$ gdje je $\omega = \omega_1 | \omega_2$ a ω_1 kopirani dio. Da bi kopirali $|$ iz radnog polja treba:

1. Na mjesto $|$ upisati $\star$ da bi zapamtili poziciju polaznog polja.
2. Ići desno dok ne nađemo drugu $\star$
3. U to polje upišemo $|$ (tj. kopirali smo još jedan simbol)
4. Ići lijevo dok ne nađemo drugu $\star$ (vraćamo se na polje od koga smo pošli)
5. U to polje upišemo $|$ (tj. vraćamo njegov stari sadržaj).

Pozicija nakon koraka 5 je $] \sim \star \omega_1 | \underset{\uparrow}{\omega_2} \star \omega_1 | \star \ldots$ tj. kopiran je još jedan simbol. Treba da se pomjerimo desno za jedno polje da bi nastavili kopiranje sledećeg simbola.

Tjuringov dijagram

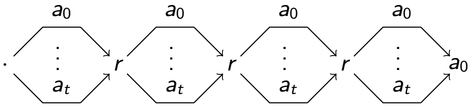
- Iz tablice mašine T_2 teško je razumjeti šta zapravo ona radi. Zato je bilo neophodno da se taj prikaz uprosti "rastavljajući" mašinu na jednostavnije (npr. elementarne) mašine za koje znamo tablicu i šta one rade. U tu svrhu koristimo Tjuringov dijagram.
- Pretpostavljamo bez umanjenja opštosti da sve TM koje se spominju rade nad istom azbukom $A_t = \{a_1, \ldots, a_t\}$. Naime ako neka TM radi nad azbukom $A_p \subset A_t$ onda modifikujemo njenu tablicu tako što za svako njeno stanje q i svaki simbol $a \in A_t \backslash A_p$ dodamo komandu $qasq$. Tj. TM se zaustavi kad naiđe na nepoznat simbol.
- Tjuringov dijagram D definiše jednu TM T. Paralelno sa opisom njegove konstrukcije (sintakse) mi opisujemo i njegovu semantiku (značenje).
- Koristimo skraćenicu TD za Tjuringov dijagram.

Sintaksa i semantika TD

- TD D ima simbol $\cdot$ (tačka) koji se pojavljuje samo jednom. Ona ukazuje gdje počinje rad mašine T.
- TD koristi i simbole $M_1, \dots, M_k$. Svaki od simbola M_i predstavlja jednu TM čija je tablica poznata (nazivamo je komponentom mašine T) i može se u TD D pojavljivati više puta.
- Za svaki simbol $a \in A_t \cup \{a_0\}$ i svaki nacrtan simbol $M \in \{M_1, \dots, M_k\} \cup \{\cdot\}$ iz simbola M polazi najviše jedna strelica označena simbolom a i ta strelica vodi do nekog drugog simbola M' TD D. Značenje je sledeće. *i*) Ako se TM M MZ iznad simbola a (tj. izvršila komandu $qasq$) onda se rad predaje TM M' ako postoji strelica inače dolazi do MZ TM T. *ii*) Ako TM M vječno radi onda će i T vjećno raditi. *iii*) Ako je došlo do PT kod mašine M onda dolazi do PT i kod mašine T.

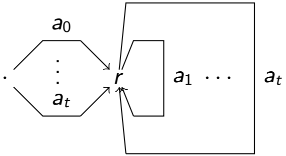
Primjeri TD

- Sledeći TD odgovara TM T_1



- Navedimo još jedan TD. On odgovara TM koju ćemo označavati sa R. TM R pomjera radno polje za jednu riječ udesno tj.

$$] \sim \underset{\uparrow}{-}\omega\star \quad \overset{R}{\Rightarrow} \quad] \sim -\omega\underset{\uparrow}{\star}$$

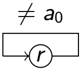


Uprošćavanje TD

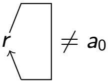
- Ako od nekog simbola M ka simbolu M' vode strelice sa simbolima a_j za svako $j \in \{0, 1, \ldots, t\}$ onda te strelice mogu biti zamijenjene sa jednom strelicom bez simbola ili izostavljene a simboli M i M' se pišu jedan za drugim tj. pišemo MM'.
- Ako od nekog simbola M ka simbolu M' vode strelice sa simbolima a_j za svako $j \in \{0, 1, \ldots, t\} \backslash \{k\}$ onda te strelice mogu biti zamjenjene sa jednom strelicom od M ka M' koja je označena sa $\neq a_k$. Slično ako nedostaju dvije strelice. Itd.
- Ako od tačke $\cdot$ ka simbolu M vode strelice sa simbolima a_j za svako $j \in \{0, 1, \ldots, t\}$ onda tačka i te strelice mogu biti izostavljene a simbol M se označi kao početni tako što se zaokruži ili napiše lijevo od svih drugih simbola.

- Pisaćemo M^n umjesto $\underbrace{M \dots M}_{n\text{-puta}}$.

- Kad uprostimo TD za TM T_1 on postaje: $r^3 a_0$.
- Takođe TD za TM R možemo prikazati sa:



ili sa:



ili na neki sličan način.

Konstrukcija tablice za dati TD

Neka u TD D učestvuju simboli TM $M_1, \ldots, M_k$ čije su tablice poznate. Stim da se isti simbol ne javlja dva puta. Na osnovu ovih tablica pravimo tablicu za TM T koja odgovara TD D.

- Zapišemo sve tablice mašina M_i jedna za drugom stim da izvršimo prenumeraciju stanja tako da se simboli stanja za razne mašine ne preklapaju i da u toj novoj numeraciji ne koristimo q_0.
- Neka je u novoj numeraciji q_{i_M} stanje koje odgovara početnom stanju mašine M. Za svaku strelicu $\cdot \xrightarrow{a} M$ dodaje se u tablicu red $q_0 a a q_{i_M}$. A za strelicu $\cdot \xrightarrow{a} \cdot$ dodaje red $q_0 a a q_0$. Ako takva strelica ne postoji za slovo a dodaje se red $q_0 a s q_0$.
- Ako u TD postoji $M' \xrightarrow{a} M$ onda u tablici svaki red koji odgovara tablici M' oblika $qasq'$ treba prepraviti u red $qaaq_{i_M}$.

Dijagrami važnih (pomoćnih) TM

- Navodimo neke TM koje koristimo pri konstruisanju složenijih TM. Navodimo simbol, opis, radnu i dijagram tih TM.
- TM R–idi udesno za jednu riječ: $] \sim \underset{\uparrow}{\star}\omega\star \overset{R}{\Rightarrow}] \sim \star\omega\underset{\uparrow}{\star}$ Dijagram:

  $\to r \circlearrowleft_{\neq \star}$

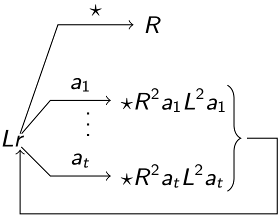
- TM L–idi ulijevo za jednu riječ: $] \sim \star\omega\underset{\uparrow}{\star} \overset{L}{\Rightarrow}] \sim \underset{\uparrow}{\star}\omega\star$ Dijagram:

  $\to l \circlearrowleft_{\neq \star}$

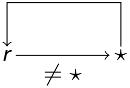
- TM $\mathcal{R}$–idi udesno za niz riječi $X = \omega_1 \star \omega_2 \star \cdots \star \omega_n$:
  $] \sim \underset{\uparrow}{\star}X\star\star \overset{\mathcal{R}}{\Rightarrow}] \sim \star X\underset{\uparrow}{\star}\star$ Dijagram: $\to Rr \circlearrowleft_{\neq \star} \xrightarrow[\star]{} l$
- TM $\mathcal{L}$–idi ulijevo za niz riječi $X = \omega_1 \star \omega_2 \star \cdots \star \omega_n$:
  $] \sim \star\star X\underset{\uparrow}{\star} \overset{\mathcal{L}}{\Rightarrow}] \sim \star\underset{\uparrow}{\star}X\star$ Dijagram: $\to Ll \circlearrowleft_{\neq \star} \xrightarrow[\star]{} r$

- TM K – kopira jednu riječ: $] \sim \star\omega \underset{\uparrow}{\star} \ldots \overset{K}{\Rightarrow}] \sim \star\omega\star\omega \underset{\uparrow}{\star} \ldots$

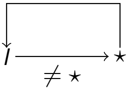
Dijagram:



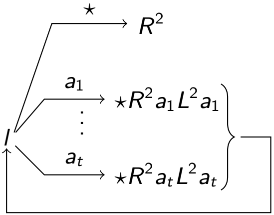
- TM W_r – briše desnu riječ: $] \sim \underset{\uparrow}{\star}\omega\star \overset{W_r}{\Rightarrow}] \sim \star\cdots\underset{\uparrow}{\star}$ Dijagram:



- TM W_l – briše lijevu riječ: $] \sim \star\omega\underset{\uparrow}{\star} \overset{W_l}{\Rightarrow}] \sim \underset{\uparrow}{\star}\cdots\star$ Dijagram:

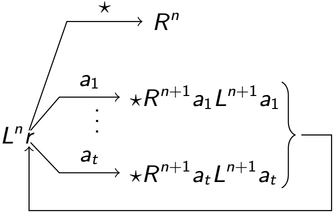


- TM $I-$stvara inverznu riječ tj. ω^{-1} koja se dobija kada se slova riječi ω zapišu u obrnutom redoslijedu (npr. za $\omega = abcd$ imamo $\omega^{-1} = dcba$): $] \sim \star\omega \underset{\uparrow}{\star} \ldots \overset{I}{\Rightarrow}] \sim \star\omega\star\omega^{-1} \underset{\uparrow}{\star} \ldots$ Dijagram:



- TM K_n-kopira prvu riječ od n $(n \geq 1)$ riječi:
$] \sim \star\omega_1\star\omega_2\star\cdots\star\omega_n \underset{\uparrow}{\star} \ldots \overset{K_n}{\Rightarrow}] \sim \star\omega_1\star\omega_2\star\cdots\star\omega_n\star\omega_1 \underset{\uparrow}{\star} \ldots$
Za $n = 1$ dobijamo mašinu K tj. $K_1 = K$.

Dijagram:

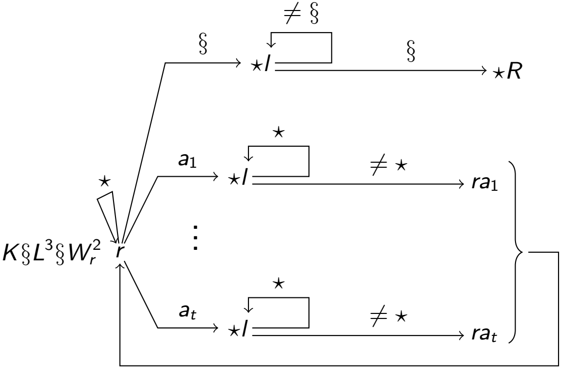


- TM $V-$isjeca jednu riječ: $] \sim \star\omega_1 \star \omega_2 \underset{\uparrow}{\star} \dots \overset{V}{\Rightarrow} \;] \sim \star\omega_2 \underset{\uparrow}{\star} \dots$

  Iz početne i završne pozicije vidimo da TM V primjenjujemo na zapis od dvije riječi i da prva riječ nestaje a drugu zadržavamo. Na primjer:
  $] \star a_3 a_5 a_1 a_2 \star a_4 a_1 a_2 \underset{\uparrow}{\star} \dots \overset{V}{\Rightarrow} \;] \star a_4 a_1 a_2 \underset{\uparrow}{\star} \dots$

  Ulazna azbuka mašine V je , kao i za prethodne TM,
  $A_t = \{a_1, \dots, a_t\}$. Međutim njena radna azbuka je
  $A_{t+1} = \{a_1, \dots, a_t, a_{t+1}\}$. Pomoćno slovo a_{t+1} ćemo označavati sa $\S$.

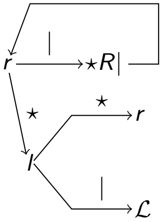
Dijagram:



- Vidimo da se pomoću $\S$ označava početak i kraj područja koje nas interesuje.
- V kopira drugu riječ i izbriše rijči koje su date. Zatim kreće sa premještanjem kopije druge riječi.

- Sledeće mašine rade nad azbukom $A_1 = \{|\}$ a sa X označavamo $X = \omega_1 \star \omega_2 \star \cdots \star \omega_n$ za $n \geq 0$, gdje je $|\omega_i| \geq 1$. Očigledno X je prazna riječ za $n = 0$.

- TM T_r−translira udesno za jedno mjesto:
$$] \sim \underset{\uparrow}{\star} X \star \ldots \quad \overset{T_r}{\Rightarrow} \quad] \sim \star \underset{\uparrow}{\star} X \star \ldots$$

Dijagram: 

- TM T_l − translira ulijevo za jedno mjesto:
  $$]\sim||\underset{\uparrow}{\star}X\star\ldots \overset{T_l}{\Rightarrow}]\sim|\underset{\uparrow}{\star}X\star\ldots$$

  Dijagram:
  $$\begin{tikzcd}
  l\star r\mathcal{R} & l \arrow[r, "|"] \arrow[d, "\star"'] & \star L| \arrow[l, bend right] \\
  & l \arrow[r, "|"] & r
  \end{tikzcd}$$

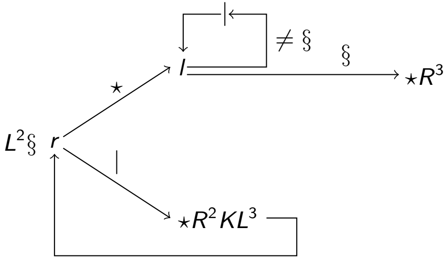
- Za prethodno navedene mašine treba znati konstruisati TD. Takođe treba znati njihovu funkciju i one mogu da se koriste pri konstrukciji složenijih TD.
- U nastavku navodimo još TD mašine za množenje dva broja.

Mašina za množenje dva broja

- Interesuje nas da računamo funkciju $f : N_0 \times N_0 \to N_0$ definisanu sa $f(n,m) = n \cdot m$.
- Neka koristimo azbuku $A_1 = \{|\}$ (tj. unarni brojni sistem). Tada je željena funkcija u stvari $f : \Omega(A_1) \times \Omega(A_1) \to \Omega(A_1)$ i $f(\omega_1, \omega_2) = \omega_3$ gdje je $|\omega_3| = |\omega_1| \cdot |\omega_2|$. Npr. $f(||,|||) = \underbrace{||||||}_{\text{6-puta}}$.
- Podsjetimo se TM K koja kopira jednu riječ:
$$] \sim \star\omega \underset{\uparrow}{\star} \ldots \overset{K}{\Rightarrow}] \sim \star\omega \star \omega \underset{\uparrow}{\star} \ldots.$$
Šta ako K primjenimo na poziciju $] \sim \star\omega \underset{\uparrow}{\star} \overline{\omega} \star \ldots$? Dobijamo:
$$] \sim \star\omega \underset{\uparrow}{\star} \overline{\omega} \star \ldots \overset{K}{\Rightarrow}] \sim \star\omega \star \overline{\omega}\omega \underset{\uparrow}{\star} \ldots.$$
- Dakle, ako već postoji neka riječ $\overline{\omega}$ iza riječi ω koja se kopira onda se kopija riječi ω nadovezuje na postojeću riječ $\overline{\omega}$.

- Ideja: Treba da na rezultat nadovežemo drugu riječ (koristeći TM K) još onoliko puta kolika je preostala dužina prve riječi.
- TM $P-$nalazi proizvod unarno zapisanih brojeva:

$$] \sim \star\omega_1 \star \omega_2 \underset{\uparrow}{\star} \ldots \overset{P}{\Rightarrow}] \sim \star\omega_1 \star \omega_2 \star \underbrace{\omega_2 \cdots \omega_2}_{|\omega_1|\text{-puta}} \underset{\uparrow}{\star} \ldots$$

Dijagram: 

TD od elementarnih mašina

- **Def.** Neka TM M i M' imaju istu radnu azbuku A_t. Kažemo da TM M' modelira (u jakom smislu) TM M ako za datu početnu poziciju ispunjavaju:
  1. dolazi do MZ (odnosno PT) mašine M akko dolazi do MZ (odnosno PT) mašine M'. U oba slučaja završne pozicije mašina M i M' se poklapaju.
  2. Niz tekućih pozicija mašine M jeste podniz niza tekućih pozicija mašine M'.
- **Teorema**. Neka je TM M nad azbukom A_t definisana tablicom T. Toj tablici može se na efikasan način (tj. algoritamski) pridružiti dijagram D sastavljen od simbola $r, l, a_0, \ldots, a_t$ i tačke, takav da TM M' definisana tim dijagramom modelira mašinu M.
- **Dokaz**. TD D se sastoji od tačke i od po jednog simbola mašine v (s imenom $v_{qaq'}$) za svaki red tablice T oblika $qavq'$ gdje je $v \neq s$.

Za svaki simbol v s imenom $v_{q_0 a_j *}$ povucimo strelicu označenu sa a_j od tačke ka tom simbolu v.

Od svakog simbola v' s imenom v_{**q} povucimo strelicu označenu sa a_j do svakog simbola v s imenom v_{qa_j*}.[1]

Ostaje da se provjeri da TD D konstruisan po prethodno navedenim pravilima ispunjava uslove 1 i 2 definicije.

- Primjer.

$$\underbrace{\begin{array}{l} q_0\, a_3\, r\, q_2 \\ \dots \\ q_4\, a_1\, l\, q_3 \\ \dots \\ q_3\, a_4\, a_1\, q_7 \\ \dots \end{array}}_{\text{tablica } T} \quad \mapsto \quad \underbrace{\begin{array}{l} \cdot \xrightarrow{a_3} r \\ \\ \\ l \xrightarrow{a_4} a_1 \\ \\ \end{array}}_{\text{dijagram } D}$$

[1]Koristimo $*$ kao zamjenu za neki od mogućih simbola.

Funkcija $\gamma_{t,s}$

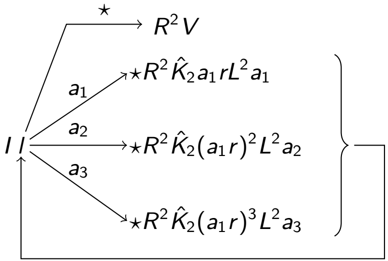
- Zelimo da nađemo funkciju $\gamma_{t,s} \ : \ \Omega(A_t) \to \Omega(A_s)$ koja zadovoljava uslove:
  1. Funkcija $\gamma_{t,s}$ je bijekcija (tj. "1-1" i "na"; $\mathrm{Dom}(\gamma_{t,s}) = \Omega(A_t)$).
  2. Funkcije $\gamma_{t,s}$ i $\gamma_{t,s}^{-1}$ su izračunljive po Tjuringu.
- Prvo konstruišemo funkciju $\gamma_t = \gamma_{t,1}$, pri tome $\Omega(A_1)$ poistovećujemo sa $\mathbb{N}_0$ (pa koristimo $+, \cdot$).

$$\gamma_t(\lambda) = 0 \tag{1}$$

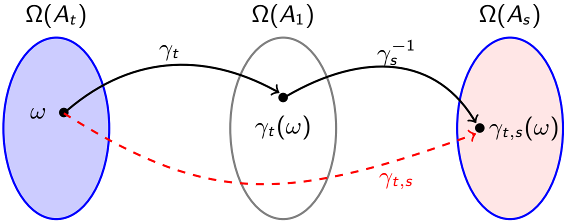
$$\gamma_t(\omega a_i) = \quad t \cdot \gamma_t(\omega) + i, \text{ za } 1 \le i \le t, \ \omega \in \Omega(A_t) \tag{2}$$

- Primjer. 1^o) $\gamma_t(a_2 a_5 a_3 a_1) = 2t^3 + 5t^2 + 3t + 1$ tj. slično kao brojni sistem sa osnovom t samo što nemamo cifru 0.
  2^o) Za $t = 3$ imamo $\gamma_3(a_1) = 1$, $\gamma_3(a_2) = 2$, $\gamma_3(a_3) = 3$,
  $\gamma_3(a_1 a_1) = 3 + 1 = 4$, $\gamma_3(a_1 a_2) = 5$, $\gamma_3(a_1 a_3) = 6$,
  $\gamma_3(a_2 a_1) = 2 \cdot 3 + 1 = 7$, $\gamma_3(a_2 a_2) = 8$, $\ldots$.
- Svaki simbol je bitan $a_1 a_3 \neq a_3$; dok broj $02 = 2$.

- Zadatak. Dokazati da je γ_t bijekcija.
- $\gamma_t^{-1}(0) = \lambda$; $\gamma_t^{-1}(n+1) = \gamma_t^{-1}(\left\lfloor \frac{n}{t} \right\rfloor)a_i$, gdje je $n \in \mathbb{N}_0$ a $i = (n \bmod t) + 1$.
- Zadatak. Dokazati da su funkcije γ_t i γ_t^{-1} izračunljive.
- Zadatak. Provjeriti da li dati TM računa γ_3. Gdje je TD za $\hat{K}_m$: $(KL)^{m-1}Kl \xrightarrow{a_1} \star L a_1 R$.



- Sada je: $\gamma_{t,s}(\omega) = \gamma_s^{-1}(\gamma_t(\omega))$, za $\omega \in \Omega(A_t)$ i $\gamma_{t,s}^{-1}(\omega) = \gamma_t^{-1}(\gamma_s(\omega))$, za $\omega \in \Omega(A_s)$.
- Slika za $\gamma_{t,s}$:



- Vidimo da je $\gamma_{t,s}^{-1} = \gamma_{s,t}$.
- Zadatak. Dokazati da je $\gamma_{t,s}$ bijekcija i da su $\gamma_{t,s}$ i $\gamma_{t,s}^{-1}$ izračunljive funkcije (po Tjuringu).

Funkcija σ_n^t

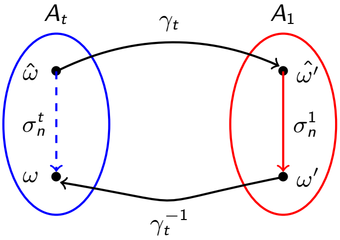
- I za funkciju $\sigma_n^t : \Omega^n(A_t) \to \Omega(A_t)$ zahtjevamo da zadovoljava uslove:
  1. Funkcija σ_n^t je bijekcija.
  2. Funkcije σ_n^t i $(\sigma_n^t)^{-1}$ su izračunljive po Tjuringu.
- Funkcija σ_n^t kodira $n-$torku riječi $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(A_t)$ s jednom riječi $\omega \in \Omega(A_t)$.
- Pisaćemo $(\sigma_n^t)^{-1} = (\sigma_{n,1}^t, \ldots, \sigma_{n,n}^t)$, gdje je $\sigma_{n,k}^t$ $k-$ta komponenta funkcije $(\sigma_n^t)^{-1}$.
- Slučaj $t > 1$ svodimo na slučaj $t = 1$, jer važi:

$$\sigma_n^t(\omega_1, \ldots, \omega_n) = \gamma_t^{-1}(\sigma_n^1(\gamma_t(\omega_1), \ldots, \gamma_t(\omega_n))) \tag{3}$$

$$\sigma_{n,i}^t(\omega) = \gamma_t^{-1}(\sigma_{n,i}^1(\gamma_t(\omega))) \tag{4}$$

- Koristimo oznake σ_n i $\sigma_{n,i}$ redom umjesto oznaka σ_n^1 i $\sigma_{n,i}^t$.

- Slika za σ_n^t:



- Takođe slučaj $n > 2$ svodimo na slučaj $n = 2$ sa:

$$\sigma_n(\omega_1, \ldots, \omega_n) = \sigma_2(\sigma_{n-1}((\omega_1, \ldots, \omega_{n-1}), \omega_n) \tag{5}$$

$$\sigma_{n,i}(\omega) = \sigma_{n-1,i}(\sigma_{2,1}(\omega)),\ i < n; \quad \sigma_{n,n}(\omega) = \sigma_{2,2}(\omega). \tag{6}$$

- Da bi definisali $\sigma_2 : \Omega^2(A_1) \to \Omega(A_1)$ prvo ćemo definisati pomoćnu funkciju $\sigma : \Omega^2(A_1) \to \Omega(A_2)$ koja je takođe bijekcija.

- Ijeja je da umjesto ”,” stavimo novo slovo a_2 tj. $(\omega_1, \omega_2) \in \Omega^2(A_1)$ kodiramo sa $\omega_1 a_2 \omega_2 \in \Omega(A_2)$. Problem je što na ovaj način nećemo dobiti sve riječi iz $\Omega(A_2)$. Da bi to postigli kodira se i $\omega_2 \in \Omega(A_1)$ sa $\gamma_2^{-1}(\omega_2) \in \Omega(A_2)$.
- Tačnije (kod ω_1 ”brišemo” jedan | da bi dobili riječi koje počinju sa a_2):

$$\sigma(\omega_1, \omega_2) = \begin{cases} \omega_2 & , \quad \text{ako je } \omega_1 = \lambda \\ (\omega_1 - 1) a_2 \gamma_2^{-1}(\omega_2) & , \quad \text{ako je } \omega_1 \neq \lambda \end{cases}$$

- Onda je $\sigma_2(\omega_1, \omega_2) = \gamma_2(\sigma(\omega_1, \omega_2))$.
- Zadatak. Dokazati da je σ_2 bijekcija i napisati formule za $\sigma_{2,1}$ i $\sigma_{2,2}$. Zatim dokazati da su funkcije σ_2, $\sigma_{2,1}$ i $\sigma_{2,2}$ izračunljive.

- σ_2 smo mogli konstruisati i na druge načine npr. redom numeracijom elemenata na dijagonalama kao što je to prikazano na slici.

$$
\begin{array}{cccccc}
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & \dots \\
(1,0) & (1,1) & (1,2) & (1,3) & (1,4) & \dots \\
(2,0) & (2,1) & (2,2) & (2,3) & (2,4) & \dots \\
(3,0) & (3,1) & (3,2) & (3,3) & (3,4) & \dots \\
(4,0) & (4,1) & (4,2) & (4,3) & (4,4) & \dots \\
 & & \vdots & & &
\end{array}
$$

- Uočimo da je zbir brojeva para na istoj dijagonali konstantan i za jedan manji od broja dijagonale, odnosno od broja elemenata na toj dijagonali.
- Tako možemo pisati $\sigma_2(m, n) = \dfrac{(m+n) \cdot (m+n+1)}{2} + n + 1.$

Normalna izračunljivost po Tjuringu

- Želimo da dokažemo da se skup normalno izračunljivih funkcija po Tjuringu poklata sa skupom izračunljivih funkcija po Tjuringu. Da bi to pokazali treba nam niz pripremnih koraka.

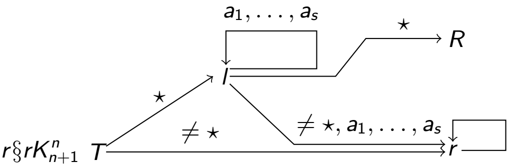
a) Konstrukcija TM na osnovu date TM T koja zadovoljava prvi uslov normalne izračunljivosti. Takvu TM označavamo sa $T^{\S}$.

b) Konastukcija mašine M' koja obavlja isti zadatak kao mašina M a koja ima radnu azbuku A_1. Ovo zahtjeva: 1^o) Konstrukciju TD od elementarnih mašina; 2^o) "modeliranje" TM l; 3^o) "modeliranje" TM r; 4^o) "modeliranje" TM a_i; 5^o) "modeliranje" strelica (veza).

c) Konstrukcija mašine koja zadovoljava i drugi uslov normalne izračunljivosti.

Mašina $T^{\S}$

- Neka je $f : \mathrm{Dom}_f \to \Omega(B)$ ($\mathrm{Dom}_f \subset \Omega^n(E)$) izračunljiva funkcija i T TM koja računa f s radnom azbukom A_t ($E \cup B \subset A_t$ tj. $B = A_s, s \leq t$).
- TD mašine $T^{\S}$ koja nastaje od TM T dat je na slici.

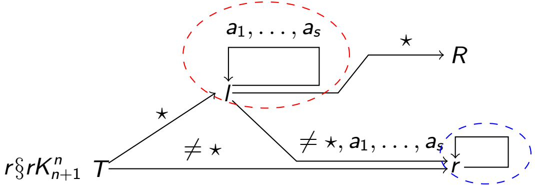


- **Tvrđenje**. Neka je $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(E)$ i neka se TM $T^{\S}$ primjeni na poziciju $]\star\omega_1\star\omega_2\star\cdots\star\omega_n\underset{\uparrow}{\star}\ldots$. Tada
  a) ako $\hat{\omega} \notin \mathrm{Dom}(f)$ onda $T^{\S}$ vječno radi i
  b) ako $\hat{\omega} \in \mathrm{Dom}(f)$ onda će se $T^{\S}$ zaustaviti i to poslije riječi $\omega = f(\hat{\omega}) \in \Omega(B)$, tj. $] \sim \star\omega\underset{\uparrow}{\star}$ je završna pozicija.

- Nakon djelovanja komponente $r\S rK^n_{n+1}$ dobijamo poziciju $]\star\omega_1\star\omega_2\star\cdots\star\omega_n\star\S\star\omega_1\star\omega_2\star\cdots\star\omega_n\underset{\uparrow}{\star}\ldots.$

  Na nju djeluje komponenta T.

- Ako T vječno radi onda i $T^\S$ vječno radi. Ako kod T dođe do PT onda ta komponenta u $T^\S$ se zaustavlja "na slovu" $\S$ i rad se predaje komponenti (zaokružena plavom bojom) u $T^\S$ koja vječno radi. Slično je ako se T zaustavi na slovu $\neq\star$.



- Ako se T zaustavlja na poziciji $]\sim\star\omega\underset{\uparrow}{\star}$ onda se rad predaje komponenti (zaokružena crveno) u $T^\S$ koja provjerava da li je $\omega\in B=A_s$. Ako nije nastavlja komponenta koja vječno radi.

Postavka zadatka o modeliranju nad azbukom A_1

- Na osnovu TM $\mathfrak{M}$ koja radi nad azbukom A_t konstruisaćemo TM $\overline{\mathfrak{M}}$ koja radi nad azbukom A_1 i koja "oponaša" TM $\mathfrak{M}$.
- Za riječ $\omega \in \Omega(A_t)$ kažemo još i da je svojstvena riječ, dok za $\overline{\omega} \in \Omega(A_t \cup \{a_0\})$ kažemo da je nesvojstvena riječ. Drugim riječima nesvojstvena riječ u sebi može da sadrži i prazne simbole.
- Ako znamo da su sva polja na traci nakon polja k prazna onda takvom zapisu možemo pridružiti nesvojstvenu riječ $\overline{\omega}$ koja sadrži sve simbole upisane na traci zaključno sa poljem k.
- Na osnovu $\overline{\omega}$ jednoznačno možemo rekonstruisati zapis. Dok jednom zapisu možemo pridružiti više nesvojstvenih riječi (koje se razlikuju samo u broju praznih simbola na kraju riječi).
- Ako nas interesuje kodiranje pozicije pomoću nesvojstvene riječi onda moramo obuhvatiti i radno polje.

Kodiranje u azbuci A_1

- Želimo da poziciju TM zapisanu u azbuci A_t, $t > 1$ kodiramo sa pozicijom koja je zapisana u azbuci A_1.
- Nesvojstvenoj riječi $\overline{\omega} \in \Omega(A_t \cup \{a_0\})$ možemo pridružiti (jednoznačno) nesvojstvenu riječ $\overline{\omega}_1 \in \Omega(\{\star, |\,\})$ tako što svakom slovu a_j pridružimo niz slova $\star\underbrace{|\cdots|}_{j+1}$. Npr. ako je $\overline{\omega} = a_0a_3a_1a_0a_5a_0$ onda je $\overline{\omega}_1 = \star| \star |||| \star || \star | \star |||||| \star |$.
- Kažemo da je nesvojstvena riječ od $\star$ i $j+1$ simbol $|$ kod za simbol a_j.
- Za zapis $\mathcal{J}$ uvedimo funkciju $I_{\mathcal{J}} : N_0 \to \{0, 1, \ldots, t\}$ određenu sa $I_{\mathcal{J}}(j) = k$ akko $\mathcal{J}(j) = a_k$. Tj. $I_{\mathcal{J}}(j)$ pamti indeks slova upisanog u polje j.
- Za funkciju $I_{\mathcal{J}}$ i broj j sa $\hat{j}_{\mathcal{J}}$ označavamo broj $\hat{j}_{\mathcal{J}} = \sum_{i=0}^{j-1}(I_{\mathcal{J}}(i) + 2)$. Broj $\hat{j}_{\mathcal{J}}$ nam u stvari kaže od kog polja bi počeli da upisujemo kod za slovo $\mathcal{J}(j)$ u azbuci $\{\star, |\,\}$.

- Za poziciju $S = (\mathcal{J}, m)$ definišimo broj $k(S) = \max\big\{\{m\} \cup \{i : \mathcal{J}(i) \neq \star\}\big\}$.
- Za polja s indeksom $j \leq k(S)$ kažemo da su upotrebljena, a $k(S)$ je najveći indeks upotrebljenog polja.
- Poziciji $S = (\mathcal{J}, m)$ u azbuci A_t, $t > 1$ pridružimo poziciju $\overline{S} = (\overline{\mathcal{J}}, \overline{m})$ u azbuci A_1, gdje je $\overline{m} = \hat{m}_{\mathcal{J}}$ a za $k = K(S)$

$$\overline{\mathcal{J}}(i) = \begin{cases} | & , \quad \text{za } i \leq \hat{k}_{\mathcal{J}} + l_{\mathcal{J}}(k) + 1 \text{ i } (\forall j \leq k) i \neq \hat{j}_{\mathcal{J}} \\ \star & , \quad \text{inače} \end{cases}$$

- Za $\overline{S}$ kažemo da je kod pozicije S.
- Nova radno polje je na početku koda (tj. na $\star$) slova koje je bilo upisano u staro radno polje.
- Primjer. Poziciji $S =] \star a_3 a_2 \star \underset{\uparrow}{\star} \ldots$ pridružujemo poziciju

  $\overline{S} =] \star | \star |||| \star ||| \star | \underset{\uparrow}{\star} | \star \ldots$

  Vidimo da je: $k = k(S) = m = 4$; $l_{\mathcal{J}}(0) = 0$, $l_{\mathcal{J}}(1) = 3$, $l_{\mathcal{J}}(2) = 2$, $l_{\mathcal{J}}(i) = 0$ za $i > 2$; $\overline{m} = \hat{m}_{\mathcal{J}} = \hat{k}_{\mathcal{J}} = 13$

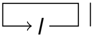
Modeliranje u slabom smislu

- **Def.** Neka TM M i $\overline{M}$ imaju redom radne azbuke A_t i A_1. Kažemo da TM $\overline{M}$ modelira (u slabom smislu) TM M ako za početnu poziciju S mašine M mašina $\overline{M}$ dobija početnu poziciju $\overline{S}$, kod pozicije S, i pri tome zadovoljava sledeće uslove:
  1. dolazi do MZ (odnosno PT) mašine M akko dolazi do MZ (odnosno PT) mašine M'. U oba slučaja završna pozicija mašine M je S_e a završna pozicija mašine $\overline{M}$ je njen kod $\overline{S_e}$.
  2. Ako je $S_0, S_1, S_2, \ldots$ (gdje je $S_0 = S$) niz tekućih pozicija mašine M onda je niz, nihovih kodova, $\overline{S}_0, \overline{S}_1, \overline{S}_2, \ldots$ podniz niza tekućih pozicija mašine $\overline{M}$.

- Sada ćemo za datu TM M konstruisati mašinu $\overline{M}$ koja modelira mašinu M u slabom smislu.
- Kao što je prethodno rečeno ovo zahtjeva pet koraka: konstrukciju TD od elementarnih mašina, modeliranje TM l, r i a_i; "modeliranje" strelica (veza).

Modeliranje nad azbukom A_1

(1) Konstruišemo TD od elementarnih mašina TM M' koja u jakom smislu modelira TM M. Ovaj korak je ranije opisan.

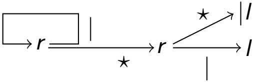
(2) Svaki simbol l u TD zamjenimo simbolom TM $\bar{l}$ čiji je TD:  .

Ova mašina radnju $] \sim a_i \underset{\uparrow}{a_j} \sim \overset{l}{\Rightarrow}] \sim \underset{\uparrow}{a_i} a_j \sim$ TM l modelira radnjom

$$] \sim \star \underbrace{|\cdots|}_{i+1} \underset{\uparrow}{\star} \underbrace{|\cdots|}_{j+1} \star \sim \overset{\bar{l}}{\Rightarrow}] \sim \underset{\uparrow}{\star} \underbrace{|\cdots|}_{i+1} \star \underbrace{|\cdots|}_{j+1} \star \sim .$$

Dok ako se primjeni na poziciju $]\underset{\uparrow}{\star} \sim$ dolazi do PT.

(3) Svaki simbol r u TD zamjenimo simbolom TM $\bar{r}$ čiji je TD:

 .

Mašina $\bar{r}$ radnju $] \sim \underset{\uparrow}{a_i} a_j \sim \overset{r}{\Rightarrow}] \sim a_i \underset{\uparrow}{a_j} \sim$ modelira sa

$$] \sim \underset{\uparrow}{\star} \underbrace{| \cdots |}_{i+1} \star \underbrace{| \cdots |}_{j+1} \star \sim \;\overset{\bar{r}}{\Rightarrow}\;] \sim \star \underbrace{| \cdots |}_{i+1} \underset{\uparrow}{\star} \underbrace{| \cdots |}_{j+1} \star \sim .$$

Dok $] \sim \underset{\uparrow}{a_i} \star \ldots \;\overset{r}{\Rightarrow}\;] \sim a_i \underset{\uparrow}{\star} \ldots$ modelira sa

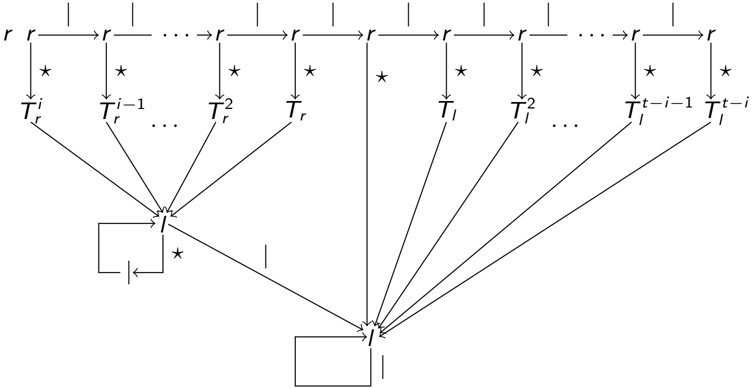
$$] \sim \underset{\uparrow}{\star} \underbrace{| \cdots |}_{i+1} \star \ldots \;\overset{\bar{r}}{\Rightarrow}\;] \sim \star \underbrace{| \cdots |}_{i+1} \underset{\uparrow}{\star} | \star \ldots .$$

(4) Svaki simbol a_i u TD zamjenimo simbolom TM $\overline{a}_i$ čiji je TD dat niže. Mašina $\overline{a}_i$ bi trebala da "upiše" kod za a_i (tj. $\star| \cdots |$ gdje se $|$ javlja $i+1$ put) na mjesto koda postojećeg simbola (to je opet $\star| \cdots |$ gdje broj je simbola $|$ jednak $j+1$ i može biti različit od $i+1$).

Znači ako je $i = j$ ne moramo mjenjati sadržaj trake.

Ako je $i > j$ onda na traku treba da umetnemo $i - j$ simbola $|$.

Ako je $i < j$ onda sa trake treba da isječemo $j - i$ simbola $|$.



Pomerimo se udesno na prvi štap. Zatim izvršimo još jedno pomjeranje udesno.

Ako je tu $\star$ (to znači da je $j = 0$) onda moramo izvršiti $i-$ pomjeranja udesno (mašina T_r^i) kako bi stvorili prostor da upišemo i štapova.

Ako je tu $|$ onda vršimo još jedno pomjeranje udesno... Znači pomjeramo se udesno dok god je $|$.

- Ako smo prešli j štapova, $j < i$ i naišli na $\star$ onda to znači da moramo izvršiti translaciju udesno za $i - j$ (mašina T_r^{i-j}) mjesta kako bi tu upisali $i - j$ štapova. Nakon translacije udesno vraćamo se ulijevo i umjesto stvorenih $\star$ upisujemo $|$. Nakon toga se vraćamo ulijevo dok ne dođemo na $\star$ od koje smo krenuli.

$$] \sim \star|\underbrace{|\cdots|}_{j}\underset{\uparrow}{\star} \sim \overset{T_r^{i-j}}{\Rightarrow}] \sim \star|\underbrace{|\cdots|}_{j}\underbrace{\star\cdots\star}_{i-j}\underset{\uparrow}{\star} \sim$$

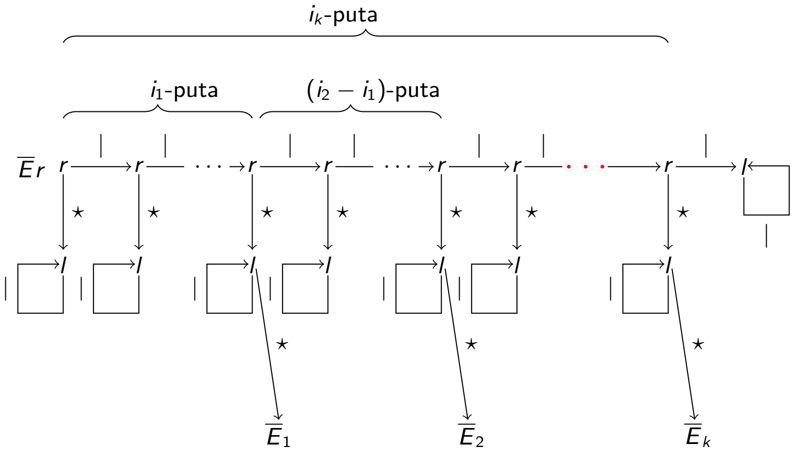
- Ako smo prešli j štapova, $j = i$ i naišli na $\star$ onda nakon toga se vraćamo ulijevo dok ne dođemo na $\star$ od koje smo krenuli.
- Ako smo prešli j štapova, $j > i$ i naišli na $\star$ onda to znači da moramo izvršiti translaciju ulijevo za $j - i$ mjesta (mašina T_l^{j-i}) kako bi izbrisali višak štapova. Nakon toga se vraćamo ulijevo dok ne dođemo na $\star$ od koje smo krenuli

(5) U prethodnim koracima svaki simbol elementarne mašine E zamjenili smo sa simbolom $\overline{E}$ a tačka je ostala tačka tj. $\bar{\cdot} = \cdot$.

Sada i sve stare veze između simbola menjamo na odgovarajući način.

Neka u starom diagramu od simbola E vodi tačno k strelica ka simbolima E_1,...,E_k. Neka je strelica od E ka E_J označena sa a_{i_j}, $j = 1, \ldots, k$. Pretpostavljamo da je numeracija takva da važi $i_1 < i_2 < \cdots < i_k$ (inače bi izvršili prenumeraciju).
U novom dijagramu simbol $\overline{E}$ povezujemo sa simbolima $\overline{E}_1$,..., $\overline{E}_k$ kao što je to prikazano u dijagramu niže.

Simbol a_{i_1} kodiramo sa $\star| \cdots |$ gdje se $|$ javlja $i_1 + 1$ puta. Zato ako iza $\star$ stoji tačno $i_1 + 1$ simbola $|$ onda treba dalji rad predati mašini $\overline{E}_1$. Naravno prije predaje, kao i prije zaustavljanja, glava treba da se vrati (ulijevo) na polaznu $\star$.



i_k-puta
i_1-puta
$(i_2 - i_1)$-puta
$\overline{E}r$
$\overline{E}_1$
$\overline{E}_2$
$\overline{E}_k$

Teorema o normalnoj izračunljivosti

- Neka je $f : \Omega(A_{t_1}) \to \Omega(A_{t_2})$ izračunljiva funkcija. Kažemo da TM T koja računa f ne koristi dodatna slova ako je njena radna azbuka A_t gdje je $t = \max\{t_1, t_2\}$.
- **Teorema**. (o normalnoj izračunljivosti) Ako je funkcija izračunljiva po Tjuringu onda je ona i normalno izračunlijva po Tjuringu. Pri tome postoji TM T_2 koja normalno računa f i ne koristi dodatna slova.
- **Dokaz**. Neka TM T računa funkciju f. Od ranije znamo da će tada mašina $T^{\S}$ računati f i zadovoljavati prvi uslov normalne izračunljivosti. Ako uzmemo mašinu $\overline{T}^{\S}$ onda ona neće koristiti dodatna slova.

  Da bi mogli da primjenimo mašinu $\overline{T}^{\S}$ koja radi nad minimalnom azbukom A_1 treba prvo da kodiramo ulaz.

Neka mašina V_n stvara kod ulaza (ne mjenjajući ulaz). Tj.

$$]\star\omega_1\star\omega_2\star\cdots\star\omega_n\underset{\uparrow}{\star}\ldots \quad \overset{V_n}{\Rightarrow}$$

$$]\star\omega_1\star\omega_2\star\cdots\star\omega_n\star\overline{\star\omega_1\star\omega_2\star\cdots\star\omega_n\underset{\uparrow}{\star}\ldots} \quad .$$

Takođe treba dekodirati izlaz (rezultat). Neka to radi mašina E. Mašina E treba da isječe i eventualne nizova X_1 i X_2 simbola $\star$, | koji se mogu pojaviti prije i nakon rezultata. Tj.

$$]\star\omega_1\star\omega_2\star\cdots\star\omega_n\star\star X_1\overline{\star\omega\underset{\uparrow}{\star}}\star X_2\star\ldots \quad \overset{E}{\Rightarrow}$$

$$]\star\omega_1\star\omega_2\star\cdots\star\omega_n\star\omega\underset{\uparrow}{\star}\ldots \quad .$$

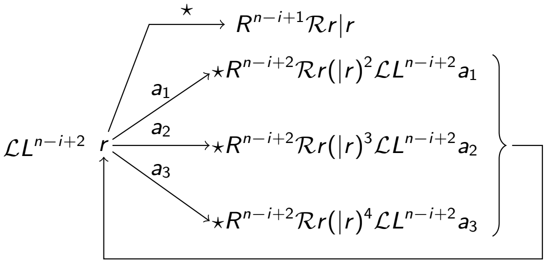
Otuda tražena mašina koja normalno računa funkciju f i ne koristi dodatna slova je $T_2 = V_n\overline{T}^{\S}E$.

- U prethodnom dokazu izostavljeni su dijagrami mašina V_n i E. Ove mašine ne treba da koriste dodatna slova.

- Za mašinu V_n možemo uzeti mašinu $r^2|rV_{n1}\dots V_{nn}l^2$ gdje mašina V_{ni} pravi kod i-te riječi i $\star$ iza te riječi tj.

  $$]\star\omega_1\star\cdots\star\omega_n\star\star X\underset{\uparrow}{\star}\dots \overset{V_{ni}}{\Rightarrow}]\star\omega_1\star\cdots\star\omega_n\star\star X\overline{\omega_i\star}\underset{\uparrow}{\star}\dots \quad .$$

  Diagram za V_{ni} kada je $t=3$:



- Dijagram mašine E napravite sami.

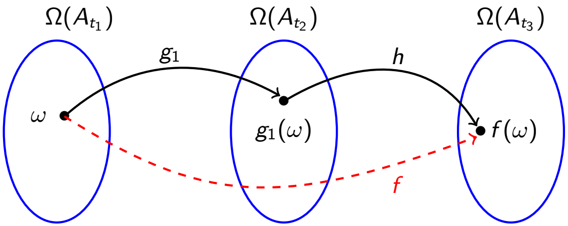
Superpozicija funkcija koje su izračunljive

Teorema. (o superpoziciji) Neka su $t_1, t_2, t_3, n, m \geq 1$. Neka su dalje g_k : $\mathrm{Dom}_{g_k} \to \Omega(A_{t_2})$, $\mathrm{Dom}_{g_k} \subset \Omega^n(A_{t_1})$, $k = 1, \dots, m$ izračunljive funkcije i neka je h : $\mathrm{Dom}_h \to \Omega(A_{t_3})$, $\mathrm{Dom}_h \subset \Omega^m(A_{t_2})$ izračunljiva funkcija. Tada je i funkcija f : $\mathrm{Dom}_f \to \Omega(A_{t_3})$, $\mathrm{Dom}_f \subset \Omega^n(A_{t_1})$ definisana sa:

$$f(\omega_1, \dots, \omega_n) = h(g_1(\omega_1, \dots, \omega_n), \dots, g_m(\omega_1, \dots, \omega_n))$$

izračunljiva funkcija.

$n = 1, m = 1$
$\omega = \omega_1$
$f(\omega) = h(g_1(\omega))$



- **Dokaz**. Kako su funkcije $g_1, \dots, g_m$ i h izračunljive to postoje Tjuringove mašine $T_1, \dots, T_m$ i T koje redom normalno računaju te funkcije. Na osnovu ovih mašina definišimo mašinu M koja će računati funkciju f sa:

$$M = T_1 K^n_{n+1} T_2 K^n_{n+1} T_3 \dots K^n_{n+1} T_m K_{(m-1)n+m} K_{(m-2)n+m} \dots K_m T$$

- Uvedimo oznake $\hat{\omega} = (\omega_1, \dots, \omega_n)$, $\hat{f} = f(\hat{\omega})$ i $\hat{g_i} = g_i(\hat{\omega})$ za $i = 1, \dots, m$
- $\hat{\omega} \in \mathrm{Dom}_f \Leftrightarrow \hat{\omega} \in \mathrm{Dom}_{g_i}, 1 \leq i \leq m$ i $(\hat{g_1}, \dots, \hat{g_m}) \in \mathrm{Dom}_h$
- Ako je $\hat{\omega} \in \mathrm{Dom}_f$ onda TM M izračuna $\hat{g_1}$ (mašina T_1) pa onda napravi kopiju ulaza (mašina K^n_{n+1}) i na toj kopiji izračuna $\hat{g_2}$ (mašina T_2) pa opet pravi kopiju ulaza -od prethodno napravljene kopije (mašina K^n_{n+1}) i izračuna $\hat{g_3}$ (mašina T_3). Ovaj postupak se nastavlja.

$$] \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots \overset{T_1}{\Rightarrow}] \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_1 \underset{\uparrow}{\star} \ldots \overset{K^n_{n+1}}{\Rightarrow}$$

$$] \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_1 \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots \overset{T_2}{\Rightarrow}$$

$$] \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_1 \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_2 \underset{\uparrow}{\star} \ldots \overset{K^n_{n+1}}{\Rightarrow}$$

$$] \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_1 \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_2 \star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots \overset{T_3}{\Rightarrow}$$

Na kraju ove faze se napravi kopija ulaza -od prethodno napravljene kopije (mašina K^n_{n+1}) i izračuna $\hat{g_m}$ (mašina T_m).

Nakon toga se kopira $\hat{g_1}$ (mašina $K_{(m-1)n+m}$, tj. prva riječ od $(m-1)$ kopija ulaza od po n riječi i m rezultata $\hat{g_i}$). Pa se zatim kopira $\hat{g_2}$ (mašina $K_{(m-2)n+m}$) itd. Na kraju ove faze se kopira $\hat{g_m}$ (mašina K_m).

$$] \star \omega_1 \star \cdots \star \omega_n \star \hat{g}_1 \star \cdots \star \hat{g}_{m-1} \star \omega_1 \cdots \star \omega_n \star \hat{g_m} \star \hat{g}_1 \star \cdots \star \hat{g_m} \underset{\uparrow}{\star} \ldots$$

Primjenom T na kraju dobijamo $\hat{f} = h(\hat{g_1}, \ldots, \hat{g_m})$.

$$] \sim \star \hat{g_m} \star \hat{g_1} \star \cdots \star \hat{g_m} \star \hat{f} \underset{\uparrow}{\star} \ldots$$

- Ako je $\hat{\omega} \notin \mathrm{Dom}_f$ onda $(\exists j)\, \hat{\omega} \notin \mathrm{Dom}_{g_j}$ (ako postoji više takvih j uzeli bi najmanje) ili $(\hat{g_1}, \ldots, \hat{g_m}) \notin \mathrm{Dom}_h$ šo znači da će mašina T_j ili mašina T vječno raditi, pa će i mašina M vječno raditi.

- Napomena. Važna stvar u prethodnom dokazu je da mašina T_i normalno računa g_i pa tako na traci iza ulaza upisuje samo vrijednost za g_i. Drugim riječima ne ostavlja neke pomoćne podatke na traci koji nam mogu smetati. Dok ne moramo zahtjevati da T normalno računa h (jer je to kraj računa).

- Funkcija g_k ne mora zavisiti od svakog ω_i tj. neki od ω_i su fiktivni parametri. Uvodimo fiktivne parametre radi simetrije.

Primjeri nerješivih skupova

- **Teorema**. Neka je $M \subset \Omega(A)$. Skup M je rješiv u odnosu na $\Omega(A)$ akko postoji TM čija je radna azbuka A a koja (normalno) računa karakterističnu funkciju skupa M u odnosu na skup $\Omega(A)$.
- Obično se izostavlja "u odnosu na skup $\Omega(A)$" jer se to podrazumjeva.
- **Dokaz**. $(\Leftarrow)$: Direktno na osnovu definicije rješivog skupa.
  $(\Rightarrow)$: Ako je M riješiv skup onda postoji TM koja računa računa karakterističnu funkciju skupa M. Na osnovu teoreme o normalnoj izračunljivosti tada postojati i TM koja ne koristi dodatna slova, tj. radi nad azbukom A, i koja (normalno) računa karakterističnu funkciju skupa M.
- Da bi dokazali postojanje nerješivog skupa mi ćemo koristiti poznatu metodu svođenja na paradoks.

- **Pr**. Berberin iz jednog puka brije tačno one vojnike toga puka koji sami sebe ne briju. Da li berberin brije samog sebe?
  Ne možemo odgovoriti na ovo pitanje jer i za odgovor "DA" i za odgovor "NE" ispostavlja se da dobijamo suprotan odgovor tj. dolazimo do paradoksa.
  Npr. ako ne brije sebe on spada u grupu vojnika koji sami sebe ne briju što znači da on brije takvog vojnika tj. brije sebe.
- Želimo da izaberemo skup tako da TM koja računa njegovu karakterističnu funkciju "igra ulogu berberina" tj. dovodi do paradoksa kad se nekako odnosi na samu sebe.
- Da bi postigli prethodno svaku TM kodiramo sa jednom rječi. Takvu riječ nazivamo mašinskom rječi jer odgovara nekoj TM.
- Sada "nekako odnosi na samu sebe" možemo i precizirati. Naime TM se primjeni na mašinsku rječ koja kodira baš nju.

Mašinske riječi

Neka je azbuka $A = A_t$ za $t \geq 1$ fiksirana. Želimo da TM T čija je radna azbuka A pridružimo rijč $\omega_T \in \Omega(A)$. To radimo u 3 koraka.

1. Tablici TM T pridružimo riječ ω'' tako što komande (četvorke simbola) redom nadovežemo jednu za drugom. Ako je skup stanja $Q = \{q_0, \ldots, q_s\}$ onda tablica ima $(t+1)(s+1)$ komandi a riječ ω'' ima $4(t+1)(s+1)$ slova. Za zapis ω'' pored slova iz A koristimo i slova $q_i \in Q$, a_0, s, r i l što je ukupno $t+s+5$ slova tj. $\omega'' \in \Omega(A_{t+s+5})$.
2. Riječi ω'' možemo pridružiti riječ $\omega' = \gamma_{t+s+5,t}(\omega'') \in \Omega(A)$. Da bi od ω' dobili ω'' treba nam s koje nije fiksirano (ne znamo ga). Zato riječi ω'' pridružujemo par (s, ω') gdje je s unarno zapisano.
3. Paru riječi $(s, \omega') \in \Omega^2(A)$ pridružujemo riječ $\omega_T = \sigma_2^t(s, \omega') \in \Omega(A)$.

- Dakle, $T \to \omega'' \to (s, \omega') \to \omega_T$. Važi i obrnuto $\omega_T \to (s, \omega') \to \omega'' \to T$.

- Mašinsku riječ za TM T smo označili sa ω_T, dok ćemo mašinsku riječ za TM M označiti sa ω_M i sl.
- Uočimo da ω_T dobijamo efektivno (algoritamski) na osnovu tablice za TM T. Važi i obrnuto, tablicu za T dobijamo efektivno na osnovu ω_T.
- Kodiranje je jednoznačno (1-1) tj. važi $M_1 \neq M_2 \Leftrightarrow \omega_{M_1} \neq \omega_{M_2}$.
- Kodiranje nije "NA". Jer nije svaka riječ $\omega \in \Omega(A)$ mašinska riječ.
- Postoji efektivna procedura koja za ulaz $\omega \in \Omega(A)$ provjerava da li je ω mašinska riječ̌. Naime na osnovu ω uz pomoć funkcija oblika σ i γ nađemo s i ω''. Zatim provjerimo da li ω'' odgovara tablici neke mašine. Uzimamo redom po 4 simbola iz ω'' i provjeravamo da li je to odgovarajuća komanda mašine. Ako smo uzeli $j(t+1)+i$-tu četvorku ona treba da bude oblika $q_j a_i v q'$ gdje je $v \in A \cup \{a_0, l, r, s\}$ i $q' \in Q$.

- **Pr**. Početak kodiranja za mašinu r u azbuci A_1 izgleda: $\omega'' = q_0 a_0 r q_1 q_0 a_1 r q_1 q_1 a_0 s q_1 q_1 a_1 s q_1$ pa $(1, 3 \cdot 7^{15} + 1 \cdot 7^{14} + 6 \cdot 7^{13} + 4 \cdot 7^{12} + \cdots + 5 \cdot 7 + 4)$, gdje je uzet poredak slova $a_0, a_1, q_0, q_1, s, r, l$ u azbuci A_7.

- Mogli smo izabrati i drugačije kodiranje. Npr. što u komandi $qavq'$ slovo a_j i stanje q_j (tj. simbole q, a, q') kodiramo sa slovom B i j simbola A, a radnju v kodiramo sa slovom B i k simbola A gdje je $k = i$ za $v = a_i$, $k = t + 1$ za $v = s$, $k = t + 2$ za $v = r$ i $k = t + 3$ za $v = l$.

  Tako kod sa prethodno spomenutu mašinu r izgleda: $BBBAAAABABBABAAAABABABBAABABABABAABA$

  Koristimo samo dva simbola tj. kod je zapisan u azbuci A_2. Dekodiranje je jednoznačno jer kodovi komandi počinju sa $4k + 1$-ve pozicije slova B.

Jedan nerješiv skup (M)

- Za TM M pisaćemo $M(\omega) = \omega'$ da označimo da se mašina M poslije primjene na ω zaustavi u konačno mnogo koraka poslije riječi ω'.
- **Def**. Za datu azbuku A definišemo skup:

$$M = \{\omega \in \Omega(A) \ : \ (\exists \text{TM } T \text{ nad } A)\, \omega = \omega_T \text{ i } T(\omega) = a_1\}.$$

- Očigledno svi elementi skupa M su mašinske riječi.
- Ako je T neka mašina i ω_T njena mašinska riječ onda važi $\omega_T \in M \Leftrightarrow T(\omega_T) = a_1$.
- **Teorema**. Skup M nije rješiv (skup M je nerješiv).
- **Dokaz**. Dopustimo da je skup M riješiv. Tada postoji TM T_0 nad A koja računa karakterističnu funkciju $\chi_M(\omega) = \begin{cases} \lambda & \text{ako} \quad \omega \in M \\ | & \text{ako} \quad \omega \notin M \end{cases}$ skupa M.

  Označim sa ω_{T_0} mašinsku riječ mašine T_0.

Pitamo se da li je $\omega_{T_0} \in M$?

1) Ako je $\omega_{T_0} \in M$ onda je $\chi_M(\omega_{T_0}) = \lambda$ tj. $T_0(\omega_{T_0}) = \lambda$ pa po definiciji skupa M dobijamo $\omega_{T_0} \notin M$. Dobili smo protivrečnost.
2) Ako je $\omega_{T_0} \notin M$ onda je $\chi_M(\omega_{T_0}) = |$ tj. $T_0(\omega_{T_0}) = |$ pa po definiciji skupa M dobijamo $\omega_{T_0} \in M$. Ponovo dobijamo protivrečnost.

Dobili smo paradoks, što nam ukazuje da je skup M nerješiv.

- Uslovi u definicija skupa M su suprotni uslovima u definiciji karakteristične funkcije. To nas i dovodi do paradoksa kao kod uslova "brije one koji sami sebe ne briju".
- Teoremu smo mogli formulisati: Karakteristična funkcija skupa M nije izračunljiva.
- Prethodnim smo dokazali da nije uvjek moguće napraviti algoritam, jer takav algoritam ne postoji.

Drugi nerješivi skupovi (M_1), (M_2)

- Za TM M pisaćemo $M(\omega) \rightsquigarrow Z$ da označimo da se mašina M poslije primjene na ω zaustavi u konačno mnogo koraka. Dozvoljeno je i PT.
- **Def**. Za datu azbuku A definišemo skupove:

$$M_1 = \{\omega \in \Omega(A) \ : \ (\exists \text{TM } T \text{ nad } A)\, \omega = \omega_T \text{ i } T(\omega) \rightsquigarrow Z\}.$$

$$M_2 = \{\omega \in \Omega(A) \ : \ (\exists \text{TM } T \text{ nad } A)\, \omega = \omega_T \text{ i } T(\lambda) \rightsquigarrow Z\}.$$

- Za M_1 i M_2 nije bitno šta TM T vraća, bitno je samo da se zaustavi u konačno mnogo koraka. Dozvoljeno je i PT.
- Jasno je da M_1 i M_2 sadrže samo mašinske riječi i da je $M \subset M_1$.
- **Teorema**. Skupovi M_1 i M_2 su nerješivi.
- **Dokaz**. Dopustimo da je M_1 rješiv. Tada postoji TM T_1 koja računa njegovu karakterističnu funkciju.
  Želimo da provjerimo da li je neka riječ $\omega \in M$.

Pomoću T_1 provjerimo da li je $\omega \in M_1$. Ako $\omega \notin M_1$ odgovorimo $\omega \notin M$. Ako $\omega \in M_1$ onda je ω mašinska riječ pa postoji TM T takva da je $\omega_T = \omega$. Sada primjenimo TM T na ω_T. Znamo $T(\omega_T) \rightsquigarrow Z$. Ako je $T(\omega_T) = a_1$ onda odgovorimo $\omega \in M$ u suprotnom odgovorimo $\omega \notin M$.

Dakle, opisali smo algoritam koji nam daje odgovor na pitanje da li je $\omega \in M$. Znamo da takav algoritam ne postoji. Dobili smo protivrečnost. Pa je M_1 nerješiv.

- ⋆ Dopustimo sada da je M_2 rješiv. Tada postoji TM T_2 koja računa njegovu karakterističnu funkciju.
  Želimo da provjerimo da li je neka riječ $\omega \in M_1$.

  Provjerimo prvo da li je ω mašinska riječ pa ako nije odgovaramo $\omega \notin M_1$. Ako je $\omega = a_{i_1} \ldots a_{i_m}$ mašinska riječ onda postoji TM T takva da je $\omega_T = \omega$. Neka je $\omega_{T'}$ mašinska riječ TM
  $T' = ra_{i_1} \ldots ra_{i_m} rT$.
  Očigledno, $\omega \in M_1 \Leftrightarrow \omega_{T'} \in M_2$.

Pomoću T_2 provjerimo da li je $\omega_{T'} \in M_2$. Ako $\omega_{T'} \notin M_2$ odgovorimo $\omega \notin M_1$. Ako $\omega_{T'} \in M_2$ onda odgovorimo $\omega \in M_1$.

Dobilismo da je M_1 rješiv, što je u suprotnosti sa prethodno dokazanim. Dakle, skup M_2 je nerješiv.

Univerzalna Tjuringova mašina

- U prethodnim dokazima na osnovu mašinske riječi ω_T zaključili smo da postoji TM T i onda T primjenjili na neku riječ ω'.
- Ne moramo da prvo rekonstruišemo tablicu za T, jer postoji TM U koju nazivamo univerzalna TM, koja kao ulaz dobija kod ω_T mašine T i ω' i modelira rad mašine T na ulazu ω'.
- Upravno savremeni računari rade kao univerzalne TM. Dobiju kod programa i ulaz i rade kao mašina koja odgovara tom programu.

- Ako je ulaz mašine T n-torka $(\omega_1, \ldots, \omega_n)$ onda TM U dobija kod te riječi $\overline{\omega} = \sigma_2^t(n, \sigma_n^t(\omega_1, \ldots, \omega_n))$.
- Funkcionisanje TM U na polaznoj poziciji $]\star \omega_T \star \overline{\omega} \underset{\uparrow}{\star} \ldots$ odgovara funkcionisanj mašine T na polaznoj poziciji $]\star \omega_1 \star \cdots \star \omega_n \underset{\uparrow}{\star} \ldots$
- Prethodno znači ili obije mašine vječno rade, ili obij prelaze preko kraja trake ili kod obije dolazi do mašinskog zaustavljanja. U slučaju MZ onda se T zaustavlja poslije neke riječi akko se U zaustavlja poslije neke riječi i pri tome se te riječi poklapaju.

Zadatak o zaustavljanju

- Neka je $\omega_T \in \Omega(A)$, $\hat{\omega} = (\omega_1, \ldots, \omega_n) \in \Omega^n(A)$ i $\overline{\omega} = \sigma_2^t(n, \sigma_n^t(\hat{\omega}))$, gdje je $A = A_t$ azbuka. Definišimo skup:

$$H = \{(\omega_T, \overline{\omega}) \, : \, (\exists \text{TM } T)\, \omega = \omega_T \text{ i } T(\hat{\omega}) \rightsquigarrow Z\}$$

- Zadatak provjere da li je $(\omega_1, \omega_2) \in H$ nazivamo zadatak o zaustavljanju ili halting problem.
- Naime zadatak se svodi na provjeru da li se TM T primjenjena na polaznu poziciju $]\star\omega_1\star\cdots\star\omega_n\underset{\uparrow}{\star}\ldots$ zaustavlja nakon konačnog broja koraka.
- **Teorema**. Zadatak o zaustavljanju je algoritamski nerješiv (tj. skup H je nerješiv).
- **Dokaz**. Ako bi dopustili da je H rješiv onda bi i skup M_1 bio rješiv, što nije tačno. Naime algoritam provjere da li je $\omega \in M_1$ bi glasio.

  Ako ω nije mašinska riječ onda je odgovor "NE". Ako je ω mašinska riječ onda uzimamo odgovor na pitanje da li je $(\omega, \overline{\omega}) \in H$, gdje je $\overline{\omega} = \sigma_2^t(1, \omega)$.
- Navedene probleme nazivamo i problemi samoraspoznavanja.

- Navedimo još nekoliko problema koji su nerješivi a slični su sa problemom zaustavljanja. 1^o) Da li je funkcija koju računa TM: konstanta, periodična, ograničena,...; 2^o) Da li data TM za dati ulaz tokom svog rada bar nešta odštampa; 2^o) Da li dati broj pripada skupu vrijednosti funkcije koju računa data TM; itd.
- Rješiv je npr. problem: Da li data mašinska riječ kodira TM koja ima više od 10 stanja?

Razni primjeri nerješivih skupova

- **Zadatak 1:** Da li opšta diofantska jednačina ima cjelobrojnih rješena? (ranije spomenuto).
- **Zadatak 2:** Konteksno slobodna gramatika. Data je gramatika G i riječ w da li riječ w pripada jeziku gramatike G? (Uči se iz programskih prevodilaca).
- **Zadatak 3:** Asocijativni račun u polugrupi. Dat je spisak dozvoljenih zamjena i dvije riječi w_1 i w_2.

Da li se w_2 može dobiti iz w_1 korišćenjem dozvoljenih zamjena (tj. da li su w_1 iw_2 ekvivalentne riječi)?

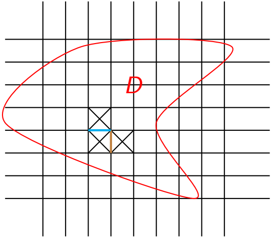
- **Zadatak 3:** Problem popločavanja.

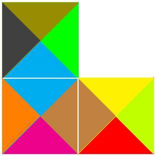
  Ploča je kvadrat dimenzija 1×1 čije su ivice obojene sa bojama iz nekog skupa boja B. Ploča ne dopušta simetrije i rotacije. Obično bojimo odgovarajuće trouglove umjesto ivica.

  Matematički ploču možemo zadavati kao uređenu četvorku $t = (l, b, r, t) \in B^4$.

  Neka je $\tau \subseteq B^4$ skup (tipova) ploča.

Opšti problem popločavanja. Dat je skup τ ploča i skup $D \subseteq \mathbb{Z} \times \mathbb{Z}$ (tj. D je dio ravni) izdeljen na kvadrate dimenzija 1×1. Da li se D može pravilno popločati sa pločama iz τ? Pravilno popločavanje znači da ploče sa zajedničkom ivicom imaju istu boju na toj ivici.





Ako je $D = \mathbb{Z}^2$ onda je problem popločavanja nerješiv.

Model izračunivosti RAM, RASP, TM

Algoritmi i njihova složenost

- Ako je funkcija izračunljiva onda postoji *TM* koja je računa. Tačnije postoji više TM (algoritama) koje računaju tu funkciju.
- Da bi izabrali najbolji algoritam (TM) koji računa datu funkciju treba da uvedemo mijeru koliko je neki algoritam dobar tj. koliko zahtjeva resursa.
- Najčešće nas interesuje koliko algoritam zahtijeva vremena i prostora, pa govorimo o vremenskoj i prostornoj složenosti.
- Vrijeme izvršavanja algoritma je vrijeme koje protekne od momenta kada algoritam počne da radi do momenta dok se algoritam ne zaustavi.
- Vrijeme izvršavanja algoritma zavisi od veličine ulaza kao i od mašine na kojoj se izvršava (njene brzine takta).
- Zelimo da procjenimo vrijeme izvršavanja algoritma bez njegovog puštanja u rad.

- Za tu procjenu nam služi vremenska složenost algoritma.
- Vremenska složenost algoritma ne zavisi od mašine na kojoj ćemo izvršavati algoritam, zavisi samo od ulaza. Zato je izražavamo kao funkciju od veličine ulaza.
- Intuitivno vremenska složenost algoritma predstavlja broj elementarnih koraka koje taj algoritam treba da izvrši.
- Treba precizirati i šta je elementarni korak. Npr. kod savremenog računara kao elementarne korake možemo uzeti: sabiranje dva broja, množenje dva broja, poređenje dva broja, i sl. Ali možemo uzeti i jedan takt (što je preciznija mjera).

  Kod TM elementarni korak bi bio prelaz sa jedne na drugu konfiguraciju.
- **Pr**. $y = x + 3$ je jedan korak algoritma. Pa je vremenska složenost 1.
- Nekad je teško prebrojati tačan broj koraka pa se koriste asimptotske oznake poput O-notacije, Ω-notacije i Θ-notacije.

- Neka su $f : \mathbb{N} \to \mathbb{N}$ i $g : \mathbb{N} \to \mathbb{N}$ dvije funkcije. Tada,

$$f(n) = O(g(n)) \Leftrightarrow (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n > n_0)\, f(n) \leq c \cdot g(n)$$

$$f(n) = \Omega(g(n)) \Leftrightarrow (\exists c \in \mathbb{R}^+)(\exists n_0 \in \mathbb{N})(\forall n > n_0)\, c \cdot g(n) \leq f(n)$$

$$f(n) = \Theta(g(n)) \;\Leftrightarrow\; \big(f(n) = O(g(n))\big) \wedge \big(f(n) = \Omega(g(n))\big).$$

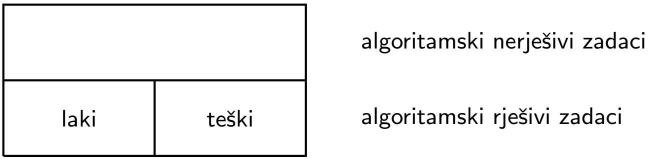
- Kod O-notacije zanemarujemo sve vrijednosti nižeg reda kao i konstantne činioce. Npr. $5n^3 - n^2 + 7n + 100 = O(n^3)$.
- Važi $O(f(n)) + O(f(n)) = O(f(n))$.
- **Pr**. Složenost ugnježdene dvije FOR petlje je $O(n^2)$.
  FRO $i = 1$ TO n DO
     FOR $j = 1$ TO n DO
          $s = s + x_{ij}$
     END FOR
  END FOR

  Teško je reći koliko tačno ima koraka. Jer postoje i skriveni koraci poput uvećavanja i za jedan, poređenja i sa n,. . .

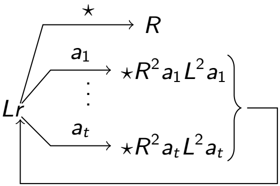
- Treba da preciziramo i kako računamo veličinu ulaza.
- Neka je data TM M.
- Veličinu ulaza kod TM M je broj upotrebljenih polja u početnoj poziciji. Smatramo da početna pozicija odgovara ulazu.
- Preciznije ako je S_0 početna pozicija TM M koja odgovara ulazu w onda je veličina ulaza $|w| = n = k(S_0) + 1$.

  Za definiciju $k(S)$ i upotrebljenog polja pogledati: Kodiranje naz azbukom A_1.
- Označimo sa $\mathcal{C}_M(w)$ izračunavanje TM M koje odgovara ulazu w a sa $\ell(\mathcal{C}_M(w))$ broj konfiguracija u izračunavanju $\mathcal{C}_M(w)$. Preciznije $\ell(\mathcal{C}_M(w)) = \infty$ ako TM M vječno radi a $\ell(\mathcal{C}_M(w)) = j$ ako se TM M zaustavlja poslije koraka j.
- Upravo $\ell(\mathcal{C}_M(w))$ predstavlja vremensku složenost TM M na ulazu w.

- **Def**. Neka je data TM M. Funkcija $T : \mathbb{N} \to \mathbb{N}$ definisana sa $T(n) = \max_{|w|=n} \ell(\mathcal{C}_M(w))$ je vremenska složenost TM T na ulazu veličine n.
- Očigledno $T(n) = \infty$ ako postoji ulaz veličine n na kome TM M vječno radi.
- Za ovakvu složenost kažemo da je složenost u najgorem slučaju jer tražimo maksimum (tj. od ulaza date veličine biramo složenost onog ulaza koja je najgora).
- Pored vremenske definiše se i prostorna složenost.
- Prostorna složenost TM M na ulazu w račuamo tako što za svaku konfiguraciju u izračunavanju $\mathcal{C}_M(w)$ nađemo broj upotrebljenih polja, a zatim nađemo maksimum tih brojeva.
- Prostornu složenost TM M na ulazu veličine n označavamo sa $S(n)$ i računamo kao maksimum prostornih složenosti po svim ulazima w veličine n.

- Kod klasičnog računara za veličinu ulaza možemo uzeti broj bajtova potrebnih da se taj ulaz zapiše. Takođe prostorna složenost prestavlja maksimalni broj bajtova koje smo iskoristili u toku izvršavanja programa.
- Složenost zadatka (vremenska/prostorna) je složenost najboljeg algoritma za taj zadatak.
- Kad kažemo samo složenost obično mislimo na vremensku složenost.
- Za zadatak kažemo da je lak ako za njega postoji algoritam čija je složenost polinomijalna. Inače kažemo da je zadatak težak. Npr. zadaci koji imaju složenost $T(n) \geq 2^n$ su teški.
- 

- Znamo da je zadatak o cjelobrojnim nulama diofantske jednačine nerješiv.
- Zadatak o poluproširenom regularnom izrazu (biće formulisan kasnije) je težak.
- Zadatak o podjeli (particiji): Dat je skup $S = \{x_1, \ldots, x_n\} \subset \mathbb{N}$. Da li postoji $S' \subset S$ tako da je suma brojeva skupa S' jednaka sumi brojeva skupa $S \setminus S'$? Za ovaj zadatak se ne zna da li je lak ili težak.
- Zadatak o Ojlerovom ciklusu: Da li u neusmjerenom grafu postoji Ojlerov ciklus? Ovaj zadatak je lak.
- Dijagram TM K:



Složenost:

$T(n) = 2n^2 + 8n + 3$