Електротехнички факултет, Универзитет у Београду

АЛАН ТЈУРИНГ И ТЈУРИНГОВА МАШИНА

- семинарски рад -

предмет: Сложеност алгоритама и одабране методе оптимизације

професори:
проф. др Бранко Малешевић
проф. др Ивана Јововић
проф. др Татјана Лутовац

студенти:
Јана Јанковић 0384/2018
Теодора Маринковић 0093/2018
Алекса Сретовић 0246/2018

Београд, 2020.

С А Д Р Ж А Ј

У В О Д

Највеће математичке умове XX века интригирала је једна идеја – могућност да се математика у целости сведе искључиво на стриктна правила, на алгоритме. Овим проблемом се позабавио и Алан Тјуринг и на њега дао одговор.

Многи историчари сматрају да је, захваљујући његовом раду, трајање Другог светског рата битно скраћено. Уз колегу Голдона Велчмана, конструисао је „Бомбу”, криптоаналитички уређај за разбијање шифри Енигме, које су задавале велике проблеме савезницима.

1. ЖИВОТ АЛАНА ТЈУРИНГА

Алан Метисон Тјуринг рођен је 23. јуна 1912. године у Лондону, као припадник више средње класе. У школи није био превише заинтересован или довољно амбициозан да удовољи образовним захтевима. За своју душу занимао се Ајнштајновом теоријом релативности и квантном механиком, као и хемијским експериментима у лабораторији. Управник школе записао је:

„Математика му не иде нарочито добро. Доста времена троши на истраживања из више математике, запостављајући основне ствари. Његови радови су неуредни. Немогуће је истовремено седети на две столице. Ако хоће да остане у приватној школи, мора да стреми да постане образован. Ако хоће да буде само специјалиста у науци, онда губи време у приватној школи.”

Тјурингова средњошколска диплома била би неизвесна, а његова научна истраживања остала би на нивоу хобија да није било неочекиваног подстрека, љубави, и то несрећне. Тјуринг се заљубио у Кристофера Моркома, младића са којим га је зближило заједничко интересовање за науку. Међутим, несрећни Морком изненада је преминуо 1930. године од последица туберкулозе. Тјуринг је тешко поднео губитак.

Тјуринг је наставио да одржава контакт са Моркомовом мајком још дуго након његове смрти. Занимљиво је да је он у том периоду живота и даље веровао у духове, који су независни од тела и настављају свој пут и након смрти. У неким круговима се чак причало да су управо Тјурингов материјализам и атеизам били оно што је убило младог Моркома. У једном од бројних писама које је послао Моркомовој мајци, записао је:

„Ја сматрам да је дух вечно повезан са материјом, али засигурно не истим телом… што се тиче везе између духа и тела, сматрам да тело задржава дух, а оно је живо и са њим чврсто повезано. Не могу наслутити шта се дешава док тело спава, али када тело умре, механизам тела који тај дух задржава нестаје, а тај дух налази ново тело пре или касније, можда чак и одмах.”

Затим је започео студије на Кембриџу, који је био окружење у коме је његова неконвеционалност, како у погледу сексуалне оријентације, тако и у приступу математичким проблемима, била охрабривана. На Кембриџу је развио склоност ка дугопругашком трчању, самопоуздање у науци и друштвену личност. Након што је дипломирао са двадесетдве године, остао је на колеџу као научни сарадник. Предавања из топологије усмерила су његово размишљање ка поблемима математичке логике. У том духу уследила је брилијантна идеја о стварању машине, данас познате као Тјурингова машина. Тјуринг је доказао да би његова машина могла да реши било који математички проблем, уколико га је могуће представити алгоритмом. Недуго потом, доказао је како *проблем одлуке* заправо нема решења, показавши да је халтинг проблем[1] за његову машину неодлучив – доказао је да општи алгоритам за решавање халтинг проблема за све могуће парове улаз–излаз не може постојати.

1938. године завршио је докторске студије на Принстону, а потом се вратио у Велику Британију.

След догађаја је такав да ће и Тјуринг, а не само његове идеје, морати да напусти врт теорије и окрене се пољу праксе. До тада пацифиста, одлучио је да се стави у службу ратне кампање против наступајуће нацистичке опасности. Крајем 1938. године почео је да ради на проблему пред којим су се нашли британски обавештајци. Машина за шифровање, „Енигма”, коју су користиле немачке морнаричке снаге за комуникацију између подморница које су дејствовале у Атлантику,

[1] Халтинг проблем је проблем одлучивања који поставља питање да ли ће програм завршити са извршавањем, или ће наставити да се извршава у бесконачност.

предтсвљала је нерешив проблем за Енглезе. Захваљујући Тјуринговом теоријском раду, ова препрека је била превазиђена. Тајна ратна резиденција британских криптоаналитичара била је Блечли парк, имање у самом срцу Енглеске. То је било место у којем је Алан провео већи део ратних година. Његов допринос коначном успеху у дешифровању Енигме лежи у нацрту „бомбе”, машине која се користила за анализу шифроване комуникације. Тјурингов рад на решавању криптоаналитичких проблема имао је за последицу конструисање првих великих електронских рачунарских машина „Колос”, конструисање уређаја за скрембловање гласовних порука, утемељење теорије информација и статистике као научних дисциплина, али у том тренутку најважнија последица била је ратна победа, у којој су информације обезбеђене дешифровањем тајне комуникације немачких поморaца имале огромну улогу.

После рата Тјуринг је радио у Националној лабораторији за физику, где је радио на нечему што је требало да постане прва универзална машина у физичком облику. 1946. године објавио је рад о дизајнирању рачунара на ускладиштеним програмима (ACE – Automatic Computing Engine). Увео је потпуно нов концепт по којем је програм складиштен у меморију и заслужан је за рад машине. Тиме је развијена идеја о вештачком мозгу или електричном мозгу који опонаша менталне процесе.
При раду на конструкцији рачунара ACE, Тјуринг се убрзо разочарао због бирократских и обавештајних препрека које су успоравале тај пројекат. 19. фебруара 1946. године презентовао је дизајн првог рачунара у Британији. Иако га је дизајнирао, ACE је извршио први програм тек 10. маја 1950. године, и то у Тјуринговом одсуству, јер је тада био на Кембриџу.

1948. године писао је шаховски програм за рачунар који још увек није постојао, тако да је 1952. године сам симулирао програм, који је једном победио и једном изгубио меч. Његов програм тада је похвалио један од најбољих шахиста свих времена, велемајстор Гари Каспаров.

1949. године водио је рачунарску лабораторију на Манчестерском универзитету, на које се бавио софтвером који је требало да покреће један од првих правих рачунара, Manchester Mark 1. Идеја је била да креира input-output систем и омогући његово програмирање користећи живину линијску меморију са кашњењем од којих је свака могла да чува до 32 бита података.

1950. године одлази у Манчестер да би се прикључио пројекту у чијем је средишту био један рачунар мањег капацитета. То је била машина у коју је по први пут имплементирана технологија катодне цеви. Тјуринг и његови сарадници углавном су је користили покушавајући да реше неке од познатих математичких проблема. Рад на машини био је компликован, напоран, често и узалудан, јер је оперативна процедура која је подразумевала трчање из просторије са машином до просторије на спрату где су се резултати исписивали остављала доста простора за грешке. Међутим, њихов пројекат се ускоро нашао у жижи интересовања јавности. Један неурохирург једном приликом је изјавио:„Све док машина не буде способна да напише сонет или компонује кончерто надахнута мислима и осећањима, а не насумичним ређањем симбола, не можемо се сложити да је дорасла мозгу”. У неку руку инспирисан овом изјавом Тјуринг је написао свој најпознатији и најнеобичнији рад „Рачунарске машине и интелигенција”, у коме се посветио питању вештачке интелигенције.

Пошто је већ поставио претпоставку да мозак ради на механичкој основи и да машина може да симулира његов рад, сматрао је да постоји формула по којој се одвија процес мишљења. Све логичке операције које се десе унутар наше главе могу да се десе и унутар машине.

Касније је предложио и тест назван „Тјурингов тест”, којим би се дошло до одговора на поменуто питање „Могу ли машине да мисле?”. Експеримент би имао форму коју је Тјуринг назвао игром имитације. У игри учествују три учесника, мушкарац, жена и испитивач, при чему се свако налази у одвојеној просторији, а сва комуникација се одвија писменим путем. Испитивач има за циљ да одреди ко је од испитаника мушко, а ко женско при чему мушки учесник има задатак да испитивача одговорима наведе на погрешан закључак, док женски учесник испитивачу помаже. Тако је питање

„Могу ли машине да мисле?” замењено питањем „Може ли машина да победи у игри имитације?”. Тјуринг је тврдио да ће у најскоријој будућности бити могућа конструкција машине која ће у игри имитације имати подједнаке шансе за победу као и човек. По његовом мишљењу, машину је могуће прогласити интелигентном ако човек не може да утврди разлику између одговора које добија од машине и другог човека. Данас често употребљаван тест на интернету, CAPTCHA тест, представља инверзан облик Тјуринговог теста – служи да утврди да ли је корисник компјутер или човек. Тјурингов тест остао је као један од најупечатљивијих доприноса дебати о вештачкој интелигенцији која и дан данас траје.

Недуго затим Тјуринг се упустио у аферу са једним младићем у Манчестеру и постао предмет пљачке и уцене, засноване баш на чињеници да су хомосексуални односи сматрани кривичним делом „велике непристојности”. Тјуринг се обратио полицији, али пред судом краљице Елизабете, од жртве је постао злочинац. Пресудом му је омогућен избор између служења затворске казне и условне казне, која је подразумевала узимање хормонских инјекција синтетичког естрогена. Тјуринг се определио за другу опцију. Поред тога, оцењен је као особа ризична по безбедност, након чега је уклоњен из пројекта у којем је био укључен.

Осмог јуна 1954. године, Тјуринг је пронађен мртав. Поред његовог тела лежала је полупоједена јабука. Иако није проверено да ли се налазио у самој јабуци, утврђено је да је узрок смрти тровање цијанидом, а истражитељи су као узрок смрти навели самоубиство. Тјурингови биографи су претпоставили да је он желео да реконструише сцену из своје омиљене бајке, *Снежана и седам патуљака.*

Неколико недеља пред смрт у писму једном пријатељу написао је:

„Тјуринг верује да машине могу да мисле
Тјуринг леже са мушкарцима
Према томе машине не могу да мисле.”

Постхумно , краљица Елизабета II га је помиловала 24. децембра 2013. године. Алан Тјуринг је тако постао четврта особа која је од 1945. године у Уједињеном Краљевству добила краљевско помиловање.

Од 1966. године Удружење за рачунарство (Association for Computing Machinery) додељује Тјурингову награду за рачунарска достигнућа. Сматра се да је та награда у свету рачунара једнака са Нобеловом наградом. У Манчестеру, граду у којем је радио до краја свог живота, одржавају се разне почасти у Тјурингово име, а његова статуа је завршена 23. јуна 2001. године у Саквил парку. Једна од манчестерских улица је 1994. године названа по њему, Alan Turing Way.

Слика 1: Статуа Алана Тјуринга у Саквил парку

2. НАУЧНИ ДОПРИНОС

2.1. ТЈУРИНГОВА МАШИНА

Да ли комплексно доношење одлука може да се механизује? Да ли је могуће радити нешто, сачувати текући резултат и у зависности од њега извршити нешто? На почетку је то рађено са такозваним „plug boards ”, таблама са прикључцима, што је омогућавало репрограмирање машина физичким путем. Идеја је била да се направе секвенце наредби и операција које би то радиле за нас.

Давид Хилберт је први покренуо питање машине која је могла на постављена питања да одговара са „да” или „не”. Машине које доносе одлуке су свима биле стране, што је довело до тога да је то свима постала тема за размишљање. Ту је допринос имао Алан Тјуринг. Био је фасциниран комплексним механизмима који стоје иза људског ума. Тјуринг је поједноставио опште питање „Како би физички изгледале те машине?!”.
1928. године објавио је рад „On Computable numbers, with an application to the Entscheidungsproblem”, директан одговор на Хилбертов изазов. У њој описује машину која може радити све што и друге машине, познату као „универзална машина”. Замишља машину као човека у кутији са оловком и папиром који је добио књигу са инструкцијама које треба да следи. Да бисте мењали шта тај човек ради морали сте му задавати инструкције у књизи. То је идеја коју је имао: мењање понашања симболима уместо механизмом. Рекао је да за дати низ симбола који би извршио посебну обраду човек анализира тај низ, али симбол по симбол. Ово је називао „свест” (awareness). Онда човек погледа листу унструкција и упореди оно што је запамтио са листом могућности у књизи док не нађе поклапање. Када нађе поклапање прелази на операциони корак (the operation step). Увидео је да без обзира на то шта пише у инструкцијама и на ком језику, радње су увек само записиване на папиру и померале пажњу на нову локацију. Последња колона у табели (IF aware/do/nextstate(page)) је ново стање, нова страница на коју треба прећи и следити инструкције (странице представљају стања машине). Програмирање је исто као придржавање страница и инструкција у књизи, извршавање задатих наредби. Та књига није иста за сваког читаоца, она зависи од претходног стања које је довело на ту страницу. За назнаку заврши (hult) књига би се остављала са стране. Можемо приметити да је за овакво доношење одлука ове машине потпуно одузело човеком утицај, сва логика се заснива на томе како је књига или алгоритам дизајнирана, и то називамо програм. Тјуринг је уочио да све ово може да се реализује са постојећом технологијом, да му није потребно учешће људи. За читање, Тјуринг је замислио померајућу, скенирајућу главу која је могла да скенира симбол, искључиво по један, и памти га (такозвана read операција). Онда прелази у друго стање и тражи одговарајући симбол, а када пронађе поклапање прелази на операциони корак. Машина садржи штампач тако да на крају може да упише обрађени резултат. То је све што ради: чита, помера се и исписује. Да би се поједноставило, Тјуринг нас подсећа да књига коју обезбеђујемо (алгоритам) може да се нађе на дугачкој траци, што доводи до поједностављивања машине на траку и читајуће-писаћу главу. Сваки симболички процес који врше данашњи компјутери може се извршити на Тјуринговој машини. Тјуринг тврди да све што његова машина може да одради идентификује се као програмибилно. Програмибилан проблем је проблем који Тјурингова машина може да заврши за довољно времена и довољно простора. Затим се запитао шта можемо, а шта не можемо написати и направити да се решава на овакав начин.

Идеја и принцип по којем ради Тјурингова машина је толико моћан да сви компјутери раде исту ствар и може се користити да би се оценила јачина компјутерског програма. Када програм може да уради оно што Тјурингова машина може, кажемо да је Тјуринг комплетан, што је на врху хијерархије у јачини програма. До данас није створено ништа што може радити нешто више него Тјурингова машина.
Тјуринг овакве машине описује као постојање начина записивања информација у кодираном облику.

Тјурингова машина је апстрактна машина, теоријски модел у служби мисаоног експеримента. Овом машином је Алан пре свега покушао да реши проблем из области математичке логике, користећи се ванматематичким средствима.

2.1.1. ВРСТЕ ТЈУРИНГОВЕ МАШИНЕ

Тјурингова машина се може сматрати за машину у којој:

– азбука којом се записује садржај ћелија траке не мора бити унарна,
– поред завршног стања q_z уводе се и нека специјална завршна стања, на пример $q_{да}$ и $q_{не}$ која, интуитивно, значе позитиван, односно негативан одговор на постављени проблем,
– је дозвољена трака која је бесконачна само на једну страну, односно постоји крајња лева ћелија, док се на десно трака пружа неограничено,
– уместо само једне, постоји више трака, а за сваку траку постоји посебна глава,
– над једном траком постоји више глава уместо само једне,
– трака је дводимензионална а не једнодимензионална, односно трака подсећа на бесконачну шаховску плочу,
– је у једној наредби машине могуће и уписати знак у ћелију и померати главу,
– не важи захтев за детерминисаношћу, односно дозвољено је да постоје наредбе које одговарају истом стању и знаку у ћелији над којом се налази глава, а које се разликују по дејству (операцији која се извршава и/или стању у које се прелази).

Занимљиво је да у смислу израчунљивости готово све од ових варијанти Тјурингове машине одговарају истој класи функција, однсно класи Тјуринг-израчунљивих функција, као и основна верзија машине. Изузетак представљају неки слабији, рестриктивни случајеви, на пример машина чија трака је ограничена са једне стране и користи унарну азбуку или машина која има само два стања и користи алфабет од два знака. Еквиваленција варијанти Тјурингове машине се доказује тако што се покаже да за сваки програм P, за неку од варијанти Тјурингове машине постоје програми за преостале варијанте који симулирају извршење програма P и израчунавају исту функцију. Избор варијанте Тјурингове машине зависи од проблема који хоћемо да решимо.

2.1.1.1. ЕЛЕМЕНТАРНЕ ТЈУРИНГОВЕ МАШИНЕ

Могу да се дефинишу три елементарне Тјурингове машине. Нека је $t \geq 1$. Све три елементарне машине a_i , r и l користе радну азбуку $\mathrm{A} = \{a_1, a_2, \ldots, a_t\}$. Резултат њихове примене на произвољну позицију јесте извесна „елементарна” промена.

1. Тјурингова машина a_i ($0 \leq i \leq t$)

   Када се та машина примени на произвољну позицију, она у радно поље уписује слово a_i , а затим се зауставља на том пољу не мењајући остале записе. Оваква Тјурингова машина користи таблицу a_i ,

2. Тјурингова машина r

   Када се машина r примени на произвољну позицију, она помери радно поље за једно место у десно и затим се зауставља не мењајући садржај траке. Оваква Тјурингова машина користи таблицу r,

3. Тјурингова машина l

   Када се машина l примени на произвољну позицију, она помери радно поље за једно место у лево и затим се зауставља не мењајући садржај траке. Оваква Тјурингова машина користи таблицу l.

Ако је у почетној позицији $m = 0$ (радно поље је нулто, односно почетно) онда ће у резултату примене машине l доћи до заустављања због изласка преко краја траке (ПТ). Заустављање машине није изазвано извршавањем радње s, односно није се десило машинско заустављање (МЗ). У завршној позицији радно поље има исти индекс $m = 0$ и садржај траке није промењен.

Пример: $\underset{\uparrow}{]}\, a_4 a_4 a_2 a_0 \cdots \overset{l}{\Rightarrow} \underset{\uparrow}{]}\, a_4 a_4 a_2 a_0 \cdots$

таблица a_i	таблица r	таблица l
$q_0\ a_0\ a_i\ q_1$ $\cdots$ $q_0\ a_t\ a_i\ q_1$ $q_1\ a_0\ s\ q_1$ $\cdots$ $q_1\ a_t\ s\ q_1$	$q_0\ a_0\ r\ q_1$ $\cdots$ $q_0\ a_t\ r\ q_1$ $q_1\ a_0\ s\ q_1$ $\cdots$ $q_1\ a_t\ s\ q_1$	$q_0\ a_0\ l\ q_1$ $\cdots$ $q_0\ a_t\ l\ q_1$ $q_1\ a_0\ s\ q_1$ $\cdots$ $q_1\ a_t\ s\ q_1$

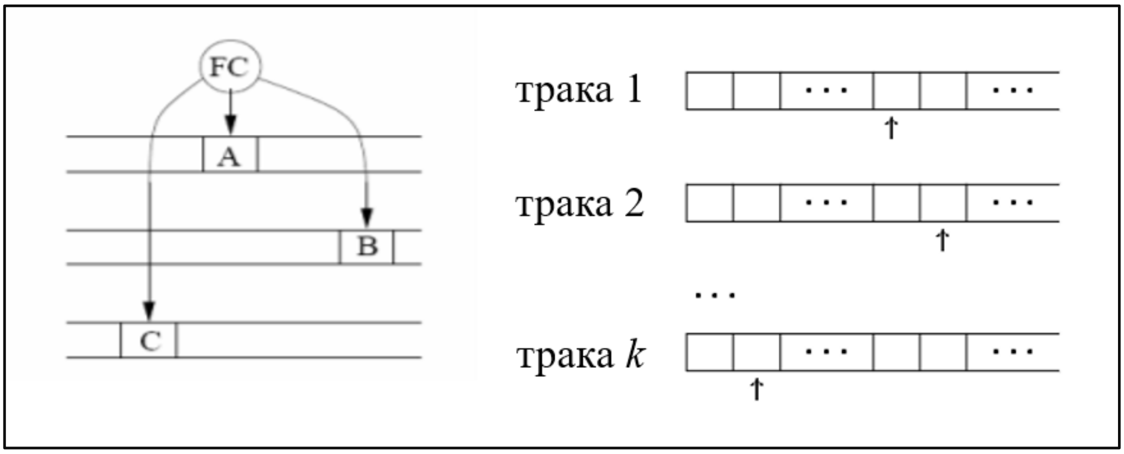
2.1.1.2. ТЈУРИНГОВА МАШИНА СА ТРАКОМ КОЈА ИМА ПОЧЕТАК СА ЛЕВЕ СТРАНЕ

Азбука код машина ове врсте садржи још један специјални знак који служи да се препозна почетна ћелија са леве стране. Преко тог симбола никада се не преписује ни један други симбол, а глава се не сме померати улево. Све функције које се рачунају помоћу Тјурингове машине чија трака има крајње леву ћелију могу израчунати и помоћу стандардне машине (неће се користити део траке који се налази са леве стране ћелије изнад које је позиционирана глава пре почетка извршавања).

2.1.1.3. ТЈУРИНГОВА МАШИНА СА ВИШЕ ТРАКА

Тјурингова машина са више трака састоји се из глава k-те траке и k трака. У једном кораку, у зависности од стања и симбола који је прочитала глава сваке траке, машина може да мења стање, штампа нови симбол у свакој ћелији коју је глава траке скенирала, независно врши померање на свакој траци у лево, у десно, или оставља непромењен положај. Оваква Тјурингова машина се у потпуности може представити Тјуринговом машином са једном траком. Конструисањем машине M_1 са k трака која извршава функцију f и машину са једном траком M_2 која ће симулирати извршавања машине са k трака. Садржај машине M_1 је уписан редом у траку машине M_2. У азбуку се додају нови симболи који означавају почетак и крај сваке трака машине M_1, као и симбол који обавештава о томе да се глава дате траке на машини M_1 налази на датом симболу (на пример подвучени симбол).

Машина M_2 симулира један корак извршавања машине M_1 тако што два пута прелази своје траке са леве на десно и назад. У првом пролазу прикупља информације о k знака који се тренутно налазе испод глава машине M_1 (то су подвучени знаци на траци машине M_2). Да би памћење било омогућено потребно је увести нова стања која одговарају комбинацијама стања машине M_1 и k-торки симбола. На пример ако се машина M_1 налази у стању q, а симболи у ћелији изнад којих су главе трака су редом $o_1, o_2, \cdots, o_k$, одговарајуће стање машине M_2 може да се означи са $q(o_1, o_2, \cdots, o_k)$. На основу свог стања након првог преласка, машина M_2 у другом преласку извршава акцију која симулира разматрани корак рада машине M_1. То се изводи тако што се прелази преко ћелија траке машине M_2 и евентуално мења садржај ћелија у које су уписани подвучени знаци. Промена се огледа или у преписивању текућег подвученог знака новим подвученим знаком (ако је операција писање) или заменом подвучене верзије знака неподвученом и променом једног од суседних знака у подвучену верзију (ако је акција померање главе). Једини проблем настаје када је тренутна позиција управо десни крај записа неке од трака машине M_1, а глава треба да се помери на десно. То значи да се у ћелију изнад које је глава, налази знак за крај. Код машине M_1 то не би представљало проблем, али овде десно од знака за крај нема слободног простора, јер се ту налазе записи садржаја осталих трака машине M_1. Зато се такав проблем превазилази подешавањем жељеног краја траке и померањем свих знакова десно од тог за по једно место у десно.



Слика 2: Апстрактни приказ Тјурингове машине са више трака

2.1.1.4. ТЈУРИНГОВА МАШИНА СА ВИШЕ ГЛАВА

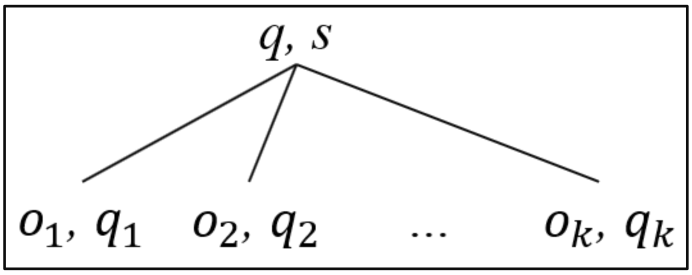
Тјурингова машина са више глава је машина са једном траком и k глава, које читају симболе са исте траке. У једном кораку свака глава врши акцију, при чему не зависе једна од друге. Машина са више глава M_1 се може симулирати машином са једном главом M_2. Нека M_1 има k глава. Тада машина M_2 има $k + 1$ сегмената на траци. Један сегмент садржи садржај траке M_1, а остали сегменти се користе да се означи позиција глава. Један потез машине M_1 се симулира проласком машине M_2 кроз траку како би се добили потребни подаци и онда накнадним проласком како би се извршиле захтеване промене. Непредвиђени проблем који се на овај начин може десити је када две главе показују на исти симбол и желе извршити различите промене. У том случају би требало да буде уведен неки приоритет међу главама које се користе.

2.1.1.5. ДВОДИМЕНЗИОНАЛНА ТЈУРИНГОВА МАШИНА

Тјурингова машина може имати и дводимензионалну траку. Када глава скенира симбол, она може да се помера лево, десно, горе или доле. Најмањи правоугаоник који садржи непразна поља је димензија $m \times n$, што значи да има m колона и n врста. Једнодимензионална Тјурингова машина којом се може симулирати оваква дводимензионална Тјурингова машина ће имати две траке. На једној од ових m колона са n симбола, свака би била представљена као m блокова величине n, раздвојене међусобно маркерима. Друга трака се користи као помоћна трака. Када се глава дводимензионалне Тјурингове машине помери лево или десно, то се симулира у блоку једнодимензионалне Тјурингове машине. Када се глава дводимензионалне Тјурингове машине помери горе или доле, глава једнодимензионалне Тјурингове машине се помера на претходни блок или следећи блок. Да би се померила на исправну позицију у том блоку, користи се друга трака. Ако се m или n повећа, број блокова или величина блокова се увећа.

2.1.1.6. НЕДЕТЕРМИНИСТИЧКА ТЈУРИНГОВА МАШИНА

Понашање Тјурингове машине које је окарактерисано као детерминистичко подразумева да за сваку комбинацију текућег стања и знака је предвиђена само једна акција. Код недетрминистичке Тјурингове машине овај захтев не постоји. У извршавању недетерминистичке Тјурингове машине постоји могућност избора: у случају да за неко стање q и неки знак s постоји више могућих наредби, треба изабрати неку од њих и наставити извршавање. На слици је приказано као део једног дрвета.



Слика 3: Недетерминистички корак у извршавању Тјурингове машине

Недетерминистичка Тјурингова машина дефинисана је као седморка $M = (Q, S, q_0, \square, q_+, q_-, \delta)$, где је:

1. Q – скуп стања
2. S – азбука (скуп симбола)
3. q_0 – почетно стање
4. $\square$ – бланко знак
5. q_+- завршно стање са позитивним одговором на проблем
6. q_- - завршно стање са негативним одговором на проблем
7. $\delta \subseteq (Q \backslash A \times S) \times (Q \times S \times \{L, S, R\})$ – ово се назива транзициона релација, где је $A = \{q_+, q_-\}$, L померање у лево, S стајање, R померање у десно

Овде се и види разлика између детерминистичке и недетрминистичке Тјурингове машине – код детерминистичке последњи члан седморке је транзициона функција $f: Q \times S \rightarrow Q \times S \times \{-1,1\}$.

Грањање у дрвету је коначно, што значи да у сваком кораку извршавања постоји само коначно много опција за избор, док гране представљају могуће редоследе извршавања програма. Детерминистичким Тјуринговим машинама дефинисана је једна класа израчунљивих функција. Код недетерминистичких машина ситуација је донекле измењена. Пре свега, погодне су за давање одговора „да” или „не” на питања облика „Да ли за улазне податке важи нешто?”. Имајући у виду идеју о увођењу нових стања $q_{да}$ и $q_{не}$, заустављање у неком од ових стања има значење позитивног, односно негативног одговора. Снага, или на језику савремених рачунара – брзина, недетерминистичких Тјурингових мачина је последица следеће асиметричне конвенције: машина потврдно одговара на питање ако се бар једно од могућих израчунавања завршава у стању $q_{да}$, док једино окончање свих могућих израчунавања у стању $q_{не}$ значи да је одговор „не”. Ако ни једно израчунавање не доводи до стања $q_{да}$ и бар једно израчунавање не доводи ни до ког завршног стања, недетерминистичка машина дивергира.

Овакав модел машине погодан је за решавање неких сложених проблема. Међутим, недетерминистичка Тјурингова машина се може симулирати детерминистичком машином, тако да се изражајност у смислу онога шта машина може одговорити не мења. Претпоставимо да је M_1 недетерминистичка Тјурингова машина. Одговарајућа детерминистичка Тјурингова машина M_2 систематски прелази све могуће редоследе извршавања машине M_1, најпре дужине 1, па дужине 2, и тако редом. Ово обезбеђује да ни једно могуће коначно извршавање неће бити прескочено. Зато, ако би се машина M_1 у неком тренутку извршавања нашла у стању $q_{да}$, то исто ће пре или касније бити случај и са машином M_2. Ако сви могући редоследи извршавања машине M_1 доводе до стаља $q_{не}$, и машина M_2 наћи ће се у том стању када исцрпи све могућности. Коначно, ако машина M_1

дивергира за дате улазне податке, машина M_2 се неће зауставити. Детерминистичка машина M_2 ће имати три траке: прва трака увек садржи улазни податак и никада се не мења, на другој траци симулираће се извршавање машине M_1, а на трећој ће се, као на неком стеку, памтити низ бројева који представљају приказ изабраних праваца у свим могућим тренуцима избора наредбе која се извршава. На почетку извршавања машине M_2 улазни податак се налази на првој траци, док су преостале две траке празне. У основном циклусу рада машина копира садржај прве на другу траку и користећи садржај треће траке, као упуство за редослед корака, симулира један детерминистички редослед извршавања машине M_1. Очигледно је да детерминистичка M_2 у најгорем случају бар једном посећује сваки чвор дрвета које приказује извршавање недетерминистичке машине M_1. За сада није познато да ли је симулацију могиће извести уз само полиномијални губитак времена. У вези са тиме је чувени проблем „Да ли је P=NP?”.

2.1.1.7. ПРОБАБИЛИСТИЧКЕ ТЈУРИНГОВЕ МАШИНЕ

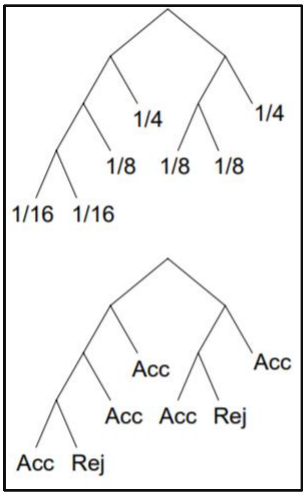
У претходном поглављу видели смо да, иако веома ефикасни, недетерминистички алгоритми нису примењљиви. Стога, потребно је увести детерминистичку апроксимацију ових алгоритама – пробабилистичку Тјурингову машину. Пробабилистичко решење недетерминистичких проблема је практично, али постоји могућност грешке. Недетерминистичке машине, у теорији, испробавају све могуће алгоритме и примењују онај који даје тачан резултат. Пробабилистичка машина бира један алгоритам, чије коначно решење може бити и нетачно, чак иако постоји алгоритам који би као резултат дао тачно решење. Сада се поставља питање: уколико машина може дати и нетачан одговор на проблем, да ли је она корисна? Одговор је да, али под одређеним условима.

Пробабилистичка Тјурингова машина је шесторка $(Q, \Sigma, \delta, \gamma, q_0, q_+, q_-)$, где је:

1. Q је скуп стања
2. Σ је азбука која укључује бланко знак
3. $\delta: Q \times \Sigma \times Q \cup \{q_+\} \cup \{q_-\} \times \Sigma \times \{L, R\} \to [0 \cdots 1]$ је транзициона функција
4. q_0 је почетно стање
5. q_+ је финално стање са тачним одговором
6. q_- је финално стање са нетачним одговором

Примећујемо да се ово разликује од детерминистичке машине по функцији δ, чији резултат представља вероватноћу.

Вероватноћа грешке мора бити строго мања од 0.5, иако може бити веома близу ове вредности. Односно, докле год је вероватноћа грешке мања од оне код нагађања, машина је корисна. Мада се овај алгоритам не чини задовољавајућим, техником амплификације битно се може смањити вероватноћа грешке, што је од велике користи уколико време извршавања није предугачко.



Идеја амплификације је веома једноставна – кроз „слаб” алгоритам се пролази неколико пута и узима се одговор који је већински присутан. Питање је: колико пута морамо проћи кроз „слаб” алгоритам пре него што будемо сигурни да је већински одговор заправо тачан?

На дијаграму са леве стране видимо која је вероватноћа прихватања (Acc), а која одбијања (Rej) одговора.

Према томе, вероватноћа да ће одговор бити тачан (прихваћен) овде износи:

$$\frac{1}{16} + \frac{1}{8} + \frac{1}{4} + \frac{1}{8} + \frac{1}{4} = \frac{13}{16}$$

док вероватноћа одбијања износи:

$$\frac{1}{16} + \frac{1}{8} = \frac{3}{16}$$

Из овога видимо да није потребно много пута извршавати алгоритам како би се дошло до задовољавајућег резултата. Наравно, што је алгоритам слабији, више пута га је потребно извршавати.
Претпоставимо да од слабе пробабилистичке Тјурингове машине P1 са грешком $\varepsilon_1 < 0.5$ желимо да направимо Тјурингову машину P2 са грешком ε_2. Означимо границу грешке са β, а унос w.

1. Рачунамо k за жељену границу грешке β
2. Извршавамо $2k$ симулација P1 за унос w
3. Уколико већина симулација P1 да тачан резултат, он се прихвата. У супротном, одбија се.

Излаз P2 је већински одговор машине P1, а биће нетачан само када је одговор машине P1 нетачан. Односно, ε_2 је вероватноћа да ће $2k$ симулација дати већински погрешан одговор. Са сваким новим извршавањем P1, ова вероватноћа експоненцијално опада. Тачан одговор на питање колика је ова вероватноћа некада може бити компликовано дати, па се налази њена горња граница.

Тврдња 1: Граница грешке ε_2 испуњава услов $\varepsilon_2 \leq \beta$.

Тврдња 2: Време извршавања P2 је полиномијално[2] за w и β.

На основу ових тврдњи, које овде узимамо без доказа, закључујемо да веома брзо можемо у великој мери умањити грешку P2, чак иако је P2 заснована на веома слабој P1.

2.1.1.8. УНИВЕРЗАЛНА ТЈУРИНГОВА МАШИНА

Универзалан програм је онај који је у стању да изврши било који програм P. Када је универзални програм уписан у Тјурингову машину онда се за њега каже да је универзална Тјурингова машина. Овакав програм Тјуринг је саставио 1926. године да би доказао нерешивост неких задатака (као што је на пример халтинг проблем). Универзална Тјурингова машина, у ознаци УТМ, је својеврстан пример програмибилног дигиталног рачунара опште намене са програмом и подацима смештеним у меморију који симулира извршавање осталих Тјурингових машина. Улазни подаци који се смештају на траку универзалне Тјурингове машине су опис неке посебне машине, односно њен програм, и улазни подаци те машине, а резултат извршавања је резултат рада симулиране посебне машине.

2.2. ЕНИГМА

У Другом светском рату немачка војска направила је машину која је омогућавала да ратне трансакције буду заштићене. Машина ради тако што генерише електричну струју када је притиснуто дугме на њој, а затим низ разних механичких догађаја доводи до тога да машина штампа неко друго слово сваког пута када је дугме стиснуто. Оваква машина позната је под називом Енигма, нјмоћније кодирајуће оружје немачких снага. Када би машина доспела у руке непријатељу, било би толико компликација декодирати поруке, које у том времену није било могуће. Енигма креира неколико вишесловних замена, брза је, механизована и прави кодове који су се чинили непробојним. Мења читљиве поруке у нечитљиве. Направљено је око милион пермутација, при чему су свакога дана мењали подешавања. Стављали су роторе на специфичан начин и имали су прстенове који су му

[2] Полиномијално време односи се на време израчунавања проблема, где време m(n) није веће од полиномијалне функције величине проблема n. Записано математички у нотацији великог O: $m(n) = O(n^k)$, где је k нека константа која може зависити од проблема. Математичари користе појам полиномијалног времена у односу на дужину улаза као дефиницију брзог рачунања.

омогућавали да се мењају, што се радило на дневној бази. Каблови су директно повезивали једно слово на друго, што је Енигму чинило имуном на фреквенцијске анализе. Мислили су да нико нема времена или математичке пособности да декодира Енигму. Први пошиљалац и прималац је морао да има Енигма машину са истим подешавањима да би слали, односно читали поруке.
Могла је да се подешава на 1019 начина. За подморнице које су биле једна од већих претњи за британску флоту, још и више јер су имале један ротор више од стандардне Енигма машине, тиме и око 1022 могућих комбинација. Енкриптовану поруку су слали Морзеовом азбуком. Енглези су се надали да ће њихови математичари успети да учине немогуће, односно да декодирају машину. Међу њима је био Алан Тјуринг. Тражило се механичко и математичко решење за то.
1940. године кренули су да тону немачки бродови на Аталантику преко којих су сви са копна комуницирали преко Енигме. Тјуринг је почео са анализом математичких и људских недостатака да би решио проблем. Елиминисао је хиљаде пермутација слова, знајући да Енигма не може криптовати слово само са собом. Тако би померали и подешавали поруке док не нађу положај у коме нема никаквих поклапања имеђу оригиналне немачке и енкриптоване примљене поруке. Енглези су развили своју машину да се супротстави немачкој, која је покушала да уназад одради процес Енигме. Енглеска машина је позната под називом „Бомба”. Поразила је пермутације помоћу познатих делова. Машина пролази све пермутације које је Енигма могла да направи. Најједноставније објашњење за оно шта је Алан тада урадио јесте да је конструисао уређај који је могао да одреди подешавања ротора и контролне табле Енигме како би дешифровао пресретнуте поруке.
Међу тим порукама биле су и оне са највишег врха. Ту су била наређења која су стизала директно од Хитлера, затим извештаји генерала са фронтова, временска прогноза и испоруке залиха. Аланов механизам је 1943. године дешифровао 84.000 порука месечно.
Машина коју је Тјуринг дизајнирао разбила је Енигму тако што је тражила тачна подешавања коришћена за писање поруке уз помоћ одговарајућег шифрата, односно дела отвореног текста. Неки примери речи које су поруке често садржале су „време данас”, „Hail Hitler”, презимена, као и војнички ранг. Била је то огромна и изузетно комплекса машина. Нацисти нису знали да Британци декодирају њихове машине и да су се са њихових бродова домогли начина на који Енигма ради. Сматра се да је дешифровање Енигме скратило трајање рата за око годину дана.

2.2.1. ПРИНЦИП ФУНКЦИОНИСАЊА ЕНИГМЕ

У овом примеру показаћемо како је Енигма коришћена за кодирање појединачних слова. Претпоставићемо да једноставнију Енигму чине три ротора:

- леви (L (I)) ,
- десни (R (III)),
- централни (M (II)).

INPUT	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
ротор I - L	E	K	M	F	L	G	D	Q	V	Z	N	T	O	W	Y	H	X	U	S	P	A	I	B	R	C	J
ротор II - M	A	J	D	K	S	I	R	U	X	B	L	H	W	T	M	C	Q	G	Z	N	P	Y	F	V	O	E
ротор III - R	B	D	F	H	J	L	C	P	R	T	X	V	Z	N	Y	E	I	W	G	A	K	M	U	S	Q	O

Табела 1

Узећемо да је input слово G. На основу табеле 1, примећујемо да ротори утичи на следећу супституцију:

- десни ротор R:

ABCDEF **G** HIJKLMNOPQRSTUVWXYZ → BDFHJL **C** PRTXVZNYEIWGAKMUSQO

– централни ротор M:

AB **C** DEFGHIJKLMNOPQRSTUVWXYZ → AJ **D** KSIRUXBLHWTMCQGZNPYFVOE

– леви ротор L:

ABC **D** EFGHIJKLMNOPQRSTUVWXYZ → EKM **F** LGDQVZNTOWYHXUSPAIBRCJ

рефлектор B	(AY) (BR) (CU) (DH) (EQ) (FS) (GL) (IP) (JX) (KN) (MO) (TZ) (VW)
рефлектор C	(AF) (BV) (CP) (DJ) (EI) (GO) (HY) (KR) (LZ) (MX) (NW) (TQ) (SU)

Табела 2

Кад струја стигне до рефлектора, под претпоставком да се користи стандардни B рефлектор, добијамо:

ABCDE **F** GHIJKLMNOPQRSTUVWXYZ
YRUHQ **S** LDPXNGOKMIEBFZCWVJAT

Како се струја при повратку у роторе враћа у супротном смеру од почетног, користимо инверзне супституције:

– леви ротор:

ABCDEFGHIJKLMNOPQR **S** TUVWXYZ
UWYGADFPVZBECKMTHX **S** LRINQOJ

– централни ротор:

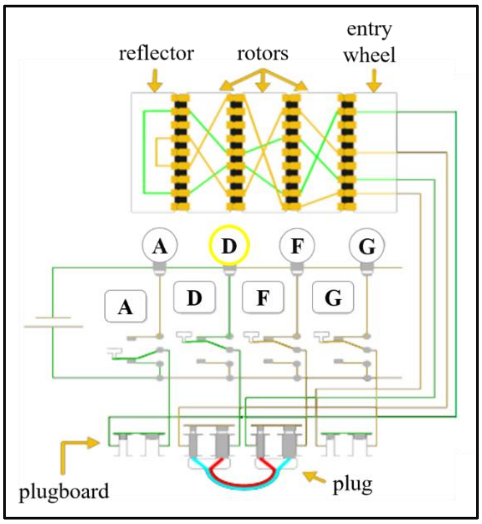
ABCDEFGHIJKLMNOPQR **S** TUVWXYZ
AJPCZWRLFBDKOTYUQG **E** NHXMIVS

– десни ротор:

ABCD **E** FGHIJKLMNOPQRSTUVWXYZ
TAGB **P** CSDQEUFVNZHYIXJWLRKOM

Овим алгоритмом за input G добијамо излаз P.

На слици 4 приказано је како долази до кодирања слова проласком кроз 3 ротора. Објаснићемо како се кодира слово при проласку кроз један ротор, због комплексности објашњења.



Слика 4: Склоп Енигме са плочом и 3 ротора

Узмимо да постоји један ротор P и рефлектор R. Нека је x неко слово. Енигма функционише по следећем принципу: $P^{-1}\left(R\big(P(x)\big)\right)$, односно: неко слово пролази кроз ротор, долази до рефлектора, а затим поново пролази кроз ротор. Имајући у виду да је $R\big(R(x)\big) = x$, односно да је два пута рефлектовано слово једнако самом себи, и поновном применом претходно наведене формуле, добијамо:

$$P^{-1}\left(R\left(P\left(P^{-1}\left(R\big(P(x)\big)\right)\right)\right)\right).$$

Како је $P\left(P^{-1}\left(R\big(P(x)\big)\right)\right) = R\big(P(x)\big)$, претходно наведени израз једнак је:

$$P^{-1}\left(R\left(R\big(P(x)\big)\right)\right) = P^{-1}\big(P(x)\big).$$

2.3. ДОПРИНОС НА ПОЉУ БИОЛОГИЈЕ

Осим машинама, Тјуринг се бавио и биологијом. Педесетих година занимао се за морфогенезу или развој шаблона и облика у организму. Фасциниран присуством Фибоначијевих бројева у распореду листова биљака и шарама на телима животиња, развио је један од првих математичких модела који одговарају на питање како се стварају облици у живом свету. У то време, молекуларна структура ДНК још није била откривена.
По Тјуринговој теорији, у организмима постоје две хемикалије од којих једна мења пигмент коже, а друга спречава мењање. У зависности од интеракције ове две хемикалије, које је назвао морфогенима, постоје и различите шаре на биљкама и животињама. До решења је дошао, а да ни једном није погледао кроз микроскоп. Хемикалије је описао једначинама. Нажалост, оставио је за собом само један рад као допринос биологији – „*Хемијска база морфогенезе*”.

2.4. ФИЛОЗОФСКА РАЗМИШЉАЊА

Истраживања на пољу квантномеханичке физике навела су га да примети да „Када се бавимо атомима и електронима уопште нисмо у могућности да знамо њихово егзактно стање пошто су и наши инструменти састављени од атома и електрона. Зато се илузија, да је сазнање егзактног стања универзума могућа, тада стварно мора срушити на микронивоу. То значи да се руши и теорија да су наши поступци предодређени.” У истом огледу приметиће да „када тело умре, „механизам” тела који држи дух, ишчезава, а дух налази ново тело пре или касније, а можда и одмах.” Борећи се са личним болом, Тјуринг је понудио своје виђење односа детерминизма и слободне воље, тог класичног парадокса.

3. ЗАДАЦИ

Задатак 1. Тјурингова машина ради са азбуком $S = \{0,1,b\}$, где је b празан симбол. Нека је на траци Тјурингове машине уписан низ јединица, а у сва остала поља уписан је празан симбол b. Нека се глава Тјурингове машине налази над крајњим левим знаком 1. Консттруисати програм за Тјурингову машину који на излазу исписује дупло већи број јединица.

Решење:

Идеја је следећа:

$$b\ \overbrace{1\ 1\ \cdots\ 1\ 1}^{x+1}\ b \rightarrow b\ \overbrace{1\ 1\ \cdots\ 1\ 1}^{2x+2}\ b$$

b	1	1	⋯	1	1	b	→	b	1	1	⋯	1	1	b

f(q0,b)=(q-,b,+1)
f(q0,1)=(q1,1,+1)
• *уписујемо 0 на крају низа јединица, као граничник*
f(q1,1)=(q1,1,+1)
f(q1,b)=(q2,0,-1)
f(q2,1)=(q2,1,-1)
f(q2,b)=(q3,b,+1)
• *прву 1 претвори у 0, а у леви бланко знак упише 1*
f(q3,1)=(q4,0,-1)
f(q4,b)=(q5,1,+1)
• *последњу 1 претвори у b и у крајњем левом бланко знаку упише 1*
f(q5,0)=(q6,0,+1)
f(q5,1)=(q5,1,+1)
f(q6,1)=(q6,1,+1)
f(q6,b)=(q7,b,-1)
f(q6,0)=(q7,0,-1)
• *овакво преписивање 1 у крајње леве бланко знаке се врши све док између нула не остану сви бланко знаци*
f(q7,1)=(q8,b,-1)
• *прву 1 коју смо записали као 0 враћамо у 1 и све b-ове између нула претварамо у 1*
f(q7,0)=(q10,1,+1)
f(q8,1)=(q8,1,-1)
f(q8,0)=(q9,0,-1)
f(q9,1)=(q9,1,-1)
f(q9,b)=(q5,1,+1)
f(q10,b)=(q10,1,+1)
f(q10,0)=(q+,b+1)

Задатак 2. Тјурингова машина ради са азбуком $S = \{0,1,b\}$, где је b празан симбол. Нека је на траци Тјурингове машине уписан унаран број, тако што је број x написан као $x + 1$ јединица, а у сва остала поља уписан је празан симбол b. Нека се глава Тјурингове машине налази над крајњим левим знаком 1. Консттруисати програм за Тјурингову машину који рачуна $f(x) = 2x$, и на излазу исписује резултат представљен у унарном запсиу.

Решење:

f(q0,b)=(q-,b,+1)
f(q0,1)=(q1,1,+1)
• *уписујемо 0 на крају низа јединица, као граничник*
f(q1,1)=(q1,1,+1)
f(q1,b)=(q2,0,-1)
f(q2,1)=(q2,1,-1)
f(q2,b)=(q3,b,+1)
• *прву 1 претвори у 0*
f(q3,1)=(q4,0,-1)
f(q4,b)=(q5,b,+1)
• *последњу 1 претвори у b и у крајњем левом бланко знаку упише 1*
f(q5,0)=(q6,0,+1)
f(q5,1)=(q5,1,+1)
f(q6,1)=(q6,1,+1)
f(q6,b)=(q7,b,-1)
f(q6,0)=(q7,0,-1)
• *овакво преписивање 1 у крајње леве бланко знаке се врши све док између нула не остану сви бланко знаци*
f(q7,1)=(q8,b,-1)
• *прву 1 коју смо записали као 0 враћамо у 1 и све b-ове између нула претварамо у 1*
f(q7,0)=(q10,1,+1)
f(q8,1)=(q8,1,-1)
f(q8,0)=(q9,0,-1)
f(q9,1)=(q9,1,-1)
f(q9,b)=(q5,1,+1)
f(q10,b)=(q10,1,+1)
f(q10,0)=(q+,b+1)

Задатак 3. Тјурингова машина ради са азбуком $S = \{0,1,b\}$, где је b празан симбол. Нека је на траци Тјурингове машине уписан бинарни број, а у сва остала поља уписан је празан симбол b. Нека се глава Тјурингове машине налази над крајњим левим знаком, 0 или 1. Консттруисати програм за Тјурингову машину који бинарни број множи са 2.

Идеја је следећа: Цео бинарни број помериђемо за једно место у лево, а затим ћемо последњи знак бинарног броја који је пребачен у лево претворити у нулу.

f(q0,b)=(q-,b,+1)
f(q0,0)=(q1,b,-1)
f(q0,1)=(q2,b,-1)
f(q1,b)=(q3,0,+1)
f(q1,0)=(q3,0,+1)
f(q1,1)=(q3,0,+1)
f(q2,b)=(q3,1,+1)
f(q2,0)=(q3,1,+1)
f(q2,1)=(q3,1,+1)
f(q3,b)=(q4,0,+1)
f(q3,0)=(q4,b,+1)
f(q3,1)=(q4,b,+1)
f(q4,b)=(q+,b,-1)
f(q4,0)=(q1,b,-1)
f(q4,1)=(q2,b,-1)

Задатак 4. Тјурингова машина ради са азбуком $S = \{b, 1\}$. Направити програм који израчунава функцију која се назива i-та пројекција, $f(x_1, x_2, \cdots, x_k) = x_i$, за фиксиране k и i $(1 \leq i \leq k)$. На почетку извршавања, унарне репрезентације бројева $(x_1, x_2, \cdots, x_k)$ су на траци раздвојене једним бланко знаком.

Решење:

Идеја је следећа: Глава се помера до краја записа броја x_{i-1} и при томе брише све јединице, а затим прелази преко записа броја x_i и поново брише записе бројева $x_{i+1}, x_{i+2}, \cdots, x_k$. Машина се затим враћа на почетак броја x_i и стаје.

• *бришемо запис броја* x_1
$f(q0,b)=(q-\ b,+1)$
$f(q0,1)=(q1,b,+1)$
$f(q1,b)=(q1,b,+1)$
$f(q1,1)=(q2,b,+1)$
⋮
• *бришемо запис броја* x_{i-1}
$f(q_j,1)=(q_{j+1},b,+1)$
$f(q_{j+1},b)=(q_{j+1},b,+1)$
$f(q_{j+1},1)=(q_{j+2},b,+1)$
• *прелазимо запис броја* x_i
$f(q_{j+2},1)=(q_{j+2},1,+1)$
$f(q_{j+1},b)=(q_{j+3},b,+1)$
• *бришемо запис броја* x_{i+1}
$f(q_{j+3},1)=(q_{j+4},b,+1)$
$f(q_{j+4},b)=(q_{j+4},b,+1)$
$f(q_{j+4},1)=(q_{j+5},b,+1)$
⋮
• *бришемо запис броја* x_k
$f(q_l,1)=(q_{l+1},b,+1)$
$f(q_{l+1},b)=(q_{l+1},b,+1)$
$f(q_{l+1},1)=(q_s,b,-1)$
• *враћамо се на почетак записа броја* x_i
$f(q_s,b)=(q_s,b,-1)$
$f(q_s,1)=(q_{s+1},1\ -1)$
$f(q_{s+1},b)=(q+,b,+1)$
$f(q_{s+1},1)=(q_{s+1},1,-1)$

Задатак 5. Тјурингова машина ради са азбуком S = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, b\}$. На почетку је уписан један бинаран број, а око њега симболи b. Конструисати програм за Тјурингову машину који претворити бинарни број у декадни.

Решење:

Идеја је следећа:

b	b	1	1	1	b	b	→	b	7	b	0	0	b	b	→	b	b	b	7	b	b	b

f(q0,b)=(q-,b,+1)
f(q0,0)=(q+,0,+1)
f(q0,1)=(q1,1,+1)

- *читамо број*

f(q1,1)=(q1,1,+1)
f(q1,0)=(q1,0,+1)
f(q1,b)=(q2,b,-1)

- *смањујемо број у бинарном запису за 1*

f(q2,1)=(q3,0,-1)
f(q2,0)=(q6,1,-1)

- *када је број непаран, прелазимо у стање q3*

f(q3,0)=(q3,0,-1)
f(q3,1)=(q3,1,-1)
f(q3,b)=(q4,b,-1)

- *након сваког смањења броја у бинарном запису за 1, број у декадном запису повећавамо за 1*

f(q4,b)=(q5,1,+1)
f(q4,0)=(q5,1,-1)
f(q4,1)=(q5,2,+1)
f(q4,2)=(q5,3,+1)
f(q4,3)=(q5,4,+1)
f(q4,4)=(q5,5,+1)
f(q4,5)=(q5,6,+1)
f(q4,6)=(q5,7,+1)
f(q4,7)=(q5,8,+1)
f(q4,8)=(q5,9,+1)
f(q4,9)=(q4,0,-1)
f(q5,b)=(q1,b,+1)
f(q5,0)=(q5,0,+1)
f(q5,1)=(q5,1,+1)
f(q5,2)=(q5,2,+1)
f(q5,3)=(q5,3,+1)
f(q5,4)=(q5,4,+1)
f(q5,5)=(q5,5,+1)
f(q5,6)=(q5,6,+1)
f(q5,7)=(q5,7,+1)
f(q5,8)=(q5,8,+1)
f(q5,9)=(q5,9,+1)

- *када је прочитан број паран, прелазимо у стање q6*

f(q6,1)=(q3,0,-1)
f(q6,0)=(q7,1,-1)

- *коментар 1*

f(q7,0)=(q7,1,-1)
f(q7,1)=(q3,0,-1)
f(q7,b)=(q8,b,+1)
f(q6,b)=(q8,b,+1)

- *када се цео број конвертује, у бинарном запису остаће само 0, које се претварају у 1*
- *како би се добило решење у жељеном формату, њих претварамо у b симболе*

f(q8,1)=(q8,b,+1)
f(q8,b)=(q+,b,+1)

Коментар 1: Ако је $q6 = 0$ може доћи до ситуације да 1000 постане 1111, стога, уводимо стање $q7$ које ће све наредне 0 претварати у 1, а при проналажењу прве 1 вратиће се у стање $q3$, које ће прочитатти број до краја (у смеру с десна на лево).

З А К Љ У Ч А К

Алан Тјуринг се сматра оцем модерног рачунарства. Утро је пут вештачкој интелигенцији и прогамирању каквог га данас познајемо. Његова истраживања и радови на пољу математике и биологије обележили су развој ових наука у 20. веку. Декодирањем Енигме задобио је поштовање и дивљење целог света, а своје име уписао је у књигу великана чији је допринос омогућио бржи напредак човечанства и оформио начин размишљања модерног човека.

Л И Т Е Р А Т У Р А

- Aho, Hopcroft, Ullman, The design and analysis of computer algorithms
- Зоран Огњановић, Ненад Крџавац, „Увод у теоријско рачунарство”, Београд – Крагујевац 2004.
- Variations of Turing Machines - nplet
- Тјурингове машине – Алан Тјуринг, портал Заштита података
- Од криптографије до вештачке интелигенције – Алан Тјуринг кроз своја највећа открића, startit.rs
- Википедија
- https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/lecture-notes/MIT6_045JS11_lec17.pdf
- http://axon.cs.byu.edu/Dan/252/misc/252-ProbTMs.pdf?fbclid=IwAR1ZuGNKT2r8Q_uBok4pLsw_IQ1yB8JWWy3J3J9N2nXXs7yKNVjZEjEteIk
- https://hackaday.com/2017/08/22/the-enigma-enigma-how-the-enigma-machine-worked/
- https://www.codesandciphers.org.uk/enigma/example1.htm