1.Compareandcontrastthecapability,representation,acceptanceandtransitionoffinite

automata,pushdownautomataandTuringmachine.

Characteristics FA TM PDA

Capacity Recognizesregularlanguages,

limitedcomputationalpower.

Universalmodelof

computation,can

recognizerecursively

enumerablelanguages

Recognizescontext-free

languages,more

computationalpowerdue

tostack

Representation Finiteautomatacanbe

representedinvariousways:

-Transitiondiagrams:States

arerepresentedasnodes,and

transitionsarerepresentedas

labeledarrowsbetweennodes.

-Transitiontables:Atabular

representationofthetransition

function,whererowsrepresent

states,columnsrepresent

inputsymbols,andtheentries

specifythenextstates.

Transitiondiagramsor

tableswithstates,

inputsymbols,and

tapemovements.

PDAsaretypically

representedusing

transitiondiagrams,similar

tofiniteautomata,where

statesarenodesand

transitionsarelabeledwith

theinputsymbol,stack

symbolpopped,andstack

symbolspushed.The

transitionfunctionis

depictedusingthese

labeledarrowsbetween

states.

Acceptance Afiniteautomatonacceptsan

inputstringifitreachesan

acceptingstateafter

processingtheentirestring.

ForDFAs,aninputisaccepted

iftheuniquecomputationends

inanacceptingstate.For

NFAs,aninputisacceptedifat

leastoneofthepossible

computationsendsinan

acceptingstate.

Acceptance:Turing

Machinesacceptinput

stringsbyreachinga

finalstatedefinedas

anacceptingstate.If

themachinehaltsin

anacceptingstate,the

inputstringis

consideredaccepted.

Acceptance:PDAscan

acceptstringsbyeither

reachinganacceptingstate

orbyhavinganempty

stackafterprocessingthe

entireinput.FAs,onthe

otherhand,onlyaccept

stringsbyreachingan

acceptingstate

Transition TransitionFunction

-δ:Q×Σ->Q,mapscurrent

stateandinputsymboltonext

state

TransitionFunction

-:Q×(Σ∪{ε})×(Γ

∪{ε})=P(Q×(Γ∪{ε})),

mapscurrentstate,

TransitionFunction-δ:

Q×Σ×Γ->Q,mapscurrent

state,tapesymbol,and

tapeheadpositiontonext

InputSymbol=

-Inputsymbolisusedto

determinethenextstate.

StackSymbol=

-Nostacksymbolisused

NextState =

-Nextstateisdeterminedby

thecurrentstateandinput

symbol

TapeMovement=

Notapemovementisinvolved.

inputsymbol,and

stacksymboltonext

state

InputSymbol=

-Inputsymbolisused

todeterminethenext

state,andstack

symbolisusedto

determinethenext

stackoperation

StackSymbol=

-Stacksymbolisused

todeterminethenext

stackoperation.

NextState=

-Nextstateis

determinedbythe

currentstate,input

symbol,andstack

symbol.l

TapeMovement=

Notapemovementis

involved..

state

InputSymbol=

-Tapesymbolisusedto

determinethenextstate,

andtapeheadpositionis

usedtodeterminethenext

tapemovement.

StackSymbol=

-Nostacksymbolisused

NextState=

-Nextstateisdetermined

bythecurrentstate,tape

symbol,andtapehead

position

TapeMovement=

Tapemovementisinvolved,

withthetapeheadmoving

leftorrightbasedonthe

transitionfunction

2.DefineCFG,Derivation(Bothleftandrightderivation)andparsetree.Demonstrateeachtopics

usingfollowingCFGwhereG:(V,T,E,P)withV={E,I},T={a,b,c,+,\*,(,)}andP:

E->T|E+T

T->F|E+F

F->I|(E)

I->a|b|c

OUTPUTSTRING:(((a+b)\*c))+a+b)

Context-FreeGrammar(CFG):

Acontext-freegrammar(CFG)isaformalgrammarconsistingofasetofproductionrulesthat

describehowtogeneratestringsinalanguage.Itconsistsofasetofvariables(V),asetof

terminals(T),asetofproductionrules(P),andastartsymbol(S).InaCFG,eachproduction

rulespecifieshowavariablecanbereplacedbyasequenceofvariablesandterminals.

ACFGGisa4-tuple(V,T,P,S)where:

-Visafinitesetofvariables(non-terminals)

-Tisafinitesetofterminals(disjointfromV)

-PisafinitesetofproductionsoftheformA->αwhereA∈Vandα∈(V∪T)\*

-S∈Visthestartsymbol

CFG:

G=(V,T,P,E)where

V={E,T,F,I}

T={a,b,c,+,\*,(,)}

P:

E->T|E+T

T->F|E+F

F->I|(E)

I->a|b|c

Derivation:

Aderivationisasequenceofstringsα1,α2,...,αnsuchthat:

-α1=S(thestartsymbol)

-αnisastringofterminals(theoutputstring)

-αi⇒αi+1byreplacingavariableAinαiwiththerightsideofaproductionA->β

LeftDerivation:Aleftderivationisasequenceofgrammarruleapplicationswheretheleftmost

non-terminalinthesententialformisalwaysreplacedateachstep.Itstartswiththestart

symbolandproceedsbyreplacingtheleftmostnon-terminalineachstepuntilonlyterminals

remain.

LeftDerivation(⇒lm):Replacetheleftmostvariableateachstep

Exampleleftderivationfor(((a+b)\*c))+a+b):

RightDerivation:Arightderivationissimilartoaleftderivationbutinvolvesreplacingthe

rightmostnon-terminalinthesententialformateachstep.

RightDerivation(⇒rm):Replacetherightmostvariableateachstep

Examplerightderivationfor(((a+b)\*c))+a+b):

ParseTree:

Aparsetreeisagraphicalrepresentationofthederivationofastringinacontext-freegrammar.

Itshowshowthestartsymbolisexpandedintoterminalsandnon-terminalsfollowingthe

productionrulesofthegrammar.Inaparsetree,eachnoderepresentsavariableorterminal

symbol,andtheedgesrepresenttheapplicationofaproductionrule.

Aparsetreeisagraphicalrepresentationofaderivation,where:

-Eachinteriornodeislabeledbyavariable

-Eachleafislabeledbyaterminalorε

-ThechildrenofanodelabeledAarethesymbolsontherightsideofaproductionA→α

-Therootislabeledbythestartsymbol

-Theleaves,readleft-to-right,givetheyieldoftheparsetree

Theparsetreefortheoutputstring(((a+b)\*c))+a+b)is:

3.Defineambiguousandshowthatthefollowinggrammarisambiguous.

E->aSbS|bSaS|ε

AmbiguousGrammar

Anambiguousgrammarisacontext-freegrammarforwhichthereexistsastringthatcanhave

morethanoneleftmostderivationorparsetree.Thismeansthatthegrammarcangenerate

multipleparsetreesforasingleinputstring,leadingtoambiguityintheparsingprocess.

4.ConvertthegrammarintoChomskynormalform.

S->abAB|CA

A->bAB|ε

B->AB|A|ε

C->Cc

5.ForthelanguageL={anb²n:n>=0}

i.DrawpushdownAutomata

ii.DrawTuringmachine

iii.ChecktheacceptanceofstringabbusingInstantaneousdescriptionforbothPDAandTM

6.WhatlanguageisacceptedbythePDAM=({q0,q1,q2},{a,b},{a,b,z},,q0,{q2})

withtransitions

(q0,a,z)={(q1,a),(q2,ε)}

(q1,b,a)={(q1,b)}

(q1,b,b)={(q1,b)}

(q1,a,b)={(q2,ε)}

PDADescription:

-States:{q0,q1,q2}

-InputAlphabet:{a,b}

-StackAlphabet:{a,b,z}(wherezistheinitialstacksymbol)

-StartState:q0

-AcceptingStates:{q2}

Transitions:

1.(q0,a,z)={(q1,a),(q2,ε)}

2.(q1,b,a)={(q1,b)}

3.(q1,b,b)={(q1,b)}

4.(q1,a,b)={(q2,ε)}

InitialStack:

-Thestackstartswiththesymbol'z'.

TransitionExplanation:

1.Fromq0:

-Onreading'a'with'z'onthestack:

-Movetoq1andpush'a'ontothestack.

-Alternatively,movetoq2anddonothingwiththestack(acceptingstate).

=(q0,a,z)={(q1,a),(q2,ε)}

2.Fromq1:

-Onreading'b'with'a'onthestack:

-Movetoq1andpush'b'ontothestack.

-Onreading'b'with'b'onthestack:

-Movetoq1andpush'b'ontothestack.

-Onreading'a'with'b'onthestack:

-Movetoq2anddonothingwiththestack(acceptingstate).

=.(q1,b,a)={(q1,b)}

(q1,b,b)={(q1,b)}

(q1,a,b)={(q2,ε)}

AcceptancebyStateq2:

-ThePDAcanmovetostateq2fromq0directlyonreading'a'with'z'onthestack.

-ThePDAcanmovetostateq2fromq1onreading'a'with'b'onthestack.

LanguageCharacteristics:

-ThePDAstartsinq0andcantransitiontoq2directlyonreading'a',acceptinganystring

startingwith'a'.

-ThePDAtransitionstoq1whenreading'a'andcontinuesprocessing'b'withstackoperations.

-Inq1,thestackgrowsasitprocesses'b'symbols.

-ThePDAcantransitiontotheacceptingstateq2ifitencountersan'a'whilethereisa'b'onthe

stack,balancingsomepreviouslyread'b's.

ThisimpliesthatthePDAacceptsstringswhere:

-Thereisatleastone'a'.

-Thenumberof'a'sreadisgreaterthanorequaltothenumberof'b'sread.

ThelanguageacceptedbythePDAMis

7.ConstructPDAthatacceptthelanguage

8.DrawTMthatcomputesadditionoftwounarynumbers.

9.Proofhaltingproblemisundecidableproblem.

HaltingProblem

Thehaltingproblemisperhapsoneofthemostfamousproblemsintheoreticalcomputer

science.Putsimply,thehaltingproblemaskswhetherthecomputationofaTuringmachine

haltsonsomegiveninputword.

Thehaltingproblemisdefinedtobetheproblem,foranyprogramRrunonalegalinputof

decidingifRhalts(terminates)onX.

HaltingProblemisUndecidable-proof

Proofbycontradiction.

Supposethereisa'program',H(insomeappropriatelanguage)thattakesasitsinputs

thetextofanotherlegalprogramRinthesamelanguage,andaninputXtoR,andafter

somefinitetime,itoutputs'yes'ifRhaltsonX,and'no'ifRdoesnot.

NowassumethatthereissomeotherprogramSthatdoesthefollowing:

-TakesasitsinputthetextofanylegalprogramR.

-Makesacopyofit.

-PassesbothcopiestothehypotheticalprogramH,whichisthusbeingasked'DoesR

haltwhenitisgivenitsowntextasinput?'

-GoesintoaninfiniteloopiftheoutputofHis'yes'(ieifRdoeshaltontext-inputR),

andhaltsiftheoutputofHis'no'(ieifithasdiscoveredthatRdoesnothaltonitsowntextas

input).

NowletShaveitsowntextasinput:

First,assumethatSdoeshalt:

H(S,S)=yes,soLgoesintoaninfiniteloop,andhenceS,theprogram

thatcontainsL,alsofailstoterminate.TheassumptionthatShaltshasled

totheconclusionthatitdoesn't.

Thenassume,thatSdoesnothalt:

H(S,S)=no,soSthenhalts.Now,evenmorestraightforwardly,the

assumptionthatSdoesnothalthasledtotheconclusionthatitdoes.

Clearlythereissomethingwrong!Henceitisprovedbycontradictionthathaltingproblemis

undecidable.

10.DefineP,NP,NPhardandNPcomplete.

P(PolynomialTime)

TheclassP,alsoknownasthesetofdecisionproblemsthatcanbesolvedbyadeterministic

Turingmachineinpolynomialtime,representsthesetofallproblemsthatcanbesolvedin

polynomialtime.Thismeansthatgivenaninstanceoftheproblem,theansweryesornocanbe

decidedinpolynomialtime.Forexample,theproblemofdeterminingwhetheraconnected

graphcanbecoloredusingtwocolorssothatnoedgeismonochromaticisinP.

NP(NondeterministicPolynomialTime)

TheclassNP,whichstandsfor"NondeterministicPolynomial-time,"representsthesetofall

decisionproblemsforwhichtheinstanceswheretheansweris"yes"haveproofsthatcanbe

verifiedinpolynomialtime.Thismeansthatifsomeonegivesusaninstanceoftheproblemand

acertificate(sometimescalledawitness)totheanswerbeingyes,wecancheckthatitis

correctinpolynomialtime.Forinstance,theproblemofdeterminingwhetheragivenintegercan

befactoredintosmallerintegersisinNP.

NP-Hard

AproblemisNP-hardifallproblemsinNParepolynomialtimereducibletoit.Thismeansthatif

asolutiontotheNP-hardproblemcanbefoundinpolynomialtime,thenallproblemsinNPcan

alsobesolvedinpolynomialtime.NP-hardproblemsdonotnecessarilybelongtoNP,andthey

canbedecisionproblemsoroptimizationproblems.Forexample,thehaltingproblem,which

determineswhetheragivenprogramwillhaltgivenaspecificinput,isNP-hard.

NP-Complete

AproblemisNP-completeifitisbothNP-hardandinNP.ThismeansthattheproblemisinNP

andallotherproblemsinNPcanbereducedtoitinpolynomialtime.NP-completeproblemsare

themostdifficultproblemsinNP,andsolvinganyoneofthemwouldimplythatallproblemsin

NPcanbesolvedinpolynomialtime.Forinstance,theproblemofdeterminingwhetheragiven

BooleanformulaissatisfiableisNP-complete.

Reference

-ComputationalComplexityTheoryManojPokharelCentralDepartmentofComputerScience

andIT,TribhuvanUniversityJune19,2020

-stackoverflow.com

-www.geeksforgeeks.org

-slideshare.com