

Lecture 8

Friday, February 1, 2019 9:02 AM

DFS - uninformed
doesn't halt unless that property is added

BFS - uses queue

shortest soln \neq shortest procedure
 \downarrow \downarrow
shortest path fewest computational steps

expanding a path vs. unfolding recursive calls (NOT backtracking)

A* search
if h is monotone, A* never repeats a state
monotone heuristics are non-increasing

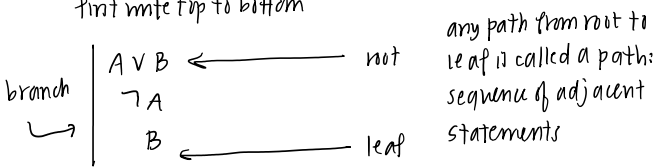
PC Trees

$P, P \rightarrow Q \vdash Q$

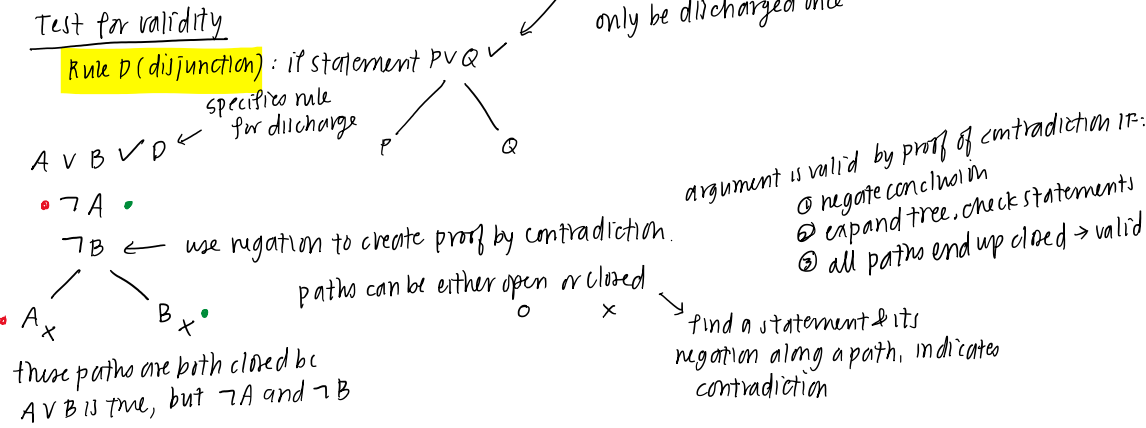
P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

relevant valuation when P is T and $P \rightarrow Q$ is T
 Q is also T \rightarrow checking all combos
is a BRUTE FORCE METHOD

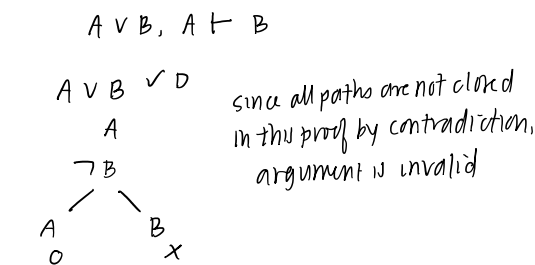
Basics of PC Trees:
 $A \vee B, \neg A \vdash B$
first write top to bottom



branches may split off into their own statements guaranteed to halt (finite length)
tree rule:
① discharges a statement
② allows us to produce $1+$ branches
check symbolizes discharge of statement and each statement can only be discharged once



Test for validity (this argument is invalid)



Negated disjunction (ND)

$\neg(P \vee Q) \checkmark$ goes along the same branch (all conjunction rules)
 $\neg P$ disjunctive statements split a branch
 $\neg Q$

Conjunction (C)

$P \wedge Q \checkmark$
 P
 Q

Negated conjunction (NC)

$\neg(P \wedge Q)$
 $\neg P$ $\neg Q$

Implication (I)

$P \rightarrow Q \checkmark$
 $\neg P$ Q

Negated Implication (NI)

$\neg(P \rightarrow Q) \checkmark$
 P
 $\neg Q$

Double Negation (DN)

$\neg \neg P$
 P

two potential issues of PC trees
① Does PCT properly represent PC?
② What properties does PCT have?
 \hookrightarrow efficient (not a brute force soln)

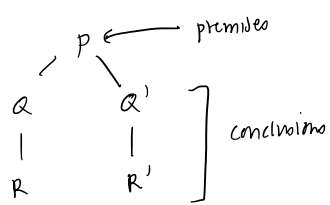
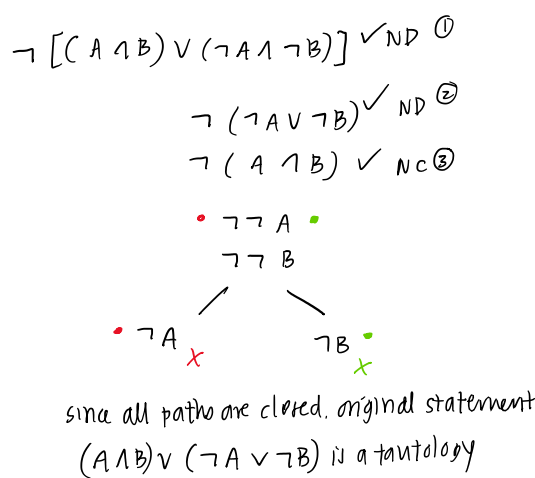
Rules Involving the closure of tree paths

negation 1: P & $\neg P$ in the same path
negation 2: $\neg P$ and then P in the same path
constructed def for closed; anything else is open

PCT can be used for more than just validity tests:
- tautology
- satisfiability
- ...

Test for Tautology

let $S \equiv (A \wedge B) \vee (\neg A \vee \neg B)$
show S is a tautology using PCT
① negate S
② show all paths are closed $\therefore S$ is a tautology



a given tree Rule R is correct iff
 $(\{P\}^v = T) \Rightarrow (\{S\}^v = T)$
for $S \in L$, where L is one of our tree's lists of conclusions

definition: R is complete iff the premises of on rtree are true whenever $\{S\}^v = T$ for all $S \in L$.

we're guaranteed to halt w/ PCT rules \leftarrow why?