Homework #2

CS 474: Fall 2024

Due on Monday Oct 14, 11:59 PM Central

Homework Policy: You are allowed to collaborate with your classmates, but declare it in your submission. Even if you work on problems together, each student must write up their solution individually in their own words in their submission. The CS 473 course's page on academic integrity is a handy reference: `https://courses.engr.illinois.edu/cs473/sp2023/integrity.html`. Late submissions are not allowed. If you have personal circumstances that will prevent you from submitting on time, write to the course staff as soon as possible and we can try to work with you.

Points: Problem 1: 20 points, Problem 2: 20 points, Problem 3: 20 points; Total: 60 points

Theme This assignment is all about resolution. You will be asked to prove formulas valid, show that formulas are satisfiable, and think about the meaning of resolution proofs.

For both problems, work out the resolution proof on paper and then use the tool available at https://github.com/muraliadithya/logichw/tree/main/resolution You just need to download the file *resolution.py* and execute it on any reasonable Python version on your machine. The tool allows you to save the proof. For exercises in this assignnment, simply submit a printout of your interaction with the tool, or submit the proof file.

Problem 1. Prove that the following propositional formula is valid using resolution:

$$(p \wedge (p \Rightarrow q)) \Rightarrow q$$

Present the various steps carefully and clearly: conversion of validity to satisfiability, conversion to CNF (perhaps using the Tseitin transformation from your earlier homework), the resolution proof showing unsatisfiaility by deriving the empty set of clauses, and a brief comment on why this shows the original formula valid. Also execute the proof in the resolution proof tool and submit the proof, giving a link to a publicly accessible github and also give submit on Gradescope a printout of the interaction.

Problem 2.

(a) In this problem, we want to show using resolution that the following formula φ is satisfiable:

$$(q \vee \neg r) \wedge (\neg p \vee r) \wedge (\neg q \vee r \vee p) \wedge (p \vee q \vee \neg q) \wedge (\neg r \vee q)$$

This formula is already in CNF. Apply resolution directly and do not introduce any new variables. Describe the steps of resolution clearly. And resolve and add clauses exhaustively, until you are left with a set of clauses that cannot be resolved any further.

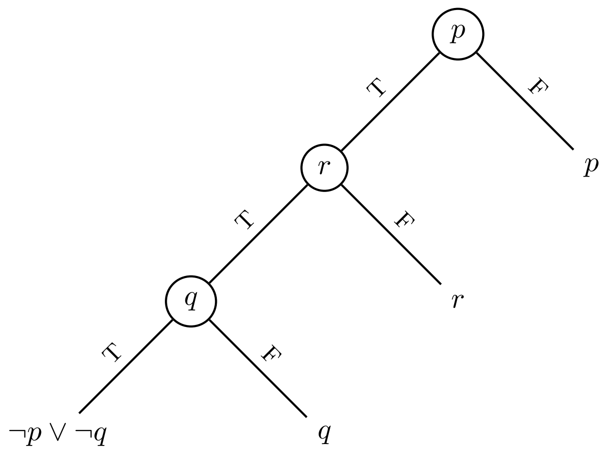
(b) Encode the original formula φ in Part (a) of this problem as well as the formula ψ you obtain at the end of resolution (i.e., the conjunction of all the clauses at the end that you could not resolve any further) in Z3. Using Z3, show that (1) φ is indeed satisfiable, and (2) φ and ψ are equivalent. The above are two separate tasks to encode in Z3. Note that since resolution only introduces implied clauses, we expect φ and ψ to be equivalent.

As before, describe your solution clearly at a high level (mathematically). Copy-paste entire Z3 input files in your submission! And also upload your encodings publicly to GitHub as separate files and provide the links in your submission.

Problem 3. Let us revisit the decision-tree view of resolution that we learned in class. Consider the formula α given by

$$p \wedge r \wedge (\neg p \vee \neg q) \wedge q$$

The following decision tree evaluates α with variable order $p > r > q$:



The idea is that we start to set values to propositions (we do p first, then r, and then q) and branch on the choices, stopping on a path as soon as we falsify some clause (i.e., the partial valuation corresponding to the path falsifies some clause). Since the outer structure of α is conjunctive, falsifying one clause is enough to falsify the formula. In this way, if all paths are "blocked", i.e., they end in a falsified clause, then we can conclude that any assignment of values to propositions will falsify α. This shows that α is unsatisfiable.

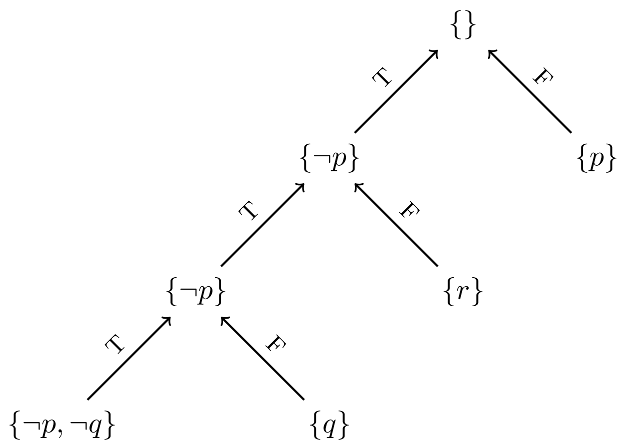
From the above tree, one can also derive a resolution proof. The idea is: in a bottom-up fashion, keep applying one step of resolution between sibling nodes. For siblings that branch on a proposition p, if both of them mention p, then you can resolve with respect to p. If this does not happen, then choose a sibling that **does not mention** p (since that will reduce the number of propositions involved). You can get to the root if and only if all paths are blocked. In this case, we will be able to get to the root in three steps (recall that we represent disjunctions as sets):

1. Take the children of node q: $\{\neg p, \neg q\}$ and $\{q\}$, and resolve them on q to obtain $\{\neg p\}$.

2. Now, the left child of node r is $\{\neg p\}$ and the right child is $\{r\}$. We choose the sibling without r in it, making it $\{\neg p\}$.

3. At this point, the left child of the root is $\{\neg p\}$ and the right child is $\{p\}$. We resolve these on p to obtain $\{\}$.

We can view the resolution proof as follows:



Task. For this problem, use a decision tree to derive a similar proof of resolution as we did for α for the CNF formula that you came up with in Problem 1. Fix a variable order of your choice. Clearly state the variable order, show the decision tree, and the resolution proof derived from it. Your explanation should correspond the decision tree with the resolution proof. Do not simply give a decision tree and a resolution proof with no other context.

Aside: The above can be turned into a proof of why resolution is complete. Take a decision tree evaluating variables in some order. It's clear that the paths will get eventually "blocked" with some clause. From this a resolution proof can always be derived using the procedure mentioned above. See the following link for such a proof: `http://www.cs.toronto.edu/~toni/Courses/438/Mynotes/res.pdf`.