INFO0054-1
Programmation fonctionnelle

Project 2022-2023

Christophe Debruyne
(c.debruyne@uliege.be)

Some details

- Project description on eCampus
- Groups of <u>three</u> people
  - <u>Three groups of two people will become two groups of three</u>
- Deadlines names of your groups: 2022-10-20 at 23:59
- Deadlines of your deliverables: 2022-12-02 at 23:59

Context: semantic tableau for propositional logics

Semantic Tableaux are a well-known and popular algorithm to *decide* (decision procedure) whether a set of formulae is *satisfiable* (has a model, or "can happen").

In propositional logic, truth tables can become quite large. We have 2^N possibilities where N corresponds with the number of propositions p, q, ... used in our set.

Propositional logics: Some background

- Given *A*, a countably infinite set of elements of proposition symbols p, q, r, … which we call the alpha set or atomic formulas.
- Given $O = O_1 \cup O_2$ where
  - $O_1 = \{\neg\}$ the set of unary logical operators
  - $O_2 = \{\wedge, \vee, \rightarrow\}$ the set of binary logical operators

- The set of formulas is *inductively* defined by the following rules:
  - Any element of A is a formula;
  - If φ is a formula, then $\neg\varphi$ is a formula;
  - If ψ and φ are formulas, then $(\psi \vee \varphi)$, $(\psi \wedge \varphi)$, and $(\psi \rightarrow \varphi)$ are formulas;
  - <u>Nothing else is a formula.</u>

Semantic Tableaux

The result of applying the *semantic tableau* is a structure containing one or more *tableaux.* A tableau ("panel") contains a set of formulas. One can call a tableau of a semantic tableau a *branch.*

Open vs. Closed tableaux ~~and branches~~:

- ***A tableau is said to be closed*** if the branch contains a contradiction, that is: the branch contains both a φ and a $\neg\varphi$
- ***A tableau is said to be open*** if all binary operators have been eliminated (i.e., it only contains proposition symbols or their negation) and it does not contain a contradiction.
- ***A semantic tableau is said to be closed*** if all its tableaux are closed.
- ***A semantic tableau is said to be open*** if at least one tableau is open.

Satisfiability of a set of formulas

A set of formulas is called satisfiable if there is at least one model. I.e., at least one mapping of truth values to proposition symbols that make all the formulas in the set true.

Is $\mathrm{F} = \{p \lor q, r, \neg q\}$ satisfiable?

Start:

$$T_1 = \{p \lor q, r, \neg q\}$$

Is T_1 closed? No

Does T_1 contain formulas we can eliminate? Yes: $p \lor q$.

The formula $p \lor q$ is true if p is true (no matter the truth value of q) OR q is true (no matter the truth value of p). The tableau thus "splits". We remove $p \lor q$ and copy the table. We place p in the original table, and q in the copy.

$$\begin{aligned} T_1 &= \{p, r, \neg q\} \\ T_2 &= \{q, r, \neg q\} \end{aligned}$$

Satisfiability of a set of formulas

$$\begin{aligned} T_1 &= \{p, r, \neg q\} \\ T_2 &= \{q, r, \neg q\} \end{aligned}$$

- T_1
  - Is T_1 closed? No
  - Does T_1 contain formulas that can be eliminated? No
  - T_1 is therefore open
- T_2
  - Is T_2 closed? Yes

- At least one tableau is open. T_1 contains at least one model and the set $\mathrm{F} = \{p \vee q, r, \neg q\}$ is therefore satisfiable!

Satisfiability of a set of formulas

$$\begin{aligned} T_1 &= \{p, r, \neg q\} \\ T_2 &= \{q, r, \neg q\} \end{aligned}$$

How many models does T_1 contain and why?

- 1, as it assigns a truth value for each proposition symbol F
- $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

Can a tableau contain multiple models? Discuss.

Elimination rules

$\neg\neg\phi$ is true/false if ϕ is true/false

$(\phi \vee \pi)$ is true if ϕ is true OR π is true
$\neg(\phi \vee \pi)$ is true if ϕ is false AND π is false

$(\phi \wedge \pi)$ is true if ϕ is true AND π is true
$\neg(\phi \wedge \pi)$ is true if ϕ is false OR π is false

$(\phi \rightarrow \pi)$ is true if ϕ is false OR π is true
$\neg(\phi \rightarrow \pi)$ is true if ϕ is true AND π is false

ϕ	π	$(\phi \rightarrow \pi)$
0	0	1
0	1	1
1	0	0
1	1	1

Logical consequence and tautologies

A formula ψ is said to be a logical consequence (or valid under) of a set of hypotheses F

- If all models of F are also models of ψ
- Or, if $F \cup \{\neg\psi\}$ is unsatisfiable!
- Why is that? Does this technique remind you of something?

- If $F \cup \{\neg\psi\}$ is satisfiable, that means we have found **counterexamples**
    - A counterexample is a model that makes F true, but ψ false!

Tautologies and contradictions

A tautology ψ is a formula that is always true.

- In other words, $\{\neg\psi\}$ is unsatisfiable

A contradiction ψ is a formula that is always false

- In other words, $\{\psi\}$ is unsatisfiable

Models and counterexamples

An open tableau may contain multiple models.

Of instance, if F uses the propositions p, q, and r and an open branch only mentions p and $\neg q$. Then that branch contains truth values for p and q independent from the proposition r

It thus contains two models:

- $v_1(p) = 1, v_1(q) = 0, v_1(r) = 1$
- $v_2(p) = 1, v_2(q) = 0, v_2(r) = 0$

Open tableaux may also share models.

The project:
Part 1: the function `semtab` (30%)

Define an ADT allowing you to represent formulas of the form

- p
- $\big((p \vee q) \wedge \neg(s \wedge \neg s)\big)$
- $\big(p \rightarrow (p \vee r)\big)$
- $\neg\neg s$
- …

Given a set of formulas (or other abstract data type), semtab applies the semantic tableau algorithm and returns a data structure that contains the tableaux.

Ensure your implementation is elegant, efficient, and extensible.

The project:
Part 2: the module (30%)

The function `semtab` provides the basis for all other functions. You are required to implement functions for:

- `isSatisfiable` ; is F satisfiable?
- `isValid` ; is a formula f valid under F?
- `isTautology` ; is a formula f a tautology?
- `isContradiction` ; is a formula f a contradiction?
- `models` ; w.r.t. the propositions in F, what are all the models?
- `counterexamples` ; what are all the counterexamples when f is not valid under F?

Specify your functions (and any auxiliary functions). Motivate your approach and chosen data structures.

The project:
Part 2: the module (30%)

- $isSatisfiable(\{a, \neg b, (\neg b \rightarrow c)\}) =$
  - $true$
- $isValid(a, \{(a \vee \neg a)\}) =$
  - $false$
- $isTautology\big((a \vee \neg a)\big) =$
  - $true$
- $isContradiction\big((a \wedge \neg a)\big) =$
  - $true$
- $models(\{a, \neg b, (\neg b \rightarrow c)\})$ = ???
  - Up to you to decide ☺
- $counterexamples(a, \{(a \vee \neg a)\})$ = ???
  - Up to you to decide ☺

The project:
Part 3: extensions (30%)

1. Conceptualize, formulate and implement elimination rules for
   - `EQUIV`
   - `XOR`
   - `NAND`
   - `XNOR`

2. Conceptualize, formalize, and implement a generalized approach for AND and OR with n => 1 arguments

Report (10%)

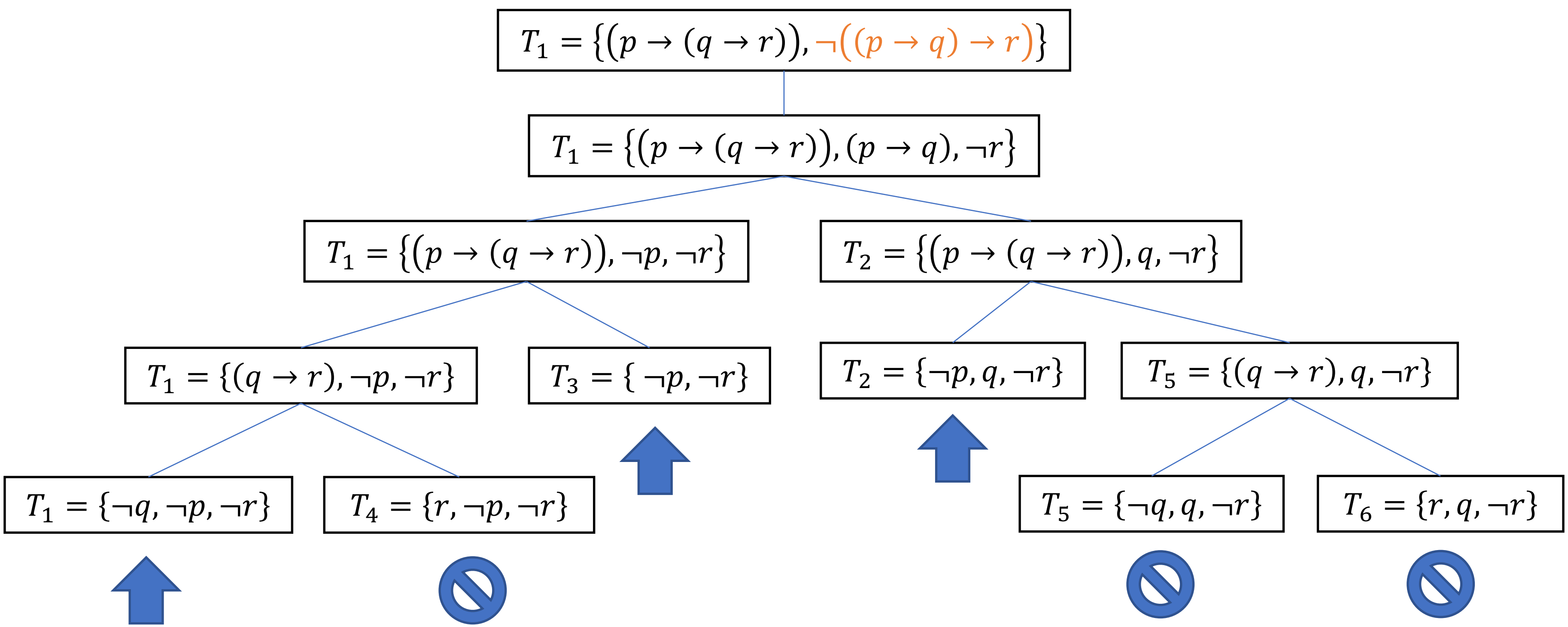
- A small technical report containing
  - A description and motivation for Parts 1, 2, 3, (and 4)
  - Describe and motivate the data representations you have chosen
    - For formulas
    - For tableaux
    - For extensibility and generalized approaches
    - For efficacy
    - …
  - **<u>Provide examples on how to use your module in a separate file!</u>**

The project:
Part 4: Parser (10% bonus)

- For "bonus" points, you can create a parser for formulas in propositional logic by using and adapting "Chapter 9. Parser combinators" from your book.
- You may limit yourself to AND, OR, NOT, and IFTHEN operations. If you attempt this part, ensure this part of the code is in a separate module.
- Serious attempts only! ☺ You will not be awarded bonus points for submitting "some code" that barely works.

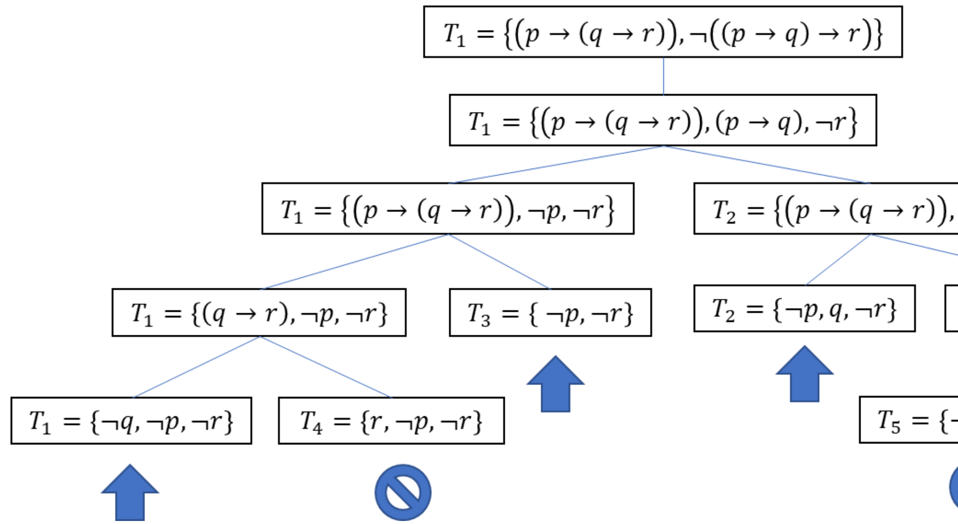
A final example!

- Is $((p \to q) \to r)$ valid under $\{(p \to (q \to r))\}$?



A final example!

- T1 contains the counterexamples
  - $v1(p) = 0, v1(q) = 0, v1(r) = 0$
- T3 contains the counterexamples
  - $v1(p) = 0, v1(q) = 0, v1(r) = 0$
  - $v2(p) = 0, v2(q) = 1, v2(r) = 0$
- T2 contains the counterexamples
  - $v2(p) = 0, v2(q) = 1, v2(r) = 0$



We thus have two counterexamples, and T3 contained both.

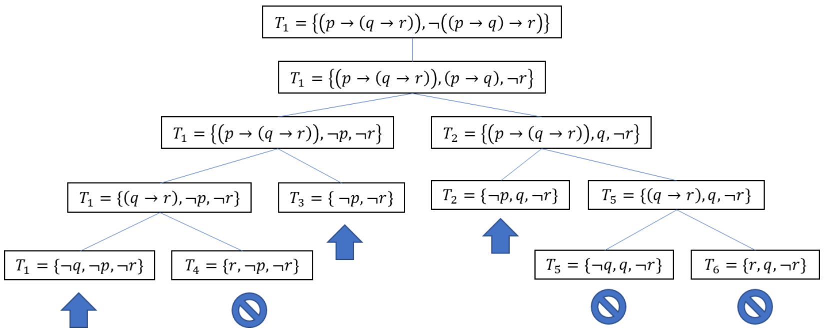
Final remarks

Semantic tableaux started as panels.

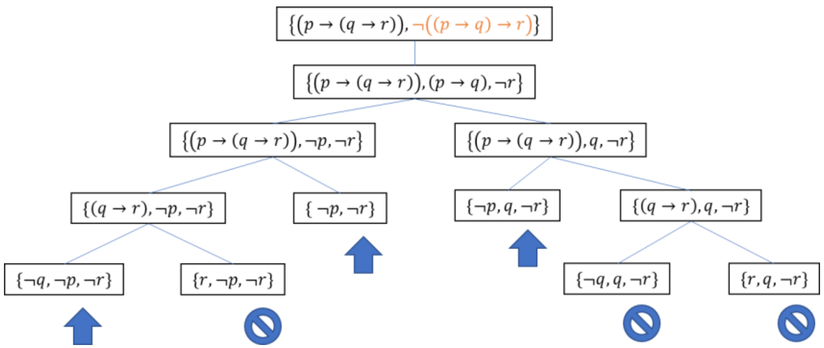
Source: https://www.museodelprado.es/en/the-collection/sxix-painting

Final remarks

Semantic tableaux started as panels. The convenient tree representations (capturing relationships between tableaux) came later.

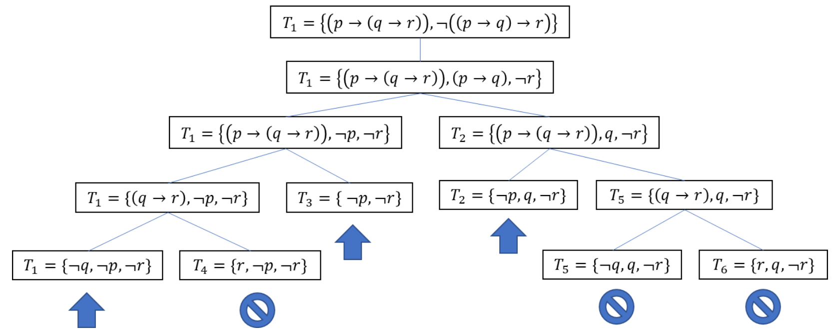


We now call the whole a semantic tableau. Each node in the tree representation contains a tableau.

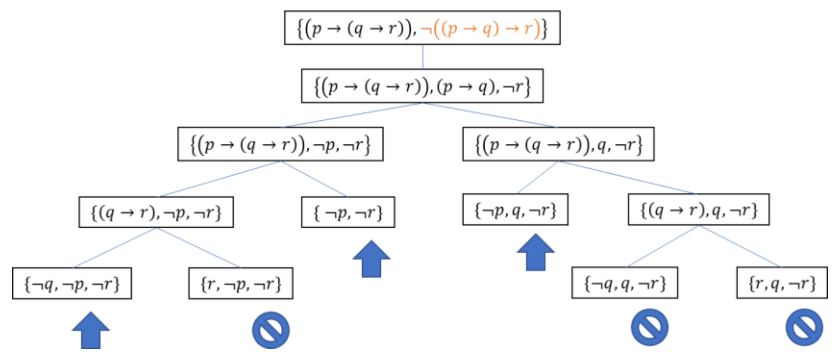


Final remarks

The indices 1,2, … are used to disambiguate tableaux and are not used to indicate order. In fact, one does not have to give "names" to branches and tableaux.



There are various ways to represent (semantic) tableaux. Each with their advantages and disadvantages. ☺



Questions?