Mathematical Logics
PL - Reasoning via Truth Tables*

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli

Lecture index

1. Truth tables – summary
2. Decision problems
3. Deciding satisfiability (SAT)
   a. CNF
   b. The DPLL SAT decision procedure
4. Satisfiability in practice: MiniSat

Reasoning / decision problems

Four tipes of questions

- **Model Checking – MC** (I, φ): $I \stackrel{?}{\models} \varphi$.
  What is the truth value of φ in I , or equivalently, does I satisfy φ or does it not satisfy φ.

- **(Un)Satisfiability – SAT/ UNSAT** (φ): $\exists I \,.\, I \stackrel{?}{\models} \varphi$
  Is there a model I that satisfies φ?

- **Validity - VAL(**φ**)**: $\stackrel{?}{\models} \varphi$. Is φ satisfied by all the models I ?

- **Logical consequence(**Γ, φ**)**: $\Gamma \stackrel{?}{\models} \varphi$
  Is φ satisfied by all the models I that satisfy all the formulas in Γ?

Model Checking

Model checking decision procedure

A model checking decision procedure, MCDP is an algorithm that checks if a formula φ is satisfied by an interpretation I. Namely

$$\text{MCDP}(\varphi, I) = \textit{true} \text{ if and only if } I \models \varphi$$

$$\text{MCDP}(\varphi, I) = \textit{false} \text{ if and only if } I \not\models \varphi \text{ (short for not } \models \varphi\text{)}$$

The procedure of model checking returns for all inputs either true or false since for all models I and for all formulas φ, we have that either $I \models \varphi$ or $I \not\models \varphi$.

A naive algorithm for model checking (example)

A simple way to check if $I \models \varphi$

(1) Replace each occurrence of a propositional variables in φ with the truth value assigned by I. I.e. replace each p with $I(p)$

(2) Recursively apply the following reduction rules for connectives:

true	$\wedge$	true	=	true	true	$\rightarrow$	true	=	true
true	$\wedge$	false	=	false	true	$\rightarrow$	false	=	false
false	$\wedge$	true	=	false	false	$\rightarrow$	true	=	true
false	$\wedge$	false	=	false	false	$\rightarrow$	false	=	true
true	$\vee$	true	=	true	true	$\equiv$	true	=	true
true	$\vee$	false	=	true	true	$\equiv$	false	=	false
false	$\vee$	true	=	true	false	$\equiv$	true	=	false
false	$\vee$	false	=	false	false	$\equiv$	false	=	true
	$\neg$	true	=	false					
	$\neg$	false	=	true					

A naive algorithm for model checking (example)

Example

- $\varphi = p \vee (q \rightarrow r)$
- $I = I(p) = \mathit{false},\ I(q) = \mathit{false},\ I(r) = \mathit{true}$

To check if $I \models p \vee (q \rightarrow r)$ we:

(I) replace, p, q, and r in φ with $I(p)$, $I(q)$ and $I(r)$, obtaining

$$\mathit{false} \vee (\mathit{false} \rightarrow \mathit{true})$$

(I) recursively apply the reduction rules

$$\mathit{false} \vee (\mathit{false} \rightarrow \mathit{true})$$
$$\mathit{false} \vee \mathit{true}$$
$$\mathit{true}$$

A simple optimization of MCDP

MCDP(I, φ) with lazy evaluation

Idea: When you evaluate a conjunction, if the first conjunct is evaluated to false, then you can jump to the conclusion that the whole conjunction is false, without evaluating the second conjunct. Similar idea can be applied to the other connectives ($\vee$, $\rightarrow$ and $\equiv$)

MCDP(I, p)
  if I(p) = *true*
    then return YES
  else return NO

MCDP(I, $\varphi \wedge \psi$)
  if MCDP(I, φ)
    then return MCDP(I, ψ)
  else return NO

MCDP(I, $\varphi \vee \psi$)
  if MCDP(I, φ)
    then return YES
  else return MCDP(I, ψ)

MCDP(I, $\varphi \rightarrow \psi$)
  if MCDP(I, φ)
    then return MCDP(I, ψ)
  else return YES

MCDP(I, $\varphi \equiv \psi$)
  if MCDP(I, φ)
    then return MCDP(I, ψ)
  else return not(MCDP(I, ψ)

Satisfiability

Satisfiability decision procedure

A satisfiability decision procedure SDP is an algorithm that takes in input a formula φ and checks if φ is (un)satisfiable. Namely

$$SDP(\varphi) = \textit{Satisfiable} \text{ if and only if } I \models \varphi \text{ for some I}$$

$$SDP(\varphi) = \textit{Unsatisfiable} \text{ if and only if } I \not\models \varphi \text{ for all I}$$

When SDP(φ) = *satisfiable*, SDP can return a (model) I, that satisfies φ.
NOTE: there can be more than one model

Validity

Validity decision procedure

A decision procedure for Validity VDC, is an algorithm that checks whether a formula is valid. VDP can be based on a satisfiability decision procedure by exploiting the equivalence

φ is valid if and only if $\neg\varphi$ is not satisfiable

$VDP(\varphi) = \mathit{true}$ if and only if $SDP(\neg\varphi) = \mathit{Unsatisfiable}$

$VDP(\varphi) = \mathit{false}$ if and only if $SDP(\neg\varphi) = \mathit{Satisfiable}$

When $SDP(\neg\varphi)$ returns an interpretation I, this interpretation is a counter-model for φ.

Logical consequence

Logical consequence decision procedure

A decision procedure for logical consequence LCDP is an algorithm that checks whether a formula φ is a logical consequence of a finite set of formulas $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. LCDP can be implemented on the basis of satisfiability decision procedure by exploiting the property

$$\Gamma \models \varphi \text{ if and only if } \Gamma \cup \{\neg\varphi\} \text{ is unsatisfiable}$$

$LCDP(\Gamma, \varphi) = true$ if and only if $SDP(\gamma_1 \wedge \cdots \wedge \gamma_n \wedge \neg\varphi) = Unatisfiable$

$LCDP(\Gamma, \varphi) = false$ if and only if $SDP(\gamma_1 \wedge \cdots \wedge \gamma_n \wedge \neg\varphi) = Satisfiable$

When $SDP(\gamma_1 \wedge \cdots \wedge \gamma_n \wedge \neg\varphi)$ an interpretation I, this interpretation is a model for Γ and a counter-model for φ.

Proof of the previous property (see also before)

Theorem

$\Gamma \models \varphi$ if and only if $\Gamma \cup \{\neg\varphi\}$ is unsatisfiable

Proof.

$\Rightarrow$ Suppose that $\Gamma \models \varphi$, this means that every interpretation I that satisfies Γ, it does satisfy φ, and therefore $I \not\models \neg\varphi$. This implies that there is no interpretations that satisfies together Γ and $\neg\varphi$.

$\Leftarrow$ Suppose that $I \models \Gamma$, let us prove that $I \models \varphi$, Since $\Gamma \cup \{\neg phi\}$ is not satisfiable, then $I \not\models \neg\varphi$ and therefore $I \models \varphi$.

□

Mathematical Logics
PL - Reasoning via Truth Tables*

Fausto Giunchiglia and Mattia Fumagalli

University of Trento

**Originally by Luciano Serafini and Chiara Ghidini*
Modified by Fausto Giunchiglia and Mattia Fumagalli