innovate achieve lead

BITS Pilani
Pilani Campus

CS/IS F214
Logic in Computer Science

MODULE: PROPOSITIONAL LOGIC

Satisfiability and Disjunctive Normal Form

Disjunctive Normal Form (DNF)

RECALL

- A propositional logic formula is said to be in **DNF** if the formula is a *disjunction of clauses*

  i.e. it is of the form $\mathbf{C_1 \vee C_2 \vee \ldots \vee C_n}$

  where each clause C_i is *a conjunction of literals*:

  i.e. . it is of the form $\mathbf{L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{im}}$

  where each literal $\mathbf{L_{ij}}$ is either *an atomic proposition* (**p**) or the *negation of an atomic proposition* (**¬p**).

- In Boolean logic, the DNF is referred to as the ***Sum-of-Products*** (***SOP***) form.

Satisfiability and DNF

- Consider a formula ϕ in DNF:
  - Let ϕ be $C_1 \vee C_2 \vee \ldots \vee C_n$
    - Then ϕ is satisfiable *if and only if* C_i is satisfiable for some i
  - Let a given clause C_i be $L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{im}$
    - Then, under what conditions will C_i be satisfiable?

Satisfiability and DNF

- Consider a formula ϕ in DNF:
  - Let ϕ be $C_1 \lor C_2 \lor \ldots \lor C_n$
    - Then ϕ is satisfiable *if and only if* C_i is satisfiable for some i
- Let a given clause C_i be $L_{i1} \land L_{i2} \land \ldots \land L_{im}$
  - Question:
    - Then, under what conditions will C_i be satisfiable?
  - Answer:
    - C_i will *not be satisfiable* only if it includes a proposition p and its negation i.e.:
      - there exist k and l such that L_{ik} is p and L_{il} is $\neg p$ for some propositional atom p

Satisfiability and DNF

- Exercises:

  (*Use the idea from the previous slide* and)

  1. Write an algorithm to check satisfiability of a given propositional logic formula in DNF
  2. Calculate the cost of (i.e. time taken by) your algorithm.
  3. Compare this cost with

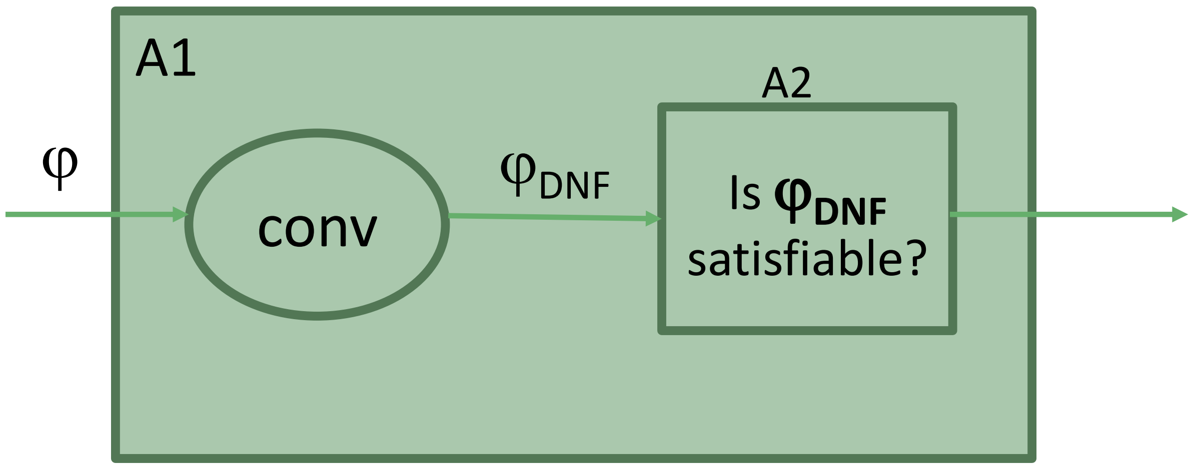
     the cost of checking satisfiability of a given propositional logic formula

     – not necessarily in DNF –

     using the truth table (or *equivalently by testing a circuit*).

SAT is not known to be in P

- Consider this approach for solving SAT:



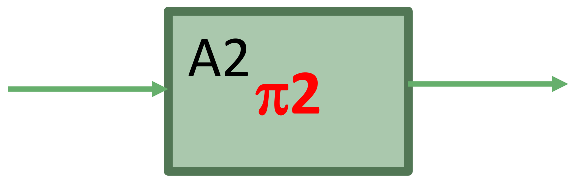
- A2 is a *polynomial-time algorithm for testing satisfiability* of a formula in DNF.
- $\varphi_{DNF} =|= \varphi$

- *What is the implication for **conv**?*

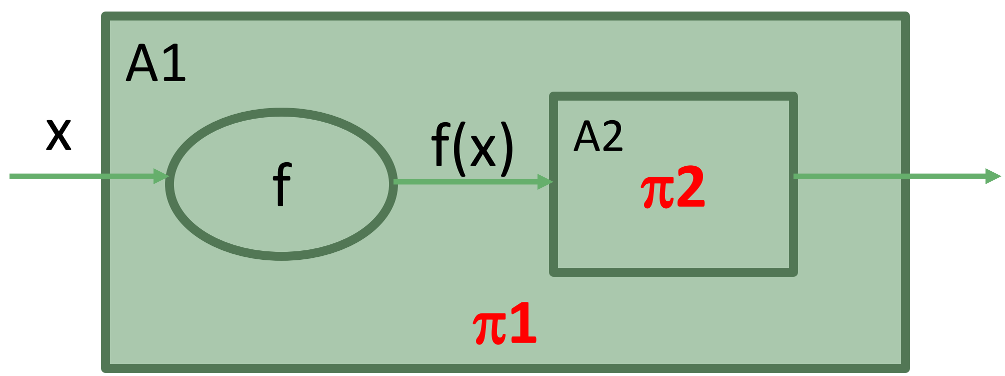
Reduction

- A ***reduction*** from a decision problem **π1** to a decision problem **π2** is
  - a mapping **f** of the inputs of **π1** to the inputs of **π2** such that
    - **π1(x)** is TRUE <u>iff</u> **π2(f(x))** is TRUE for all inputs **x**

- In algorithmic terms, <u>*if there is an algorithm* A2 *for solving* **π2**</u> **:**



- <u>*then one can construct an algorithm* A1 *for* **π1**</u>:



- Assumption:
  - <u>*There is an algorithm for* **f**</u>.

Reductions:

- Recall the approach used to prove that a problem is undecidable using a known undecidable problem (e.g Halting problem).
- Can you generalize the implication of a reduction?