PROPOSITIONAL LOGIC

based on

Huth & Ruan
Logic in Computer Science:
Modelling and Reasoning about Systems
Cambridge University Press, 2004

The Language of Logic

- Logic: symbolic language for making declarative statements about the state of the world
  - declarative statements:

The train arrives late.

If the train arrives late and there are no taxis, John is late at his meeting.

John is not late at his meeting.

There are taxis.

The Language of Logic

- Logic: symbolic language for making declarative statements about the state of the world
  - statements which are not declarative :

Fetch the train!

Have you seen the train?

I hope your train will be in time!

The Language of Logic

- Logic: symbolic language for making declarative statements about state of the world
  - symbolic language: we will use symbols to express our beliefs about the world

p =	The train arrives late.
q =	There are taxis.
r =	John is late at his meeting.
$p \wedge \neg q \rightarrow r$ =	If the train arrives late and there are no taxis, John is late at his meeting.

Uses of Logic

- Allows for formal specification:
  - of what we know (knowledge representation)

The train arrives late.

If the train arrives late and
there are no taxis,
John is late at his meeting.

  - of what we want to achieve

John is not late at his meeting.

Uses of Logic

- Allows for reasoning:
  - drawing conclusions from observations

The train arrives late.
There are no taxis.
→ John is late at his meeting.

Red light detected.
Road detected.
→ Pedestrian crossing detected.

Uses of Logic

- Allows for reasoning:
  - finding inconsistencies in our knowledge

The train arrives late.	There are no taxis.	John is not late at his meeting.
If the train arrives late and there are no taxis, John is late at his meeting.		

Program A terminates.	Program B terminates.	Program C does not terminate.
If program A terminates and program B terminates, program C also terminates.		

Uses of Logic

- Allows for reasoning:
  - finding models (worlds in which a desired situation is true)

Knowledge + desired situation	**Model**
If the train arrives late and there are no taxis, John is late at his meeting. →	The train does not arrive late.
John is not late at his meeting. →	The train arrives late. There are taxis.
All cities are visited →	Visit A, then B, then C, …

Propositional logic

- **<u>Well-formed formulas</u>** in propositional logic are obtained by using the following construction rules, and only these rules, a finite number of times:
  - propositional atoms $p, q, \ldots, p_1, p_2, \ldots$ are well-formed formulas
  - if ϕ is a well-formed formula, then so is $(\neg \phi)$
  - if ϕ and ψ are well-formed formulas, then so is $(\phi \wedge \psi)$
  - if ϕ and ψ are well-formed formulas, then so is $(\phi \vee \psi)$
  - if ϕ and ψ are well-formed formulas, then so is $(\phi \rightarrow \psi)$

$\neg, \wedge, \vee, \rightarrow$ are called *connectives*

Propositional logic

- Examples of well-formed formulas:

$$((p \to q) \wedge (p \to \neg q))$$

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

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

- Examples of badly-formed formulas:

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

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

$$((P \cup Q) \vee R)$$

Propositional logic

- Notational convenience: we often drop (…) based on the precedence between operators: $\neg$ (highest), $\wedge \vee$ (equal), $\rightarrow$ (lowest)

$$p \wedge q \rightarrow \neg r \vee s \iff ((p \wedge q) \rightarrow ((\neg r) \vee s))$$

$$p \rightarrow q \wedge r \rightarrow t \iff (p \rightarrow ((q \wedge r) \rightarrow t))$$

Implication is right associative

However, formulas in this notation are **not** well-formed!

Semantics of Propositional logic

- A *valuation* or *interpretation* of a formula ϕ is an assignment of each propositional atom in ϕ to a truth value
- A *truth value* is a value in the domain {true, false} or {T,F}

4 valuations for the formula $p \vee \neg q$:

p	*q*
T	T
T	F
F	T
F	F

Evaluating formulas

- The truth value of a formula for a given valuation is determined using *truth tables* for the connectives

ϕ	ψ	$\phi \wedge \psi$
T	T	T
T	F	F
F	T	F
F	F	F

ϕ	ψ	$\phi \vee \psi$
T	T	T
T	F	T
F	T	T
F	F	F

ϕ	ψ	$\phi \rightarrow \psi$
T	T	T
T	F	F
F	T	T
F	F	T

ϕ	$\neg\phi$
T	F
F	T

$\top$
T

$\bot$
F

“true” formula ($\top$)

“false” formula ($\bot$)

Evaluating formulas

4 valuations for the formula $p \vee \neg q$:

p	q	Truth value formula
T	T	T
T	F	T
F	T	F
F	F	T

Semantic entailment

- If, for all valuations in which all $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to T, ψ evaluates to T as well, we say that

$$\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$$

holds and that $\phi_1, \phi_2, \ldots, \phi_n$ **semantically entail** ψ

p	q	q $\vDash$	$(p \wedge q) \vee \neg p$
T	T	T	T
T	F	F	F
F	T	T	T
F	F	F	T

Semantic entailment

- If $\top \models \phi_1, \phi_2, \ldots, \phi_n$ then $\phi_1, \phi_2, \ldots \phi_n$ are said to be a **tautology**

- If $\phi_1, \phi_2, \ldots, \phi_n \models \bot$ then $\phi_1, \phi_2, \ldots \phi_n$ are said to be a **contradiction**

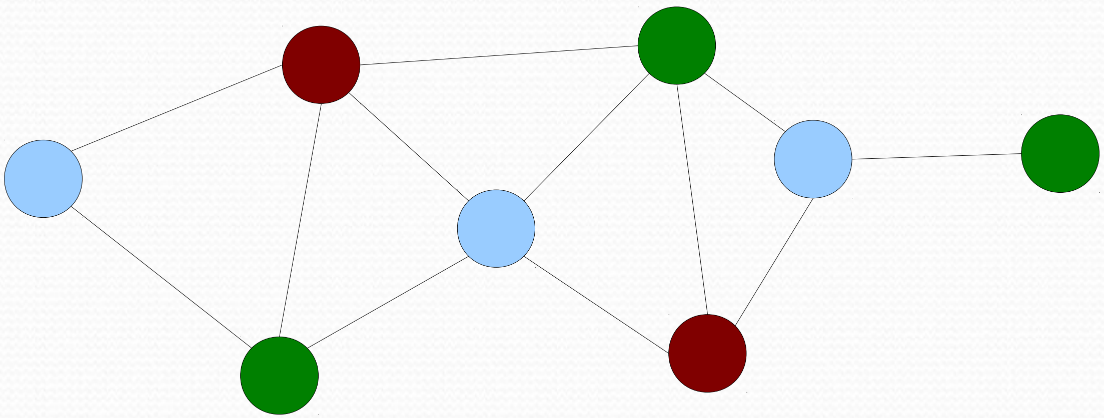
$(p \to q) \vee (p \wedge \neg q)$ is a tautology

$(p \to q) \wedge (p \wedge \neg q)$ is a contradiction

If there is a valuation that makes a formula true, the formula is said to be **satisfiable** (i.e. there is no contradiction)

Using satisfiability for solving CSPs

- Example: graph coloring



Given a graph *G* and a parameter *k*
Find a color assignment to each node
Such that
- no two adjacent have the same color
- not more than *k* colors are used

Using satisfiability for solving CSPs

- We can encode this problem as a satisfiability (or entailment) problem, by creating atoms and formulas based on the graph
  - for each node, create *k* atoms p_{ic} indicating that node *i* has color *c*
  - for each node, create a formula
    $$\phi_i = p_{i1} \vee p_{i2} \vee \cdots \vee p_{ik}$$
    indicating that each node *i* must have a color
  - for each node and different pair of colors, create a formula
    $$\phi_{ic_1c_2} = \neg(p_{ic_1} \wedge p_{ic_2})$$
    indicating a node may not have more than 1 color

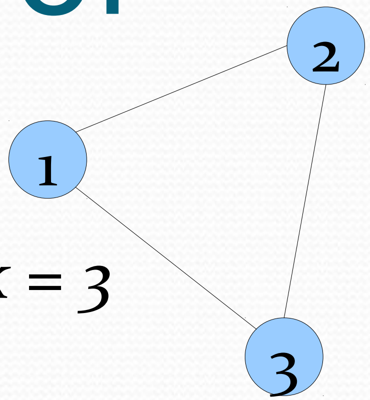
Using satisfiability for solving CSPs

- We can encode this problem as a satisfiability (or entailment) problem, by creating atoms and formulas based on the graph
  - …
  - for each edge, create *k* formulas

    $$\phi_{ijc} = \neg(p_{ic} \wedge p_{jc})$$

    indicating that a pair connected nodes *i* and *j* may not both have color *c* at the same time

Using satisfiability for solving CSPs



- Assume we wish to color this graph for *k* = 3

$\phi_1 = p_{11} \vee p_{12} \vee p_{13}$

$\phi_2 = p_{21} \vee p_{22} \vee p_{23}$

$\phi_3 = p_{31} \vee p_{32} \vee p_{33}$

$\phi_{121} = \neg(p_{11} \wedge p_{21})$ $\phi_{122} = \neg(p_{12} \wedge p_{22})$ $\phi_{123} = \neg(p_{13} \wedge p_{23})$

$\phi_{131} = \neg(p_{11} \wedge p_{31})$ $\phi_{132} = \neg(p_{12} \wedge p_{32})$ $\phi_{133} = \neg(p_{13} \wedge p_{33})$

$\phi_{231} = \neg(p_{21} \wedge p_{31})$ $\phi_{232} = \neg(p_{22} \wedge p_{32})$ $\phi_{233} = \neg(p_{23} \wedge p_{33})$

If this holds:

$\phi_1, \phi_2, \phi_3, \phi_{121}, \phi_{122}, \phi_{123}, \phi_{131}, \phi_{132}, \phi_{133}, \phi_{231}, \phi_{232}, \phi_{233} \models \perp$

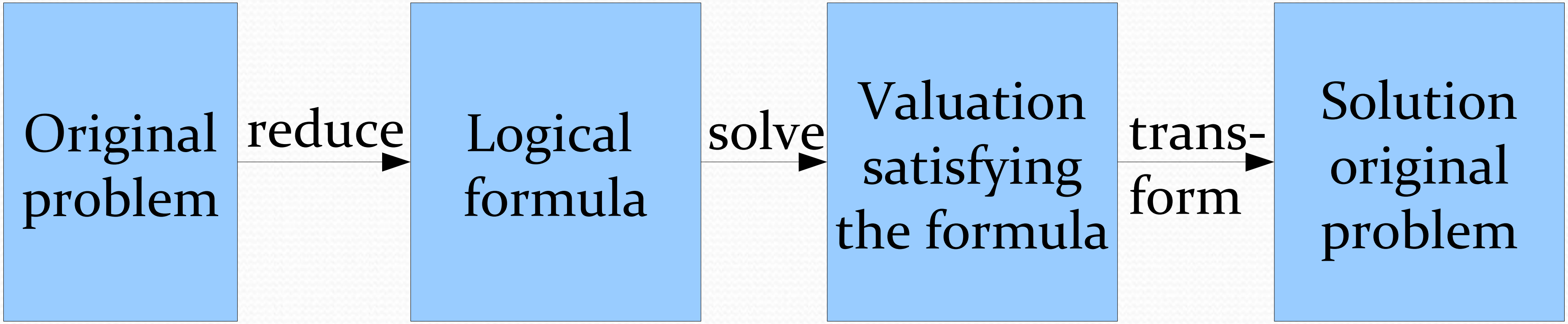
there is no coloring.

However, there is a coloring; set to T: p_{11}, p_{22}, p_{33}

set to F: $p_{12}, p_{13}, p_{21}, p_{23}, p_{31}, p_{32}$

Using satisfiability for solving CSPs

- General idea: **reduce** a constraint satisfaction problem to a satisfiability problem



Solving entailment problems

- How to decide whether one formula semantically entails another?

- If the number of atoms is n, the number of possible valuations is 2^n → enumerating all valuations is usually not feasible to prove entailment.

- Two solutions:
  - finding proofs using **syntactic entailment** *
  - **SAT solvers** for specific types of formulas

*: rarely used in computers, but used by humans

Syntactic entailment & Natural deduction

- Essentially, we will introduce a number of proof rules (the proof rules of *natural deduction*) that allow to derive new formulas from old formulas. We say that

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$$

holds and that $\phi_1, \phi_2, \ldots, \phi_n$ **syntactically entail** formula ψ.

- It can be shown that these rules are
  - **sound:** if $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ then $\phi_1, \phi_2, \ldots, \phi_n \models \psi$
  - **complete**: if $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ then $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$

Natural deduction: And-introduction

- If ψ and ϕ are true, then $\psi \wedge \phi$ is true

Notation:

1.	ϕ	premise
2.	ψ	premise
3.	$\psi \wedge \phi$	$\wedge$i 1,2

Line number → (the numbers 1., 2., 3.)

what is given → (premise)

what follows from rule application → ($\wedge$i 1,2)

Natural deduction: And-elimination

- If $\psi \wedge \phi$ is true, then ϕ is true
- If $\psi \wedge \phi$ is true, then ψ is true

1.	$\psi \wedge \phi$	premise
2.	ϕ	$\wedge e_1$ 1
3.	ψ	$\wedge e_2$ 1

Natural deduction: And-example

Proof that: $p \wedge q, r \vdash q \wedge r$

1.	$p \wedge q$	premise
2.	r	premise
3.	q	$\wedge e_1$ 1
4.	$q \wedge r$	$\wedge i$ 2, 3

Natural deduction: Double negation

- If ϕ is true, then $\neg\neg\phi$ is true
- If $\neg\neg\phi$ is true, then ϕ is true

1.	ϕ	premise
2.	$\neg\neg\phi$	$\neg\neg$ i 1

1.	$\neg\neg\phi$	premise
2.	ϕ	$\neg\neg$ e 1

Natural deduction: Implication elimination

- If ψ and $\psi \to \phi$ are true, then ϕ is true

1.	ψ	premise
2.	$\psi \to \phi$	premise
3.	ϕ	→ e 1,2

Now prove that $p, p \to q, p \to (q \to r) \vdash r$

Natural deduction: Modus tolens

- If $\neg\phi$ and $\psi \rightarrow \phi$ are true, then $\neg\psi$ is true

1.	$\neg\phi$	premise
2.	$\psi \rightarrow \phi$	premise
3.	$\neg\psi$	MT 1,2

Now prove that $p \rightarrow (q \rightarrow r), p, \neg r \vdash \neg q$

Natural deduction: Implication introduction

- If under the assumption that ϕ is true, also ψ is true, then $\phi \to \psi$

1.	$\psi \to \phi$	premise
2.	$\neg\phi$	assumption
3.	$\neg\psi$	MT 1,2
4.	$\neg\phi \to \neg\psi$	→i 2-3

Box around temporary conclusions

Now prove that $\vdash (q \to r) \to ((\neg q \to \neg p) \to (p \to r))$

$$p \wedge q \to r \vdash p \to (q \to r)$$