Introduction to Theoretical Computer Science

Zdeněk Sawa

Department of Computer Science, FEI,
Technical University of Ostrava
17. listopadu 15, Ostrava-Poruba 708 33
Czech republic

February 14, 2020

Lecturer

Name: doc. Ing. Zdeněk Sawa, Ph.D.

E-mail: `zdenek.sawa@vsb.cz`

Room: EA413

Web: `http://www.cs.vsb.cz/sawa/uti/index-en.html`

On these pages you will find:

- Information about the course
- Study texts
- Slides from lectures
- Exercises for tutorials
- Recent news for the course
- A link to a page with animations

Requirements

- **Credit** (22 points):
  - Written test (16 points) — it will be written on a tutorial
    - The minimal requirement for obtaining the credit is 9 points.
    - A correcting test for 14 points.
  - Activity on tutorials (6 points)
    - The minimal requirement for obtaining the credit is 3 points.
- **Exam** (78 points)
  - A written exam consisting of three parts (26 points for each part); it is necessary to obtain at least 10 points for each part.

Theoretical Computer Science

Theoretical computer science — a scientific field on the border between computer science and mathematics

- investigation of general questions concerning algorithms and computations
- study of different kinds of formalisms for description of algorithms
- study of different approaches for description of syntax and semantics of formal languages (mainly programming languages)
- a mathematical approach to analysis and solution of problems (proofs of general mathematical propositions concerning algorithms)

Examples of some typical questions studied in theoretical computer science:

- Is it possible to solve the given problem using some algorithm?
- If the given problem can be solved by an algorithm, what is the computational complexity of this algorithm?
- Is there an efficient algorithm solving the given problem?
- How to check that a given algorithm is really a correct solution of the given problem?
- What kinds instructions are sufficient for a given machine to perform a given algorithm?

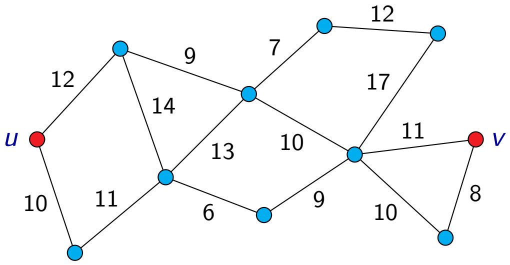
An example of an algorithmic problem

Problem "'Finding the shortest path in an (undirected) graph'

Input: An undirected graph $G = (V, E)$ with edges labelled with numbers, and a pair of nodes $u, v \in V$.

Output: The shortest path from node u to node v.

Example:



Algorithms and Problems

Theoretical computer science overlaps with many other areas of mathematics and computer science:

- graph theory
- number theory
- computational geometry
- searching in text
- game theory
- ...

Logic

Logic — study of reasoning and argumentation

- it studies when a conclusion follows from given assumptions
- it studies questions concerning proofs and provability
- it provides a basic language for mathematics and all sciences based on mathematics
- it is connected with study of foundations of mathematics
- it is used in computer science on many different levels

Propositional Logic

Logical Inference

- *If the train arrives late and there are no taxis at the station, then John is late for his meeting.*
- *John is not late for his meeting.*
- *The train did arrive late.*

- *There were taxis at the station.*

- *If it is raining and Jane does not have her umbrella with her, then she will get wet.*
- *Jane is not wet.*
- *It is raining.*

- *Jane has her umbrella with her.*

Logical Inference

p	The train is late.	It is raining.
q	There are taxis at the station.	Jane has her umbrella with her.
r	John is late for his meeting.	Jane gets wet.

If p and not q, then r.
Not r.
p.

---

q.

Propositions

Examples of propositions:

- *"Jane gets wet."*
- *"If it is raining and Jane does not have her umbrella with her, then she will get wet."*
- *"Paris is the capital of Japan."*
- *"There are infinitely many primes."*
- "$1 + 1 = 3$"
- *"Number $\sqrt{2}$ is irrational."*

Logical Connectives

Atomic proposition — it cannot be decomposed into simpler propositions

"It is raining."

Compound proposition — it is composed from some simpler propositions

"If it is raining and Jane does not have her umbrella with her, then she will get wet."

It consists of propositions:

- *"It is raining."*
- *"Jane has her umbrella with her."*
- *"Jane gets wet."*

Logical Connectives

More complicated propositions can be constructed from simpler propositions using **logical connectives**:

Symbol	Log. connective	Example of use	Informal meaning
$\neg$	negation	$\neg p$	*"not p"*
$\wedge$	conjunction	$p \wedge q$	*"p and q"*
$\vee$	disjunction	$p \vee q$	*"p or q"*
$\rightarrow$	implication	$p \rightarrow q$	*"if p then q"*
$\leftrightarrow$	equivalence	$p \leftrightarrow q$	*"p if and only if q"*

Atomic propositions — p, q, r, ...
(possibly with indexes — p_0, p_1, p_2, ...)

Logical Connectives

Propositions are represented using **formulas** — formulas have a precisely defined syntax and semantics.

"If it is raining and Jane does not have her umbrella with her, then she will get wet."

The proposition written as a formula:

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

Atomic propositions:

- p — *"It is raining."*
- q — *"Jane has her umbrella with her."*
- r — *"Jane gets wet."*

Truth Values

1	0
true	false
yes	no

We will use 0 and 1 to denote the truth values.

The truth values are also called **boolean** values.

Negation

The **negation** of a proposition φ is the proposition *"not φ"*, or the proposition *"it is not the case that φ"*. For example, the negation of

"the number 5 is a prime"

is a proposition

"it is not true that the number 5 is a prime"

or

"the number 5 is not a prime".

In formulas, negation is denoted by symbol "$\neg$".

Formally: $\neg\varphi$

φ	$\neg\varphi$
0	1
1	0

Conjunction

The **conjunction** of propositions φ and ψ is proposition "φ *and* ψ".

Example: The conjuction of propositions
"Copenhagen is the capital of Denmark" and "$2 + 2 = 4$" is proposition

"Copenhagen is the capital of Denmark and $2 + 2 = 4$."

In formulas, conjuction is denoted by symbol "$\wedge$".

$$p \wedge q$$

- p — *"Copenhagen is the capital of Denmark"*
- q — "$2 + 2 = 4$"

Conjunction

φ	ψ	$\varphi \wedge \psi$
0	0	0
0	1	0
1	0	0
1	1	1

Examples of false propositions:

- *"Helsinki are the capital of Italy and Charles University was founded in 1348."*
- *"Asia is the largest continent and* $3 + 5 = 14$*."*
- *"There are only finitely many primes and Pilsen is the capital of USA."*

Disjunction

The **disjunction** of propositions φ and ψ is proposition "φ *or* ψ".

Example: The disjunction of propositions *"whales are mammals"* and *"Czeck Republic is in Europe"* is proposition

"whales are mammals or Czech Republic is in Europe".

In formulas, disjunction is denoted by symbol "$\vee$".

$$p \vee q$$

- p — *"whales are mammals"*
- q — *"Czech Republic is in Europe"*

Disjunction

Disjunction is *"or"* in **non-exclusive** sense.

φ	ψ	$\varphi \vee \psi$
0	0	0
0	1	1
1	0	1
1	1	1

Implication

Implication — *"if φ then ψ"*

- φ — assumption (hypothesis)
- ψ — conclusion

Example:

> *"If Peter was well-prepared for the exam, he obtained a good grade."*

Implication is denoted by symbol "$\rightarrow$".

$$p \rightarrow q$$

- p — *"Peter was well-prepared for the exam"*
- q — *"Peter obtained a good grade"*

Implication

φ	ψ	$\varphi \rightarrow \psi$
0	0	1
0	1	1
1	0	0
1	1	1

Remark: Formula $p \rightarrow q$ is true exactly in those cases where the following formula is true:

$$\neg p \vee q$$

Implication

Implication does **not** express causal dependence.

Example:

- *"If Washington is the capital of USA, then* $1 + 1 = 2$*."*
- *"If Washington is the capital of USA, then* $1 + 1 = 3$*."*
- *"If Tokyo is the capital of USA, then* $1 + 1 = 2$*."*
- *"If Tokyo is the capital of USA, then* $1 + 1 = 3$*."*

Implication

Implication $p \rightarrow q$ can be expressed in a natural language in many different ways:

- *"q if p"*
- *"p only if q"*
- *"p implies q"*
- *"q provided that p"*
- *"from p follows q"*
- *"p is a sufficient condition for q"*
- *"q is a necessary condition for p"*

Implication

If $\varphi \rightarrow \psi$ is true and also φ is true, we can infer from this that also ψ is true.

Example: When it holds that

- *"if today is Tuesday, then tomorrow is Wednesday"*
- *"today is Tuesday"*

we can infer from this that

- *"tomorrow is Wednesday"*

Equivalence

Equivalence — *"φ if and only if ψ"*

Example:

"Triangle ABC has all three sides of the same length if and only if it has all three angles of the same size."

The logical connective equivalence is denoted by symbol "$\leftrightarrow$"

$$p \leftrightarrow q$$

- p — *"triangle ABC has all three sides of the same length"*
- q — *"triangle ABC has all three angles of the same size"*

Equivalence

φ	ψ	$\varphi \leftrightarrow \psi$
0	0	1
0	1	0
1	0	0
1	1	1

Remark: Formula $p \leftrightarrow q$ says basically the same thing as

$$(p \rightarrow q) \wedge (q \rightarrow p)$$

Equivalence

Alternatives for expressing equivalence $p \leftrightarrow p$:

- *"p is a necessary and sufficient condition for q"*
- *"p iff q"*

Equivalence is often used in **definitions** of new notions:

Example:

- *"A triangle is isoscele if and only if at least two of its sides are equal in length."*
- *"A triangle is isoscele if at least two of its sides are equal in length."*

Formulas of Propositional Logic

- **Syntax** — what are well-formed formulas of propositional logic
- **Semantics** — assigns meaning to formulas and to individual symbols occurring in these formulas

Syntax of Formulas of Propositional Logic

Formulas — sequences of symbols from a given **alphabet**:

- **atomic propositions** — for example symbols "p", "q", "r", etc.
- **logical connectives** — symbols "$\neg$", "$\wedge$", "$\vee$" "$\rightarrow$", "$\leftrightarrow$"
- **parentheses** — symbols "$($" and "$)$"

Not every sequence of these symbols is a formula.

For example, this is not a formula:

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

Syntax of Formulas of Propositional Logic

Definition

Well-formed **formulas of propositional logic** are sequences of symbols constructed according to the following rules:

1. If p is an atomic proposition, then p is a well-formed formula.
2. If φ and ψ are well-formed formulas, then also $(\neg\varphi)$, $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$ a $(\varphi \leftrightarrow \psi)$ are well-formed formulas.
3. There are no other well-formed formulas than those constructed according to two previous rules.

Syntax of Formulas of Propositional Logic

Examples of well-formed formulas:

- q
- $(\neg q)$
- r
- $((\neg q) \rightarrow r)$
- p
- $(p \leftrightarrow r)$
- $(\neg(p \leftrightarrow r))$
- $(((\neg q) \rightarrow r) \wedge (\neg(p \leftrightarrow r)))$

An example of a sequence of symbols, which is not a well-formed formula:

- $(p \wedge \vee q)$

Syntax of Formulas of Propositional Logic

Formula ψ is a **subformula** of formula φ if at least one of the following possibilities holds:

- Formula ψ is the same formula as formula φ.
- Formula φ is of the form $(\neg\chi)$ and ψ is a subformula of formula χ.
- Formula φ is of the form $(\chi_1 \wedge \chi_2)$, $(\chi_1 \vee \chi_2)$, $(\chi_1 \rightarrow \chi_2)$, or $(\chi_1 \leftrightarrow \chi_2)$, and ψ is a subformula of formula χ_1 or a subformula of formula χ_2.

Example: Subformulas of formula $((\neg(p \wedge q)) \leftrightarrow r)$:

$$p \qquad q \qquad r \qquad (p \wedge q) \qquad (\neg(p \wedge q)) \qquad ((\neg(p \wedge q)) \leftrightarrow r)$$

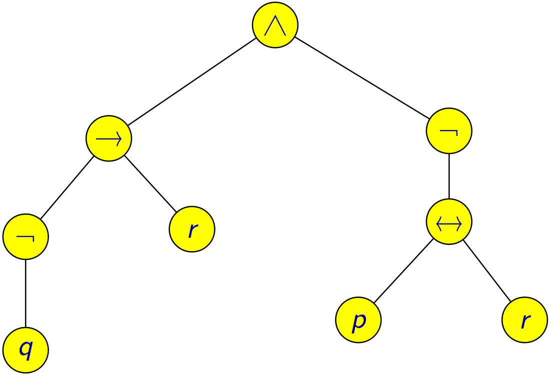
Syntax of Formulas of Propositional Logic

Alternative symbols for logical connectives:

Connective	Symbol	Alternative symbols
negation	$\neg$	$\sim$
conjunction	$\wedge$	$\&$
implication	$\rightarrow$	$\Rightarrow$, $\supset$
equivalence	$\leftrightarrow$	$\Leftrightarrow$, $\equiv$

Syntax of Formulas of Propositional Logic

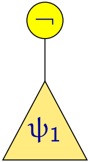
An abstract syntax tree of formula $(((\neg q) \rightarrow r) \wedge (\neg(p \leftrightarrow r)))$:



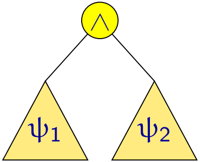
Syntax of Formulas of Propositional Logic

p

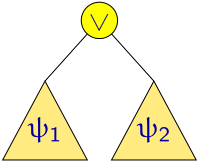
$(\neg\psi_1)$



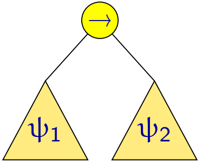
$(\psi_1 \wedge \psi_2)$



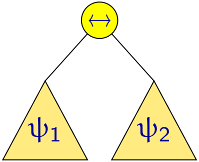
$(\psi_1 \vee \psi_2)$



$(\psi_1 \rightarrow \psi_2)$



$(\psi_1 \leftrightarrow \psi_2)$



Syntax of Formulas of Propositional Logic

Arity of logical connectives:

- **unary connective** (arity 1): $\neg$
- **binary connectives** (arity 2): $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Syntax of Formulas of Propositional Logic

Conventions for omitting parentheses:

- Outermost pair of parentheses can be omitted.
- Priority of logical connectives (from the highest to the lowest):

$$\neg \qquad \wedge \qquad \vee \qquad \rightarrow \qquad \leftrightarrow$$

- Instead of $\neg(\neg\varphi)$, it is possible to write $\neg\neg\varphi$.

Example: Instead of $((\neg p) \wedge (r \rightarrow (q \vee s)))$, it is possible to write

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

Remark: Other conventions will be described later.

Semantics of Propositional Logic

At — a set of atomic propositions

For example

- $At = \{p, q, r\}$, or
- $At = \{p_0, p_1, p_2, \ldots\}$

Definition

A **truth valuation** is an assignment of truth values (i.e., values from the set $\{0, 1\}$) to all atomic propositions from the set At.
(Formally, a truth valuation can be defined as a function $v : At \to \{0, 1\}$.)

Example: A truth valuation v for $At = \{p, q, r\}$, where

$$v(p) = 1 \qquad v(q) = 0 \qquad v(r) = 1$$

If set At is finite and contains n atomic propositions, then there are 2^n truth valuations.

Example: $At = \{p, q, r\}$

$$
\begin{array}{llll}
v_0: & v_0(p) = 0, & v_0(q) = 0, & v_0(r) = 0 \\
v_1: & v_1(p) = 0, & v_1(q) = 0, & v_1(r) = 1 \\
v_2: & v_2(p) = 0, & v_2(q) = 1, & v_2(r) = 0 \\
v_3: & v_3(p) = 0, & v_3(q) = 1, & v_3(r) = 1 \\
v_4: & v_4(p) = 1, & v_4(q) = 0, & v_4(r) = 0 \\
v_5: & v_5(p) = 1, & v_5(q) = 0, & v_5(r) = 1 \\
v_6: & v_6(p) = 1, & v_6(q) = 1, & v_6(r) = 0 \\
v_7: & v_7(p) = 1, & v_7(q) = 1, & v_7(r) = 1
\end{array}
$$

Semantics of Propositional Logic

At truth valuation v, formula φ has truth value 1:

$$v \models \varphi$$

At truth valuation v, formula φ has truth value 0:

$$v \not\models \varphi$$

Semantics of Propositional Logic

Definition

Truth values of formulas of propositional logic in a given truth valuation v are defined as follows:

- For atomic proposition p, $v \models p$ iff $v(p) = 1$.
  (So, if $v(p) = 0$, then $v \not\models p$.)
- $v \models \neg\varphi$ iff $v \not\models \varphi$.
- $v \models \varphi \wedge \psi$ iff $v \models \varphi$ and $v \models \psi$.
- $v \models \varphi \vee \psi$ iff $v \models \varphi$ or $v \models \psi$.
- $v \models \varphi \rightarrow \psi$ iff $v \not\models \varphi$ or $v \models \psi$.
- $v \models \varphi \leftrightarrow \psi$ iff $v \models \varphi$ and $v \models \psi$, or $v \not\models \varphi$ and $v \not\models \psi$.

Semantics of Propositional Logic

φ	$\neg\varphi$
0	1
1	0

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$	$\varphi \leftrightarrow \psi$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

Semantics of Propositional Logic

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ

$$\varphi \;:=\; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$$

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

- $v \models p$
- $v \not\models q$
- $v \models r$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1					

$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

- $v \models p$
- $v \not\models q$
- $v \models r$
- $v \models \neg q$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1				

$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

- $v \models p$
- $v \not\models q$
- $v \models r$
- $v \models \neg q$
- $v \models \neg q \rightarrow r$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1			

$\varphi \; := \; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

- $v \models p$
- $v \not\models q$
- $v \models r$
- $v \models \neg q$
- $v \models \neg q \rightarrow r$
- $v \models p \leftrightarrow r$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1		

$\varphi \; := \; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Semantics of Propositional Logic

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

- $v \models p$
- $v \not\models q$
- $v \models r$
- $v \models \neg q$
- $v \models \neg q \rightarrow r$
- $v \models p \leftrightarrow r$
- $v \not\models \neg(p \leftrightarrow r)$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1	0	

$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Semantics of Propositional Logic

Example: $At = \{p, q, r\}$
valuation v, where $v(p) = 1$, $v(q) = 0$, and $v(r) = 1$

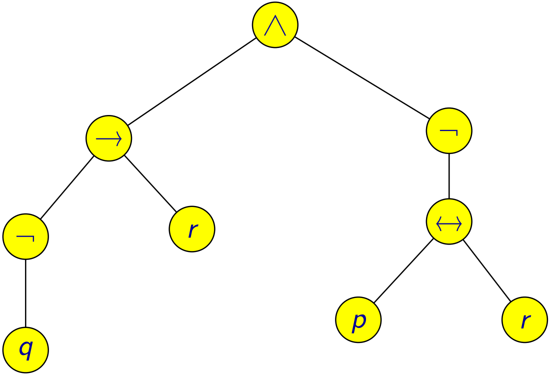
- $v \models p$
- $v \not\models q$
- $v \models r$
- $v \models \neg q$
- $v \models \neg q \rightarrow r$
- $v \models p \leftrightarrow r$
- $v \not\models \neg(p \leftrightarrow r)$
- $v \not\models (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1	0	0

$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

Semantics of Propositional Logic

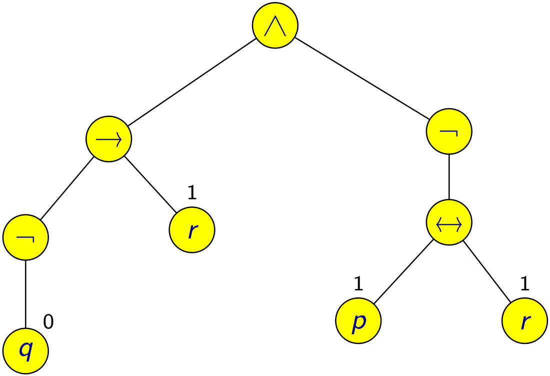
$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ

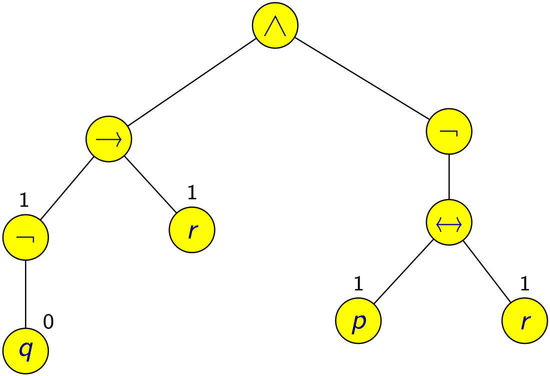
Semantics of Propositional Logic

$\varphi \; := \; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



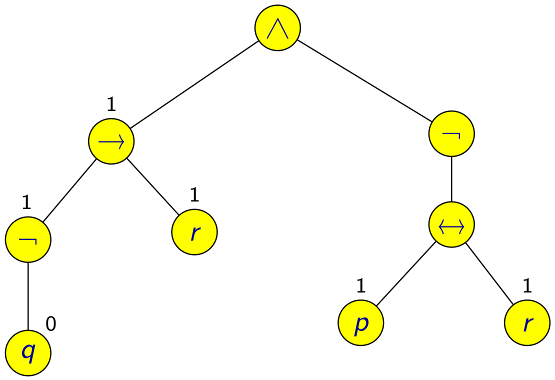
p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1					

$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1				

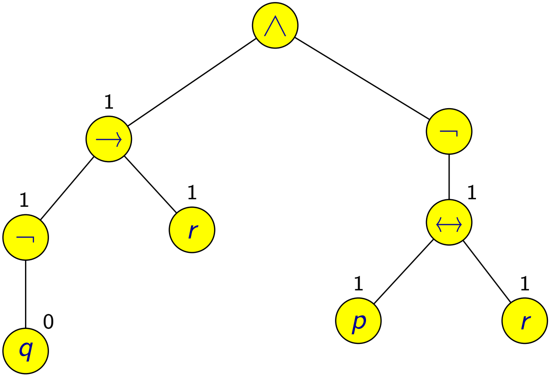
$\varphi \; := \; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1			

Semantics of Propositional Logic

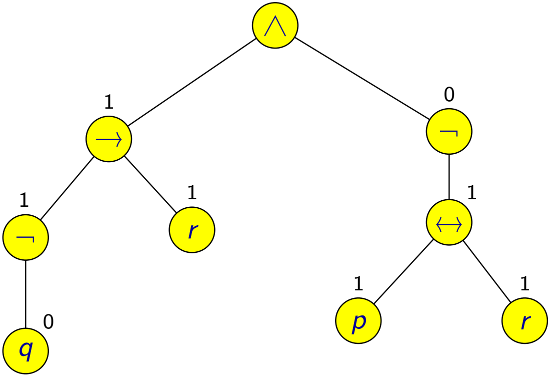
$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1		

Semantics of Propositional Logic

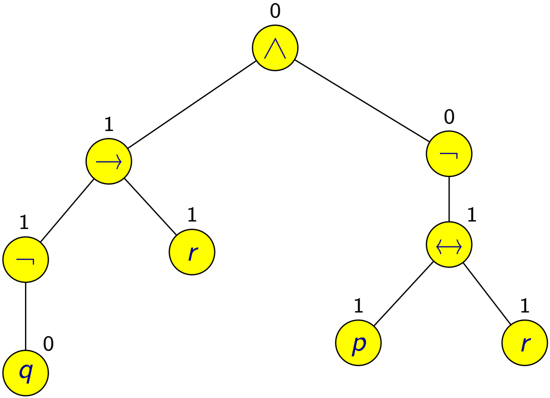
$$\varphi \;:=\; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1	0	

Semantics of Propositional Logic

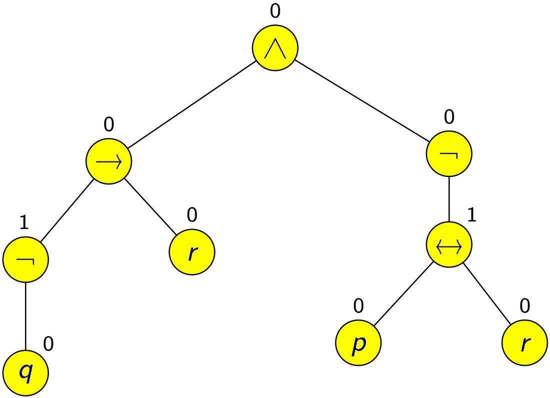
$\varphi \;:=\; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
1	0	1	1	1	1	0	0

Semantics of Propositional Logic

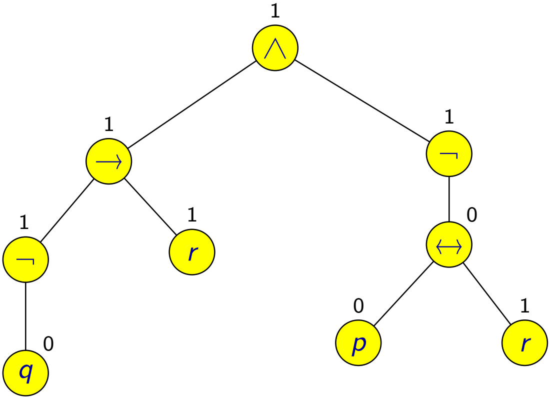
$\varphi \; := \; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_0(p) = 0 \qquad v_0(q) = 0 \qquad v_0(r) = 0$

Semantics of Propositional Logic

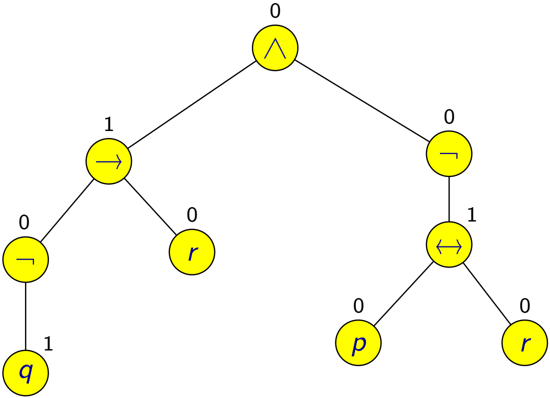
$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_1(p) = 0 \qquad v_1(q) = 0 \qquad v_1(r) = 1$

Semantics of Propositional Logic

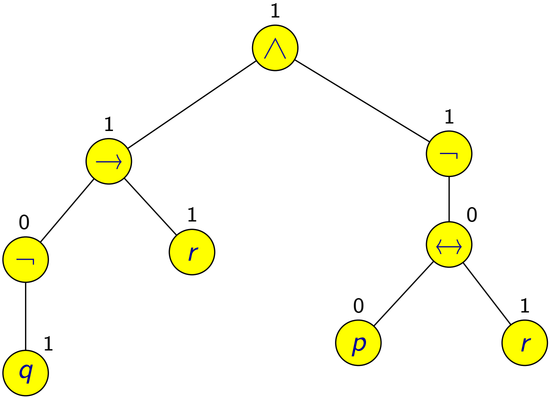
$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_2(p) = 0 \qquad v_2(q) = 1 \qquad v_2(r) = 0$

Semantics of Propositional Logic

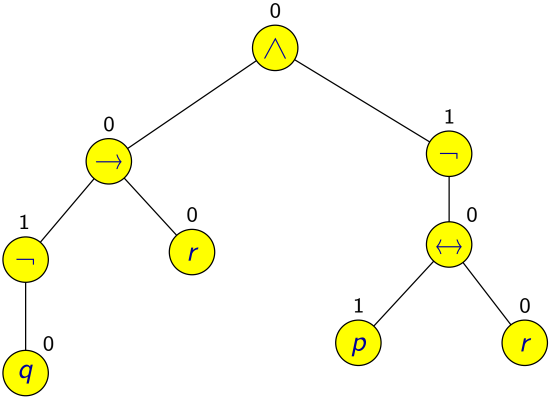
$\varphi \ := \ (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_3(p) = 0 \qquad v_3(q) = 1 \qquad v_3(r) = 1$

Semantics of Propositional Logic

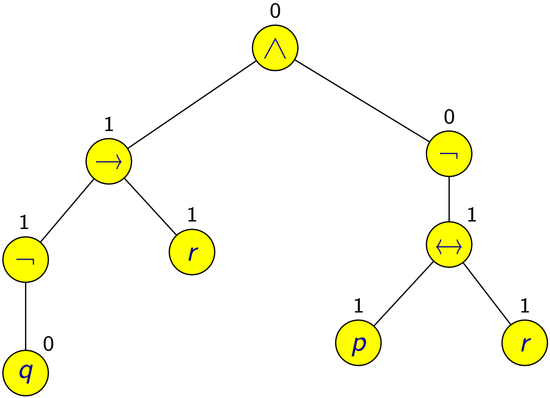
$\varphi \;:=\; (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_4(p) = 1 \qquad v_4(q) = 0 \qquad v_4(r) = 0$

Semantics of Propositional Logic

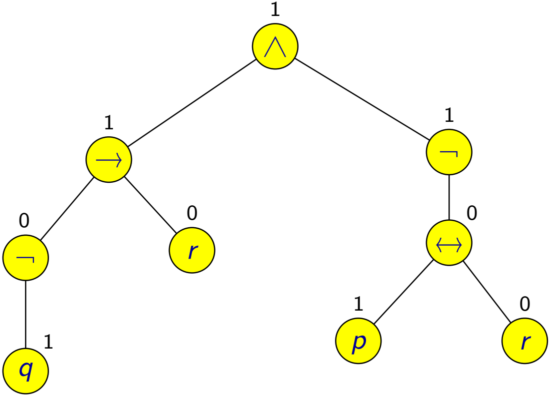
$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_5(p) = 1 \qquad v_5(q) = 0 \qquad v_5(r) = 1$

Semantics of Propositional Logic

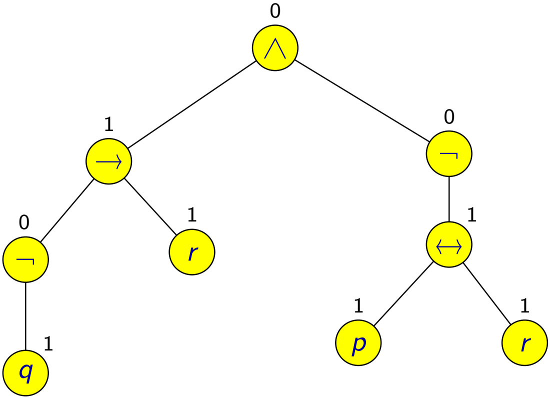
$\varphi := (\neg q \to r) \wedge \neg(p \leftrightarrow r)$



$v_6(p) = 1 \qquad v_6(q) = 1 \qquad v_6(r) = 0$

Semantics of Propositional Logic

$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$



$v_7(p) = 1 \qquad v_7(q) = 1 \qquad v_7(r) = 1$

$\varphi := (\neg q \rightarrow r) \wedge \neg(p \leftrightarrow r)$

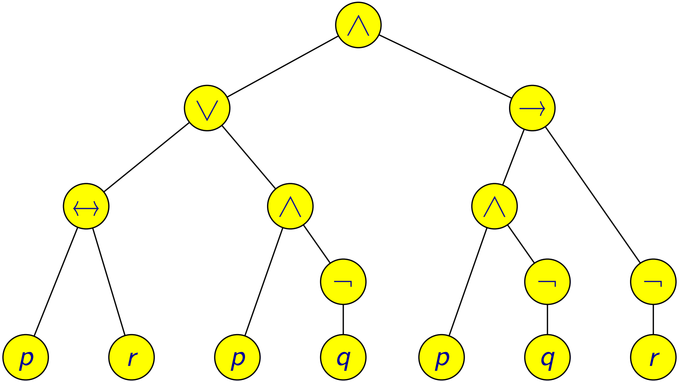
p	q	r	$\neg q$	$\neg q \rightarrow r$	$p \leftrightarrow r$	$\neg(p \leftrightarrow r)$	φ
0	0	0	1	0	1	0	0
0	0	1	1	1	0	1	1
0	1	0	0	1	1	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	0	1	0
1	0	1	1	1	1	0	0
1	1	0	0	1	0	1	1
1	1	1	0	1	1	0	0

Those valuations, where the given formula is true, are called its **models**:

$$
\begin{array}{llll}
v_1: & v_1(p) = 0, & v_1(q) = 0, & v_1(r) = 1, \\
v_3: & v_3(p) = 0, & v_3(q) = 1, & v_3(r) = 1, \\
v_6: & v_6(p) = 1, & v_6(q) = 1, & v_6(r) = 0,
\end{array}
$$

Formulas Represented as Directed Acyclic Graphs

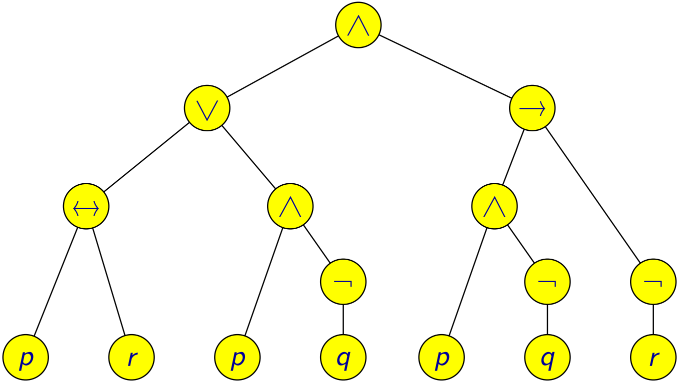
$\varphi \; := \; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \; \wedge \; ((p \wedge \neg q) \rightarrow \neg r)$



In an abstract syntax tree, nodes correspond to **occurrences** of subformulas.

Formulas Represented as Directed Acyclic Graphs

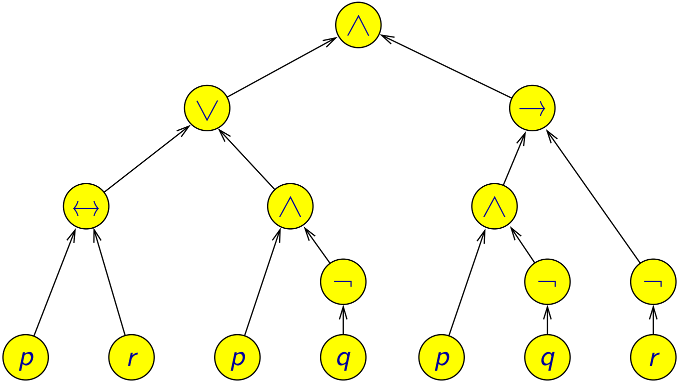
$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



Alternatively, a formula can be represented as a **directed acyclic graph**.

Formulas Represented as Directed Acyclic Graphs

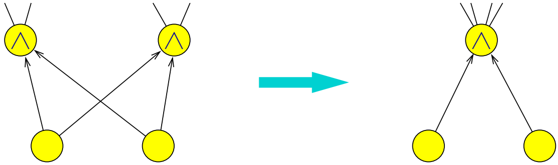
$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



Edges are oriented from children to their parents.

Formulas Represented as Directed Acyclic Graphs

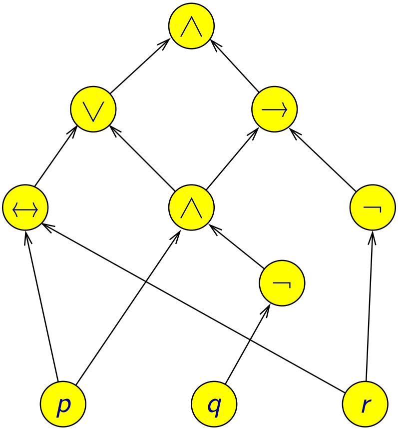
- Leafs labelled with the same atomic proposition are merged together.
- Those nodes, which are labelled with the same symbol and which have the same predecessors, can be merged.



- Such merging of nodes can be done repeatedly — when some nodes are merged, merging of some other nodes may become possible.

Formulas Represented as Directed Acyclic Graphs

$$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \wedge ((p \wedge \neg q) \rightarrow \neg r)$$

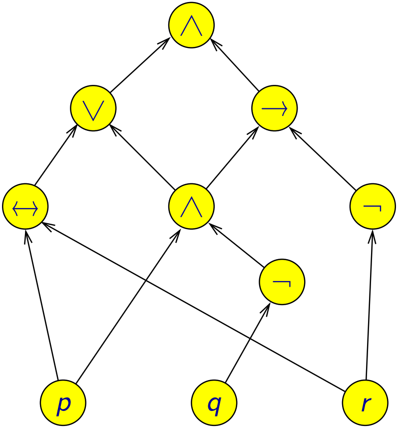


Formulas Represented as Directed Acyclic Graphs

- When we merge all nodes that can be possibly merged this way, we obtain a graph where individual nodes correspond to different subformulas of a given formula.
- A directed acyclic graph representing a given formula can be viewed as a **logic circuit**:
    - **Inputs** — nodes labelled with atomic propositions
    - **Output** — the node corresponding to the whole formula

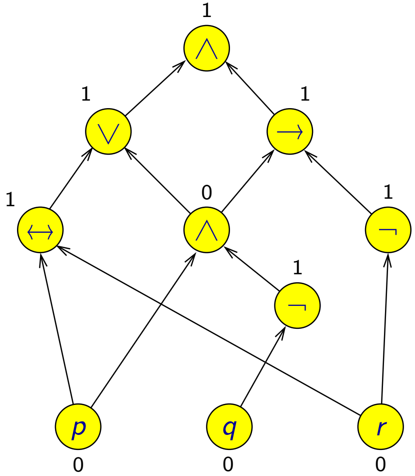
Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



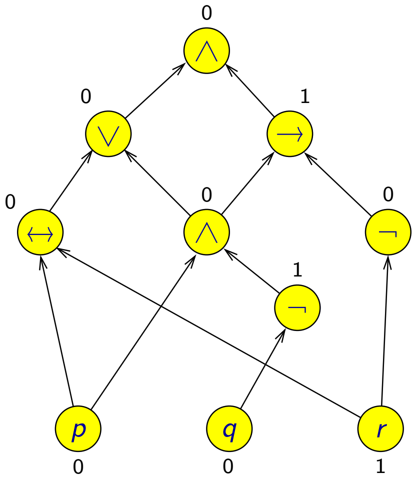
Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



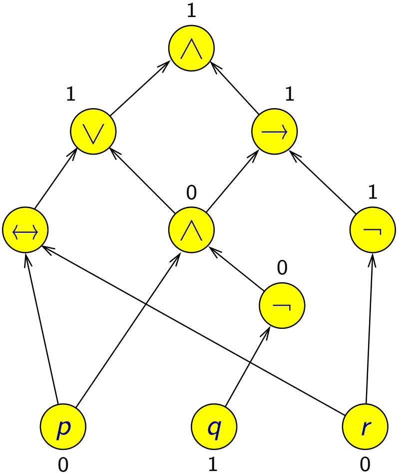
Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



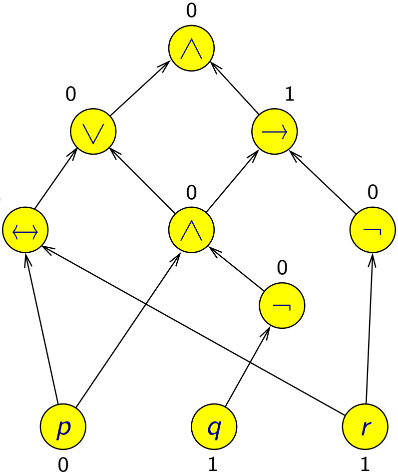
Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



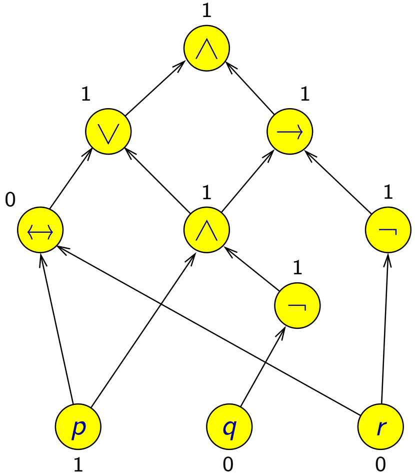
Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



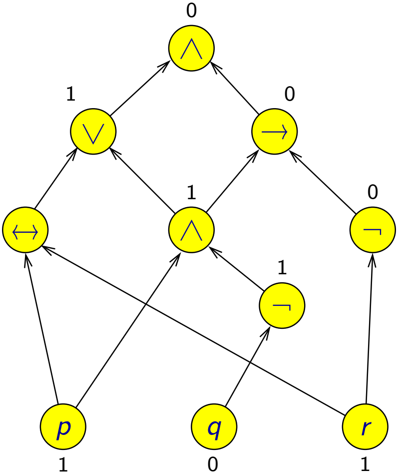
Formulas Represented as Directed Acyclic Graphs

$\varphi \ := \ ((p \leftrightarrow r) \vee (p \wedge \neg q)) \ \wedge \ ((p \wedge \neg q) \rightarrow \neg r)$



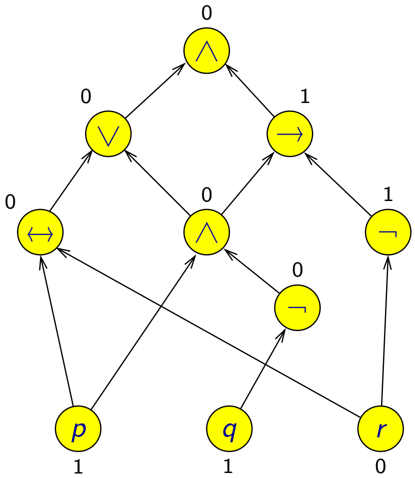
Formulas Represented as Directed Acyclic Graphs

$\varphi \ := \ ((p \leftrightarrow r) \vee (p \wedge \neg q)) \ \wedge \ ((p \wedge \neg q) \rightarrow \neg r)$



Formulas Represented as Directed Acyclic Graphs

$\varphi \;:=\; ((p \leftrightarrow r) \vee (p \wedge \neg q)) \;\wedge\; ((p \wedge \neg q) \rightarrow \neg r)$



Formulas Represented as Directed Acyclic Graphs

$\varphi \ := \ ((p \leftrightarrow r) \vee (p \wedge \neg q)) \ \wedge \ ((p \wedge \neg q) \rightarrow \neg r)$