

Natural Language Inference: from Aristotle to AI

Ian Pratt-Hartmann

Department of Computer Science, Manchester University, UK
Instytut Informatyki, Uniwersytet Opolski, Polska
email: ian.pratt@manchester.ac.uk

(Lawrence S. Moss, Allan Third, Viktor Schlegel, Tharindu Madusanka)

ISC 2026 Summer School
Knowledge, Reasoning, and Decision-Making
Montréal, 8th June, 2026

- The **classical (Aristotelian) syllogistic** is a formal language featuring the following sentence-forms

Every p is q

No p is q

Some p is q

Some p is not a q



- The **classical** (**Aristotelian**) **sylogistic** is a formal language featuring the following sentence-forms

Every p is q Some p is q
No p is q Some p is not a q



- The letters p and q are to be substituted by common (count) nouns such as *artist*, *beekeeper*, *carpenter*

- The **classical** (**Aristotelian**) **sylogistic** is a formal language featuring the following sentence-forms

Every p is q Some p is q
No p is q Some p is not a q



- The letters p and q are to be substituted by common (count) nouns such as *artist*, *beekeeper*, *carpenter*
- These are called **terms**, so the classical sylogistic is a **term logic**.

- The **classical** (**Aristotelian**) **sylogistic** is a formal language featuring the following sentence-forms

Every p is q Some p is q
No p is q Some p is not a q



- The letters p and q are to be substituted by common (count) nouns such as *artist*, *beekeeper*, *carpenter*
- These are called **terms**, so the classical sylogistic is a **term logic**.
- Aristotle invented it (*ab initio*) in the fourth century B.C. It is set out in the *Prior Analytics*, book A.

- Today, we would write the classical syllogistic in first-order logic:

Every p is q
No p is q

Some p is q
Some p is not a q

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall x(p(x) \rightarrow q(x))$ Some p is q $\exists x(p(x) \wedge q(x))$
No p is q $\forall x(p(x) \rightarrow \neg q(x))$ Some p is not a q $\exists x(p(x) \wedge \neg q(x))$

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall (p(x) \rightarrow q(x))$ Some p is q $\exists (p(x) \wedge q(x))$
 No p is q $\forall (p(x) \rightarrow \neg q(x))$ Some p is not a q $\exists (p(x) \wedge \neg q(x))$

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall(p, q)$ Some p is q $\exists(p, q)$
No p is q $\forall(p, \bar{q})$ Some p is not a q $\exists(p, \bar{q})$

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall(p, q)$ Some p is q $\exists(p, q)$
No p is q $\forall(p, \bar{q})$ Some p is not a q $\exists(p, \bar{q})$

Denote this language by \mathcal{S} .

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall(p, q)$ Some p is q $\exists(p, q)$
 No p is q $\forall(p, \bar{q})$ Some p is not a q $\exists(p, \bar{q})$

Denote this language by \mathcal{S} .

- There are some missing possibilities:

Every **non- p** is q $\forall(\bar{p}, q)$ Some **non- p** is not a q $\exists(\bar{p}, \bar{q})$

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall(p, q)$ Some p is q $\exists(p, q)$
 No p is q $\forall(p, \bar{q})$ Some p is not a q $\exists(p, \bar{q})$

Denote this language by \mathcal{S} .

- There are some missing possibilities:

Every **non- p** is q $\forall(\bar{p}, q)$ Some **non- p** is not a q $\exists(\bar{p}, \bar{q})$

But we can add these formulas. Call this extended language the **extended classical syllogistic**, and denote it by \mathcal{S}^+ .

- Today, we would write the classical syllogistic in first-order logic:

Every p is q $\forall(p, q)$ Some p is q $\exists(p, q)$
 No p is q $\forall(p, \bar{q})$ Some p is not a q $\exists(p, \bar{q})$

Denote this language by \mathcal{S} .

- There are some missing possibilities:

Every **non- p** is q $\forall(\bar{p}, q)$ Some **non- p** is not a q $\exists(\bar{p}, \bar{q})$

But we can add these formulas. Call this extended language the **extended classical syllogistic**, and denote it by \mathcal{S}^+ .

- Aristotle considered these forms in *De Interpretatione*, but did nothing with them.

- The Prior Analytics analyses logical relations involving the formulas of \mathcal{S} , codified as **syllogisms**:

$$\frac{\forall(p, \bar{q}) \quad \exists(o, p)}{\exists(o, \bar{q})}$$

$$\frac{\forall(q, p) \quad \exists(o, \bar{p})}{\exists(o, \bar{q})}$$

- The Prior Analytics analyses logical relations involving the formulas of \mathcal{S} , codified as **syllogisms**:

$$\frac{\forall(p, \bar{q}) \quad \exists(o, p)}{\exists(o, \bar{q})}$$

$$\frac{\forall(q, p) \quad \exists(o, \bar{p})}{\exists(o, \bar{q})}$$

- These syllogisms can be chained together to form arguments:

$$\frac{\frac{\exists(\text{artist}, \text{bkpr}) \quad \forall(\text{beekpr}, \overline{\text{cpntr}})}{\exists(\text{artst}, \overline{\text{cptnr}})} \quad \forall(\text{gdnr}, \text{cpntr})}{\exists(\text{artst}, \overline{\text{gdnr}})}$$

- The Prior Analytics analyses logical relations involving the formulas of \mathcal{S} , codified as **sylogisms**:

$$\frac{\forall(p, \bar{q}) \quad \exists(o, p)}{\exists(o, \bar{q})}$$

$$\frac{\forall(q, p) \quad \exists(o, \bar{p})}{\exists(o, \bar{q})}$$

- These syllogisms can be chained together to form arguments:

$$\frac{\exists(\text{artist}, \text{bkpr}) \quad \forall(\text{beekpr}, \overline{\text{cptnr}})}{\exists(\text{artst}, \overline{\text{cptnr}})} \quad \forall(\text{gdnr}, \text{cptnr})$$

$$\frac{\exists(\text{artst}, \overline{\text{cptnr}}) \quad \forall(\text{gdnr}, \text{cptnr})}{\exists(\text{artst}, \overline{\text{gdnr}})}$$

- Aristotle listed the 24 valid syllogisms (of the 256 possible).

- The Prior Analytics analyses logical relations involving the formulas of \mathcal{S} , codified as **sylogisms**:

$$\frac{\forall(p, \bar{q}) \quad \exists(o, p)}{\exists(o, \bar{q})}$$

$$\frac{\forall(q, p) \quad \exists(o, \bar{p})}{\exists(o, \bar{q})}$$

- These syllogisms can be chained together to form arguments:

$$\frac{\frac{\exists(\text{artist}, \text{bkpr}) \quad \forall(\text{beekpr}, \overline{\text{cptnr}})}{\exists(\text{artst}, \overline{\text{cptnr}})} \quad \forall(\text{gdnr}, \text{cptnr})}{\exists(\text{artst}, \overline{\text{gdnr}})}$$

- Aristotle listed the 24 valid syllogisms (of the 256 possible).
- We write $\Phi \vdash_X \psi$ if, from premises Φ we can derive ψ using the rules in some set of rules X .

- Nowadays, we think in terms of **models**

Some artists are beekeepers

No beekeepers are carpenters

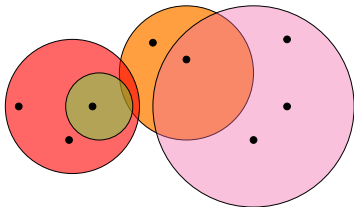
All gardeners are carpenters

- Nowadays, we think in terms of **models**

Some **artists** are **beekeepers**

No **beekeepers** are **carpenters**

All **gardeners** are **carpenters**



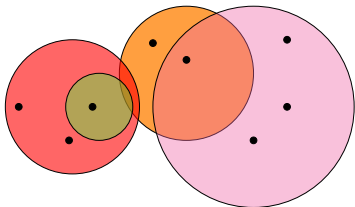
- Nowadays, we think in terms of **models**

Some **artists** are **beekeepers**

No **beekeepers** are **carpenters**

All **gardeners** are **carpenters**

Some **artists** are not **gardeners**.



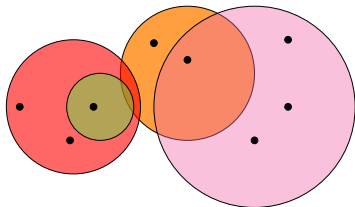
- Nowadays, we think in terms of **models**

Some **artists** are **beekeepers**

No **beekeepers** are **carpenters**

All **gardeners** are **carpenters**

Some **artists** are not **gardeners**.



- We write $\Phi \models \psi$ if every model of Φ is a model of ψ .

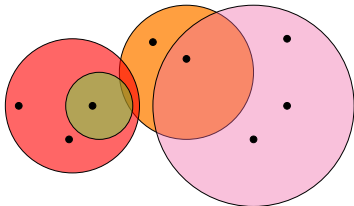
- Nowadays, we think in terms of **models**

Some **artists** are **beekeepers**

No **beekeepers** are **carpenters**

All **gardeners** are **carpenters**

Some **artists** are not **gardeners**.



- We write $\Phi \models \psi$ if every model of Φ is a model of ψ .
- Note that $\Phi \models \psi$ just in case the set $\Phi \cup \{\bar{\psi}\}$ is **unsatisfiable** (has no model).

- Are Aristotle's rules adequate?
- A set of rules X is:
 - X is **sound** if $\Phi \vdash_X \psi$ entails $\Phi \models \psi$;
 - X is **complete** if $\Phi \models \psi$ entails $\Phi \vdash_X \psi$.
- With some goodwill, Aristotle's rules can be shown to be sound and complete for \mathcal{S} .
 - J. Łukasiewicz (J. Śłupecki) (39/57);
 - J. Corcoran (72) and T.J. Smiley (73)

The following set of rules is sound and complete for \mathcal{S}
 (PH and Moss 09)

$$\begin{array}{c}
 \frac{\forall(q, \ell) \quad \exists(p, q)}{\exists(p, \ell)} \qquad \frac{\forall(p, q) \quad \forall(q, \ell)}{\forall(p, \ell)} \qquad \frac{\forall(p, \bar{p})}{\forall(p, \ell)} \\
 \\
 \frac{\exists(p, \ell) \quad \forall(p, q)}{\exists(q, \ell)} \qquad \frac{\psi \quad \bar{\psi}}{\varphi} \\
 \\
 \frac{\forall(q, \bar{\ell}) \quad \exists(p, \ell)}{\exists(p, \bar{q})} \qquad \frac{}{\forall(p, p)} \qquad \frac{\exists(p, \ell)}{\exists(p, p)}
 \end{array}$$

Here we denote the **negation** of ψ by $\bar{\psi}$, thus: $\overline{\forall(p, q)} = \exists(p, \bar{q})$.

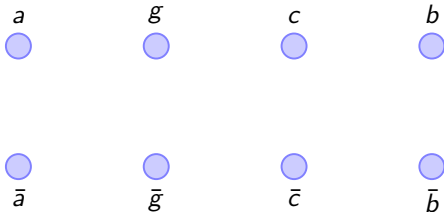
- A better method: consider the valid argument

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \models \quad \exists(a, \bar{g}).$$

- A better method: consider the (unsatisfiable) set

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \forall(a, g).$$

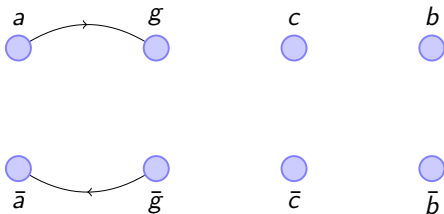
- We map the universal statements to a directed graph:



- A better method: consider the (unsatisfiable) set

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \forall(a, g).$$

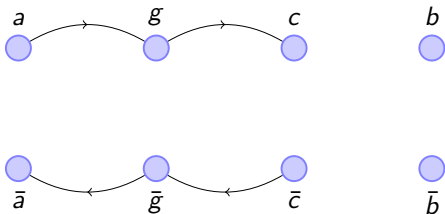
- We map the universal statements to a directed graph:



- A better method: consider the (unsatisfiable) set

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \forall(a, g).$$

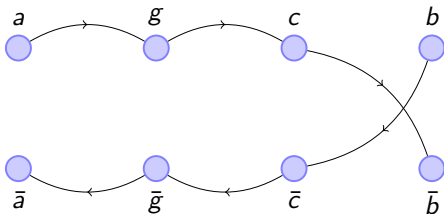
- We map the universal statements to a directed graph:



- A better method: consider the (unsatisfiable) set

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \forall(a, \bar{g}).$$

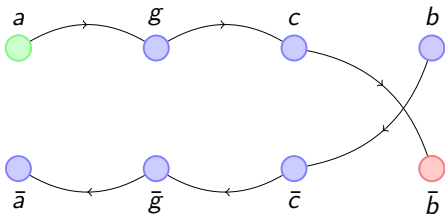
- We map the universal statements to a directed graph:



- A better method: consider the (unsatisfiable) set

$$\exists(a, b) \quad \forall(g, c) \quad \forall(b, \bar{c}) \quad \forall(a, g).$$

- We map the universal statements to a directed graph:



- A better method: consider the (unsatisfiable) set

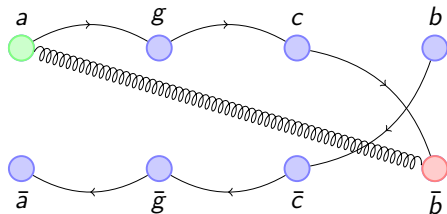
$\exists(a, b)$

$\forall(g, c)$

$\forall(b, \bar{c})$

$\forall(a, \bar{g})$.

- We map the universal statements to a directed graph:



- Now look to see if there is an existential statement with an 'opposing path'.

- A better method: consider the (unsatisfiable) set

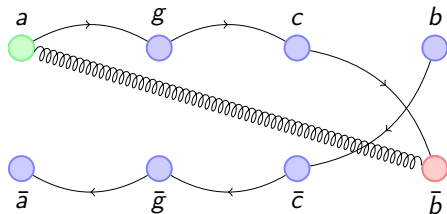
$\exists(a, b)$

$\forall(g, c)$

$\forall(b, \bar{c})$

$\forall(a, \bar{g})$.

- We map the universal statements to a directed graph:



- Now look to see if there is an existential statement with an 'opposing path'.
- All we need to do is to find certain cycles in a directed graph. This method works for \mathcal{S} and \mathcal{S}^+ .

- Reachability in directed graphs is in **NLOGSPACE**.
- Hence, the satisfiability problems for \mathcal{S} and \mathcal{S}^+ are in **NLOGSPACE**.

- Reachability in directed graphs is in **NLOGSPACE**.
- Hence, the satisfiability problems for \mathcal{S} and \mathcal{S}^+ are in **NLOGSPACE**.
- With a little work, a reduction in the reverse direction can be given.
- Hence the satisfiability problems for \mathcal{S} and \mathcal{S}^+ are **NLOGSPACE-complete**.

- Mediæval logicians ([Buridan](#), [Ockham](#), ...) frequently pondered on sentences such as:

omnis homo videt omnem asinum.

- Mediæval logicians (Buridan, Ockham, ...) frequently pondered on sentences such as:

omnis homo videt omnem asinum.

- Why? Because the syllogistic struggles with relations.

- Mediæval logicians (Buridan, Ockham, ...) frequently pondered on sentences such as:

omnis homo videt omnem asinum.

- Why? Because the syllogistic struggles with relations.
- Such inferences were also of interest to the Port-Royalists (Arnauld, Nicole, Pascale):

The sun is an insensible thing

The Persians worshipped the sun

The Persians worshipped an insensible thing



- Mediæval logicians ([Buridan](#), [Ockham](#), ...) frequently pondered on sentences such as:

omnis homo videt omnem asinum.

- Why? Because the syllogistic struggles with relations.
- Such inferences were also of interest to the Port-Royalists ([Arnauld](#), [Nicole](#), [Pascale](#)):

The sun is an insensible thing
The Persians worshipped the sun
The Persians worshipped an insensible thing



- A. De Morgan most clearly saw that relational reasoning requires new principles of inference:

Every horse is an animal
Every horse's head is an animal's head.



- Here is a simpler example:

Every beekeeper admires some artist
Some artist admires no beekeeper
Some artist is not a beekeeper.

- Here is a simpler example:

Every beekeeper admires some artist
Some artist admires no beekeeper
Some artist is not a beekeeper.

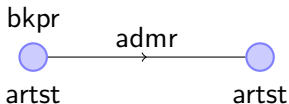
bkpr



artst

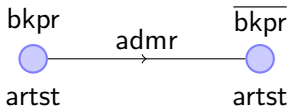
- Here is a simpler example:

Every beekeeper admires some artist
Some artist admires no beekeeper
Some artist is not a beekeeper.



- Here is a simpler example:

Every beekeeper admires some artist
Some artist admires no beekeeper
Some artist is not a beekeeper.



- Let us define the **relational syllogistic**, \mathcal{R} , to be the language consisting of the classical syllogistic together with statements of the following forms:

{Every/some} p r 's {every/some} q

No p r 's {any/every} q

Some p r 's no q

Some p does not r every q .

- The presentation in first-order logic is more uniform ($\pm r(x, y)$ stands for $r(x, y)$ or $\neg r(x, y)$):

$$\forall x(p(x) \rightarrow \forall y(q(y) \rightarrow \pm r(x, y)))$$

$$\forall x(p(x) \rightarrow \exists y(q(y) \wedge \pm r(x, y)))$$

$$\exists x(p(x) \wedge \forall y(q(y) \rightarrow \pm r(x, y)))$$

$$\exists x(p(x) \wedge \exists y(q(y) \wedge \pm r(x, y))),$$

where p and q are unary predicates and r a binary predicate.

- The presentation in first-order logic is more uniform ($\pm r(x, y)$ stands for $r(x, y)$ or $\neg r(x, y)$):

$$\begin{aligned} & \forall (p \quad \forall (q \quad \pm r \quad)) \\ & \forall (p \quad \exists (q \quad \pm r \quad)) \\ & \exists (p \quad \forall (q \quad \pm r \quad)) \\ & \exists (p \quad \exists (q \quad \pm r \quad)), \end{aligned}$$

where p and q are unary predicates and r a binary predicate.

- The presentation in first-order logic is more uniform ($\pm r(x, y)$ stands for $r(x, y)$ or $\neg r(x, y)$):

$$\forall(p, \forall(q, \pm r))$$

$$\forall(p, \exists(q, \pm r))$$

$$\exists(p, \forall(q, \pm r))$$

$$\exists(p, \exists(q, \pm r)),$$

where p and q are unary predicates and r a binary predicate.

- The presentation in first-order logic is more uniform ($\pm r(x, y)$ stands for $r(x, y)$ or $\neg r(x, y)$):

$$\forall(p, \forall(q, \pm r))$$

$$\forall(p, \exists(q, \pm r))$$

$$\exists(p, \forall(q, \pm r))$$

$$\exists(p, \exists(q, \pm r)),$$

where p and q are unary predicates and r a binary predicate.

- Thus, for example,

Every beekeeper admires some artist

$$\forall(\text{bkpr}, \exists(\text{artst}, \text{admr})).$$

- It is possible to devise a rule-system for \mathcal{R} that is sound and **refutation**-complete (P-H and Moss 2010), featuring rules corresponding to 'oblique syllogisms':

$$\frac{\forall(p, \forall(o, t)) \quad \exists(o, q)}{\forall(p, \exists(q, t))} \qquad \frac{\exists(p, \forall(o, t)) \quad \forall(q, o)}{\exists(p, \forall(q, t))}$$

- It is possible to devise a rule-system for \mathcal{R} that is sound and **refutation**-complete (P-H and Moss 2010), featuring rules corresponding to ‘oblique syllogisms’:

$$\frac{\forall(p, \forall(o, t)) \quad \exists(o, q)}{\forall(p, \exists(q, t))} \qquad \frac{\exists(p, \forall(o, t)) \quad \forall(q, o)}{\exists(p, \forall(q, t))}$$

- With some work, one can again analyse the problem of determining satisfiability in \mathcal{R} in terms of directed graphs, just as we did for \mathcal{S} .
- It can thus be shown that this problem remains **NLOGSPACE-complete**.

- As with the classical syllogistic, we can add ‘noun-level’ negation to \mathcal{R} :

Every artist admires some non-beekeeper
Some carpenter hates no non-dentist.

Call this language the **extended relational syllogistic**, \mathcal{R}^+ .

- As with the classical syllogistic, we can add ‘noun-level’ negation to \mathcal{R} :

Every artist admires some non-beekeeper
Some carpenter hates no non-dentist.

Call this language the **extended relational syllogistic**, \mathcal{R}^+ .

- There is no sound and (refutation-) complete set of syllogism-like rules for \mathcal{R} (P-H and Moss, 2010).

- As with the classical syllogistic, we can add ‘noun-level’ negation to \mathcal{R} :

Every artist admires some non-beekeeper
Some carpenter hates no non-dentist.

Call this language the **extended relational syllogistic**, \mathcal{R}^+ .

- There is no sound and (refutation-) complete set of syllogism-like rules for \mathcal{R} (P-H and Moss, 2010).
- The satisfiability problem for \mathcal{R}^+ is **EXPTIME-complete**.

- One could alternatively extend \mathcal{R} by allowing relational terms in the subject, thus

$$\forall(\exists(p, r), \exists(q, r)).$$

- In the resulting fragment, which we might denote \mathcal{R}^* , we can express De Morgan's example:

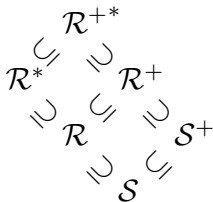
Every horse is an animal

Every horse's head is an animal's head.

This language is close to one investigated by [McAllister](#) and [Givan](#) 92.

- We can of course extend \mathcal{R}^* with noun-level negation to obtain the language \mathcal{R}^{+*} .

- Summary of the Classical and relational syllogistics:



\mathcal{S}	SC	NLOGSPACE
\mathcal{S}^+	SC	NLOGSPACE
\mathcal{R}	SRC	NLOGSPACE
\mathcal{R}^+	No SC indirect	EXPTIME
\mathcal{R}^*	SC indirect	NPTIME
\mathcal{R}^{+*}	No SC indirect	EXPTIME

- There is no need to stop! We can add **relative clauses** to \mathcal{S} :

Every philosopher **who** is not a stoic is a cynic

$$\forall x(\text{philosopher}(x) \wedge \neg \text{stoic}(x) \rightarrow \text{cynic}(x)).$$

- Or we can extend \mathcal{R} by allowing **ditransitive verbs**:

No stoic **recommends** every sceptic to some cynic

$$\forall x(\text{stoic}(x) \rightarrow \neg \forall y(\text{sceptic}(y) \rightarrow \exists z(\text{cynic}(z) \wedge \text{recommend}(x, y, z))))).$$

- New validities are then expressible. In $\mathcal{S}+\text{Rel}$:

Every philosopher who is not a stoic is a cynic

Every stoic is a man

Every cynic is a man

Every philosopher is a man,

- And in $\mathcal{R}+\text{DTV}+\text{Rel}$:

Every sceptic recommends every sceptic to every cynic

No sceptic recommends any stoic who hates some cynic to any philosopher

Diogenes is a cynic whom every sceptic hates

Every cynic is a philosopher

No stoic is a sceptic.

- Bigger summary (PH and Third, 2006)

Fragment	Complexity
$\mathcal{R}+DTV$	P TIME
$\mathcal{S}+Rel$	NP
$\mathcal{R}+Rel$	EXPTIME
$\mathcal{R}+Rel+DTV$	NEXPTIME
$\mathcal{R}+Rel+RA$	NEXPTIME
$\mathcal{R}+Rel+GA$	co-r.e.
$\mathcal{R}+Rel+DTV+RA$	co-r.e.

- Bigger summary (PH and Third, 2006)

Fragment	Complexity
$\mathcal{R}+DTV$	P TIME
$\mathcal{S}+Rel$	NP
$\mathcal{R}+Rel$	EXPTIME
$\mathcal{R}+Rel+DTV$	NEXPTIME
$\mathcal{R}+Rel+RA$	NEXPTIME
$\mathcal{R}+Rel+GA$	co-r.e.
$\mathcal{R}+Rel+DTV+RA$	co-r.e.

- The basic idea can be found in A. Morawiec (1961) *Podstawy logiki nazw* and also in various papers by Patrick Suppes in the 1970s.

- What about **numbers**? Let us re-arrange the forms of the Classical syllogistic as follows:

No p is a q

Every p is a q

Some p is a q

Some p is not a q ,

- and re-write them using numerical quantifiers, as follows:

At most 0 ps are qs

At most 0 ps are not qs

More than 0 ps are qs

More than 0 ps are not qs .

- This suggests a natural generalization in which we can formalize:

At least thirteen **artists** are **beekeepers**

At most three **beekeepers** are **carpenters**

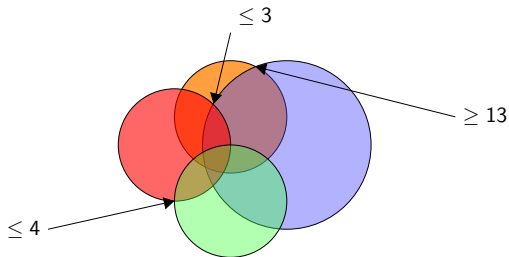
At most four **gardeners** are not **carpenters**

- This suggests a natural generalization in which we can formalize:

At least thirteen **artists** are **beekeepers**

At most three **beekeepers** are **carpenters**

At most four **gardeners** are not **carpenters**



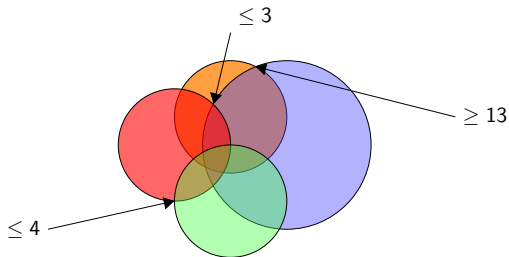
- This suggests a natural generalization in which we can formalize:

At least thirteen **artists** are **beekeepers**

At most three **beekeepers** are **carpenters**

At most four **gardeners** are not **carpenters**

At least six **artists** are not **gardeners**.



- De Morgan (1848) proposed some plausible generalizations of the Classical syllogisms:

At most m o are not p

At most n p are not q

At most $(m + n)$ o are not q

At least m o are p

At most n p are not q

At least $(m - n)$ o are q

These rules are not complete.

- De Morgan (1848) proposed some plausible generalizations of the Classical syllogisms:

At most m o are not p

At most n p are not q

At most $(m + n)$ o are not q

At least m o are p

At most n p are not q

At least $(m - n)$ o are q

These rules are not complete.

- A better method, suggested by [W.S. Jevons](#), is to reduce to integer linear programming feasibility



- De Morgan (1848) proposed some plausible generalizations of the Classical syllogisms:

At most m o are not p

At most n p are not q

At most $(m + n)$ o are not q

At least m o are p

At most n p are not q

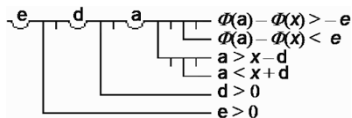
At least $(m - n)$ o are q

These rules are not complete.

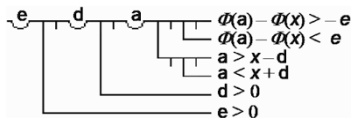
- A better method, suggested by [W.S. Jevons](#), is to reduce to integer linear programming feasibility
- This is how it's done nowadays: satisfiability for the numerical syllogistic is **NPTIME-complete** and for the numerical relational syllogistic, **NEXPTIME-complete**.



- In 1879, G. Frege published his *Begriffsschrift*: a language designed to formalize mathematics.



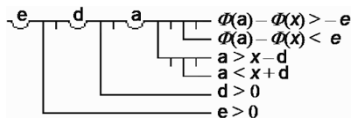
- In 1879, **G. Frege** published his *Begriffsschrift*: a language designed to formalize mathematics.



- Frege's language was taken up by **B. Russell** and **A.N. Whitehead**.



- In 1879, **G. Frege** published his *Begriffsschrift*: a language designed to formalize mathematics.

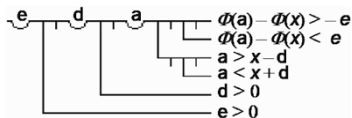


- Frege's language was taken up by **B. Russell** and **A.N. Whitehead**.



- Their key innovation: **bound-variable quantification**.

- In 1879, **G. Frege** published his *Begriffsschrift*: a language designed to formalize mathematics.

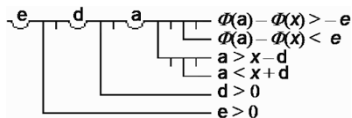


- Frege's language was taken up by **B. Russell** and **A.N. Whitehead**.



- Their key innovation: **bound-variable quantification**.
Every boy loves some girl.

- In 1879, **G. Frege** published his *Begriffsschrift*: a language designed to formalize mathematics.



- Frege's language was taken up by **B. Russell** and **A.N. Whitehead**.

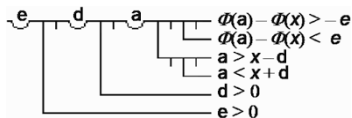


- Their key innovation: **bound-variable quantification**.

Every boy loves some girl.

For every x , if x is a boy, then there exists some y such that y is a girl and x loves y .

- In 1879, **G. Frege** published his *Begriffsschrift*: a language designed to formalize mathematics.



- Frege's language was taken up by **B. Russell** and **A.N. Whitehead**.



- Their key innovation: **bound-variable quantification**.

Every boy loves some girl.

For every x , if x is a boy, then there exists some y such that y is a girl and x loves y .

$$\forall x(\text{boy}(x) \rightarrow \exists y(\text{girl}(y) \wedge \text{loves}(x, y)))$$

- If we restrict quantification to the domain of objects, we obtain **first-order logic**, or \mathcal{FO} .
- In their *Grundzüge der theoretischen Logik* (1928), **D. Hilbert** and **W. Ackermann** formulated the *Entscheidungsproblem* (decision problem):

Does there exist an algorithm to determine whether a given formula of first-order logic is satisfiable?



- This question was answered negatively by **A. Church** and **A. Turing** in 1936/7.

- On the other hand, a positive answer to the *Entscheidungsproblem* is available for various special cases.

- On the other hand, a positive answer to the *Entscheidungsproblem* is available for various special cases.
- This yields a classification problem: for which **fragments** of first-order logic is (finite) satisfiability decidable?

- On the other hand, a positive answer to the *Entscheidungsproblem* is available for various special cases.
- This yields a classification problem: for which **fragments** of first-order logic is (finite) satisfiability decidable?
- If \mathcal{L} is a fragment of first-order logic, we write $\text{Sat}(\mathcal{L})$ for the satisfiability problem for \mathcal{L} .

- On the other hand, a positive answer to the *Entscheidungsproblem* is available for various special cases.
- This yields a classification problem: for which **fragments** of first-order logic is (finite) satisfiability decidable?
- If \mathcal{L} is a fragment of first-order logic, we write $\text{Sat}(\mathcal{L})$ for the satisfiability problem for \mathcal{L} .
- When this problems is decidable, we want to know its **computational complexity**.

- Many fragments of first-order logic are known to be decidable for satisfiability and finite satisfiability.

- Many fragments of first-order logic are known to be decidable for satisfiability and finite satisfiability.
- The most commonly encountered fragments:
 - **quantifier-prefix fragments**: $[\forall^*\exists^*]_=$, $[\exists^*\forall\exists^*]_=$ and $[\exists^*\forall^2\exists^*]$;
 - the **two variable fragment**: \mathcal{FO}^2 and its variants ...
 - the **guarded fragment**: \mathcal{GF} and its variants ...
 - the **unary negation fragment**, \mathcal{UN}
 - ...

- Many fragments of first-order logic are known to be decidable for satisfiability and finite satisfiability.
- The most commonly encountered fragments:
 - **quantifier-prefix fragments**: $[\forall^* \exists^*]_=$, $[\exists^* \forall \exists^*]_=$ and $[\exists^* \forall^2 \exists^*]$;
 - the **two variable fragment**: \mathcal{FO}^2 and its variants ...
 - the **guarded fragment**: \mathcal{GF} and its variants ...
 - the **unary negation fragment**, \mathcal{UN}
 - ...
- None of these owe their salience to natural language syntax.
- As we have seen, such fragments can be sensibly investigated.

- How to solve logic problems (if you're a bad student):

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)
 - Are there very few negations?

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)
 - Are there very few negations?
 - Are there loads more existential quantifiers than universal quantifiers?

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)
 - Are there very few negations?
 - Are there loads more existential quantifiers than universal quantifiers?
 - Is the non-logical vocabulary large compared to the number of sentences?

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)
 - Are there very few negations?
 - Are there loads more existential quantifiers than universal quantifiers?
 - Is the non-logical vocabulary large compared to the number of sentences?
 - Try to remember the examples from class: is this vaguely similar to any of them?

- How to solve logic problems (if you're a bad student):
- (We focus on the problem of determining **satisfiability** of a collection of English sentences.)
 - Are there very few negations?
 - Are there loads more existential quantifiers than universal quantifiers?
 - Is the non-logical vocabulary large compared to the number of sentences?
 - Try to remember the examples from class: is this vaguely similar to any of them?
- These methods can work depressingly well ...

- How does a bad student solve logical satisfiability problems *nowadays?*

- How does a bad student solve logical satisfiability problems *nowadays?*



- In the last decade, various attempts have been made to explore the ability of language models to recognize logical entailments (e.g. Bowman et al., 2015).
- In most early experiments, problem-instances were presented in the form of premise hypothesis pairs:

A soccer game with multiple males playing
Some men are playing a sport

- The SNLI corpus contains 500K such sentence-pairs:
 - logical inference and factual knowledge are (deliberately) inter-twined;
 - the 'entailments' are in many cases probabilistic in character;
 - the data are crowd-sourced.
- Questions were immediately raised about dependence of these results on data artefacts.

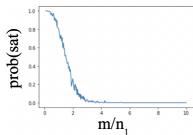
- An alternative approach is to investigate the ability of language models to recognize logical entailments *sensu stricto*.

- An alternative approach is to investigate the ability of language models to recognize logical entailments *sensu stricto*.
- For various fragments (\mathcal{S}^+ , $\mathcal{S}^++\text{Rel}$, \mathcal{R}^+ , $\mathcal{R}^++\text{Rel}$...) we can construct large corpora ($\approx 140\text{K}$) of sets of sentences.

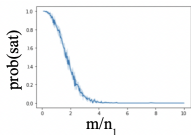
- An alternative approach is to investigate the ability of language models to recognize logical entailments *sensu stricto*.
- For various fragments (\mathcal{S}^+ , $\mathcal{S}^++\text{Rel}$, \mathcal{R}^+ , $\mathcal{R}^++\text{Rel}$...) we can construct large corpora ($\approx 140\text{K}$) of sets of sentences.
- We can tag each data point (set of sentences) as satisfiable/unsatisfiable.

- An alternative approach is to investigate the ability of language models to recognize logical entailments *sensu stricto*.
- For various fragments (\mathcal{S}^+ , $\mathcal{S}^++\text{Rel}$, \mathcal{R}^+ , $\mathcal{R}^++\text{Rel}$...) we can construct large corpora ($\approx 140\text{K}$) of sets of sentences.
- We can tag each data point (set of sentences) as satisfiable/unsatisfiable.
- We can train various language models, and test them on unseen data.

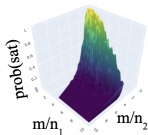
- If we are not careful, this problem is too easy:



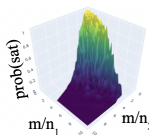
(a)



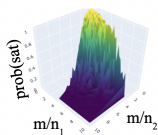
(b)



(c)



(d)



(e)

- We sample problems from the region of parameter space where the probability of satisfiability is ≈ 0.5 .

- Here are results showing that learning to reason with more complex fragments is harder than with easier fragments:

Fragment	T5-large	DeBERTa-v3-large
\mathcal{S}	88.3	89.0
\mathcal{W}	80.7	78.6
\mathcal{V}	79.5	77.2
\mathcal{Z}	75.1	75.2
\mathcal{A}	70.1	70.9

Source: Madusanka et al., Proc. ACM. 2024. pp. 15278 ff.

- Here are results we obtained when we trained an LM on ‘easy’ problems and tested on ‘harder’ problems:

Train l	Eval l	\mathcal{S}^+	\mathcal{R}	\mathcal{W}^-	\mathcal{W}	\mathcal{R}^+
$l \geq 6$ (<i>easy</i>)	satisfiable	61	87	90	53	58
	$l \leq 12$	98	99	99	70	98
	$l \geq 42$	84	74	70	42	70
$l \geq 22$ (<i>hard</i>)	satisfiable	57	89	86	50	52
	$l \leq 12$	68	61	33	61	72
	$l \geq 42$	100	94	99	96	73

Source: Schlegel et al., Proc. EMNLP. 2022. pp. 11184 ff.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.
- Until the time of Frege, logic consisted mainly of sporadic observations on entailments in natural language.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.
- Until the time of Frege, logic consisted mainly of sporadic observations on entailments in natural language.
- First-order logic gave us expanded expressive power, but with it, greater computational complexity.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.
- Until the time of Frege, logic consisted mainly of sporadic observations on entailments in natural language.
- First-order logic gave us expanded expressive power, but with it, greater computational complexity.
- A significant strand of research in first-order logic in recent years has concerned the satisfiability problem for fragments.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.
- Until the time of Frege, logic consisted mainly of sporadic observations on entailments in natural language.
- First-order logic gave us expanded expressive power, but with it, greater computational complexity.
- A significant strand of research in first-order logic in recent years has concerned the satisfiability problem for fragments.
- There is no reason these fragments cannot owe their salience to the syntax of natural language.

- I have traced the origins of logic as a tool for making sense of reasoning with natural languages.
- Until the time of Frege, logic consisted mainly of sporadic observations on entailments in natural language.
- First-order logic gave us expanded expressive power, but with it, greater computational complexity.
- A significant strand of research in first-order logic in recent years has concerned the satisfiability problem for fragments.
- There is no reason these fragments cannot owe their salience to the syntax of natural language.
- With the rise of transformer-based LMs, we can expect the focus to shift again, perhaps towards understanding the ability of these models to approximate logical reasoning for natural language fragments.