Computer Science as the continuation of Logic by other means

Georg Gottlob
Oxford University
& TU Vienna

Formal Logic

All Sciences classify statements as *true* or *false.*

Formal Logic establishes rules for deriving true statements from other true statements, and for refutation.

Logic is the mother of all sciences.

→Some formal logic should be taught in all sciences and disciplines.

Goethe´s Faust

STUDENT ASKS:
I want to be a true scholar,
I want to grasp, by the collar,
What´s on earth, in heaven above
In Science, and in Nature too.
…

ANSWER: ...
My dear friend, I´d advise in sum
First, the *<u>Collegium Logicum</u>.*
There your mind will be trained,
As if in Spanish boots, constrained,
So that painfully, as it ought,
It creeps along the way of thought,
Not flitting about all over,
Wandering here and there.
....

Goethe´s Faust

STUDENT ASKS:
I want to be a true scholar,
I want to grasp, by the collar,
What´s on earth, in heaven above
In Science, and in Nature too.
…

ANSWER: ...
My dear friend, I´d advise in sum
First, the *Collegium Logicum*.
There your mind will be trained,
As if in Spanish boots, constrained,
So that painfully, as it ought,
It creeps along the way of thought,
Not flitting about all over,
Wandering here and there.
....

Computer Science

While logical methods are omnipresent in all sciences, there is a unique "new" discipline that:

- is historically rooted in Logic
- uses predominantly logical methods
- continually poses logical problems and challenges to Formal Logic
- takes logic further

Computer Science

While logical methods are omnipresent in all sciences, there is a unique "new" discipline that:

- is historically rooted in Logic
- uses predominantly logical methods
- continually poses logical problems and challenges to Formal Logic
- takes logic further

Computer Science:

The continuation of Logic by other means.

Historic Roots of Computer Science

Leibniz:
Calculus Ratiocinator

Wittgenstein

Frege.

Tarski.

Post

Markov

Büchi

Rabin

Russell & Whitehead.

Gentzen.

K. Zuse
Z3 - 1941

Mc Carthy
AI –Lisp

Automata Theory 60ies

Hilbert:
H.Programme
Entscheidungs-problem

Gödel
1931

Turing
TM 1936

Von Neumann
v.N. Architecture
1945

Chomsky
Formal Lang. 1956

Complexity Theory 70es

Boole

Other computers

Karp, Cook
P/NP

Rosser.

Church
Lambda Calculus
Un/decidability results
Church's Thesis

Mauchly & Eckert
ENIAC -1946
EDVAC

Hoare
H.Logic, CSP

Babbage
Difference engine,
Analytical Engine
1823+

ALGOL
58, 60, 68

R. Péter
Prim Recursive F

Wirth
Pascal

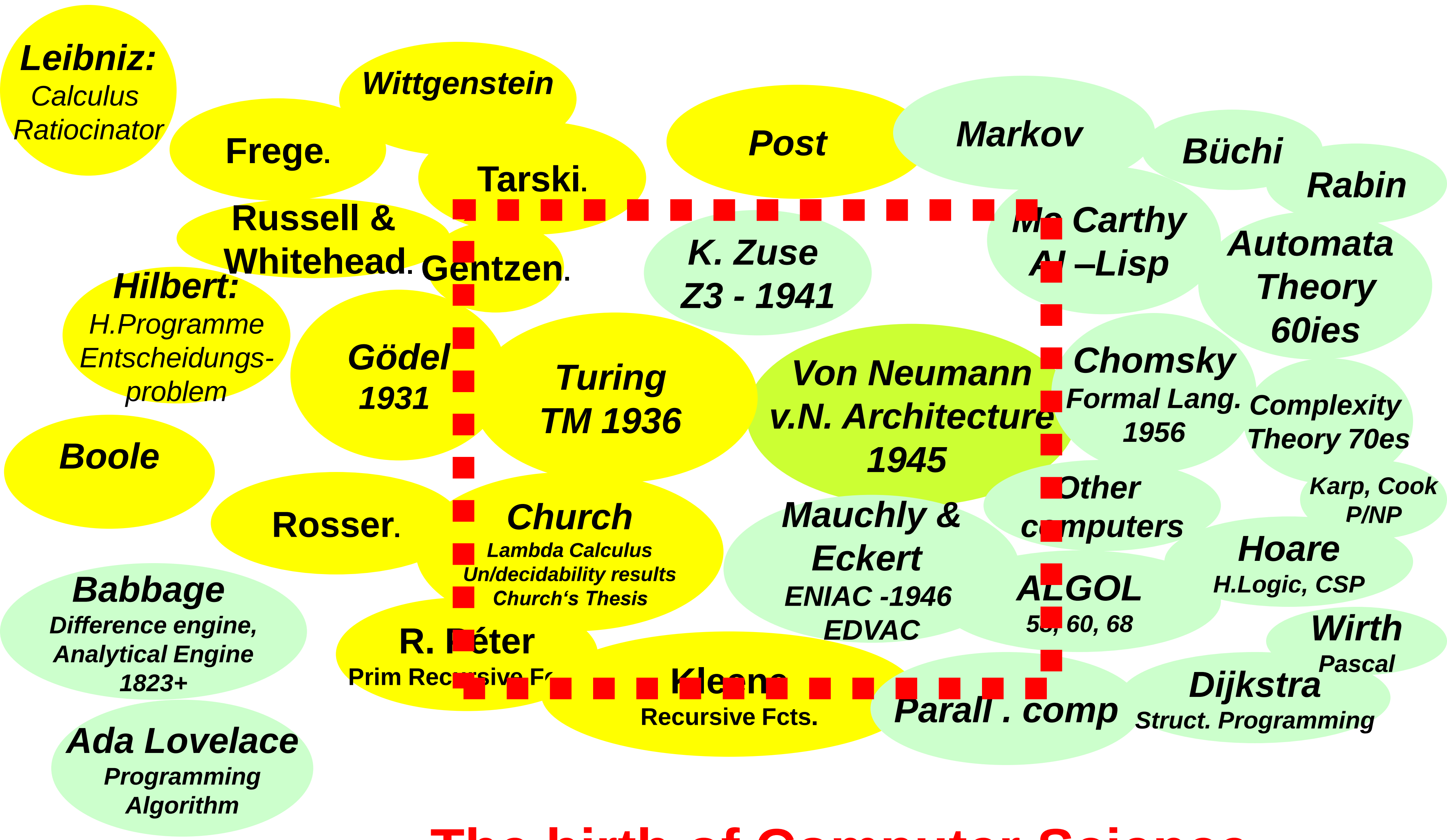
Kleene
Recursive Fcts.

Parall . comp

Dijkstra
Struct. Programming

Ada Lovelace
Programming Algorithm

Historic Roots of Computer Science



The birth of Computer Science

Paradigm & Method Shifts

Many methods of Logic carry over to Computer Science and are further developed and enriched in this discipline.

For example:

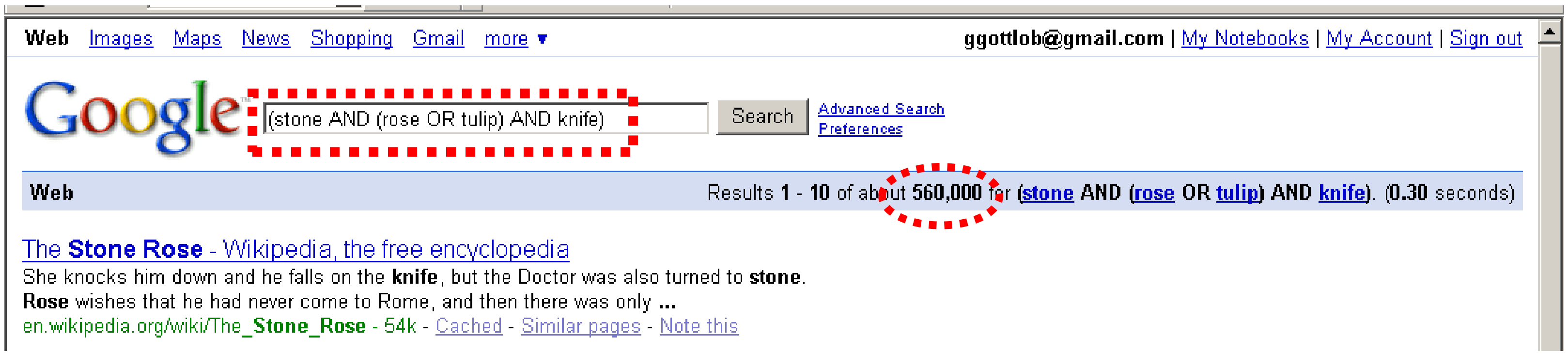
- Coding ·······► Data Compression
- Diagonalization ·······► Complexity Theory
- Formal syntax (wffs) ·······► Program BNF
- Formal semantics, etc.

But there are also several shifts that are used *in addition* to the original methods or paradigms.

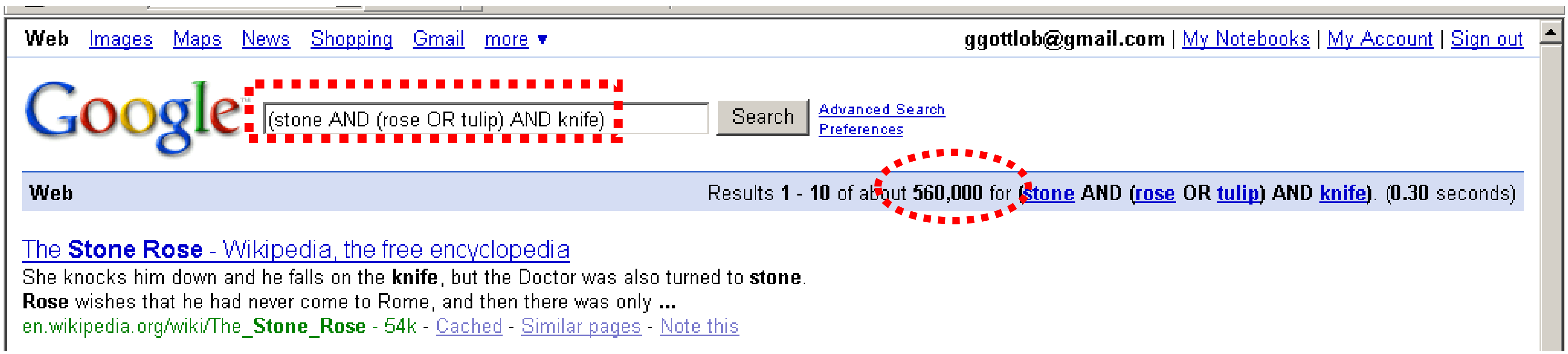
Paradigm & Method Shifts

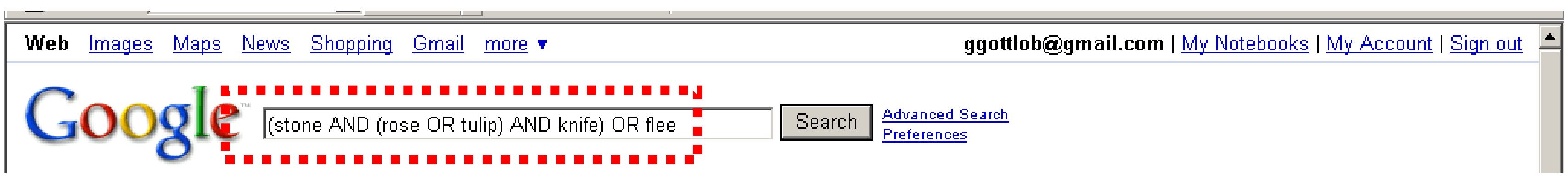
- Existence of solution → Solution algorithm
- Recursive → Efficiently computable in time & space
- Model → Finite Model such as: list, tree, array, database
- Satisfiability → Model Checking (Database query)
- Infinitary methods → Finite Combinatorics
- Sound & Complete → Satisfies the users

The logical flaws of Google:

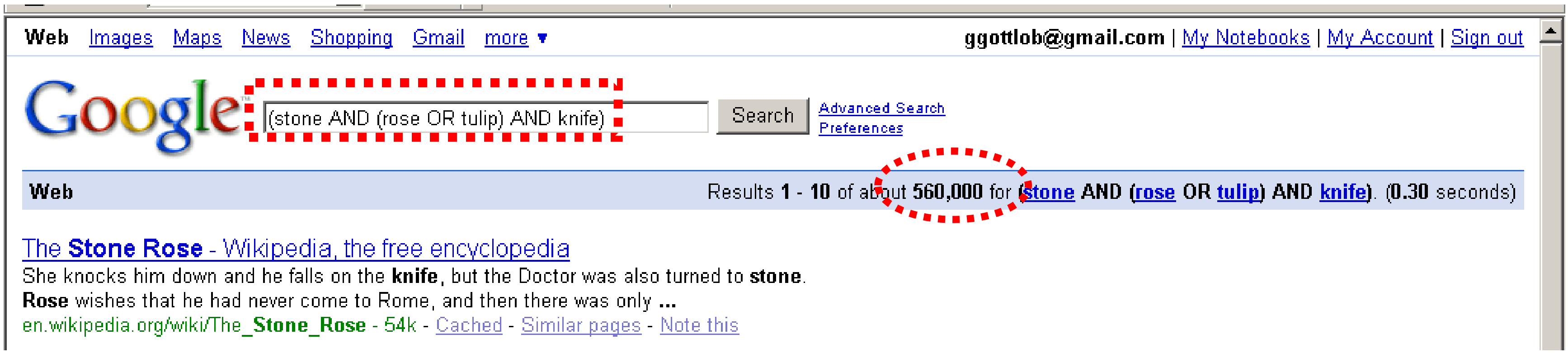


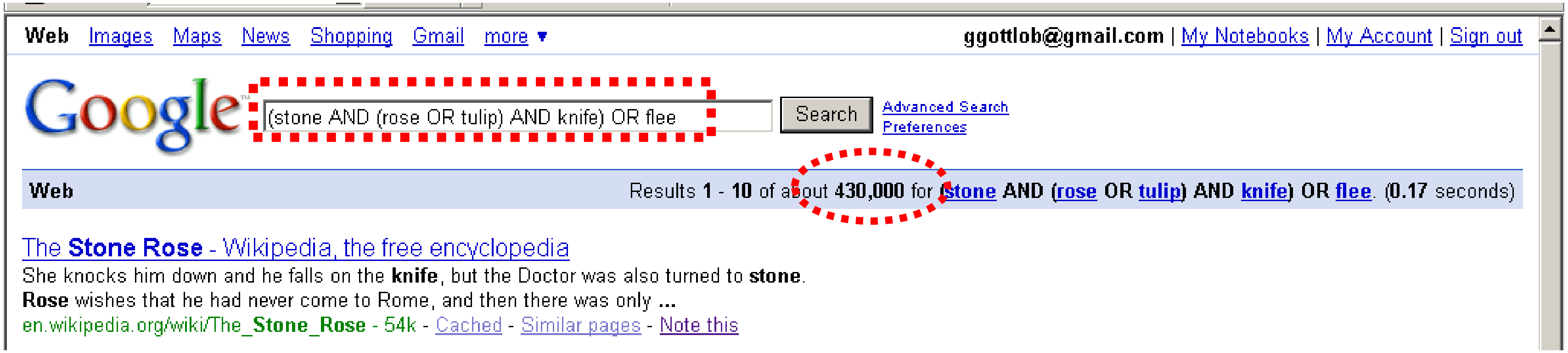
The logical flaws of Google:



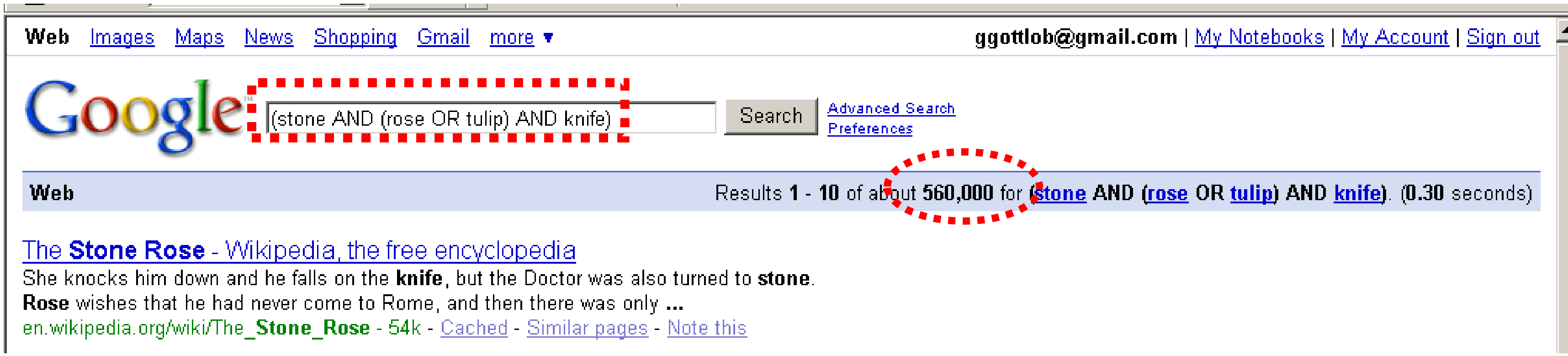


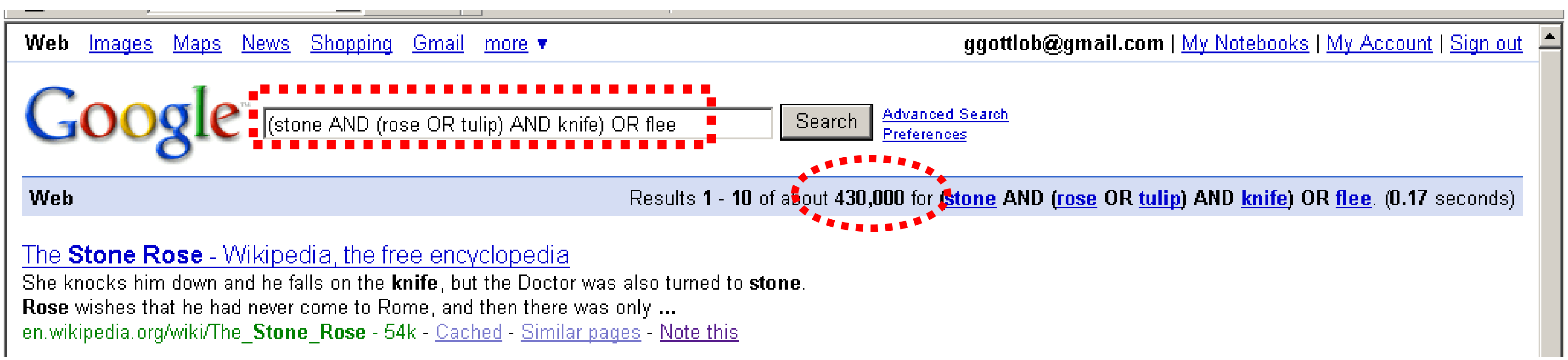
The logical flaws of Google:





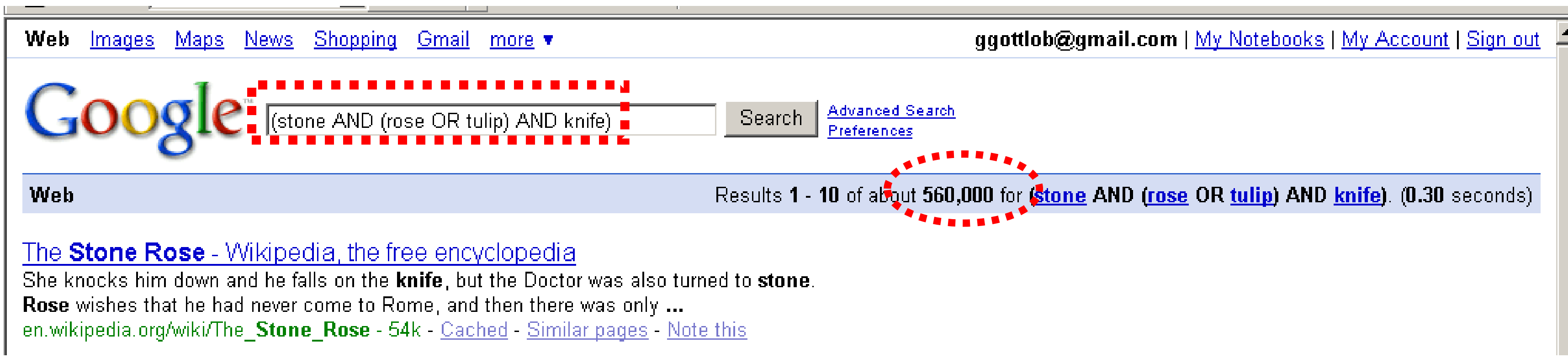
The logical flaws of Google:

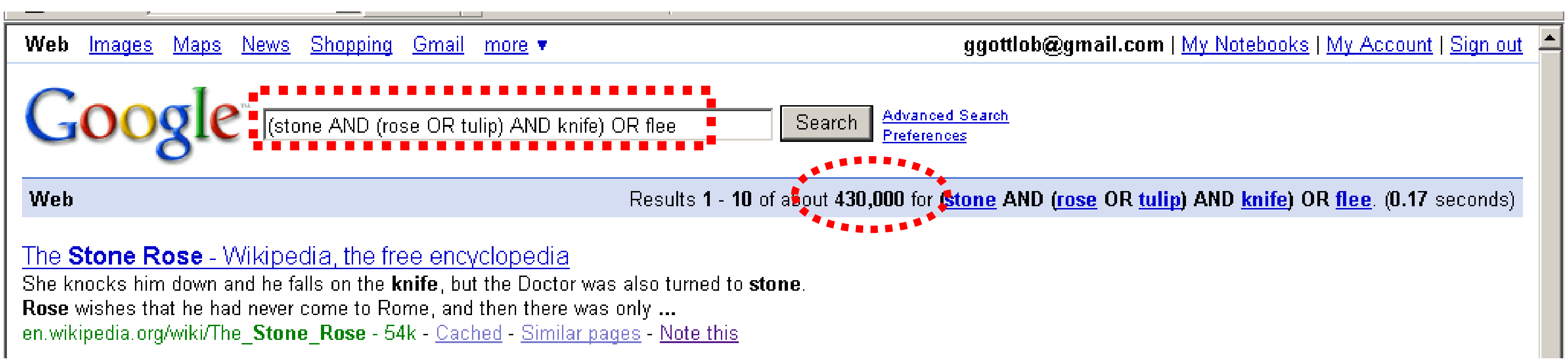




But Brin & Page still did a good job ...

The logical flaws of Google:





But Brin & Page still did a good job:
the first 30.000 or so results coincide

Continuation of sub-disciplines of Logic

Examples:

- Recursion theory (computability theory)
  - → Complexity theory
  - → Recursive program Analysis

- Proof theory
  - → Type checking & inference
  - → Automated theorem proving **resolution, cut elimination**
  - → Linear Logic, proof nets,...

- Model Theory
  - → Program semantics
    e.g. fixed-point based,
    logic programming,...
  - → Finite Model Theory -
    Descriptive Complexity
    Database Theory
- Propositional Logic
  - → Hardware
  - → Switching Theory

- Modal logic

Temporal logics for
computer aided verification
(system = Kripke structure)

Modal nonmonotonic logics,
epistemic logics in AI
(K and B operators)

....

- Set theory

? (Transfinite ordinals)

Two surveys

M. Davis 1988:

Influences of Mathematical Logics on Computer Science

In Herken Ed. *The Univesal Turing Machine:Half a Century Survey*, OUP

Halpern, Harper, Immerman, Kolaitis, Vardi, Vianu 2001:

On the Unusual Effectiveness of Logic in Computer Science

Bulletin of Symbolic Logic 7, 2001

Areas Mentioned by Davis and Vardi

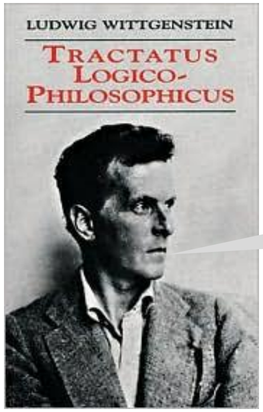
- Formal syntax
- Boolean Logic
- Programming languages & typing
- Logic programming
- Descriptive complexity
- Database query languages
- Reasoning about knowledge
- Automated verification

Areas Mentioned by Davis and Vardi

- Formal syntax
- Boolean Logic
- Programming languages & typing
- Logic programming
- Descriptive complexity
- Database query languages
- Reasoning about knowledge
- Automated verification

What about programming and software engineering?

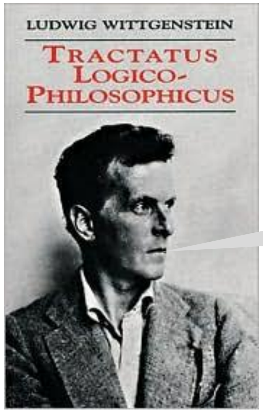
Programming & Software Engineering



The world is all that is the case.

A unique true model

Programming & Software Engineering

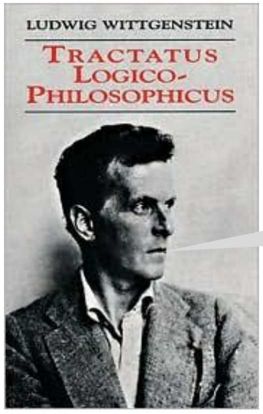


The world is all that is the case.

A unique <u>true</u> model

¬bridged(Strait-of-Messina)

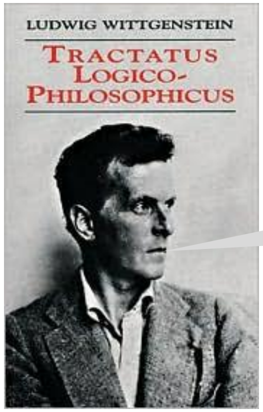
Programming & Software Engineering



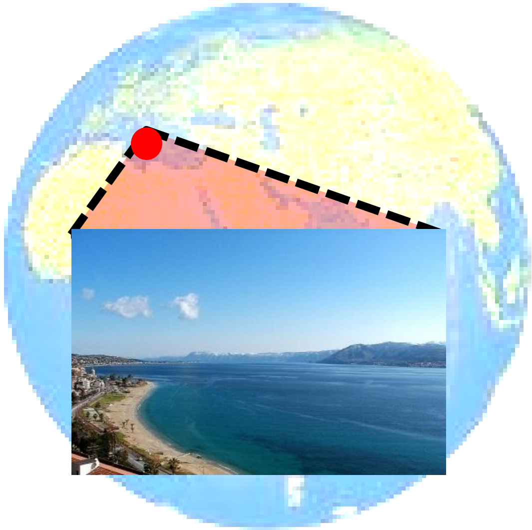
The world is all that is the case.

A unique <u>true</u> model

Programming & Software Engineering



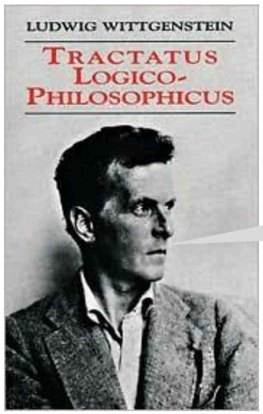
The world is all that is the case.



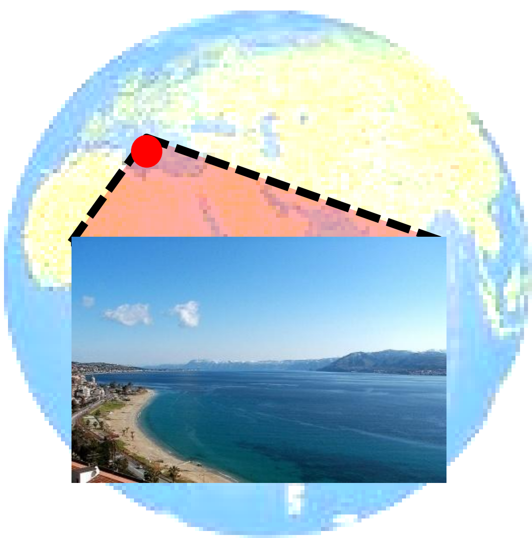
A unique <u>true</u> model

¬bridged(Strait-of-Messina)

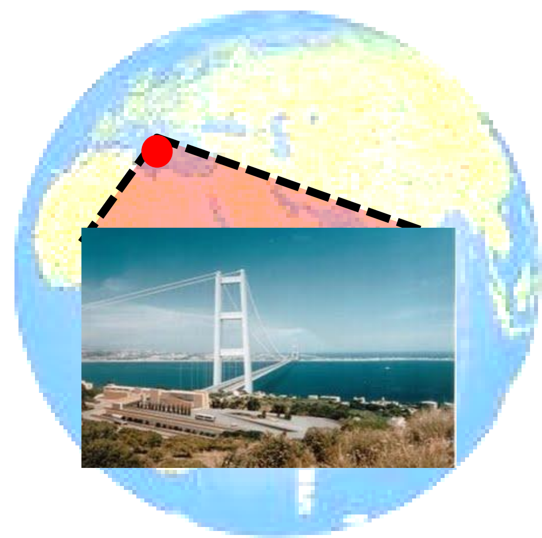
Programming & Software Engineering



The world is all that is the case.

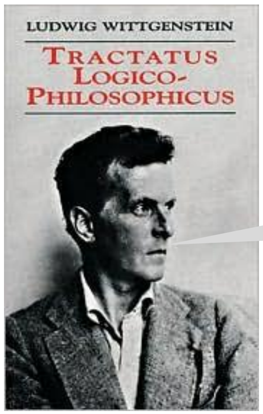


¬bridged(Strait-of-Messina)

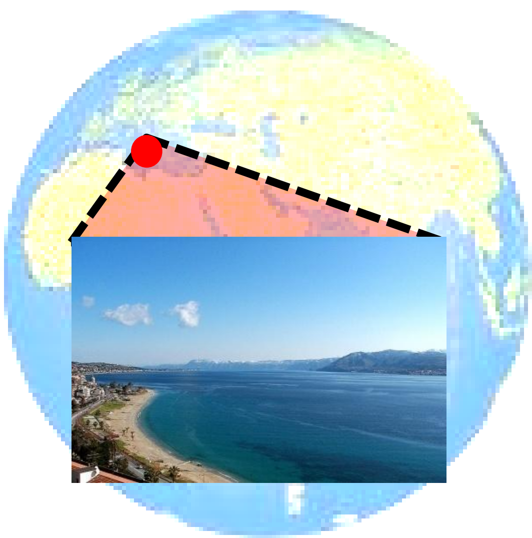


bridged(Strait-of-Messina)

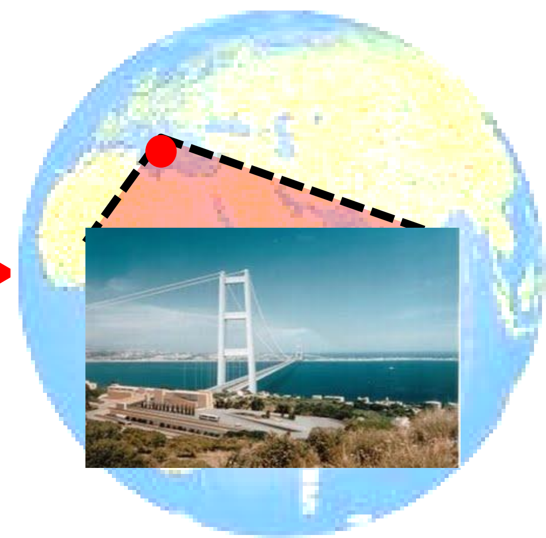
Programming & Software Engineering



The world is all that is the case.



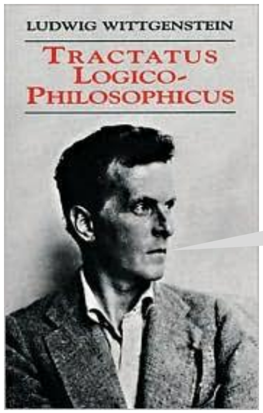
ENGINEERING



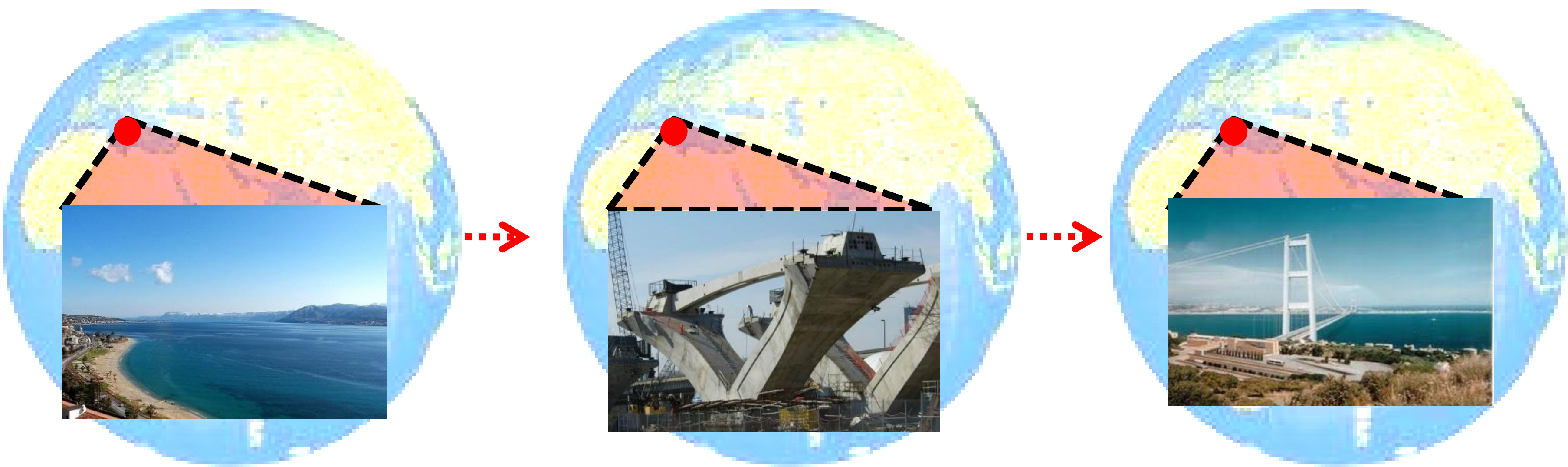
¬bridged(Strait-of-Messina)

bridged(Strait-of-Messina)

Programming & Software Engineering

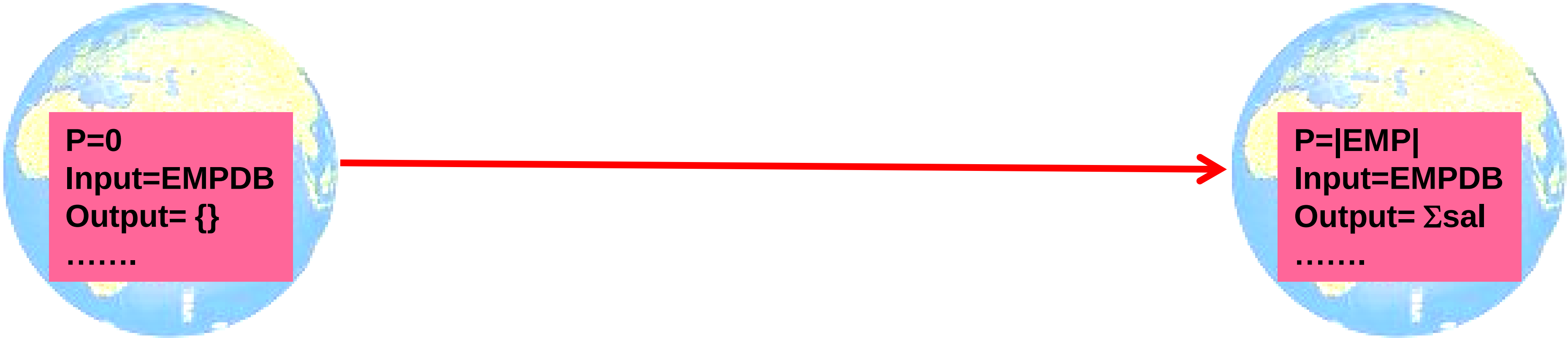


The world is all that is the case.

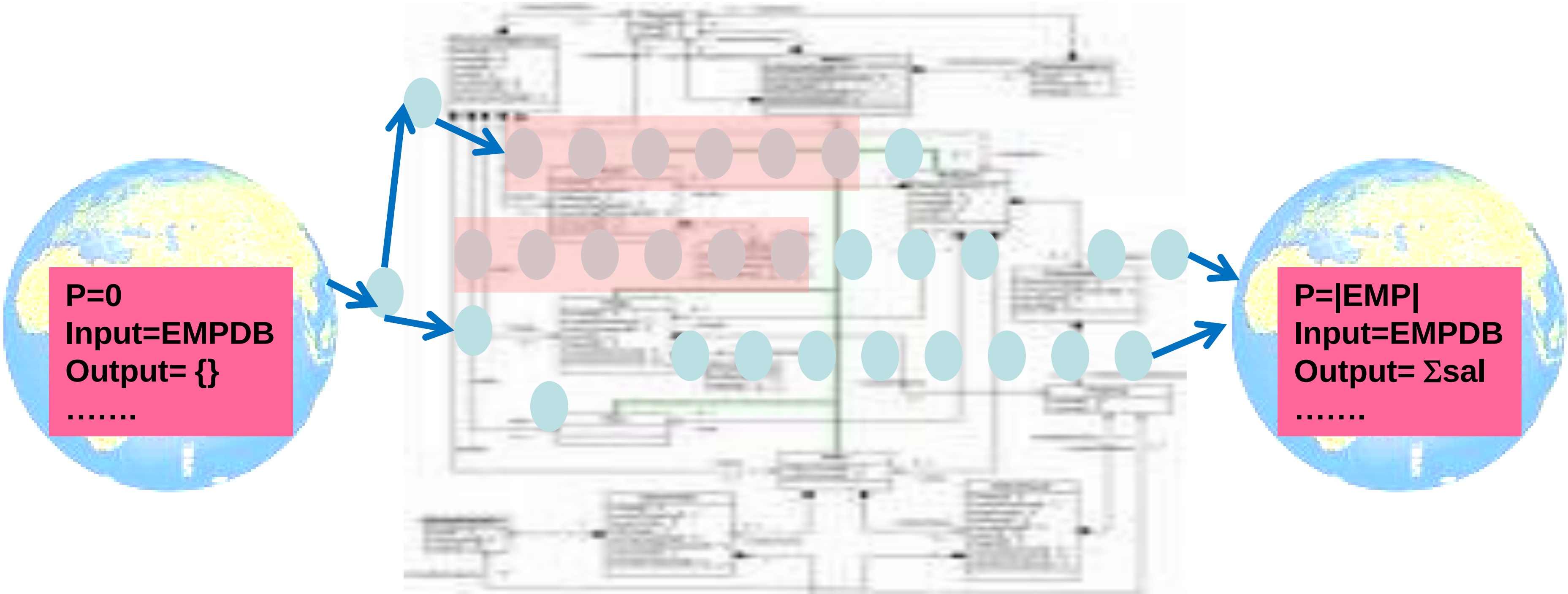


¬bridged(Strait-of-Messina) *bridged(Strait-of-Messina)*

Software Engineering

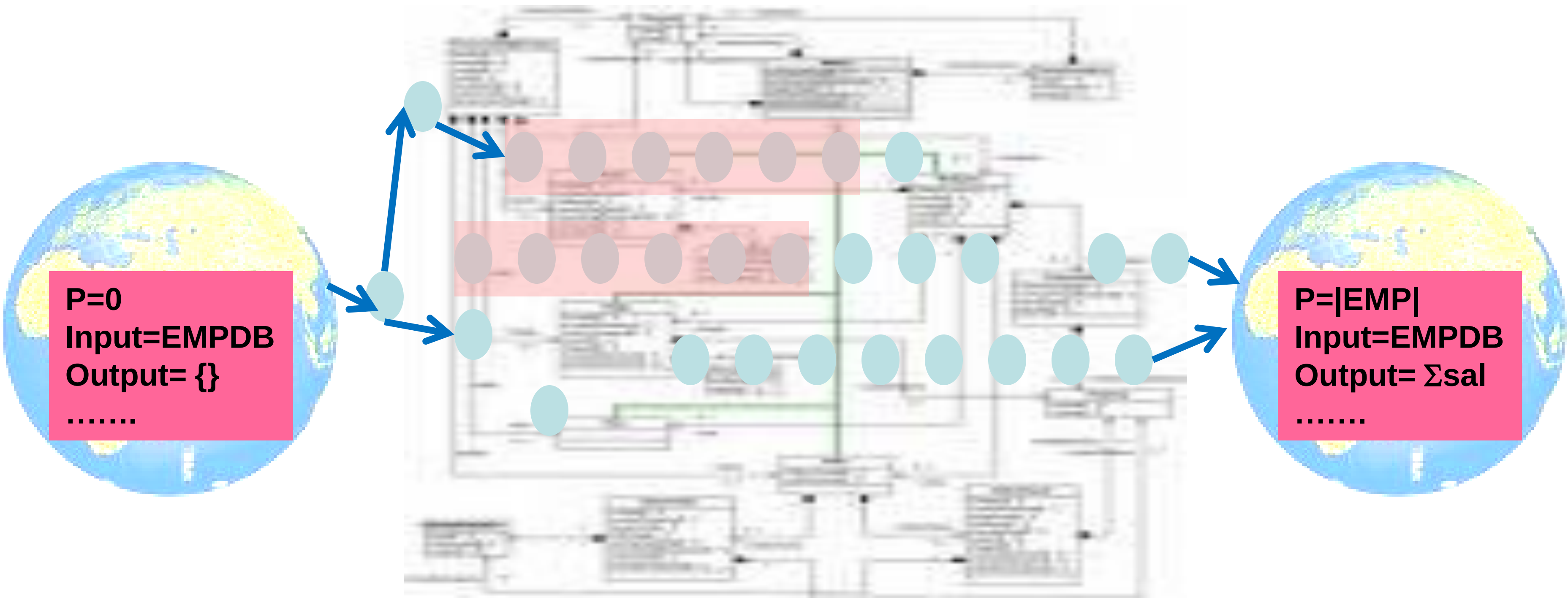


Software Engineering



The science of defining, implementing, testing & maintaining complex parameterised transitions between logical worlds (Wittgensteinian models).

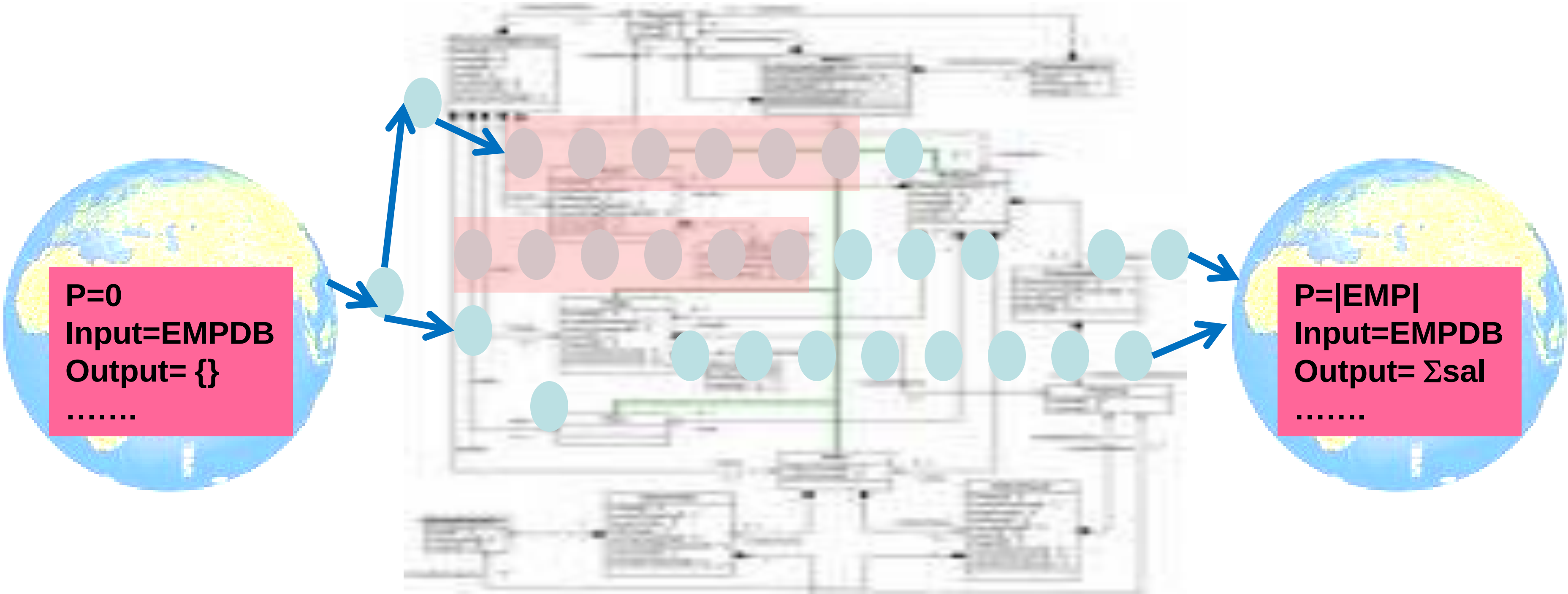
Software Engineering



The science of defining, implementing, testing & maintaining complex parameterised transitions between logical worlds (Wittgensteinian models).

SE takes logic further!

Software Engineering



The science of defining, implementing, testing & maintaining complex parameterised transitions between logical worlds (Wittgensteinian models).

software = logiciel

Rest of this Talk

Two examples of logic in computer science

- Logical aspects of NP vs P

- Logic and the Semantic Web

$$NP \stackrel{?}{=} P$$

The most important problem of Theoretical Computer Science

.... and arguably an extremely important problem of Applied CS

This problem has many logical facets

I will mention some.

Logical Aspects of $P \stackrel{?}{=} NP$

- Gödel's letter to von Neumann
- Cook-Levin-Karp Theorem
- Propositional proof systems / Frege systems
- Fagin's Theorem: NP = Existential SO
- Courcelle's theorem
- Recognizing tractable problems
- Probabilistically Checkable Proofs (PCP)
- Independence of P=NP

Gödel's 1956 letter to von Neumann

Princeton, 20./III. 1956

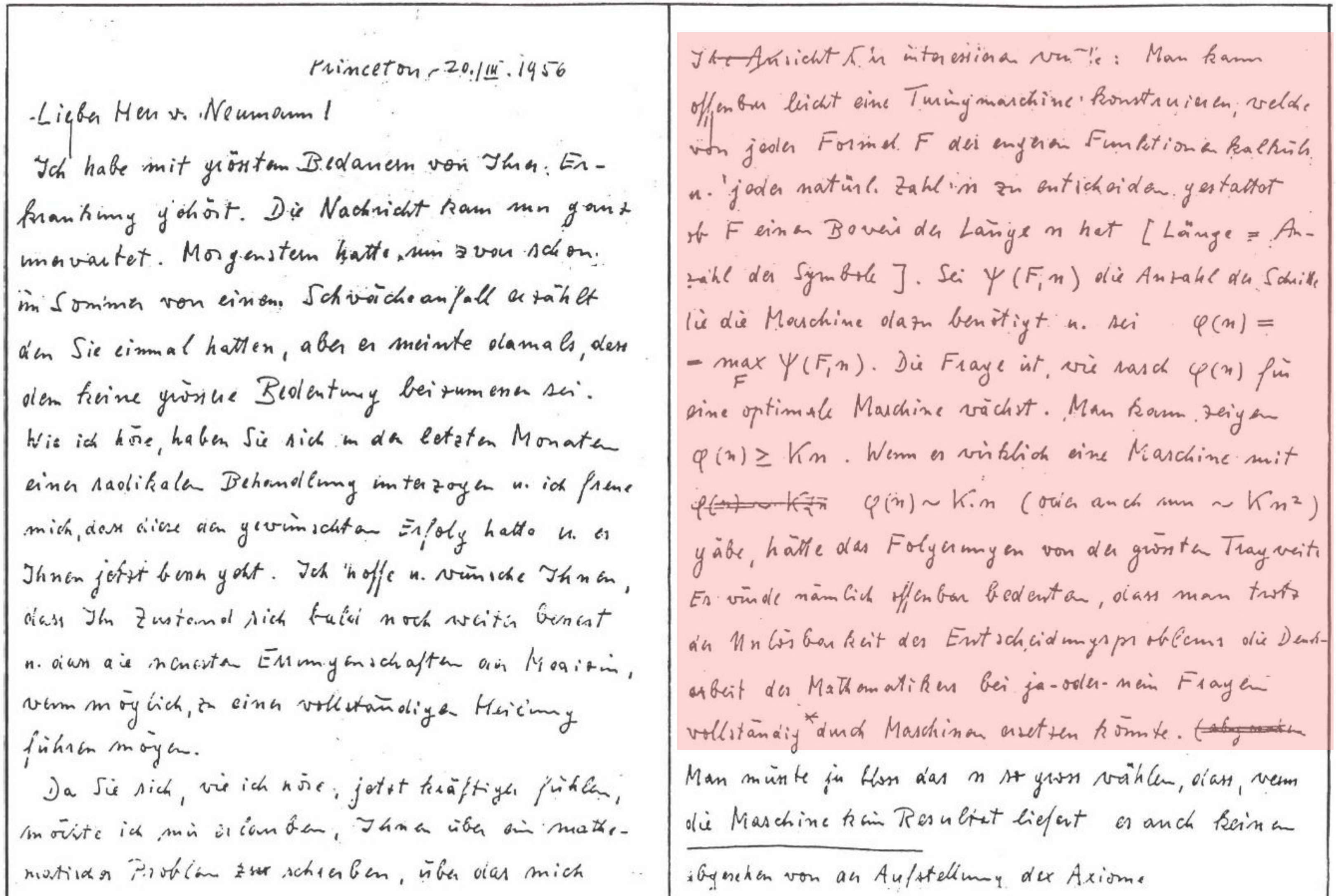
Lieber Herr v. Neumann!

Ich habe mit grösstem Bedauern von Ihrer Erkrankung gehört. Die Nachricht kam mir ganz unerwartet. Morgenstern hatte mir zwar schon im Sommer von einem Schwächeanfall erzählt, den Sie einmal hatten, aber er meinte damals, dass dem keine grössere Bedeutung beizumessen sei. Wie ich höre, haben Sie sich in den letzten Monaten einer radikalen Behandlung unterzogen u. ich freue mich, dass diese den gewünschten Erfolg hatte u. es Ihnen jetzt besser geht. Ich hoffe u. wünsche Ihnen, dass Ihr Zustand sich bald noch weiter bessert u. dass die neuesten Errungenschaften der Medizin, wenn möglich, zu einer vollständigen Heilung führen mögen.

Da Sie sich, wie ich höre, jetzt kräftiger fühlen, möchte ich mir erlauben, Ihnen über ein mathematisches Problem zu schreiben, über das mich Ihre Ansicht sehr interessieren würde: Man kann offenbar leicht eine Turingmaschine konstruieren, welche von jeder Formel F des engeren Funktionenkalküls u. jeder natürl. Zahl n zu entscheiden gestattet ob F einen Beweis der Länge n hat [Länge = Anzahl der Symbole]. Sei $\psi(F,n)$ die Anzahl der Schritte die die Maschine dazu benötigt u. sei $\varphi(n) = \max_F \psi(F,n)$. Die Frage ist, wie rasch $\varphi(n)$ für eine optimale Maschine wächst. Man kann zeigen $\varphi(n) \geq Kn$. Wenn es wirklich eine Maschine mit ~~$\varphi(n) \sim Kn^2$~~ $\varphi(n) \sim K.n$ (oder auch nur $\sim Kn^2$) gäbe, hätte das Folgerungen von der grössten Tragweite. Es würde nämlich offenbar bedeuten, dass man trotz der Unlösbarkeit des Entscheidungsproblems die Denkarbeit des Mathematikers bei ja-oder-nein Fragen vollständig* durch Maschinen ersetzen könnte. Man müsste ja bloss das n so gross wählen, dass, wenn die Maschine kein Resultat liefert es auch keinen

* abgesehen von der Aufstellung der Axiome

Gödel's 1956 letter to von Neumann



Princeton, 20./III. 1956

Lieber Herr v. Neumann!

Ich habe mit grösstem Bedauern von Ihrer Erkrankung gehört. Die Nachricht kam mir ganz unerwartet. Morgenstern hatte mir zwar schon im Sommer von einem Schwächeanfall erzählt, den Sie einmal hatten, aber er meinte damals, dass dem keine grössere Bedeutung beizumessen sei. Wie ich höre, haben Sie sich in den letzten Monaten einer radikalen Behandlung unterzogen u. ich freue mich, dass diese den gewünschten Erfolg hatte u. es Ihnen jetzt besser geht. Ich hoffe u. wünsche Ihnen, dass Ihr Zustand sich bald noch weiter bessert u. dass die neuesten Errungenschaften der Medizin, wenn möglich, zu einer vollständigen Heilung führen mögen.

Da Sie sich, wie ich höre, jetzt kräftiger fühlen, möchte ich mir erlauben, Ihnen über ein mathematisches Problem zu schreiben, über das mich

Ihre Ansicht sehr interessieren würde: Man kann offenbar leicht eine Turingmaschine konstruieren, welche von jeder Formel F des engeren Funktionenkalküls u. jeder natürl. Zahl n zu entscheiden gestattet, ob F einen Beweis der Länge n hat [Länge = Anzahl der Symbole]. Sei $\psi(F,n)$ die Anzahl der Schritte die die Maschine dazu benötigt u. sei $\varphi(n) = \max_F \psi(F,n)$. Die Frage ist, wie rasch $\varphi(n)$ für eine optimale Maschine wächst. Man kann zeigen $\varphi(n) \geq Kn$. Wenn es wirklich eine Maschine mit ~~$\varphi(n) \sim K.n$~~ $\varphi(n) \sim K.n$ (oder auch nur $\sim Kn^2$) gäbe, hätte das Folgerungen von der grössten Tragweite. Es würde nämlich offenbar bedeuten, dass man trotz der Unlösbarkeit des Entscheidungsproblems die Denkarbeit des Mathematikers bei ja-oder-nein Fragen vollständig* durch Maschinen ersetzen könnte. ~~abgesehen~~

Man müsste ja bloss das n so gross wählen, dass, wenn die Maschine kein Resultat liefert es auch keinen

*abgesehen von der Aufstellung der Axiome

Lieber Herr v. Neumann: [...]

I would like to allow myself to write you about a mathematical problem, of which your opinion would very much interest me:

One can obviously easily construct a Turing machine, which for every formula *F* in first order predicate logic and every natural number n, allows one to decide if there is a proof of F of length n (length = number of symbols). Let $\Psi(F,n)$ be the number of steps the machine requires for this and let $\varphi(n)=\max_F \Psi(F,n)$.

The question is how fast $\varphi(n)$ grows for an optimal machine.

One can show that $\varphi(n) \geq k.n$. If there really were a machine with $\varphi(n) \sim k.n$ (or even $\varphi(n) \sim k.n^2$), this would have consequences of the greatest importance.

NP=P

Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine.

[.....]

Cook´s Theorem (1971)

SAT is NP-complete

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

Cook´s Theorem (1971)

SAT is NP-complete

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

By Turing – Machine reduction

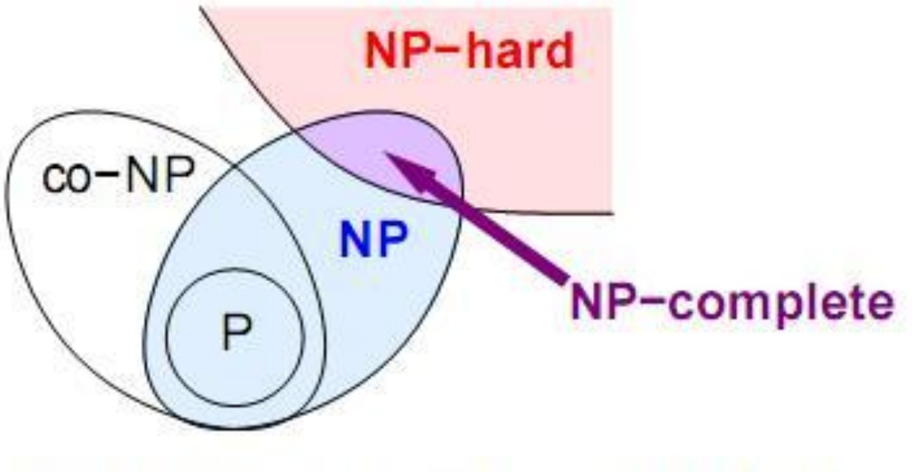
This reduction is <u>solidly grounded in logic.</u>
The idea of reducing TMs to logical formulae was already present in Turing´s 1937 paper...
Many such reductions were used by logicians in the 60es

Propositional proof systems

Are there short proofs for co-NP ?

⇔ Are there short proofs for propositional tautologies ?

If not, then NP ≠ co-NP and thus NP ≠ P



Propositional proof systems

Are there short proofs for co-NP ?

⇔ Are there short proofs for propositional tautologies ?

If not, then NP $\neq$ co-NP

For which proof systems can we show that there are only superpolynomially sized proofs?

Solved for some proof systems e.g. Resolution (Haken 84)

For Frege proof systems and many others this is still open.

Fagin´s Theorem

NP = ESO

A property of finite structures is decidable in NP if and only if it is expressible in existential second-order logic.

Fagin´s Theorem

NP = ESO

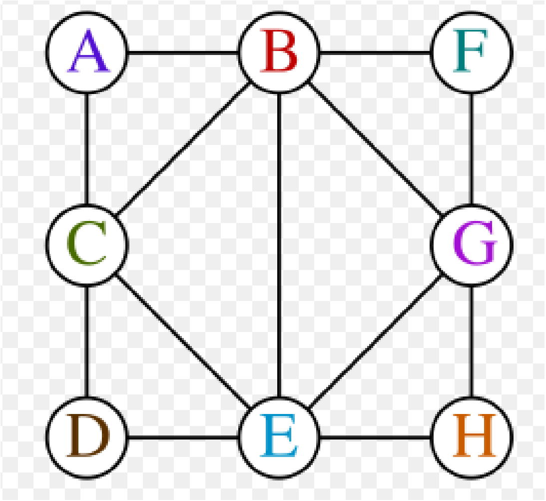
Example: Formulating Graph 3-colourability in Monadic ESO

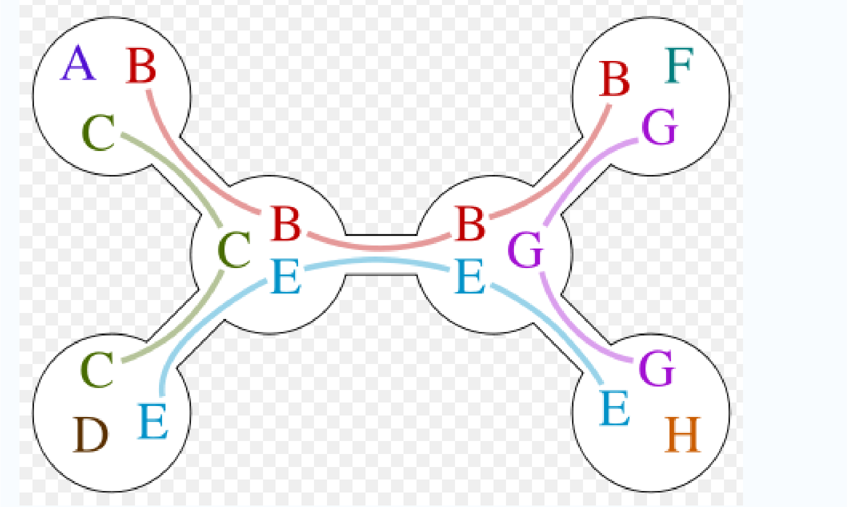
$$
\begin{array}{lll}
(\exists R, G, B) & [& (\forall x\, (R(x) \vee G(x) \vee B(x))) \\
& \wedge & (\forall x (R(x) \Rightarrow (\neg G(x) \wedge \neg B(x)))) \\
& \wedge & \ldots \\
& \wedge & \ldots \\
& \wedge & (\forall x, y (E(x,y) \Rightarrow (R(x) \Rightarrow (G(x) \vee B(y))))) \\
& \wedge & (\forall x, y (E(x,y) \Rightarrow (G(x) \Rightarrow (R(x) \vee B(y))))) \\
& \wedge & (\forall x, y (E(x,y) \Rightarrow (B(x) \Rightarrow (R(x) \vee G(y))))) \,]
\end{array}
$$

Restricting the formula and the structures

Courcelle's Theorem (1993)

All problems expressible in Monadic Second Order Logic are solvable in *linear time* on input structures of *bounded treewidth* .



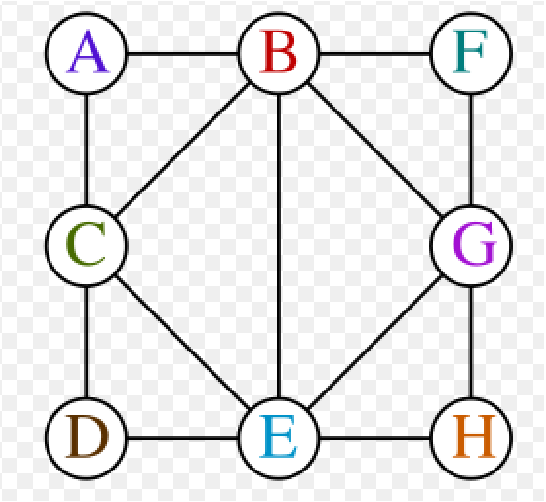


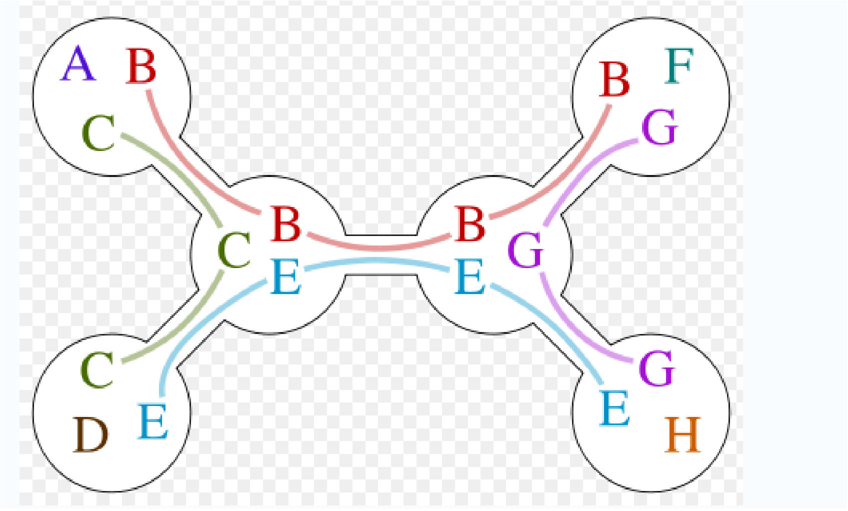
Treewidth 2

Restricting the formula and the structures

Courcelle's Theorem (1993)

All problems expressible in Monadic Second Order Logic are solvable in *linear time* on input structures of *bounded treewidth* .



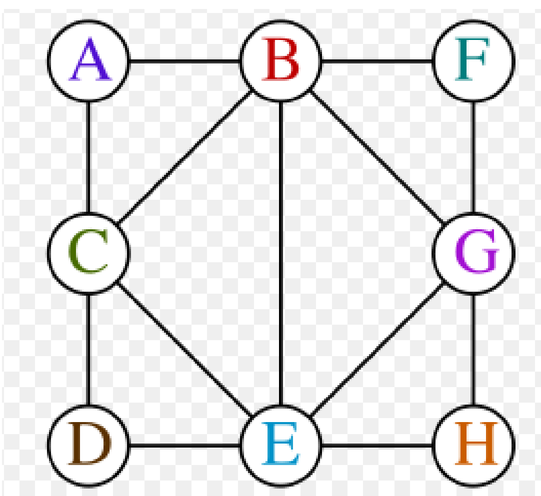


+ Interesting follow-up work by Courcelle , Makowsky, et al.

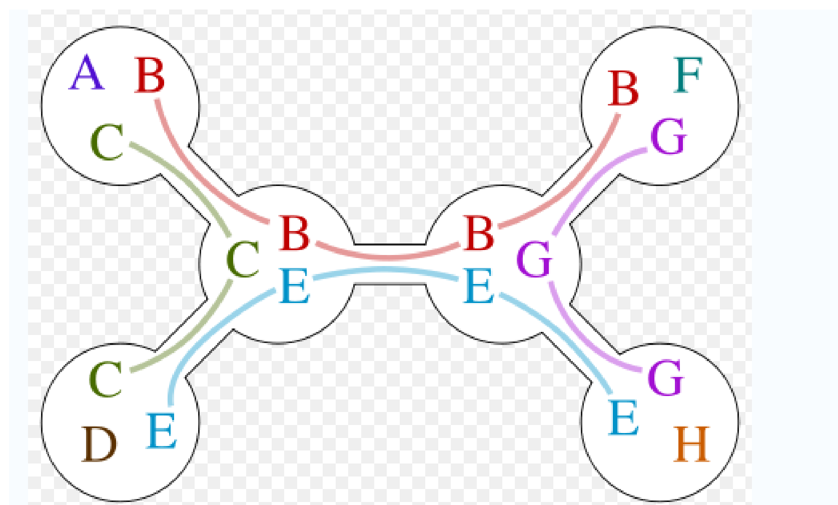
Restricting the formula and the structures

Courcelle's Theorem (1993)

All problems expressible in Monadic Second Order Logic are solvable in *linear time* on input structures of *bounded treewidth* .



Treewidth 2



Many applications.
E.g. Graph multicut problems [G., Lee, 2007]

Recognizing Tractable Problems

Logic can help!

What about Prefix classes?

Often one can immediately recognize that a problem belongs to a specific ESO prefix class.

A “simple” Facility Placement Problem

Every room should be equipped with a computer.

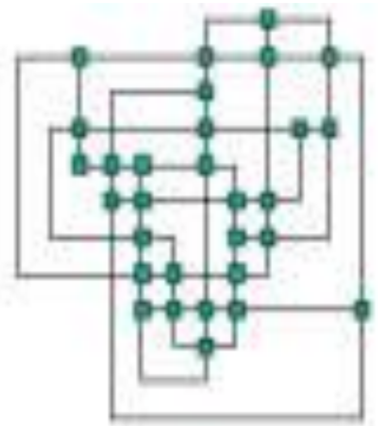
If a printer is not present in a room, then one should be available in an adjacent room.

No room with a printer should be a meeting room.

Every room is at most 5 rooms distant from a meeting room.

[…]

Given an office layout as a graph, decide whether the facility placement constraints are satisfiable.



$\exists P \exists M \dots \ \forall x \exists y\, ((P(x) \vee E(x,y)\ \&\ P(y))\ \&\ \dots$

Observe that this is an E_1^*ae formula

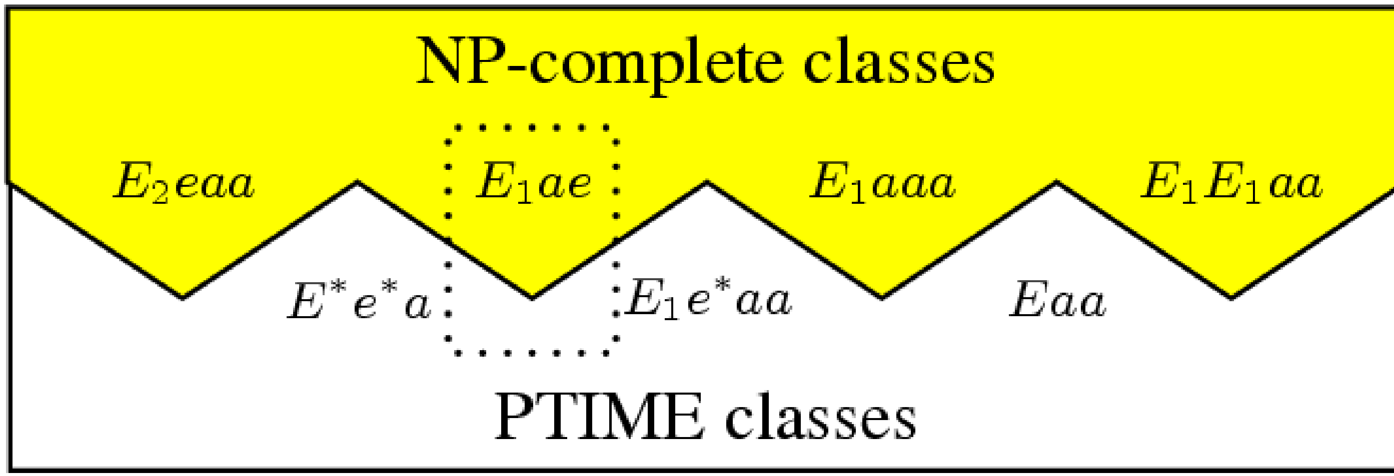
This leads to the questions:

Are formulas of the type E_1^*ae or even E^*ae polynomially verifiable over <u>graphs?</u>

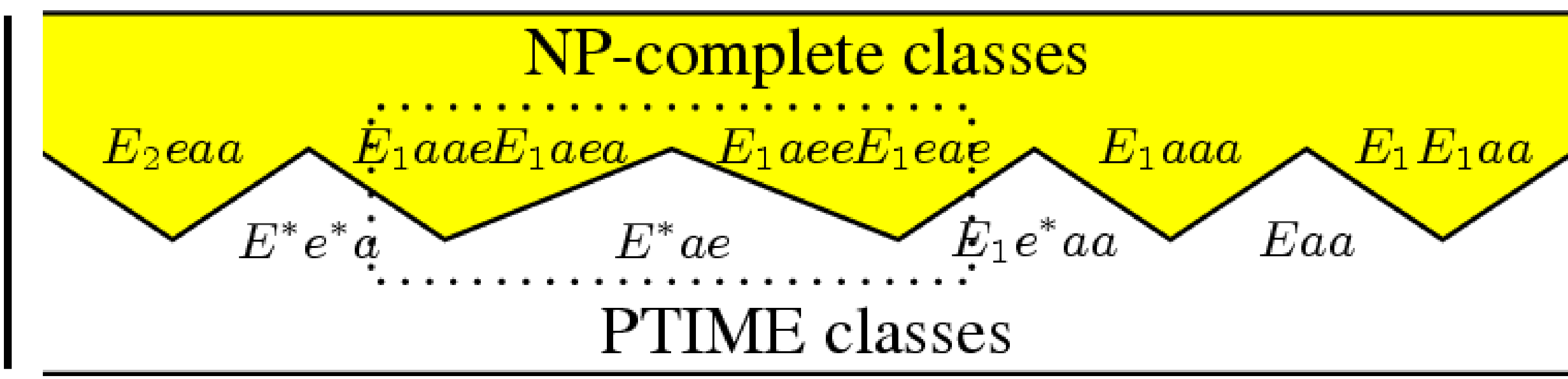
What about other fragments of ESO or SO?

Complexity characterization of ESO prefix classes [G.,Kolaitis, Schwentick 2000]

Directed graphs (or undirected graphs with self-loops):



Undirected graphs w/o self-loops:



The Most Difficult Result

E*ae : PTIME model checking on undirected graphs

This class expresses as special cases problems such as:

Given an undirected graph G,
Does G contain a cycle whose length is a multiple of k ?

Tractability had been an an open problem for many years.
Solved positively in 1988 by Carsten Thomassen.

The Most Difficult Result

E*ae : PTIME model checking on undirected graphs

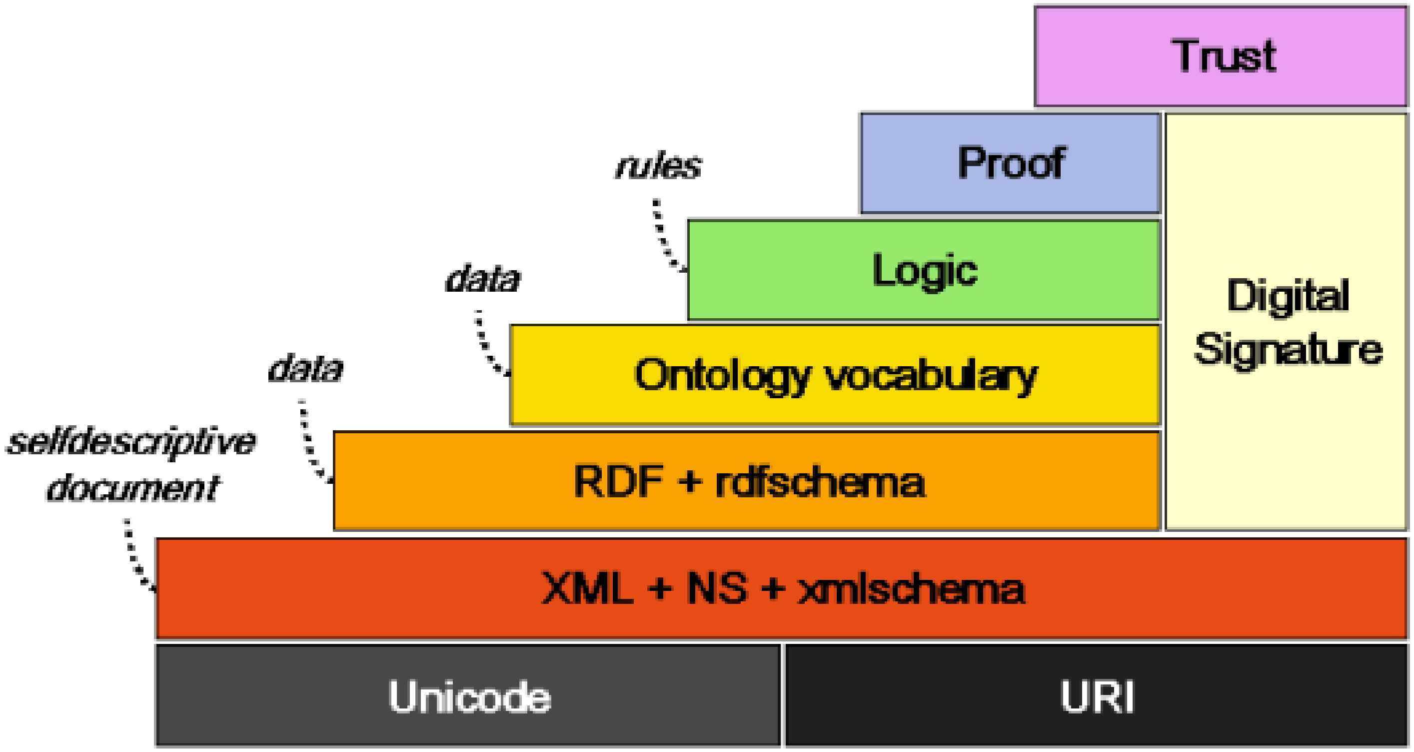
This class expresses as special cases problems such as:

Given an undirected graph G,
Does G contain a cycle whose length is a multiple of k ?

Tractability had been an an open problem for many years.
Solved positively in 1988 by Carsten Thomassen.

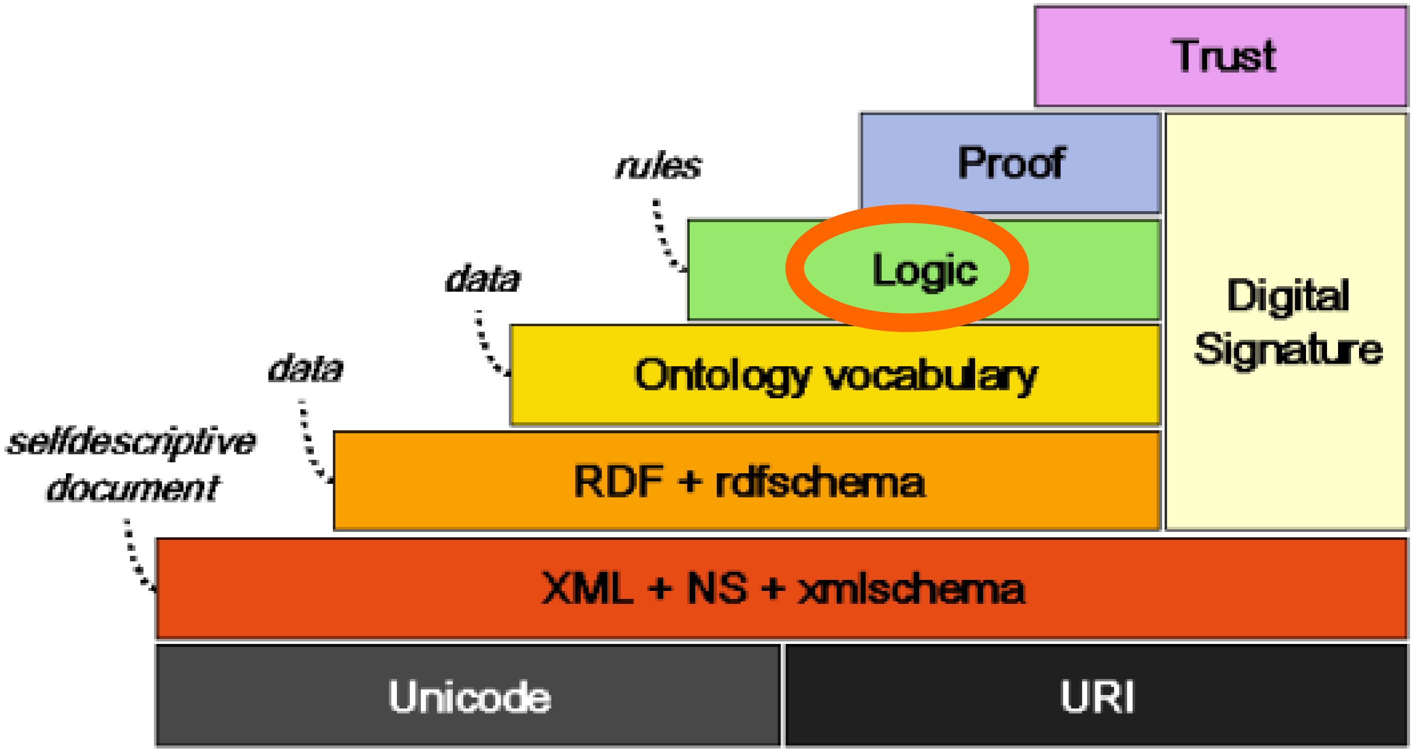
$$\exists P\, \exists Q\, \exists R\ \forall x\, \exists y\ \big(\ (P(x) \oplus Q(x) \oplus R(x)\,)\ \&$$
$$P(x) \rightarrow (E(x,y)\ \&\ Q(y))\ \&$$
$$Q(x) \rightarrow (E(x,y)\ \&\ R(y))\ \&$$
$$Q(x) \rightarrow (E(x,y)\ \&\ P(y)).$$

Logic & the Semantic Web



Tim Berners Lee et al.

Logic & the Semantic Web



Tim Berners Lee et al.

Each dog is an animal:

Predicate Logic: $\forall x.\ dog(x) \rightarrow animal(x)$

RDFS: (<onto2:dog> <rdfs:subClassOf> <onto2:animal>)

Simple Terminological inference

(<in:Snoopy> <rdf:type> <an:Dog>)

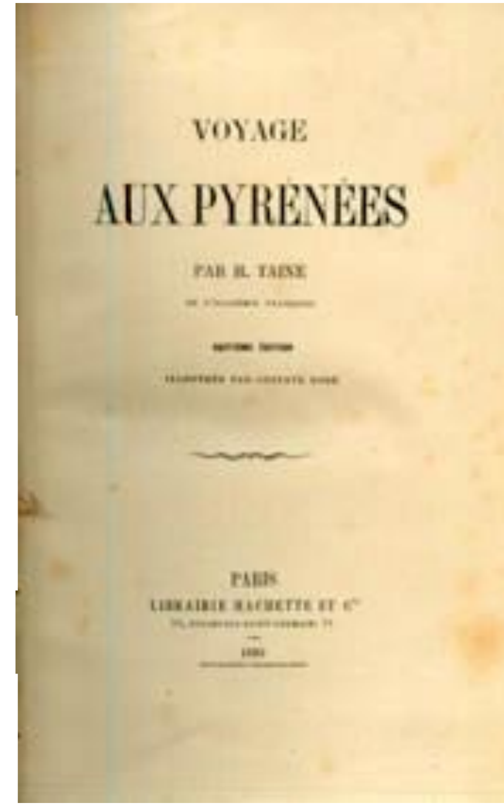
(<an:Dog> <rdfs:subClassOf> <an:Animal>)

(<in:Snoopy> <rdf:type> <an:Animal>)

Académie Française RDF Web knowledge base "af"

(<af:Taine> <rdf:type><prof:historian>)

(<af:Taine><bibl:authored><isbn-asin:B00135ND4G>),…



General ISBN-ASIN Book Catalogue"isbn-asin"

(<B00135ND4G > <ck:has-title><"Voyage aux Pyrenees">)

(<B00135ND4G > <ck: has-author><wp:Taine>)

(<B00135ND4G > <ck: is-about><ck:Pyrenees>), ………

Common Knowledge Ontology "ck"

(<ck> <ck:has-author><owl:inverse-of><ck:has-written>)

(<ck:Pyrenees> <rdfs:subclass-of> <ck:mountains>),…..

***Query:** Find titles of books about mountains written by a historian*

T : (N, has-title, T) & (A, has-written, N) &
(N, is-about, mountains) & (A, type, historian)

Google

french historian book on pyrenees | Search | Advanced Search | Preferences

Search: (•) the web ( ) pages from the UK

Web Results **1** - **10** of about **58,800** for french histo

Pyrenees holiday | Skiing, Gites, Hotels **Pyrenees**

Skiing Holidays in the **Pyrenees**, we are able to provide hotels and help with **...** 'hors categorie' climb making its way into the Tour de France **history books ...**
www.**epyrenees**.com/ - 22k - Cached - Similar pages

Pyrenees (France) Travel Writing & Other Literature from Stanfords ...

World's largest stock of Travel Writing & Other Literature for **Pyrenees** (France) including Food & Drink Travel Writing, **History Books**, Travel Writing and **...**
www.stanfords.co.uk/go/**pyrenees**-france/travel-writing-other-literature/ - 22k - Cached - Similar pages

Greatpyr.com |All the best Great **Pyrenees books** and Great **Pyrenees ...**

Greatpyr.com review: Complete **book** on Great **Pyrenees** focusing on its **history**, health, and orgins. Extensive information on the dogs from France that went **...**
www.greatpyr.com/store.php - 33k - Cached - Similar pages

Hiking With **History** in the **Pyrenees** - New York Times

Any comparison with the picturesque Alps beggars the rugged satisfactions of what the **French** call "la frontiere sauvage." The rustic **Pyrenees** are best **...**
query.nytimes.com/gst/fullpage.html?sec=travel&res=9C0CE5DE1439F933A15756C0A966958260 - 69k - Cached - Similar pages

Deduction Example

SIEMENS	has-client	BT
SIEMENS	has-client	ACME
BT	has-client	ACME
ACME	has-client	Michelin
BT	is	British
MICHELIN	is-not	British

RDF/OWL

Does SIEMENS have a British client that itself has a non-British client?

Deduction Example

SIEMENS	has-client	BT
SIEMENS	has-client	ACME
BT	has-client	ACME
ACME	has-client	Michelin
BT	is	British
MICHELIN	is-not	British

RDF/OWL

Does SIEMENS have a British client that itself has a non-British client?

ACME is British or not British – tertium non datur

Simple Terminological inference

Reasoning under ontologies is extremely complex.

Using the general formalism: Undecidable!

Using standard DLs: 2-EXPTIME complete, i.e., $O(2^{(2^{|KB|})})$

We need to find fragments that

- are enough expressive
- are tractable

This is what we and others currently working on.

Default Reasoning

If a positive property is not mentioned (or is not known to hold), should one infer "by default" that it doesn't hold?

Being British: Positive property

LIXO is not known to be British ≈> LIXO is not British

This is what we do all the time…

Nonmonotonic /Default Reasoning

This is incorrect according to classical logic… but often useful.

Research problem: Find the right logic for the Semantic Web

Conclusion

I HOPE I COULD CONVINCE YOU THAT

Computer Science is a continuation of Logic by other means.

Consequences →

Consequences for CS Curricula

- Do not eliminate logic courses from CS curricula

In practice: Logic courses are like jewels:
the last thing you buy, the first thing you sell

- Include practical examples in "Logic for CS" courses

- Add more fun to logic courses courses

Liftboy: Up or down?

Liftboy: Up or down?
Logician: Yes.

A Logic Lesson in Latin

E falso quodlibet

A Logic Lesson in Latin

E falso quodlibet

Verum ex quodlibet

A Logic Lesson in Latin

E falso quodlibet

Verum ex quodlibet

Simplex semper sigillum veri

A Logic Lesson in Latin

E falso quodlibet

Verum ex quodlibet

Simplex semper sigillum veri

In vino veritas