The 30 Years War

- reduction in German population 15–30%
- in some terrirories 3/4 of the population died
- male population reduced by almost half
- population of Czech lands reduced by 1/3

Gottfried Wilhelm Leibniz (1646-1716)

Leibniz
Logic in Practice
Satisfiability
ILP

Leibniz's dream:

"a general method in which all truths of the reason would be reduced to a kind of calculation. At the same time this would be a sort of universal language or script, but infinitely different from all those projected hitherto; for the symbols and even the words in it would direct reason; and errors, except those of fact, would be mere mistakes in calculation."

If controversies were to arise, "there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, and say to each other: Let us calculate."
Dissertio de Arte Combinatoria, 1666

Leibniz
Logic in Practice
Satisfiability
ILP

1 handout: slides
730W journal entries were due

Leibniz
Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL
Satisfiability
ILP

Logic in Practice

Alfred Horn (1951)

Leibniz

Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL

Satisfiability

ILP

$x \wedge y \rightarrow z$

Alfred Horn (1951)

Leibniz
Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL
Satisfiability
ILP

$x \wedge y \rightarrow z \equiv \neg x \vee \neg y \vee z$

at most one positive literal (eaxctly one = 'definite clause')

```
Cat(x) :- Furry(x), Meows(x).
Cat(y) :- Feline(y).
Furry(A).
Meows(A).
?  Cat(z).
```

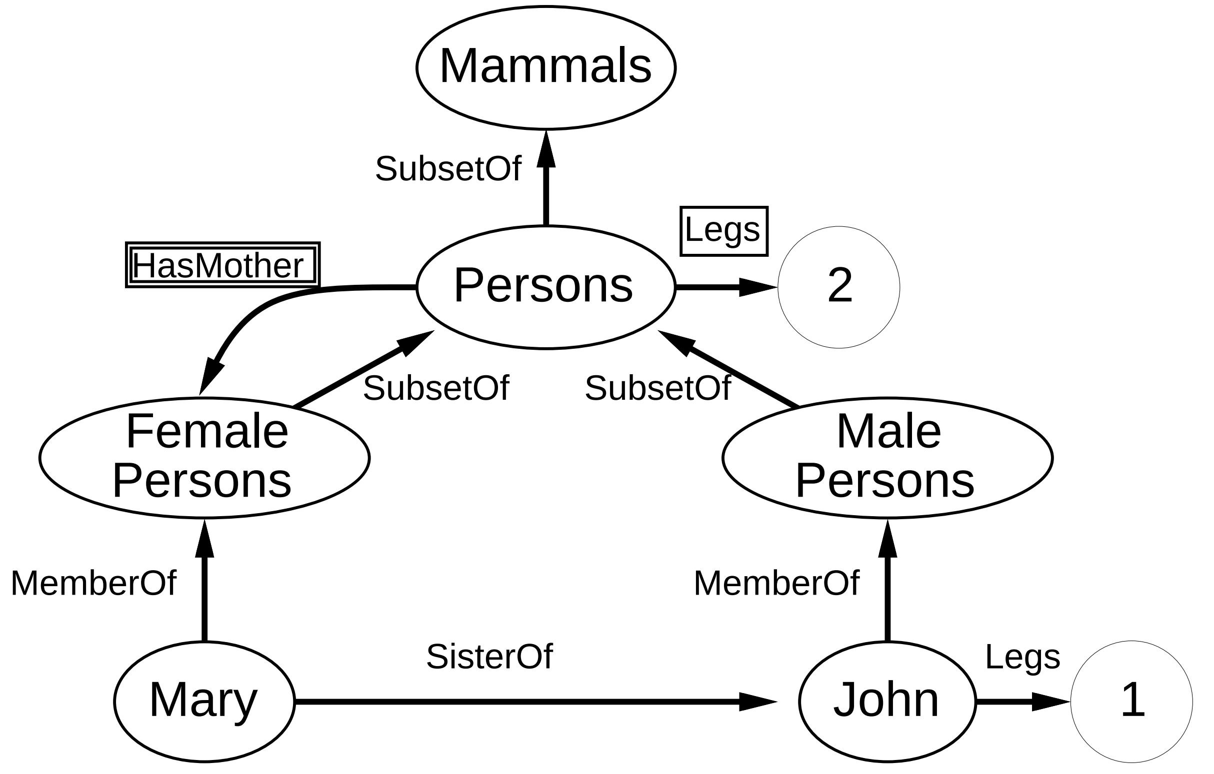
Still undecidable in first-order case.

Propositional: Unit resolution (Modus Ponens) is sound and complete in linear time for Horn theories: 'forward chaining'. Each rule 'fires' at most once, each variable 'processed' at most once

'expert systems'

Semantic Networks

Leibniz
Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL
Satisfiability
ILP



Semantic Networks

Leibniz

Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL

Satisfiability

ILP

Multiple aspects:

- A visual notation
- A restricted logic
- A set of implementation tricks

Typically:

- Efficient indexing
- Precomputation
- Methods for defaults or typicality

Aka: frames, inheritence networks, semantic graphs, description logics, terminological logics, ontologies

Description Logic

Leibniz

Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL

Satisfiability

ILP

computing categories and membership including:

1. subsumption
2. classification
3. inheritance

missing:

1. negation
2. disjunction
3. nested functions
4. existentials
5. intractability

Example DL

Leibniz

Logic in Practice
- Horn Clauses
- Semantic Nets
- Description Logic
- Example DL

Satisfiability

ILP

1. concepts (primitive and derived), instances
2. roles (definitional) and properties (assertional)
3. subsumption: *subsumes (x, y)* iff
   (a) x is a concept, and
   (b) same primitive concept ancestor, and
   (c) for each role of x with restriction r_x
      i. y has same role with restriction r_y, and
      ii. r_x subsumes r_y

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

Satisfiability

Terminology

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

Model of P: an interpretation in which P is true

Satisfiable: $\exists$ a model

Entailment: if Q is true in every model of P, then $P \models Q$

Valid: true in any interpretation

Boolean Satisfiability

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT

ILP

Given a formula of boolean logic, is there any assignment of T/F to its variables that makes the entire formula true?

$$(a \vee b \vee c) \wedge (\neg a \vee b \vee \neg c) \wedge (\neg a \vee \neg b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$$

Davis-Putnam-Logemann-Loveland

Leibniz

Logic in Practice

Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT

ILP

Davis-Putnam-Logemann-Loveland

Leibniz

Logic in Practice

Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT

ILP

boolean logic is a CSP: clause $\rightarrow$ nogood

Davis-Putnam-Logemann-Loveland

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

boolean logic is a CSP: clause $\rightarrow$ nogood

DPLL(α):
if α has no clauses, return true
if α has an empty clause, return false
if α contains a unit clause, return DPLL(Simplify(α, *literal*))
$v \leftarrow$ choose a variable in α
if DPLL(Simplify(α, v)) is true, return true
else, return DPLL(Simplify(α, $\neg v$))

Simplify(α, *literal*):
remove clauses in α where *literal* is positive
remove $\neg literal$ from clauses where it appears
return new $alpha$

'unit propagation', model-finding

Break

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

- asst 2
- exam 1
- office hours
- final projects

Local Search for SAT

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

1. Start with a random solution
2. Repeatedly flip variable to satisfy the most clauses
   (a) If same as previous, try second-best.
3. When tired, restart

(Selman and Kautz (GSAT, WalkSAT), *. . .)*

Local Search for SAT

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

1. Start with a random solution
2. Repeatedly flip variable to satisfy the most clauses
   (a) If same as previous, try second-best.
3. When tired, restart

(Selman and Kautz (GSAT, WalkSAT), *. . .)*

DPLL: 50 vars = 1.4 secs, 100 vars = 2.8 min, 140 vars = 4.7 hrs

Local Search for SAT

Leibniz
Logic in Practice
Satisfiability
- Terminology
- SAT
- DPLL
- Break
- GSAT
ILP

1. Start with a random solution
2. Repeatedly flip variable to satisfy the most clauses
   (a) If same as previous, try second-best.
3. When tired, restart

(Selman and Kautz (GSAT, WalkSAT), *...)*

DPLL: 50 vars = 1.4 secs, 100 vars = 2.8 min, 140 vars = 4.7 hrs

GSAT: 100 vars = 6 secs, 140 vars = 14 secs, 500 vars = 1.6 hrs

Leibniz

Logic in Practice

Satisfiability

ILP

- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

Inductive Logic Programming

Learning

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

Three types:

Supervised: classification (= prediction of class)
Unsupervised: compression (= prediction of actual value)
Reinforcement: sequence of decisions with occasional reward

Each can be on-line (incremental) or off-line (batch).

Terminology:

1. Hypothesis space
2. Training data (vs test data, for off-line case)
3. Performance metric (often on validation data)

Inductive Logic Programming

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

Given: ground facts and background definitions
Find: short (almost Horn) clauses that cover positive examples and not negative ones

$$Background \land Hypothesis \land Descriptions \models Classifications$$

Input

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

Descriptions:

Father(Philip, Charles)	*Father(Philip, Anne)*
Mother(Mum, Margaret)	*Mother(Mum, Elizabeth)*
Married(Diana, Charles)	*Married(Elizabeth, Philip)*
Male(Philip)	*Male(Charles)*
Female(Beatrice)	*Female(Margaret)*

Classifications:

Grandparent(Mum, Charles)	*Grandparent(Elizabeth, Beatrice)*
$\neg$*Grandparent(Mum,Harry)*	$\neg$*Grandparent(Spencer,Peter)*

Background: $Parent(x,y) \leftrightarrow Mother(x,y) \vee Father(x,y)$

Target: $Grandparent(x,y) \leftrightarrow \exists\, z\ Parent(x,z) \wedge Parent(z,y)$

FOIL

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

Given: ground facts and background definitions
Find: short (almost Horn) clauses that cover positive examples and not negative ones

Sequential covering ('FOIL')
rules ← { }
Until no remaining positives (or good enough):
    new ← empty rule
    While false positives (eg, covers any negatives):
        Add best single literal precondition
    Add *new* to *rules*
    Remove positive examples covered by *new*

Example

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

$\rightarrow$ *Grandfather(x,y)*

Example

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

$\rightarrow$ *Grandfather(x,y)*

Father(x,y) $\rightarrow$ *Grandfather(x,y)* (always wrong)
Parent(x,y) $\rightarrow$ *Grandfather(x,y)* (many false $+$)
Father(x,z) $\rightarrow$ *Grandfather(x,y)* (selected)

Example

Leibniz

Logic in Practice

Satisfiability

ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

$\rightarrow$ *Grandfather(x,y)*

Father(x,y) $\rightarrow$ *Grandfather(x,y)* (always wrong)
Parent(x,y) $\rightarrow$ *Grandfather(x,y)* (many false +)
Father(x,z) $\rightarrow$ *Grandfather(x,y)* (selected)

Father(x,z) $\wedge$ *Parent(z,y)* $\rightarrow$ *Grandfather(x,y)* (target)

Specializing

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

New literals:

1. Any predicate over any variables, where at least one of the variables is in previous literal or head
2. Equal(x, y), where x and y are already in rule
3. Negation of any of the above

Best: maximizes 'information gain'

Clause must be shorter than positives it explains (cf Ockham's razor).

ILP Applications

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

1. Mutagenesis
2. Toxicity
3. Rules of chess
4. Protein structure
5. Parsers

EOLQs

Leibniz
Logic in Practice
Satisfiability
ILP
- Learning
- ILP
- Input
- FOIL
- Example
- Specializing
- ILP Applications
- EOLQs

- What question didn't you get to ask today?
- What's still confusing?
- What would you like to hear more about?

Please write down your most pressing question about AI and put it in the box on your way out.

Thanks!