15-780: Graduate AI
Knowledge Representation Logic

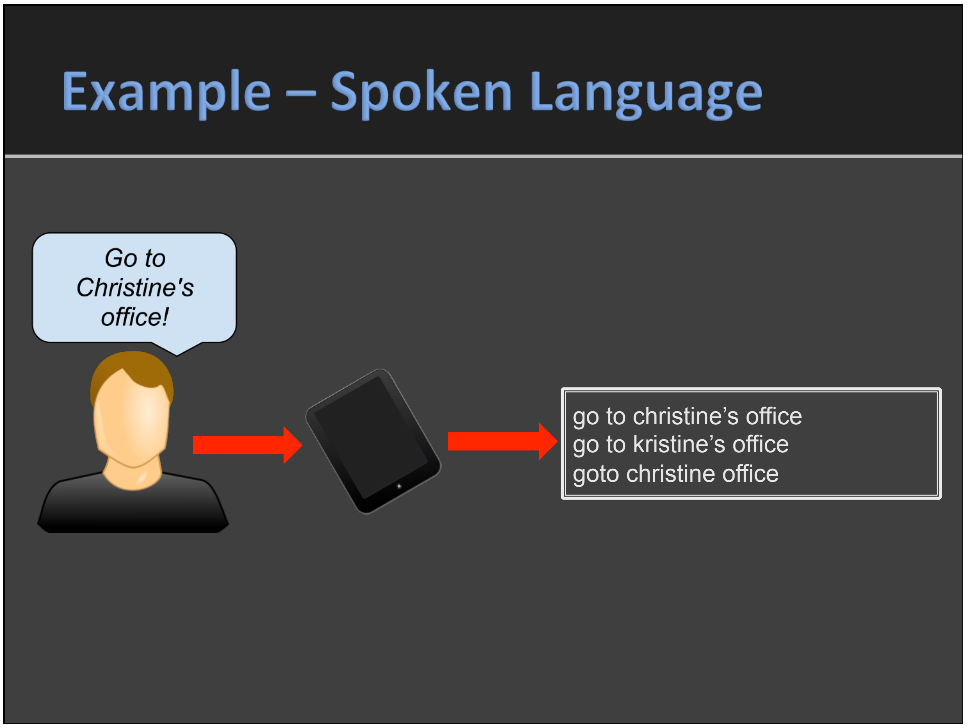
Manuela Veloso (this lecture)
Tuomas Sandholm
Tas: John Dickerson and Prateek Tandon

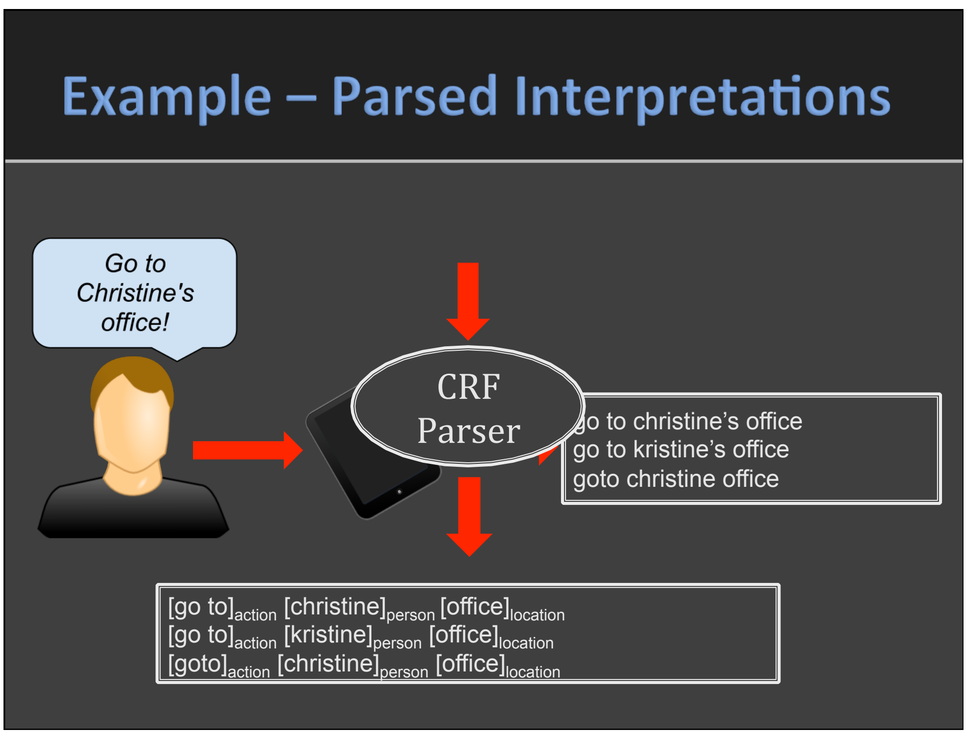
(Thanks to Geoff Gordon)
Chapters 7 and 9, Russell & Norvig

http://www.cs.cmu.edu/~15780
Carnegie Mellon University

CoBot – Collaborative Robot

- Tasks
  - Go to a location
  - Deliver a message
  - Escort a visitor
  - Transport object between locations
  - Semi-autonomous telepresence
  - Visitor companion
  - Deliver mail
  - Tour guide


Example – Spoken Language
Go to Christine's office!
go to christine's office
go to kristine's office
goto christine office


Example – Parsed Interpretations
Go to Christine's office!
CRF Parser
go to christine's office
go to kristine's office
goto christine office
[go to]$_{action}$ [christine]$_{person}$ [office]$_{location}$
[go to]$_{action}$ [kristine]$_{person}$ [office]$_{location}$
[goto]$_{action}$ [christine]$_{person}$ [office]$_{location}$

Example – Action Grounding

In order to correctly ground the parsed sentence we use a Knowledge Base.

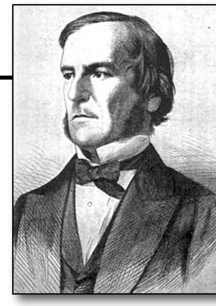
actionGroundsTo("go to", GoToLocation);5
actionGroundsTo("goto", GoToLocation);2

[go to]$_{action}$ [christine]$_{person}$ [office]$_{location}$
[go to]$_{action}$ [kristine]$_{person}$ [office]$_{location}$
[goto]$_{action}$ [christine]$_{person}$ [office]$_{location}$

Logic for Knowledge Representation

- Want to be able to relate facts:
  - Manuela is a faculty
  - All faculty have an office
  - Consequence: Manuela has an office.
- Reasoning about consequences of given, learned knowledge by agent
- Also compact representation for search
- … and, logical inference is a special case of probabilistic inference (we will see later)

Propositional logic



George Boole
1815–1864

- Constants: T or F
- Variables: x, y (values T or F)
- Connectives: ∧, ∨, ¬
  - Can get by w/ just NAND
  - Sometimes also add others:
    - ⊕, ⇒, ⇔, …

Propositional logic

- Build up expressions like ¬ x ⇒ y
- Precedence: ¬, ∧, ∨, ⇒
- Terminology: variable or constant with or w/o negation = **literal**
- Whole thing = **formula** or **sentence**

But what does it mean?

- A formula defines a mapping
  - (assignment to variables) $\mapsto$ {T, F}
- Assignment to variables = **model**
- **Truth Table**
  - **Model:**
    - row of truth table
  - **Value of formula in model:**
    - result column of truth table

Truth tables

x	y	$x \wedge y$
T	T	T
T	F	F
F	T	F
F	F	F

x	y	$x \vee y$
T	T	T
T	F	T
F	T	T
F	F	F

Truth table for implication

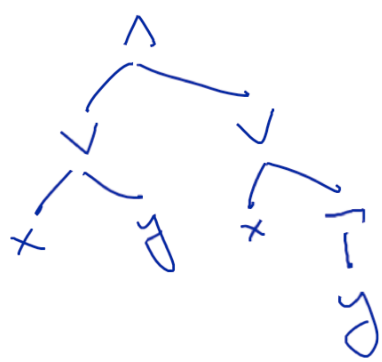
- Not a OR b

a	b	$a \Rightarrow b$
T	T	T
T	F	F
F	T	T
F	F	T

Complex formulas

- To evaluate a bigger formula
  - $(x \vee y) \wedge (x \vee \neg y)$ when $x = F$, $y = F$

- Build a parse tree
- Fill in variables at leaves using model
- Work upwards using truth tables for connectives
  - Linear time in size of parse tree



Models and sentences

- How many models make a sentence true?
  - Sentence is **satisfiable** if true in *some* model (famous NP-complete problem)
    - "Satisfying model" – one that makes the formula true
  - If not satisfiable, it is a **contradiction** (false in every model)
  - A sentence is **valid** if it is true in every model (called a **tautology**)

Variables in Models

- How is the variable X set in {some, all} satisfying models?
- Relevant and frequent question for an agent: given my assumptions, can I conclude X? Can I rule X out?
- SAT and CSP (next classes) can answer above questions
  - More examples later
    - 3-coloring, sudoku, planning, etc…

Truth tables get big fast

x	y	z	a	$(x \vee y \vee a) \Rightarrow z$
T	T	T	T	
T	T	F	T	
T	F	T	T	
T	F	F	T	
F	T	T	T	
F	T	F	T	
F	F	T	T	
F	F	F	T	
T	T	T	F	
T	T	F	F	
T	F	T	F	
T	F	F	F	
F	T	T	F	
F	T	F	F	
F	F	T	F	
F	F	F	F	

Definitions

- Two sentences are **equivalent**, A ≡ B, if they have same truth value in every model
  - (rains ⇒ pours) ≡ (←rains ∨ pours)
  - reflexive, transitive, symmetric
- **Simplifying** = transforming a formula into a simpler, equivalent formula

Transformation rules

$$
\begin{aligned}
(\alpha \wedge \beta) &\equiv (\beta \wedge \alpha) && \text{commutativity of } \wedge \\
(\alpha \vee \beta) &\equiv (\beta \vee \alpha) && \text{commutativity of } \vee \\
((\alpha \wedge \beta) \wedge \gamma) &\equiv (\alpha \wedge (\beta \wedge \gamma)) && \text{associativity of } \wedge \\
((\alpha \vee \beta) \vee \gamma) &\equiv (\alpha \vee (\beta \vee \gamma)) && \text{associativity of } \vee \\
\neg(\neg\alpha) &\equiv \alpha && \text{double-negation elimination}
\end{aligned}
$$

$$
\begin{aligned}
(\alpha \wedge (\beta \vee \gamma)) &\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma)) && \text{distributivity of } \wedge \text{ over } \vee \\
(\alpha \vee (\beta \wedge \gamma)) &\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma)) && \text{distributivity of } \vee \text{ over } \wedge
\end{aligned}
$$

α, β, γ are arbitrary formulas

More rules

$$
\begin{aligned}
(\alpha \Rightarrow \beta) &\equiv (\neg\beta \Rightarrow \neg\alpha) && \text{contraposition} \\
(\alpha \Rightarrow \beta) &\equiv (\neg\alpha \vee \beta) && \text{implication elimination} \\
(\alpha \Leftrightarrow \beta) &\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)) && \text{biconditional elimination} \\
\neg(\alpha \wedge \beta) &\equiv (\neg\alpha \vee \neg\beta) && \text{de Morgan} \\
\neg(\alpha \vee \beta) &\equiv (\neg\alpha \wedge \neg\beta) && \text{de Morgan}
\end{aligned}
$$

α, β are arbitrary formulas

Still more rules…

- … can be derived from truth tables
- For example: ¬
  - (a ∨ ←a) ≡ True
  - (True ∨ a) ≡ True (T elim)
  - (False ∧ a) ≡ False (F elim)

Example

- *Given*
  $(a \vee \neg b) \wedge (a \vee \neg c) \wedge (\neg(b \vee c) \vee \neg a)$
- Reverse distribution
  $\underline{(a \vee (\neg b \wedge \neg c))} \wedge (\neg(b \vee c) \vee \neg a)$
- De Morgan
  $(a \vee (\neg b \wedge \neg c)) \wedge (\underline{(\neg b \wedge \neg c)} \vee \neg a)$
- Commutativity
  $(a \vee (\neg b \wedge \neg c)) \wedge \underline{(\neg a \vee (\neg b \wedge \neg c))}$
- Reverse distribution
  $(a \wedge \neg a) \vee (\neg b \wedge \neg c)$
- Contradiction
  $F \vee (\neg b \wedge \neg c)$
- F elimination
  $(\neg b \wedge \neg c)$

Normal forms

- A normal form is a standard way of writing a formula
- E.g., **conjunctive normal form** (CNF)
  - conjunction of disjunctions of literals
  - (x ∨ y ∨ ←z) ∧ (x ∨ ←y) ∧ (z)
  - Each disjunct called a **clause**
- Any formula can be transformed into CNF w/o changing meaning

CNF cont'd

happy(John) ∧
(←happy(Bill) ∨ happy(Sue)) ∧
man(Socrates) ∧
(←man(Socrates) ∨ mortal(Socrates))

- Knowledge base (KB) often in CNF
- Can add new clauses as we find them out
- Each clause in KB is separately true (if KB is)

Another normal form: DNF

- DNF = disjunctive normal form = disjunction of conjunctions of literals
- Doesn't compose the way CNF does: can't just add new conjuncts w/o changing meaning of KB

Transforming to CNF or DNF

- Naive algorithm:
  - replace all connectives with ∧∨←
  - move negations inward using De Morgan's laws and double-negation
  - repeatedly distribute over ∧ over ∨ for DNF (∨ over ∧ for CNF)

Discussion

- Problem with naive algorithm: it's exponential! (Space, time, size of result.)
- Each use of distributivity can almost double the size of a subformula
- Tseitsin transformation (On the complexity of derivation in propositional calculus. Studies in Constrained Mathematics and Mathematical Logic, 1968)
  - Avoids exponential blowup by introducing new variables

Semantics

- Recall: meaning of a formula is a function
  - models ↦ {T, F}
- Meanings are **compositional**
- Write [α] for meaning of formula α
- [α ∧ β](M) = [α](M) ∧ [β](M)
- Similarly for ∨, ¬, etc.

Entailment

- Sentence A **entails** sentence B, $A \models B$, if B is true in every model where A is
  - same as saying that $(A \Rightarrow B)$ is valid

Inference rule

- To make a proof tree, we need to be able to figure out new formulas entailed by KB
- Method for finding entailed formulas = **inference rule**
- We've implicitly been using one already

Modus ponens

$$\frac{(a \wedge b \wedge c \Rightarrow d) \quad a \quad b \quad c}{d}$$

- Probably most famous inference rule: all men are mortal, Socrates is a man, therefore Socrates is mortal
- Quantifier-free version:
- man(Socrates) ∧
- (man(Socrates) ⇒ mortal(Socrates))

Another inference rule

$$\frac{(a \Rightarrow b) \quad \neg b}{\leftarrow a}$$

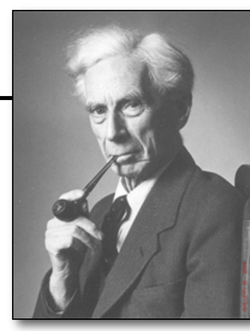
- **Modus tollens**
- If it's raining the grass is wet; the grass is not wet, so it's not raining

One more…

$$\frac{(\alpha \vee c) \quad (\neg c \vee \beta)}{\alpha \vee \beta}$$

- **Resolution**
  - α, β are arbitrary subformulas
- Combines two formulas that contain a literal and its negation
- Not as commonly known as modus ponens / tollens

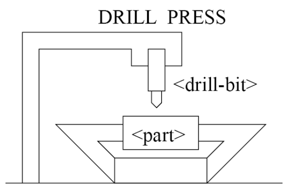
First-order logic

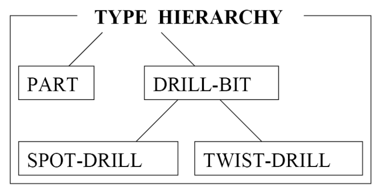


Bertrand Russell
1872-1970

- Enables representing and reasoning about knowledge, such as:
- "*All* faculty have an office"
- "If Pat is happy, then *there is someone else* happy too"

Example – Agent Action Model





drill-spot (<part>, <drill-bit>)
<part>: type PART
<drill-bit>: type SPOT-DRILL
Pre: (holding-tool <drill-bit>)
(holding-part <part>)
Add: (has-spot <part>)

put-drill-bit (<drill-bit>)
<drill-bit>: type DRILL-BIT
Pre: tool-holder-empty
Add: (holding-tool <drill-bit>)
Del: tool-holder-empty

put-part(<part>)
<part>: type PART
Pre: part-holder-empty
Add: (holding-part <drill-bit>)
Del: part-holder-empty

drill-hole(<part>, <drill-bit>)
<part>: type PART
<drill-bit>: type TWIST-DRILL
Pre: (has-spot <part>)
(holding-tool <drill-bit>)
(holding-part <part>)
Add: (has-hole <part>)

remove-drill-bit(<drill-bit>)
<drill-bit>: type DRILL-BIT
Pre: (holding-tool <drill-bit>)
Add: tool-holder-empty
Del: (holding-tool <drill-bit>)

remove-part(<part>)
<part>: type PART
Pre: (holding-part <drill-bit>)
Add: part-holder-empty
Del: (holding-part <drill-bit>)

Summary: Propositional logic

- Syntax
  - variables, constants, operators
  - literals, clauses, sentences
- Semantics (model $\mapsto$ {T, F})
- Truth tables, how to evaluate formulas
- Satisfiable, valid, contradiction
- Relationship to CSPs

Summary: Propositional logic and Beyond

- Manipulating formulas (e.g., de Morgan)
- Normal forms (e.g., CNF)
- Tseitin transformation to CNF
- Compositional semantics

- Entailment
- Inference rules
- Later: first order logic (in classical planning)