Knowledge-based systems in Bioinformatics, 1MB602

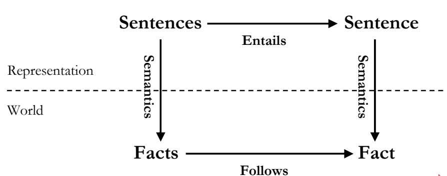
Lecture 7:
Logical inference

Lecture overview

- Logical reasoning
- Inference rules
- Example proofs
- Natural deduction
- SLD Resolution

Reasoning

- The property of one fact following from some other facts is mirrored by the property of one sentence being entailed by some other sentences



Reasoning cont.

- Entailment relation between sentences
  - We want to generate new sentences that are necessarily true, given that the old sentences are true
- An inference procedure that generates only entailed sentences is called sound (truth-preserving)
  - The inference steps should respect the semantics of the sentences they operate upon
- The record of operation of a sound inference procedure is called a proof
- A proof theory specifies which reasoning steps that are sound
- An inference procedure is called complete if it can find a proof for any sentence that is entailed

Inference

- Sound reasoning
  - Logical inference
  - Deduction
- A sentence is valid if and only if it is true under all possible interpretations in all possible worlds
  - "There is a wall in front of me OR there is not a wall in front of me"
- A sentence is satisfiable if and only if there is some interpretation in some world for which it is true
- A sentence that is not satisfiable is called contradictory (unsatisfiable)
  - "There is a wall in front of me AND there is not a wall in front of me"

Inference in propositional logic

- Logical implication: $|=$
  - A set of wfps $\{A_1,\ldots,A_n\}$ logically implies the wfp B (written $A_1,\ldots,A_n \models B$) if and only if B is True in every situation in which every A_i is True
  - $A_1,\ldots,A_n \models B$ if and only if $\models A_1 \wedge \ldots \wedge A_n \rightarrow B$
  - Could be checked by truth tables:
    - Check if B is True in every situation in which every A_i is True
    - Check if the wfp $A_1 \wedge \ldots \wedge A_n \rightarrow B$ is valid (tautology)

Propositional Logic – logical implication

- Check if $(P \vee H), \neg H \models P$
- Check validity of the following implication:
  $((P \vee H) \wedge \neg H) \rightarrow P$
- Valid: True in every situation

P	H	$P \vee H$	$(P \vee H) \wedge \neg H$	$((P \vee H) \wedge \neg H) \rightarrow P$
True	True	True	False	True
True	False	True	True	True
False	True	True	False	True
False	False	False	False	True

$\models ((P \vee H) \wedge \neg H) \rightarrow P$

$(P \vee H), \neg H \models P$

Inference rules for propositional logic

- Formula with 10 propositions has truth table with $2^{10} = 1024$ rows, too big to do by hand!
- Avoid the tedious work of building truth tables by using inference rules
- Inference rule:
  - A rule stating how sentence β can be derived from sentence α by inference ($\alpha \vdash \beta$)

$$\frac{\alpha}{\beta}$$

- Soundness
  - An inference rule (derivation) is sound if the conclusion is true in all cases where the premises are true
    $A_1, \ldots, A_n \vdash B \;\Rightarrow\; A_1, \ldots, A_n \models B$
- Completeness
  - $A_1, \ldots, A_n \models B \;\Rightarrow\; A_1, \ldots, A_n \vdash B$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules for propositional logic

$$\frac{\alpha, \neg\alpha}{\bot}(\bot I)$$

$$\frac{\begin{array}{|l}\alpha\\ \bot\end{array}}{\neg\alpha}(\neg I) \qquad \frac{\begin{array}{|l}\neg\alpha\\ \bot\end{array}}{\alpha}(\neg E) \qquad \frac{\alpha \rightarrow \beta, \beta \rightarrow \alpha}{\alpha \leftrightarrow \beta}(\leftrightarrow I)$$

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}(\wedge I) \qquad \frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}(\wedge E) \qquad \frac{\alpha \leftrightarrow \beta}{\alpha \rightarrow \beta}(\leftrightarrow E)$$

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}(\vee I) \qquad \frac{\alpha \vee \beta, \alpha \rightarrow \chi, \beta \rightarrow \chi}{\chi}(\vee E)$$

$$\frac{\begin{array}{|l}\alpha\\ \beta\end{array}}{\alpha \rightarrow \beta}(\rightarrow I) \qquad \frac{\alpha \rightarrow \beta,\ \alpha}{\beta}(\rightarrow E)$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules for propositional logic

- $\bot$-Introduction
  - From two contradictory sentences you can infer contradiction (bottom) ($\bot$)

$$\frac{\alpha, \neg\alpha}{\bot}$$

- $\neg$-Elimination (reductio ad absurdum)
  - From the negation of a sentence (premise) and contradiction you can infer the sentence

$$\frac{\begin{array}{|l}\neg\alpha\\ \bot\end{array}}{\alpha}$$

- $\neg$-Introduction
  - From a sentence (premise) and contradiction you can infer the negation of the sentence

$$\frac{\begin{array}{|l}\alpha\\ \bot\end{array}}{\neg\alpha}$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules for propositional logic

- And-Elimination
  - From a conjunction, you can infer any of the conjuncts

$$\frac{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}{\alpha_i}$$

- And-Introduction
  - From a list of sentences, you can infer their conjunction

$$\frac{\alpha_1, \alpha_2, \ldots, \alpha_n}{\alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_n}$$

- Or-Elimination

$$\frac{\alpha \vee \beta, \alpha \rightarrow \chi, \beta \rightarrow \chi}{\chi}$$

- Or-Introduction
  - From a sentence, you can infer its disjunction with anything else at all

$$\frac{\alpha_i}{\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n}$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules for propositional logic

- $\rightarrow$-Elimination (Modus ponens)
  - From an implication and the antecedent of the implication, you can infer the consequent

$$\frac{\alpha \rightarrow \beta,\ \alpha}{\beta}$$

- $\rightarrow$-Introduction
  - From a premise and a consequent of the premise you can infer implication

$$\frac{\begin{array}{|l}\alpha\\ \beta\end{array}}{\alpha \rightarrow \beta}$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules for propositional logic

- $\leftrightarrow$-Introduction
  - From two implications in opposite directions you can infer equivalence

$$\frac{\alpha \rightarrow \beta,\ \beta \rightarrow \alpha}{\alpha \leftrightarrow \beta}$$

- $\leftrightarrow$-Elimination
  - From an equivalence you can infer implication in either direction

$$\frac{\alpha \leftrightarrow \beta}{\alpha \rightarrow \beta}$$

Inference rules for propositional logic (derived)

- Double-Negation Elimination

$$\frac{\neg\neg\alpha}{\alpha}$$

- Unit Resolution
  - From a disjunction, if one of the disjuncts is false, then you can infer the other one is true

$$\frac{\alpha \vee \beta,\ \neg\beta}{\alpha}$$

- Resolution
  - Because β cannot be both true and false, one of the other disjuncts must be true

$$\frac{\alpha \vee \beta,\ \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Logical equivalences (derived rules)

$P \wedge (Q \vee R)$	$\Leftrightarrow$	$(P \wedge Q) \vee (P \wedge R)$
$P \vee (Q \wedge R)$	$\Leftrightarrow$	$(P \vee Q) \wedge (P \vee R)$
$\neg(P \wedge Q)$	$\Leftrightarrow$	$\neg P \wedge \neg Q$
$\neg(P \vee Q)$	$\Leftrightarrow$	$\neg P \wedge \neg Q$
$P \Rightarrow Q$	$\Leftrightarrow$	$\neg Q \Rightarrow \neg P$
$P \Rightarrow Q$	$\Leftrightarrow$	$\neg P \vee Q$
$P \Leftrightarrow Q$	$\Leftrightarrow$	$(P \Rightarrow Q) \wedge (Q \Rightarrow P)$
$P \Leftrightarrow Q$	$\Leftrightarrow$	$(P \wedge Q) \vee (\neg P \wedge \neg Q)$
$P \wedge \neg P$	$\Leftrightarrow$	False
$P \vee \neg P$	$\Leftrightarrow$	True

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Deduction in practice

- Rules for 'cancellation' using brackets ($\neg$I, $\neg$E, $\rightarrow$I)

$$\frac{\begin{matrix}[\alpha] \\ \bot\end{matrix}}{\neg\alpha}(\neg I) \qquad \frac{\begin{matrix}[\neg\alpha] \\ \bot\end{matrix}}{\alpha}(\neg E) \qquad \frac{\begin{matrix}[\alpha] \\ \beta\end{matrix}}{\alpha \rightarrow \beta}(\rightarrow I)$$

1. Only premises are allowed to be cancelled
2. No formula within a bracket could be used afterwards
3. No non-cancelled premise may lie within a bracket
4. Brackets may not be crossed
5. (Consequence) Premises must be cancelled bottom-up

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Deduction in practice

Three strategies for deduction

1. Direct derivation
   - Identify the parts of the conclusion
   - Derive the parts using elimination rules
   - Combine the parts into the conclusion using introduction rules
2. Indirect derivation
   - Assume the negation of the conclusion (extra premise)
   - Derive contradiction (bottom)
3. Hypothetical derivation
   - You are supposed to prove $A_1,\ldots,A_n$ |- B$\rightarrow$C
   - Assume B (extra premise)
   - Derive C: $A_1,\ldots,A_n$, B |- C
   - Derive implication using $\rightarrow$I

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Example deduction I (direct derivation)

Prove that

$A \leftrightarrow B \wedge C,\ D \rightarrow B,\ D \wedge C$ |- A

1.	$A \leftrightarrow B \wedge C$	P	Premises
2.	$D \rightarrow B$	P	
3.	$D \wedge C$	P	
4.	C	3, ($\wedge$E)	
5.	D	3, ($\wedge$E)	
6.	B	2, 5, ($\rightarrow$E)	
7.	$B \wedge C$	4, 6, ($\wedge$I)	
8.	$B \wedge C \rightarrow A$	1, ($\leftrightarrow$E)	
9.	A	7, 8, ($\rightarrow$E)	

THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Example deduction II (hypothetical derivation)

Prove that

$A \to (B \to C) \mid\!- (A \to B) \to (A \to C)$

1.	$A \to (B \to C)$	P
2.	$A \to B$	P (extra)
3.	A	P (extra)
4.	$B \to C$	1, 3, ($\to$E)
5.	B	2, 3, ($\to$E)
6.	C	4, 5, ($\to$E)
7.	$A \to C$	3-6, ($\to$I)
8.	$(A \to B) \to (A \to C)$	2-7, ($\to$I)

Example deduction III (indirect derivation)

- Prove that
  $A \to B \mid\!- \neg B \to \neg A$

1.	$A \to B$	P
2.	$\neg B$	P (extra)
3.	A	P (extra)
4.	B	1, 3, ($\to$E)
5.	$\perp$	2, 4, ($\perp$I)
6.	$\neg A$	3-5, ($\neg$I)
7.	$\neg B \to \neg A$	2-6, ($\to$I)

LCB
THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Some heuristics

- Assume you are supposed to show $A_1,\ldots,A_n \mid\!- A$
- $A = \neg B$
  - Assume B as an extra premise. Derive $\perp$. Use ($\neg$E).
- $A = B \wedge C$
  - Derive B and C separately. Use ($\wedge$I).
- $A = B \vee C$
  - Derive B or C. Use ($\vee$I).
- $A = B \to C$
  - Assume B. Derive C. Use ($\to$I)
- $A = B \leftrightarrow C$
  - Derive $B \to C$ and $C \to B$ separately. Use ($\leftrightarrow$ I).

LCB
THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules in FO predicate logic

- Same rules as in propositional logic
- Additional rules:
  - Universal elimination
  - Universal introduction
  - Existential elimination
  - Existential introduction

LCB
THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Inference rules in FO predicate logic

- Universal introduction ($\forall$I)
  - If X is not free in any premise that P(X) depends on

$$\frac{P(X)}{\forall X\, P(X)}$$

- Universal elimination ($\forall$E)
  - t free for X in P(X) (no variable in t may be bound in P(X))

$$\frac{\forall X\, P(X)}{P(t)}$$

- Existential elimination ($\exists$E)
  - X not free in any non-cancelled premise before P(X)
  - X not free in Q

$$\frac{\exists X\, P(X) \qquad \begin{array}{|l} P(X) \\ Q \end{array}}{Q}$$

- Existential introduction ($\exists$I)
  - t free for X in P(X)

$$\frac{P(t)}{\exists X\, P(X)}$$

LCB
THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Example deduction

- Prove that
  $\forall X\, \neg P(X) \mid\!- \neg \exists X\, P(X)$

1.	$\forall X\, \neg P(X)$	P
2.	$\exists X\, P(X)$	P (Extra)
3.	$P(X)$	P (Extra)
4.	$\neg P(X)$	1, ($\forall$E)
5.	$\perp$	3, 4, ($\perp$I)
6.	$\perp$	2, 3-5, ($\exists$E)
7.	$\neg \exists X\, P(X)$	2-6, ($\neg$I)

LCB
THE LINNAEUS CENTRE FOR BIOINFORMATICS
http://www.lcb.uu.se

Reasoning with definite logic programs

Given a program P and a goal G, find instances of G which are logical consequences of P

```
gchild(X,Y) ⇐ child(X,Z) ∧ child(Z,Y)
child(john,mary) ⇐
child(mary,bob) ⇐
child(mary,sue) ⇐

⇐ child(john,X) ∧ child(X,bob)
 X = mary
⇐ gchild(john,X)
 X = bob, X = sue
```


SLD refutation

```
gchild(X,Y) ⇐ child(X,Z) ∧
   child(Z,Y)

child(john,mary) ⇐
child(mary,sue) ⇐
child(mary,bob) ⇐
```

gchild(V,T)

gchild(X,Y) ⇐ child(X,Z) ∧ child(Z,Y)   X=V Y=T

child(V,Z) ∧ child(Z,T)

child(john,mary) ⇐   V=john Z=mary

child(mary,T)

child(mary,sue) ⇐   T=sue

□

Answer: V=john T=sue

Unification

- **Consider the following** substitution σ **(binding of variables to terms)**
  {f(a, Z)/X, g(Z)/Y}
- Instance tσ **of a term** t **under** σ:
  f(X, h(X, f(Y, V)))σ = f(f(a, Z), h(f(a, Z), f(g(Z), V))
- **A** unifier **of terms** t **and** s **is any** σ **such that** tσ = sσ
  f(f(a, Z),h(X, f(g(X), W)))   f(V, h(V, f(g(f(Y, U)), V)))
  {f(a,U)/V, f(a,U)/X, a/Y, U/Z, f(a,U)/W}

  f(f(a, U), h(f(a, U), f(g(f(a, U)), f(a, U))))
- **This is a** most general unifier

Unification cont.

- **Unification procedure:**

f(f(a, Z), h(X, f(g(X), W)))   f(V, h(V, f(g(f(Y, U)), V)))

{V=f(a,Z), X=V, X=f(Y,U), W=V}
{V=f(a,Z), X=f(a,Z), X=f(Y,U), W=f(a,Z)}
{V=f(a,Z), X=f(a,Z), f(a,Z)=f(Y,U), W=f(a,Z)}
{V=f(a,Z), X=f(a,Z), Y=a, Z=U, W=f(a,Z)}
{V=f(a,U), X=f(a,U), Y=a, Z=U, W=f(a,U)}

{f(a,U)/V, f(a,U)/X, a/Y, U/Z, f(a,U)/W}

SLD resolution principle

$$\frac{\Leftarrow A_1 \wedge \dots \wedge A_{i-1} \wedge A_i \wedge A_{i+1} \wedge \dots \wedge A_n, \quad B_0 \Leftarrow B_1 \wedge \dots \wedge B_n}{(\Leftarrow A_1 \wedge \dots \wedge A_{i-1} \wedge B_1 \wedge \dots \wedge B_n \wedge A_{i+1} \wedge \dots \wedge A_n)\,\theta}$$

where θ is the most general unifier of A_i and B_0
Which i to be chosen: computation rule

SLD tree

```
gchild(X,Y) ⇐ child(X,Z) ∧
   child(Z,Y)

child(john,mary) ⇐
child(mary,sue) ⇐
child(mary,bob) ⇐
```

gchild(V,T)

child(V,Z) ∧ child(Z,T)

child(mary,T)

□   □

V = john T = sue   V = john T = bob

SLD tree shows all resolvents of each selected atom

References

- K.B. Hansen, Grundläggande logik, 3:e upplagan, 2001, Studentlitteratur
- S. Russell, P. Norvig, Artificial intelligence: a modern approach, Prentice-Hall, Upper Saddle River, New Jersey, 1995
- S. C. Shapiro, Propositional, First-Order And Higher-Order Logics: Basic Definitions, Rules of Inference, and Examples, in L. L.M. Iwanska, S. C. Shapiro, Eds., Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language, AAAI Press/The MIT Press, Menlo Park, CA, 2000, 379-395