Natural Deduction

EECS 344
Winter 2008

Overview

- Natural deduction as a domain for thinking about problem solver ideas
- The KM* system
- Examples with KM*

Natural deduction as microworld

- Was in fact studied intensively at various times in AI research
  - Originally developed by logicians as a model for how people reason
- Rarely used in practical systems today
  - You’ll see some better techniques soon
- But still useful for understanding tradeoffs in designing reasoning systems

Kalish & Montegue

- Developed a set of heuristics for doing logical proofs
- Introduction rules
  - **not** introduction, **and** introduction, **or** introduction, **conditional** introduction, **biconditional** introduction
- Elimination rules
  - **not** elimination, **and** elimination, **or** elimination, **conditional** elimination and **biconditional** elimination
- KM* is based on their formalism

Kalish & Montegue cont.

- Organized along five different connectives:
  - **`not`**, **`and`**, **`or`**, **`implies`** ($\Rightarrow$), and **`iff`** ($\Leftrightarrow$).
- Rules can either *eliminate* relations or *introduce* new relations
  - Elimination rules are much like the simplification rules in Bundy system
  - Introduction rules are require more control knowledge--cannot be used randomly

Structure of a proof

- A proof is a set of numbered lines
- Each line has the form
  <line number> <statement> <justification>
- Example:

```
27 (implies P Q)          Asn
28 P                    given
29 Q               (CE 27 28)
```

Proof Rules for KM*

- Use *boxes* to denote logical contexts
- Example: Indirect proof schemata

h	~~show~~ P	(IP i j k)
i	(not P)	asn
	.	
	.	
j	Q	
	.	
	.	
k	(not Q)	

Not Elimination

```
i    (not (not P))
     P                 (NE i)
```

AND Elimination

```
i    (and P Q)
     P                 (AE i)
     Q                 (AE i)
```

OR Elimination

```
i    (or P Q)
j    (implies P R)
k    (implies Q R)
       R               (OE i j k)
```

Conditional Elimination

```
i    (implies P Q)
j    P
     Q                  (CE i j)
```

Biconditional Elimination

```
i    (iff P Q)
     (implies P Q)          (BE i)
     (implies Q P)          (BE i)
```

Not Introduction

```
      show  (not P)            (NI i j k)
i           P                        Asn
            .
j           Q
            .
k           (not Q)
```

And Introduction

```
i    P
j    Q
     (and P Q)          (AI i j)
     (and Q P)          (AI j i)
```

Or Introduction

```
i   P
    (or P Q)          (OI i)
    (or Q P)          (OI i)
```

Conditional Introduction

```
  show (implies P Q)       (CI i j)
i      P                        Asn
j      show Q
```

Biconditional Introduction

```
i   (implies P Q)
j   (implies Q P)
    (iff P Q)         (BI i j)
```

KM* Examples

- Several of them in the book
- How to read them:
  - Don't go directly to the proofs: Try proving them yourself first, then peek.
  - Think about what you are doing in the process of doing a proof. That is the process you will be modeling soon...

Simple KM* Example

- Premises
  - If it is spring, there cannot be snow
  - It snowed this week
- Show
  - It cannot be spring
- Formalization
  - `(implies spring (not snow))`
  - `snow`
  - `(show (not spring))`

1.	(implies spring (not snow))	Premise
2.	snow	Premise
3.	(show (not spring))	

1.	(implies spring (not snow))	Premise
2.	snow	Premise
3.	(show (not spring))	
4.	spring	Asn

1.	(implies spring (not snow))	Premise
2.	snow	Premise
3.	(show (not spring))	
4.	spring	Asn
5.	(not snow)	(CE 1 4)

1.	(implies spring (not snow))	Premise
2.	snow	Premise
3.	(~~snow~~ (not spring))	(NI 4 2 5)
4.	spring	Asn
5.	(not snow)	(CE 1 4)

Implementing KM* in a reasoning system

- How to encode the rules?
  - Some kind of rule would make sense
  - How do we control them?
- How to do boxes?