FIT5047 Week 3
Knowledge representation

授课老师：Joe

Propositional Logic

FIT5047平时班 – Week 3

Literal: a proposition or its negation

- E.g., P, $\neg P$

Clause: a disjunction of literals

- E.g., $\neg P \vee Q \vee A$

Negation: If S is a sentence, $\neg S$ is a sentence

Conjunction:
If S_1 and S_2 are sentences, $S_1 \wedge S_2$ is a sentence

Disjunction:
If S_1 and S_2 are sentences, $S_1 \vee S_2$ is a sentence

Implication:
If S_1 and S_2 are sentences, $S_1 \Rightarrow S_2$ is a sentence

Biconditional:
If S_1 and S_2 are sentences, $S_1 \Leftrightarrow S_2$ is a sentence

FIT5047平时班 – Week 3

$\neg S$	is true iff S is false	
$S_1 \wedge S_2$	is true iff S_1 is true and S_2 is true	
$S_1 \vee S_2$	is true iff S_1 is true or S_2 is true	
$S_1 \Rightarrow S_2 \equiv \neg S_1 \vee S_2$	is true iff S_1 is false or S_2 is true	
$S_1 \Leftrightarrow S_2$	is true iff $S_1 \Rightarrow S_2$ is true and $S_2 \Rightarrow S_1$ is true	

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
false	*false*	*true*	*false*	*false*	*true*	*true*
false	*true*	*true*	*false*	*true*	*true*	*false*
true	*false*	*false*	*false*	*true*	*false*	*false*
true	*true*	*false*	*true*	*true*	*true*	*true*

FIT5047平时班 – Week 3

- Two sentences are **logically equivalent** iff they are both true in the same models: $\alpha \equiv \beta$ iff $\alpha \models \beta$ and $\beta \models \alpha$

$(\alpha \wedge \beta)$	$\equiv (\beta \wedge \alpha)$	commutativity of $\wedge$
$(\alpha \vee \beta)$	$\equiv (\beta \vee \alpha)$	commutativity of $\vee$
$((\alpha \wedge \beta) \wedge \gamma)$	$\equiv (\alpha \wedge (\beta \wedge \gamma))$	associativity of $\wedge$
$((\alpha \vee \beta) \vee \gamma)$	$\equiv (\alpha \vee (\beta \vee \gamma))$	associativity of $\vee$
$\neg(\neg\alpha)$	$\equiv \alpha$	double-negation elimination
$(\alpha \Rightarrow \beta)$	$\equiv (\neg\beta \Rightarrow \neg\alpha)$	contraposition
$(\alpha \Rightarrow \beta)$	$\equiv (\neg\alpha \vee \beta)$	implication elimination
$(\alpha \Leftrightarrow \beta)$	$\equiv ((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$	biconditional elimination
$\neg(\alpha \wedge \beta)$	$\equiv (\neg\alpha \vee \neg\beta)$	de Morgan
$\neg(\alpha \vee \beta)$	$\equiv (\neg\alpha \wedge \neg\beta)$	de Morgan
$(\alpha \wedge (\beta \vee \gamma))$	$\equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$	distributivity of $\wedge$ over $\vee$
$(\alpha \vee (\beta \wedge \gamma))$	$\equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$	distributivity of $\vee$ over $\wedge$

- **A sentence is valid if it is true in all models**
  - E.g., *True*, $A \lor \neg A$, $A \Rightarrow A$
- **Validity is connected to inference via the Deduction Theorem**
  - $\alpha \models \beta$ iff $\alpha \Rightarrow \beta$ is valid
- **A sentence is satisfiable if it is true in some model**
  - E.g., $A \lor B$, C
- **A sentence is unsatisfiable if it is true in no model**
  - E.g., $A \land \neg A$
- **Satisfiability and validity are connected**
  - α is valid iff $\neg\alpha$ is unsatisfiable
  - α is satisfiable iff $\neg\alpha$ is not valid
  - $\alpha \models \beta$ iff $\alpha \land \neg\beta$ is unsatisfiable

FIT5047平时班 – Week 3

Prove that $(A \wedge (A \Rightarrow B)) \Rightarrow B$ **is valid**

- $(A \wedge (\neg A \vee B)) \Rightarrow B$
- $((A \wedge \neg A) \vee (A \wedge B)) \Rightarrow B$
- $(A \wedge B) \Rightarrow B$
- $\neg(A \wedge B) \vee B$
- $\neg A \vee \neg B \vee B$
- True

FIT5047平时班 – Week 3

- **Prove that α is valid iff ¬α is unsatisfiable**
  - If α is valid, it is true in all models ➔ there does not exist a model for which ¬α is true ➔ ¬α is unsatisfiable
  - If ¬α is unsatisfiable ➔ there does not exist a model for which ¬α is true ➔ α is true in all models ➔ α is valid
- **Prove that α is satisfiable iff ¬α is not valid**
  - If α is satisfiable, it is true in some models ➔ ¬α is not true in these models ➔ ¬α is not valid
  - If ¬α is not valid➔ there exist some models for which ¬α is false ➔ α is true in these models ➔ α is satisfiable

Resolution

1. Eat $\Rightarrow$ HaveLessMoney
2. $\neg$Eat $\Rightarrow$ Hungry

Resolvent: HaveLessMoney $\vee$ Hungry

Conversion to conjunctive normal form

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta)\wedge(\beta \Rightarrow \alpha)$
2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha\vee\beta$
3. Move $\neg$ inwards by repeated application of the following equivalences:
   - double-negation: $\neg(\neg\alpha) \equiv \alpha$
   - de Morgan $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$
   - de Morgan $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$
4. Apply distributivity law ($\wedge$ over $\vee$) and flatten

FIT5047平时班 – Week 3

$A \Leftrightarrow (B \vee C)$

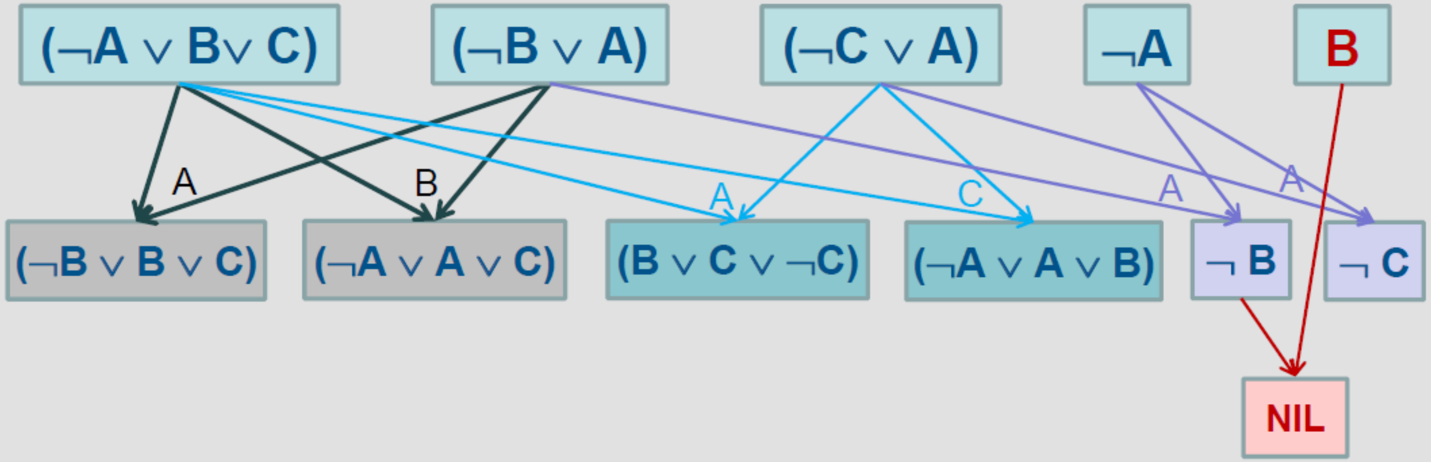
1. Eliminate $\Leftrightarrow$: $(A \Rightarrow (B \vee C)) \wedge ((B \vee C) \Rightarrow A)$
2. Eliminate $\Rightarrow$: $(\neg A \vee (B \vee C)) \wedge (\neg (B \vee C) \vee A)$
3. Move $\neg$ inwards: $(\neg A \vee (B \vee C)) \wedge ((\neg B \wedge \neg C) \vee A)$
   $(\neg A \vee B \vee C) \wedge ((\neg B \wedge \neg C) \vee A)$
4. Apply distributivity law:
   $(\neg A \vee B \vee C) \wedge (\neg B \vee A) \wedge (\neg C \vee A)$

FIT5047平时班 – Week 3

- **Proof by refutation**
  1. Negate the goal and add the negation to the set of clauses
  2. Apply resolution to the clauses in the set of clauses until a contradiction is reached
- **Answer extraction**
  1. Build a tautology by appending the goal itself to the negation of the goal
  2. When the negated goal is contradicted, the answer resides in the goal

FIT5047平时班 – Week 3

- ***KB*** **= (A ⇔ (B∨ C)) ∧ ¬A**
  (¬A ∨ B∨ C) ∧ (¬B ∨ A) ∧ (¬C ∨ A) ∧ ¬A
- **Prove: α = ¬B**



FIT5047平时班 – Week 3

- **KB:**
  - R1: $P \Rightarrow Q$
  - R2: $L \wedge M \Rightarrow P$
  - R3: $B \wedge L \Rightarrow M$
  - R4: $A \wedge P \Rightarrow L$
  - R5: $A \wedge B \Rightarrow L$
  - A
  - B
- **Prove Q**

- **KB:**
  - R1: $\neg P \vee Q$
  - R2: $\neg L \vee \neg M \vee P$
  - R3: $\neg B \vee \neg L \vee M$
  - R4: $\neg A \vee \neg P \vee L$
  - R5: $\neg A \vee \neg B \vee L$
  - A
  - B
- **Negate Q: $\neg Q$**

$\underline{\neg \mathbf{Q}}$ R1: $\neg P \vee \underline{Q}$

$\underline{\neg \mathbf{P}}$

R2: $\neg L \vee \neg M \vee \underline{P}$

$\neg \mathbf{L} \vee \underline{\neg \mathbf{M}}$

R3: $\neg B \vee \neg L \vee \underline{M}$

$\neg \mathbf{B} \vee \underline{\neg \mathbf{L}}$

R5: $\neg A \vee \neg B \vee \underline{L}$

$\underline{\neg \mathbf{A}} \vee \neg \mathbf{B}$

$\underline{A}$

$\underline{\neg \mathbf{B}}$

$\underline{B}$

nil

FIT5047平时班 – Week 3

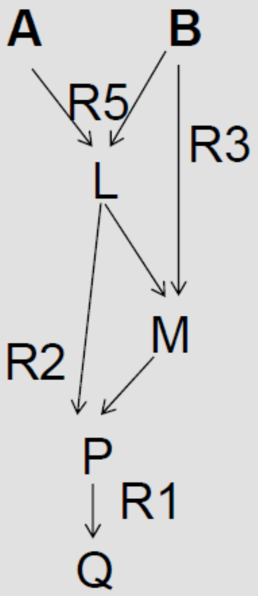
Definite clause – a disjunction of literals of which <u>exactly one</u> is positive

$\neg A \lor \neg B \lor C$

Horn clause – a disjunction of literals of which <u>at most one</u> is positive

FIT5047平时班 – Week 3

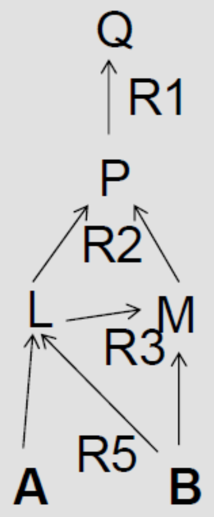
- **KB:**
  - R1: $P \Rightarrow Q$
  - R2: $L \wedge M \Rightarrow P$
  - R3: $B \wedge L \Rightarrow M$
  - R4: $A \wedge P \Rightarrow L$
  - R5: $A \wedge B \Rightarrow L$
  - A
  - B
- **Prove Q**



Agenda	Count					Inferred
	R1	R2	R3	R4	R5	
A B	1	2	2	2	2	
L						
M						

FIT5047平时班 – Week 3

- **KB:**
  - R1: $P \Rightarrow Q$
  - R2: $L \wedge M \Rightarrow P$
  - R3: $B \wedge L \Rightarrow M$
  - R4: $A \wedge P \Rightarrow L$
  - R5: $A \wedge B \Rightarrow L$
  - A
  - B
- **Prove Q**



First order logic

Notation

Predicate symbol – MARRIED
Constant – *John* or *A*
Function – *Mother* or *f*
Variable – *x*

Predicate – a function that only returns Boolean
Function can return any value

FIT5047平时班 – Week 3

- disjunction – $P(x) \vee Q(y) \vee W(x,y)$
- conjunction – $P(x) \wedge Q(y)$
- implication – $P(x) \Rightarrow W(x,y) \equiv \neg P(x) \vee W(x,y)$

FIT5047平时班 – Week 3

Quantification

predicate constant

- Universal (∀) –
  $\forall x\ [\text{ELEPHANT}(x) \Rightarrow \text{COLOUR}(x,\text{Gray})]$
- Existential (∃) –
  $\exists x\ \text{WRITE}(x,\text{Computer-Chess})$

variable

FIT5047平时班 – Week 3

$$\exists x(\forall y[P(x,y)\wedge Q(x,y)]\Rightarrow R(x))$$

$$\neg f(A)$$

$$\neg P(A,g(A,B,A))$$

$$f(P(A))$$

$$\forall P\ P(A)$$

FIT5047平时班 – Week 3

- Existential inside the scope of universal
  E.g., $\forall s \exists c$ Eats(s,c)

- Universal inside the scope of existential
  E.g., $\exists c \forall s$ Eats(s,c)

- $\neg(\exists x)P(x) \equiv (\forall x)\,[\neg P(x)]$
  There does not exist an x such that P(x) is true ≡
  For all x, P(x) is false
- $\neg(\forall x)P(x) \equiv (\exists x)\,[\neg P(x)]$
  It is not true that for all x P(x) is true ≡
  There exists an x, such that ¬P(x)
- $(\forall x)[P(x) \land Q(x)] \equiv (\forall x)P(x) \land (\forall y)Q(y)$
  For all x, P(x) and Q(x) are true ≡
  For all x, P(x) is true, and for all y Q(y) is true
- $(\exists x)[P(x) \lor Q(x)] \equiv (\exists x)P(x) \lor (\exists y)Q(y)$
  There is an x, such that P(x) is true or Q(x) is true ≡
  There is an x, such that P(x) is true, or
  there is a y, such that Q(y) is true

$(\forall x)[P(x) \lor Q(x)]$ $(\forall x)P(x) \lor (\forall y)Q(y)$

Substitution

Substitution is a set of ordered pairs
$s=\{v_1|t_1, v_2|t_2, \ldots, v_n|t_n\}$
where $v_i|t_i$ means that term t_i substitutes variable v_i throughout

- Example:
  $P(x,y)\ \{x|A, y|B\} \rightarrow P(A,B)$

Semantics: all elements are applied *simultaneously*

- Example:
  $P(w, y, g(z), x)\ \{x|g(y), y|h(z), z|x\} \rightarrow P(w, h(z), g(x), g(y))$

Substitution

$s_i s_j$ – composition of two substitutions

- Apply s_j to the terms of s_i
- Add any pairs of s_j having variables not in s_i

Example: term variable

s1 = {z|g(x,y)}   s2 = {x|A, y|B, w|C, z|D}

s1s2 = {z|g(x,y)}{x|A, y|B, w|C, z|D}={z|g(A,B), x|A, y|B, w|C}

s2s1 = {x|A, y|B, w|C, z|D} {z|g(x,y)} = {x|A, y|B, w|C, z|D}

NOT commutative: s1s2 ≠ s2s1

Unification

s={x|A, y|B} unifies P(x,f(y)) with P(x,f(B))

You can substitute anything into variable, not the other way around

Converting wffs into clauses

1. Eliminate implication symbols
2. Reduce scopes of negation symbols
3. Standardize variables (for each quantifier)
4. Eliminate existential quantifiers (*skolemize*)
5. Move all universal quantifiers to the front
6. Put result in conjunctive normal form (CNF)
7. Eliminate universal quantifiers
8. Eliminate $\wedge$ symbols
9. Rename variables (standardize variables apart for each clause)

FIT5047平时班 – Week 3

$\forall x$ [CanRead(x) $\Rightarrow$ Intelligent(x)]

1. Eliminate $\Rightarrow$: $\forall x$ [$\neg$CanRead(x) v Intelligent(x)]

7. Eliminate $\forall$: $\neg$CanRead(x) v Intelligent(x)

FIT5047平时班 – Week 3

$\forall x\ [\neg\ (\forall y)\ [\ P(x,y) \Rightarrow Q(x,y)]\ ]$

1. Eliminate $\Rightarrow$: $\forall x\ [\neg\ (\forall y)\ [\ \neg P(x,y) \vee Q(x,y)]\ ]$
2. Reduce scope of $\neg$: $\forall x\ [\ \exists y\ \neg\ [\ \neg P(x,y) \vee Q(x,y)]\ ]$
   $\forall x\ \ \exists y\ \ [\ P(x,y) \wedge \neg Q(x,y)\ ]$
4. Eliminate $\exists$: $\forall x\ [\ P(x,g(x)) \wedge \neg Q(x,g(x))\ ]$
7. Eliminate $\forall$: $P(x,g(x)) \wedge \neg Q(x,g(x))$
8. Eliminate $\wedge$ symbols: $\{\ P(x,g(x)),\ \neg Q(x,g(x))\ \}$
9. Standardize variables apart: $\{\ P(x_1,g(x_1)),\ \neg Q(x_2,g(x_2))\ \}$

FIT5047平时班 – Week 3

1. **If a unit is easy, there are some students who are enrolled in it who are happy**

   $\forall u\ [\ Easy(u) \Rightarrow \exists s\ [\ Enrolled(s,u) \land Happy(s)\ ]\ ]$

2. **If a unit has a final exam, no students that are enrolled in it are happy**

   $\forall u\ [\ HasFinal(u) \Rightarrow \neg \exists s\ [\ Enrolled(s,u) \land Happy(s)\ ]\ ]$

3. **Prove that if a unit has a final exam, the unit is not easy**

   $\forall u\ [\ HasFinal(u) \Rightarrow \neg\ Easy(u)\ ]$

FIT5047平时班 – Week 3

1. $\forall u\ [\ Easy(u) \Rightarrow \exists s\ [\ Enrolled(s,u) \wedge Happy(s)\ ]\ ]$

Eliminate $\Rightarrow$: $\forall u\ [\ \neg\ Easy(u) \vee \exists s\ [\ Enrolled(s,u) \wedge Happy(s)\ ]\ ]$

Eliminate $\exists$: $\forall u\ [\ \neg\ Easy(u) \vee [\ Enrolled(g(u),u) \wedge Happy(g(u))\ ]\ ]$

Convert to CNF: $\forall u\ [\ [\ \neg\ Easy(u) \vee Enrolled(g(u),u)\ ] \wedge$
$[\ \neg\ Easy(u) \vee Happy(g(u))\ ]\ ]$

Eliminate $\forall$: $[\ \neg\ Easy(u) \vee Enrolled(g(u),u)\ ] \wedge$
$[\ \neg\ Easy(u) \vee Happy(g(u))\ ]$

Eliminate $\wedge$ and standardize variables apart:

1.1 $\neg\ Easy(u_1) \vee Enrolled(g(u_1),u_1)$

1.2 $\neg\ Easy(u_2) \vee Happy(g(u_2))$

FIT5047平时班 – Week 3

2. $\forall u\ [\ HasFinal(u) \Rightarrow \neg\ \exists s\ [\ Enrolled(s,u) \wedge Happy(s)\ ]\ ]$

Eliminate $\Rightarrow$: $\forall u\ [\ \neg\ HasFinal(u) \vee$

$\neg\ \exists s\ [\ Enrolled(s,u) \wedge Happy(s)\ ]\ ]$

Reduce scope of $\neg$:

$\forall u\ [\neg\ HasFinal(u) \vee \forall s\ \neg[\ Enrolled(s,u) \wedge Happy(s)\ ]\ ]$

$\forall u\ [\neg\ HasFinal(u) \vee \forall s[\ \neg Enrolled(s,u) \vee \neg Happy(s)\ ]\ ]$

Move $\forall$ to the front: $\forall u\ \forall s\ [\ \neg\ HasFinal(u) \vee$

$\neg\ Enrolled(s,u) \vee \neg Happy(s)\ ]$

Eliminate $\forall$ and standardize variables apart:

$\neg\ HasFinal(u_3) \vee \neg Enrolled(s_3,u_3) \vee \neg Happy(s_3)$

FIT5047平时班 – Week 3

Using resolution to prove statement 3

Negate the goal:

3'. $\neg \forall u\, [\, HasFinal(u) \Rightarrow \neg\, Easy(u)\,]$

Eliminate $\Rightarrow$: $\neg \forall u\, [\neg\, HasFinal(u) \vee \neg\, Easy(u)\,]$

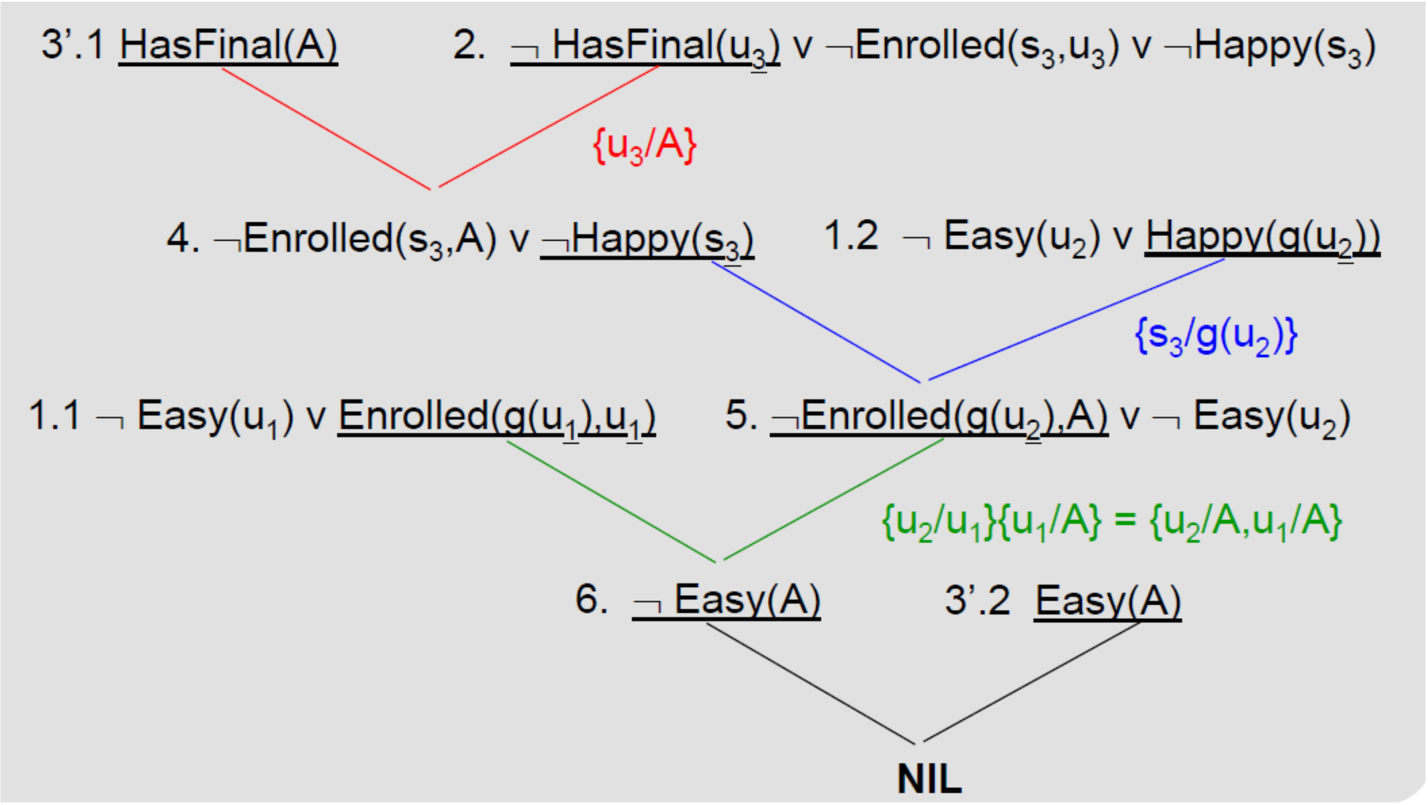
Reduce scope of $\neg$: $\exists u\, [\, HasFinal(u) \wedge Easy(u)\,]$

Eliminate $\exists$: $HasFinal(A) \wedge Easy(A)$

Eliminate $\wedge$: 3'.1 $HasFinal(A)$

3'.2 $Easy(A)$

FIT5047平时班 – Week 3



FIT5047平时班 – Week 3

Question answering

1. MANAGER(Purchasing-dept., John-Jones)
2. WORKSIN(Purchasing-dept., Joe-Smith)
3. $\forall x\ \forall y\ \forall z\ [\text{WORKSIN}(x,y) \land \text{MANAGER}(x,z)] \Rightarrow \text{BOSSOF}(y,z)$
4. **Goal:** Who is the boss of Joe Smith?

   $\exists x\ \text{BOSSOF}(\text{Joe-Smith},x)$

FIT5047平时班 – Week 3

3. ¬ WORKSIN(x_1,y) v ¬ MANAGER(x_1,z) v BOSSOF(y,z)

4. ¬ BOSSOF(Joe-Smith,x_2) v BOSSOF(Joe-Smith,x_2)

{y|Joe-Smith, x_2|z}

5. ¬ WORKSIN(x_1,Joe-Smith) v ¬ MANAGER(x_1,z) BOSSOF(Joe-Smith,z)

1. MANAGER(Purchasing-dept., John-Jones)

{x_1|Purchasing-dept, z|John-Jones}

6. ¬ WORKSIN(Purchasing-dept,Joe-Smith) BOSSOF(Joe-Smith,John-Jones)

2. WORKSIN(Purchasing-dept,Joe-Smith)

NIL