Week 9 Tutorial

Predicate Logic & Natural Deduction Rules for Predicate Logic

Joe Scott / Jan Gorzny

Prepared based off of the notes of CS245 Instructors, past and present.

3 March 2017

Plan

Predicate Translations 1

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Everybody has a mother

Predicate Translations 1

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Everybody has a mother

$(\forall y(\exists x M(x, y))$

Predicate Translations 1

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Everybody has a mother

$(\forall y(\exists x M(x, y))$

Problem

Everybody has a mother and Father

Predicate Translations 1

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Everybody has a mother

$(\forall y(\exists x M(x, y))$

Problem

Everybody has a mother and Father

$(\forall y(\exists x M(x, y) \wedge (\exists z F(z, y)))$

Predicate Translations 2

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Whoever has a mother has a father

Predicate Translations 2

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Whoever has a mother has a father

$$(\forall y(\exists x M(x, y) \implies (\exists z F(z, y))))$$

Predicate Translations 3

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Ed is a grandfather

Predicate Translations 3

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

Ed is a grandfather

$(\exists y(\exists x(F(ed, x) \wedge ((M(x, y) \vee F(x, y))))))$

Predicate Translations 4

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

No uncle is an aunt

$\alpha(x) =$"Is an uncle"$=$

Predicate Translations 4

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

No uncle is an aunt

$\alpha(x) =$"Is an uncle"$= (\exists y(B(x, y) \wedge (\exists z(F(y, z) \vee M(y, z)))))$
$\beta(x) =$"Is an aunt"$=$

Predicate Translations 4

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

No uncle is an aunt

$\alpha(x) =$"Is an uncle"$= (\exists y(B(x, y) \wedge (\exists z(F(y, z) \vee M(y, z)))))$
$\beta(x) =$"Is an aunt"$= (\exists y(S(x, y) \wedge (\exists z(F(y, z) \vee M(y, z)))))$

Predicate Translations 4

1. Let $F(x, y)$ denote that x is the father of y
2. Let $M(x, y)$ denote that x is the mother of y.
3. Let $H(x, y)$ denote that x is the husband of y.
4. Let $S(x, y)$ denote that x is the sister of y
5. Let $B(x, y)$ denote that x is the brother of y

Problem

No uncle is an aunt

$\alpha(x) =$"Is an uncle"$= (\exists y(B(x, y) \wedge (\exists z(F(y, z) \vee M(y, z)))))$
$\beta(x) =$"Is an aunt"$= (\exists y(S(x, y) \wedge (\exists z(F(y, z) \vee M(y, z)))))$
$(\forall x(\alpha \implies (\neg\beta))$

Predicate Translations 5

Create your own predicates.

Problem

An attacker can persuade a server that a successful login has occurred, even if it hasnt.

Predicate Translations 5

Create your own predicates.

Problem

An attacker can persuade a server that a successful login has occurred, even if it hasnt.

Define

1. Let $P(x)$ denote user x logs in legitimately.
2. Let $Q(x)$ denote user x persuades the server that has logged in legitimately

Predicate Translations 5

Create your own predicates.

Problem

An attacker can persuade a server that a successful login has occurred, even if it hasnt.

Define

1. Let $P(x)$ denote user x logs in legitimately.
2. Let $Q(x)$ denote user x persuades the server that has logged in legitimately

$(\exists x((\neg P(x)) \wedge Q(x))$

Predicate Translations 6

Create your own predicates.

Problem

An attacker can overwrite someone elses credentials on the server.

Predicate Translations 6

Create your own predicates.

Problem

An attacker can overwrite someone elses credentials on the server.

1. Let $P(x, y)$ denote user x can overwrite user y's credentials on the server.
2. Let $Q(x, y)$ denote for users x, y that $x \neq y$

Predicate Translations 6

Create your own predicates.

Problem

An attacker can overwrite someone elses credentials on the server.

1. Let $P(x, y)$ denote user x can overwrite user y's credentials on the server.
2. Let $Q(x, y)$ denote for users x, y that $x \neq y$

$(\exists x(\exists y(P(x, y)) \wedge Q(x, y)))$

(Basic) New Rules for Natural Deduction

- $(\forall -)$ If $\Sigma \vdash \forall x A(x)$, then $\Sigma \vdash A(t)$
  - t is some term, and replaces *all* occurrences of x in $A(x)$.
- $(\forall +)$ If $\Sigma \vdash A(u)$ where u does not occur in Σ, then $\Sigma \vdash \forall x A(x)$.
  - x must not occur in $A(u)$!

(Basic) New Rules for Natural Deduction II

- $(\exists -)$ If $\Sigma, A(u) \vdash B$ and u does not occur in Σ or B, then $\Sigma, \exists x A(x) \vdash B$.
- $(\exists +)$ If $\Sigma \vdash A(t)$ then $\Sigma \vdash \exists x A(x)$ where $A(x)$ results by replacing some (not necessarily all) occurrences of t in $A(t)$ by x.

(Basic) New Rules for Natural Deduction III

- $(\approx -)$ If $\Sigma \vdash A(t_1)$ and $t_1 \approx t_2$ where $A(t_2)$ results from $A(t_1)$ by replacing some (but not necessarily all) occurrences of t_1 in $A(t_1)$ by t_2.
- $(\approx +) \vdash u \approx u$.

(Extended) New Rules for Natural Deduction

- $(\forall -)$ If $\Sigma \vdash \forall x_1, \ldots, x_n A(x_1, \ldots, x_n)$, then $\Sigma \vdash A(t_1, \ldots, t_n)$
  - t is some term, and replaces *all* occurrences of x in $A(x)$.
- $(\forall +)$ If $\Sigma \vdash A(u_1, \ldots, u_n)$ where $u_1, \ldots, u_n$ do not occur in Σ, then $\Sigma \vdash \forall x_1, \ldots, x_n A(x_1, \ldots, x_n)$.
  - u_i should be distinct!

(Extended) New Rules for Natural Deduction II

- $(\exists -)$ If $\Sigma, A(u_1, \ldots, u_n) \vdash B$ where $u_1, \ldots, u_n$ do not occur in Σ or B, then $\Sigma, \exists x_1, \ldots, x_n A(x_1, \ldots, x_n) \vdash B$.
  - u_i should be distinct!
- $(\exists +)$ If $\Sigma \vdash A(t_1, \ldots, t_n)$ then $\Sigma \vdash \exists x_1, \ldots, x_n A(x_1, \ldots, x_n)$ where $A(x_1, \ldots, x_n)$ results from simultaneously replacing some (not necessarily all) occurrences of t_i in $A(t_1, \ldots, t_n)$ by x_i for $i \in [1..n]$.

Plan

The end

Thats it folks. Feel free to hang out and ask questions.

These slides are based off of the tutorial notes and lecture slides provided to you online.
If you want a copy feel free to email me. The are also available on my personal website joe-scott.net

IA Email:

IA	email
Jan Gorzny	jgorzny
Joe Scott	j29scott

Jan and Joe have an office hour Mondays at 3pm in the Tutorial Center in MC.

Instructor Office Hours:

Instructor	Time	Room	Email
Trefler	Tue, Thur 4:00pm	DC 2336	trefler
Rahkooy	Tue, Thur 4:00pm	DC 2302B	hamid.rahkooy