

(1)

First order resolution

- 1) Reduction of FOL to clausal form
- 2) Dealing with \exists s : Skolemisation
- 3) FO resolution rule : resolution + unification

1)

$$\alpha \supset \beta \Rightarrow \neg \alpha \vee \beta$$

$$\alpha \equiv \beta \Rightarrow (\alpha \supset \beta) \wedge (\beta \supset \alpha)$$

$$\neg \neg \alpha \Rightarrow \alpha$$

$$\neg(\alpha \wedge \beta) \Rightarrow \neg \alpha \vee \neg \beta$$

$$\neg(\alpha \vee \beta) \Rightarrow \neg \alpha \wedge \neg \beta$$

TBC \Rightarrow

1) continued

(2)

$$\neg \forall x \alpha \Rightarrow \exists x \neg \alpha$$

$$\neg \exists x \alpha \Rightarrow \forall x \neg \alpha$$

make sure each quantifier has a
distinct variable (new, not in the formula)

$$\forall x P(x) \wedge \exists x R(x) \Rightarrow \forall x P(x) \wedge \exists y R(y)$$

$$\exists x (R(x) \wedge S(y)) \neq \exists y (R(y) \wedge S(y))$$

- get rid of $\exists s$ (later today)
- push $\forall s$ outside the clause

$$\forall x P(x) \vee \forall y R(y)$$

$$\alpha \vee \forall x \beta \Rightarrow \forall x (\alpha \vee \beta)$$

(3)

if α does not contain x free

$$\alpha \wedge \forall x \beta \Rightarrow \forall x (\alpha \wedge \beta)$$

if α does not contain x free

- make clauses

$$\forall x \forall y \forall z \alpha \quad \alpha \text{ only has } \vee, \wedge, \neg$$

use distributivity to make α clauses:

$$\alpha \wedge (\beta \vee \gamma) \Rightarrow (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$$

$$\alpha \vee (\beta \wedge \gamma) \Rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

After α is in CNF, make clauses

$$\forall x \forall y (R(x) \wedge (S(y) \vee R(y))) \Rightarrow [R(x)], [S(y), R(y)]$$

$$\forall x (\underline{\text{Member}(x) \wedge \neg \text{Skier}(x)} \supset \text{Climber}(x))$$
$$\forall x (\neg (\text{Member}(x) \wedge \neg \text{Skier}(x)) \vee \text{Climber}(x))$$
$$\forall x (\neg \text{Member}(x) \vee \neg \neg \text{Skier}(x) \vee \text{Climber}(x))$$
$$\forall x (\neg \text{Member}(x) \vee \text{Skier}(x) \vee \text{Climber}(x))$$
$$[\neg \text{Member}(x), \text{Skier}(x), \underline{\text{Climber}(x)}]$$

(4)

$$\forall x (\text{Dog}(x) \supset \forall y (\text{Fruit}(y) \supset \text{Eats}(x,y))) \quad (5)$$

$$\begin{aligned} & \forall x (\neg \text{Dog}(x) \vee \underline{\forall y (\neg \text{Fruit}(y) \vee \text{Eats}(x,y))}) \\ & \quad \alpha \vee \forall_x \beta \quad \frac{x \text{ not in } \alpha}{\Rightarrow \forall_x (\alpha \vee \beta)} \end{aligned}$$

$$\begin{aligned} & \forall x \forall y (\neg \text{Dog}(x) \vee \neg \text{Fruit}(y) \vee \text{Eats}(x,y)) \\ & \quad [\neg \text{Dog}(x), \neg \text{Fruit}(y), \text{Eats}(x,y)] \end{aligned}$$

2) Getting rid of \exists

(6)

$$\exists x (\text{Dog}(x) \wedge \text{Likes}(x, \text{fish}))$$

$$\forall x \exists y \text{LivesAt}(x, y)$$

Skolemisation

not equivalent rewriting

if S' satisfiable, then (if and only if)

Skolemisation of S is satisfiable

So if $S \cup T_d$ unsatisfiable, then its Skolemisation is, and vice versa

$\exists x \varphi \Rightarrow \varphi(c) \quad c \text{ is a new constant}$ (7)

if $\mathcal{J} = (D, I)$ makes $\exists x \varphi$ true
exist $d \in D$ which makes φ true
make $I(c) = d$
 $\varphi(c)$ is true

If \mathcal{J} makes $\varphi(c)$ true, then in the same interpretation $\exists x \varphi(x)$ is true

(8)

$$\forall x_1 \dots \forall x_2 \dots \forall x_n - \exists x \varrho(x, x_1, \dots, x_n)$$

$$\forall x_1 \exists x \text{ LivesAt}(x_1, x)$$

there exist a function which
for every x_1 returns (possibly
different) value for x

$$\forall x_1 \text{ LivesAt}(x_1, f(x_1))$$

↑
a new
Skolem function

$$\forall x_1 \dots \forall x_2 \dots \forall x_n \varrho(f(x_1, \dots, x_n), x_1, \dots, x_n)$$

Last step in reduction to clausal form

$$\forall_x P(x) \vee \forall_y P(y)$$

$$\forall_x (P(x) \vee \forall_y P(y))$$

$$\forall_x \forall_y (P(x) \vee P(y))$$

$$[P(x), P(y)] \Rightarrow [P(x)]$$

Collect terms:
 $[P(x), P(x)] \Rightarrow [P(x)]$

Factoring : if $P(x)$ and $P(y)$ are substitution instances of each other, drop one of them