CS:4980 Topics in Computer Science II

Introduction to Automated Reasoning

Proof systems for First-order Logic

Cesare Tinelli

Spring 2024

The University of Iowa

Credits

These slides are based on slides originally developed by **Cesare Tinelli** at the University of Iowa, and by **Clark Barrett**, **Caroline Trippel**, and **Andrew (Haoze) Wu** at Stanford University. Adapted by permission.

Outline

- Semantic arguments for FOL
- PCNF (ML 9.2) and Clausal Form
- First-order Resolution (ML 10)

Proofs in first-order logic

Proof systems for FOL are usually extensions of those for PL

For example, we can extend the semantic arguments system by

Proofs in first-order logic

Proof systems for FOL are usually extensions of those for PL

For example, we can extend the semantic arguments system by

- replacing the truth assignment v with an interpretation $\mathcal{I}$ and
- adding proof rules for quantifiers
- adding proof rules for equality (for FOL with equality)

Proofs in first-order logic

Proof systems for FOL are usually extensions of those for PL

For example, we can extend the semantic arguments system by

- replacing the truth assignment v with an interpretation $\mathcal{I}$ and
- adding proof rules for quantifiers
- adding proof rules for equality (for FOL with equality)

Proofs in first-order logic

Proof systems for FOL are usually extensions of those for PL

For example, we can extend the semantic arguments system by

- replacing the truth assignment v with an interpretation $\mathcal{I}$ and
- adding proof rules for quantifiers
- adding proof rules for equality (for FOL with equality)

Proofs in first-order logic

Proof systems for FOL are usually extensions of those for PL

For example, we can extend the semantic arguments system by

- replacing the truth assignment v with an interpretation $\mathcal{I}$ and
- adding proof rules for quantifiers
- adding proof rules for equality (for FOL with equality)

Semantic arguments for FOL: propositional rules

(a) $$\frac{\mathcal{I} \models \neg\alpha}{\mathcal{I} \not\models \alpha}$$

(b) $$\frac{\mathcal{I} \not\models \neg\alpha}{\mathcal{I} \models \alpha}$$

(c) $$\frac{\mathcal{I} \models \alpha \wedge \beta}{\mathcal{I} \models \alpha, \mathcal{I} \models \beta}$$

(d) $$\frac{\mathcal{I} \not\models \alpha \wedge \beta}{\mathcal{I} \not\models \alpha \mid \mathcal{I} \not\models \beta}$$

(e) $$\frac{\mathcal{I} \models \alpha \vee \beta}{\mathcal{I} \models \alpha \mid \mathcal{I} \models \beta}$$

(f) $$\frac{\mathcal{I} \not\models \alpha \vee \beta}{\mathcal{I} \not\models \alpha, \mathcal{I} \not\models \beta}$$

(g) $$\frac{\mathcal{I} \models \alpha \Rightarrow \beta}{\mathcal{I} \not\models \alpha \mid \mathcal{I} \models \beta}$$

(h) $$\frac{\mathcal{I} \not\models \alpha \Rightarrow \beta}{\mathcal{I} \models \alpha, \mathcal{I} \not\models \beta}$$

(i) $$\frac{\mathcal{I} \models \alpha \quad \mathcal{I} \not\models \alpha}{\mathcal{I} \models \bot}$$

(k) $$\frac{\mathcal{I} \models \alpha \Leftrightarrow \beta}{\mathcal{I} \models \alpha, \mathcal{I} \models \beta \mid \mathcal{I} \not\models \alpha, \mathcal{I} \not\models \beta}$$

(j) $$\frac{\mathcal{I} \not\models \alpha \Leftrightarrow \beta}{\mathcal{I} \not\models \alpha, \mathcal{I} \models \beta \mid v \models \alpha, \mathcal{I} \not\models \beta}$$

Semantic arguments for FOL: quantifier rules

Notation: if v is a variable, ε is a term/formula, and t is a term, $\varepsilon[v \leftarrow t]$ denotes the term/formula obtained from ε by replacing every free occurrence of v in ε by t

Semantic arguments for FOL: quantifier rules

Notation: if v is a variable, ε is a term/formula, and t is a term, $\varepsilon[v \leftarrow t]$ denotes the term/formula obtained from ε by replacing every free occurrence of v in ε by t

Examples:

$$
\begin{array}{rclcrcl}
x[x \leftarrow \mathsf{S}(y)] & = & \mathsf{S}(y) & \qquad & (x + y)[x \leftarrow y] & = & y + y \\
x[x \leftarrow \mathsf{S}(x)] & = & \mathsf{S}(x) & \qquad & (x \doteq y)[x \leftarrow 0] & = & 0 \doteq y \\
x[x \leftarrow y] & = & y & \qquad & (x \doteq x)[x \leftarrow \mathsf{S}(x)] & = & \mathsf{S}(x) \doteq \mathsf{S}(x)
\end{array}
$$

$$
\begin{array}{rcl}
(x \doteq y \ \vee \ x < y)[x \leftarrow \mathsf{S}(0)] & = & \mathsf{S}(0) \doteq y \ \vee \ \mathsf{S}(0) < y \\
(x \doteq y \ \vee \ \forall x.\, x < y)[x \leftarrow \mathsf{S}(y)] & = & \mathsf{S}(y) \doteq y \ \vee \ \forall x.\, x < y
\end{array}
$$

Semantic arguments for FOL: quantifier rules

Notation: if v is a variable, ε is a term/formula, and t is a term, $\varepsilon[v \leftarrow t]$ denotes the term/formula obtained from ε by replacing every free occurrence of v in ε by t

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{\,\mathsf{A}\,\}$, $\Sigma^F = \{\,P\,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool}\rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, P \,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{ A \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle A, \mathrm{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \perp$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, P \,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 1

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, P \,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, P(x) \Rightarrow \exists y. P(y)$ is valid

$$\text{(m)}\ \frac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]} \text{ for any term } t \text{ of sort } \sigma$$

$$\text{(n)}\ \frac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]} \text{ for any term } t \text{ of sort } \sigma$$

$$\text{(o)}\ \frac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]} \text{ for a fresh variable } k \text{ of sort } \sigma$$

$$\text{(p)}\ \frac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]} \text{ for a fresh variable } k \text{ of sort } \sigma$$

1. $\mathcal{I} \not\models \exists x.\, P(x) \Rightarrow \exists y.\, P(y)$
2. $\mathcal{I} \models \exists x.\, P(x)$ by (h) on 1
3. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 1
4. $\mathcal{I} \models P(x_0)$ by (o) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 3
6. $\mathcal{I} \models \bot$ by (i) on 4, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{\,\mathsf{A}\,\}$, $\Sigma^F = \{\,P\,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y.\, P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \bot$ by (i) on 3, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$ is valid

$$\text{(m)}\ \frac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]} \text{ for any term } t \text{ of sort } \sigma$$

$$\text{(n)}\ \frac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]} \text{ for any term } t \text{ of sort } \sigma$$

$$\text{(o)}\ \frac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]} \text{ for a fresh variable } k \text{ of sort } \sigma$$

$$\text{(p)}\ \frac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]} \text{ for a fresh variable } k \text{ of sort } \sigma$$

1. $\mathcal{I} \not\models \forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y.\, P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \bot$ by (i) on 3, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, P \,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$ is valid

(m) $$\frac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$$ for any term t of sort σ

(n) $$\frac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$$ for any term t of sort σ

(o) $$\frac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$$ for a fresh variable k of sort σ

(p) $$\frac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y.\, P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y.\, P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \bot$ by (i) on 3, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \forall x.\, (P(x) \Rightarrow \exists y.\, P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y.\, P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y. P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \bot$ by (i) on 3, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ P \}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\forall x. \, (P(x) \Rightarrow \exists y. \, P(y))$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma. \, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma. \, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma. \, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma. \, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \forall x. \, (P(x) \Rightarrow \exists y. \, P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y. \, P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y. P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \bot$ by (i) on 3, 5

Proof by deduction: Example 2

Consider signature Σ with $\Sigma^S = \{\,\mathsf{A}\,\}$, $\Sigma^F = \{\,P\,\}$, $\mathrm{rank}(P) = \langle \mathsf{A}, \mathsf{Bool}\rangle$, and all vars of sort A

Prove that $\forall x.\,(P(x) \Rightarrow \exists y.\,P(y))$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\,\alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\,\alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\,\alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\,\alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \forall x.\,(P(x) \Rightarrow \exists y.\,P(y))$
2. $\mathcal{I} \not\models P(x_0) \Rightarrow \exists y.\,P(y)$ by (p) on 1
3. $\mathcal{I} \models P(x_0)$ by (h) on 2
4. $\mathcal{I} \not\models \exists y.P(y)$ by (h) on 2
5. $\mathcal{I} \not\models P(x_0)$ by (n) on 4
6. $\mathcal{I} \models \perp$ by (i) on 3, 5

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, Q \,\}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\forall y.Q(x, y) \Rightarrow \forall y.\exists x.Q(x, y)$
2. $\mathcal{I} \models \exists x.\forall y.Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y.\exists x.Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y.Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x.Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ Q \}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x. \forall y. Q(x, y) \Rightarrow \forall y. \exists x. Q(x, y)$
2. $\mathcal{I} \models \exists x. \forall y. Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y. \exists x. Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y. Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x. Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{\, \mathsf{A} \,\}$, $\Sigma^F = \{\, Q \,\}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\forall y.Q(x, y) \Rightarrow \forall y.\exists x.Q(x, y)$
2. $\mathcal{I} \models \exists x.\forall y.Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y.\exists x.Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y.Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x.Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{ A \}$, $\Sigma^F = \{ Q \}$, $\mathrm{rank}(Q) = \langle A, A, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\forall y.Q(x, y) \Rightarrow \forall y.\exists x.Q(x, y)$
2. $\mathcal{I} \models \exists x.\forall y.Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y.\exists x.Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y.Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x.Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ Q \}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

$$\text{(m)}\ \frac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}\ \text{for any term } t \text{ of sort } \sigma$$

$$\text{(n)}\ \frac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}\ \text{for any term } t \text{ of sort } \sigma$$

$$\text{(o)}\ \frac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}\ \text{for a fresh variable } k \text{ of sort } \sigma$$

$$\text{(p)}\ \frac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}\ \text{for a fresh variable } k \text{ of sort } \sigma$$

1. $\mathcal{I} \not\models \exists x.\forall y.Q(x, y) \Rightarrow \forall y.\exists x.Q(x, y)$
2. $\mathcal{I} \models \exists x.\forall y.Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y.\exists x.Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y.Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x.Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ Q \}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x,y) \Rightarrow \forall y.\, \exists x.\, Q(x,y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x. \forall y. Q(x,y) \Rightarrow \forall y. \exists x. Q(x,y)$
2. $\mathcal{I} \models \exists x. \forall y. Q(x,y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y. \exists x. Q(x,y)$ by (h) on 1
4. $\mathcal{I} \models \forall y. Q(x_0,y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x. Q(x,y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0,y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0,y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{ \mathsf{A} \}$, $\Sigma^F = \{ Q \}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool} \rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x, y) \Rightarrow \forall y.\, \exists x.\, Q(x, y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x. \forall y. Q(x, y) \Rightarrow \forall y. \exists x. Q(x, y)$
2. $\mathcal{I} \models \exists x. \forall y. Q(x, y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y. \exists x. Q(x, y)$ by (h) on 1
4. $\mathcal{I} \models \forall y. Q(x_0, y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x. Q(x, y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0, y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0, y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Proof by deduction: Example 3

Consider signature Σ with $\Sigma^S = \{\,\mathsf{A}\,\}$, $\Sigma^F = \{\,Q\,\}$, $\mathrm{rank}(Q) = \langle \mathsf{A}, \mathsf{A}, \mathsf{Bool}\rangle$, and all vars of sort A

Prove that $\exists x.\, \forall y.\, Q(x,y) \Rightarrow \forall y.\, \exists x.\, Q(x,y)$ is valid

(m) $\dfrac{\mathcal{I} \models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow t]}$ for any term t of sort σ

(n) $\dfrac{\mathcal{I} \not\models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow t]}$ for any term t of sort σ

(o) $\dfrac{\mathcal{I} \models \exists v{:}\sigma.\, \alpha}{\mathcal{I} \models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

(p) $\dfrac{\mathcal{I} \not\models \forall v{:}\sigma.\, \alpha}{\mathcal{I} \not\models \alpha[v \leftarrow k]}$ for a fresh variable k of sort σ

1. $\mathcal{I} \not\models \exists x.\forall y.Q(x,y) \Rightarrow \forall y.\exists x.Q(x,y)$
2. $\mathcal{I} \models \exists x.\forall y.Q(x,y)$ by (h) on 1
3. $\mathcal{I} \not\models \forall y.\exists x.Q(x,y)$ by (h) on 1
4. $\mathcal{I} \models \forall y.Q(x_0,y)$ by (o) on 2
5. $\mathcal{I} \not\models \exists x.Q(x,y_0)$ by (p) on 3
6. $\mathcal{I} \models Q(x_0,y_0)$ by (m) on 4
7. $\mathcal{I} \not\models Q(x_0,y_0)$ by (n) on 5
8. $\mathcal{I} \models \bot$ by (i) on 6,7

Refutation Soundness and Completeness

Theorem 1 (Soundness)

For all Σ-formulas α, if there is a closed derivation tree with root $\mathcal{I} \not\models \alpha$ then α is valid

Theorem 2 (Completeness)

For all Σ-formulas α without equality, if α is valid, then there is a closed derivation tree with root $\mathcal{I} \not\models \alpha$

Refutation Soundness and Completeness

Theorem 1 (Soundness)
For all Σ-formulas α, if there is a closed derivation tree with root $\mathcal{I} \not\models \alpha$ then α is valid

Theorem 2 (Completeness)
For all Σ-formulas α without equality, if α is valid, then there is a closed derivation tree with root $\mathcal{I} \not\models \alpha$

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. …

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. $\ldots$

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. $\ldots$

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. $\ldots$

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. $\ldots$

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. $\ldots$

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. …

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x. q(x, x)$
2. $\mathcal{I} \not\models q(x_0, x_0)$ by (m) on 1
3. $\mathcal{I} \not\models q(x_1, x_1)$ by (m) on 1
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. …

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Termination?

Does the semantic argument method describe a decision procedure then?

No, for an invalid formula, the semantic argument proof system might not terminate

Intuition: Consider the invalid formula $\forall x.\, q(x, x)$

1. $\mathcal{I} \not\models \forall x$
2. $\mathcal{I} \not\models q($
3. $\mathcal{I} \not\models q($
4. $\mathcal{I} \not\models q(x_2, x_2)$ by (m) on 1
5. …

FOL is only *semi-decidable*: you can always show validity algorithmically but not invalidity

There is no strategy that guarantees termination in all cases of invalid formulas

This shortcoming is not specific to this proof system

Prenex Normal Form (PNF)

For AR purposes, it is useful in FOL too impose syntactic restrictions on formulas

A Σ-formula α is in *prenex normal form* (PNF) if it has the form

$$Q_1 x_1. \cdots Q_n x_n. \beta$$

where each Q_i is a quantifier and β is a quantifier-free formula

Formula α above is in *prenex conjunctive normal form* (PCNF) if, in addition, β is in conjunctive normal form[1]

Example: The formula below is in PCNF

$$\forall y.\, \exists z.\, (\overbrace{(\underbrace{p(f(y))}_{A_1} \lor \underbrace{q(z)}_{A_2})}^{C_1} \land \overbrace{(\underbrace{\neg q(z)}_{A_2} \lor \underbrace{q(x)}_{A_3})}^{C_2})$$

[1] If we treat every atomic formula of β as if it was a propositional variable

Prenex Normal Form (PNF)

A Σ-formula α is in *prenex normal form* (PNF) if it has the form

$$Q_1 x_1. \cdots Q_n x_n. \beta$$

where each Q_i is a quantifier and β is a quantifier-free formula

Formula α above is in *prenex conjunctive normal form* (PCNF) if, in addition, β is in conjunctive normal form[1]

Example: The formula below is in PCNF

$$\forall y. \exists z. \Big(\overbrace{\big(\underbrace{p(f(y))}_{A_1} \vee \underbrace{q(z)}_{A_2} \big)}^{C_1} \wedge \overbrace{\big(\underbrace{\neg q(z)}_{A_2} \vee \underbrace{q(x)}_{A_3} \big)}^{C_2} \Big)$$

[1] If we treat every atomic formula of β as if it was a propositional variable

Prenex Normal Form (PNF)

A Σ-formula α is in *prenex normal form* (PNF) if it has the form

$$Q_1 x_1. \cdots Q_n x_n. \beta$$

where each Q_i is a quantifier and β is a quantifier-free formula

Formula α above is in *prenex conjunctive normal form* (PCNF) if, in addition, β is in conjunctive normal form[1]

Example: The formula below is in PCNF

$$\forall y. \exists z. \Big(\overbrace{\big(\underbrace{p(f(y))}_{A_1} \vee \underbrace{q(z)}_{A_2} \big)}^{C_1} \wedge \overbrace{\big(\neg \underbrace{q(z)}_{A_2} \vee \underbrace{q(x)}_{A_3} \big)}^{C_2} \Big)$$

[1] If we treat every atomic formula of β as if it was a propositional variable

Prenex Normal Form (PNF)

A Σ-formula α is in *prenex normal form* (PNF) if it has the form

$$Q_1 x_1. \cdots Q_n x_n. \beta$$

where each Q_i is a quantifier and β is a quantifier-free formula

Formula α above is in *prenex conjunctive normal form* (PCNF) if, in addition, β is in conjunctive normal form[1]

Example: The formula below is in PCNF

$$\forall y. \exists z. (\overbrace{(\underbrace{p(f(y))}_{A_1} \lor \underbrace{q(z)}_{A_2})}^{C_1} \land \overbrace{(\neg \underbrace{q(z)}_{A_2} \lor \underbrace{q(x)}_{A_3})}^{C_2})$$

[1] If we treat every atomic formula of β as if it was a propositional variable

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\ \exists z.\ (p(f(y)) \land \neg q(y,z))$ ✗
- $\forall y.\ \forall z.\ (p(f(y)) \land \neg q(x,z))$ ✗
- $\forall y.\ \forall z.\ (p(f(y)) \land \neg q(y,z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\, \exists z.\, (p(f(y)) \wedge \neg q(y,z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(x,z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(y,z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\ \exists z.\ (p(f(y)) \land \neg q(y, z))$ ✗
- $\forall y.\ \forall z.\ (p(f(y)) \land \neg q(x, z))$ ✗
- $\forall y.\ \forall z.\ (p(f(y)) \land \neg q(y, z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\, \exists z.\, (p(f(y)) \land \neg q(y, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \land \neg q(x, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \land \neg q(y, z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\, \exists z.\, (p(f(y)) \wedge \neg q(y, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(x, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(y, z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\, \exists z.\, (p(f(y)) \land \neg q(y,z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \land \neg q(x,z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \land \neg q(y,z))$ ✓

Clausal Form

A Σ-formula is in *clausal form* if

1. it is in PCNF
2. it is closed (i.e., it has no free variables)
3. all of its quantifiers are universal

Examples: Which of the following formulas are clausal form?

- $\forall y.\, \exists z.\, (p(f(y)) \wedge \neg q(y, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(x, z))$ ✗
- $\forall y.\, \forall z.\, (p(f(y)) \wedge \neg q(y, z))$ ✓

Clausal Form: transformation

Theorem 3 (Skolem's Theorem)
Any sentence can be transformed to an equi-satisfiable formula in clausal form.

The high level transformation strategy is the following:

$$\text{Sentence} \Rightarrow \text{PNF} \Rightarrow \text{PCNF} \Rightarrow \text{Clausal Form}$$

Running example: $(\forall x.(p(x) \Rightarrow q(x))) \Rightarrow (\forall x.p(x) \Rightarrow \forall x.q(x))$

Clausal Form: transformation

Theorem 3 (Skolem's Theorem)
Any sentence can be transformed to an equi-satisfiable formula in clausal form.

The high level transformation strategy is the following:

$$\text{Sentence} \Rightarrow \text{PNF} \Rightarrow \text{PCNF} \Rightarrow \text{Clausal Form}$$

Running example: $(\forall x.(p(x) \Rightarrow q(x))) \Rightarrow (\forall x.p(x) \Rightarrow \forall x.q(x))$

Clausal Form: transformation

Theorem 3 (Skolem's Theorem)

Any sentence can be transformed to an equi-satisfiable formula in clausal form.

The high level transformation strategy is the following:

$$\text{Sentence} \Rightarrow \text{PNF} \Rightarrow \text{PCNF} \Rightarrow \text{Clausal Form}$$

Running example: $(\forall x.(p(x) \Rightarrow q(x))) \Rightarrow (\forall x.p(x) \Rightarrow \forall x.q(x))$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$(\forall x.\,(p(x) \Rightarrow q(x))) \Rightarrow (\forall x.\,p(x) \Rightarrow \forall x.\,q(x))$$

Step 1: Rename the bounded variables apart so that

1. the bounded variables are disjoint from free variables
2. different quantifiers use different bound variables

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$(\forall x.\, (p(x) \Rightarrow q(x))) \Rightarrow (\forall x.\, p(x) \Rightarrow \forall x.\, q(x))$$

Step 1: Rename the bounded variables apart so that

1. the bounded variables are disjoint from free variables
2. different quantifiers use different bound variables

$$(\forall x.\, (p(x) \Rightarrow q(x))) \Rightarrow (\forall x.\, p(x) \Rightarrow \forall x.\, q(x)) \quad \longrightarrow \cdots \longrightarrow$$
$$(\forall x.\, (p(x) \Rightarrow q(x))) \Rightarrow (\forall y.\, p(y) \Rightarrow \forall z.\, q(z))$$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$(\forall x.\, (p(x) \Rightarrow q(x))) \Rightarrow (\forall y.\, p(y) \Rightarrow \forall z.\, q(z))$$

Step 2: Eliminate all occurrences of $\Rightarrow$ and $\Leftrightarrow$ using the rewrites:

- $\alpha_1 \Leftrightarrow \alpha_2 \longrightarrow (\alpha_1 \Rightarrow \alpha_2) \wedge (\alpha_2 \Rightarrow \alpha_1)$
- $\alpha_1 \Rightarrow \alpha_2 \longrightarrow \neg\alpha_1 \vee \alpha_2$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$(\forall x.\,(p(x) \Rightarrow q(x))) \Rightarrow (\forall y.\,p(y) \Rightarrow \forall z.\,q(z))$$

Step 2: Eliminate all occurrences of $\Rightarrow$ and $\Leftrightarrow$ using the rewrites:

- $\alpha_1 \Leftrightarrow \alpha_2 \longrightarrow (\alpha_1 \Rightarrow \alpha_2) \wedge (\alpha_2 \Rightarrow \alpha_1)$
- $\alpha_1 \Rightarrow \alpha_2 \longrightarrow \neg\alpha_1 \vee \alpha_2$

$$(\forall x.\,(p(x) \Rightarrow q(x))) \Rightarrow (\forall y.\,p(y) \Rightarrow \forall z.\,q(z)) \quad \longrightarrow \cdots \longrightarrow$$

$$\neg(\forall x.\,(\neg p(x) \vee q(x))) \vee (\neg\forall y.\,p(y) \vee \forall z.\,q(z))$$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$\neg(\forall x.\, (\neg p(x) \vee q(x))) \vee (\neg\forall y.\, p(y) \vee \forall z.\, q(z))$$

Step 3: Push negations inward as much as possible using the rewrites:

- $\neg(\alpha \wedge \beta) \longrightarrow \neg\alpha \vee \neg\beta$ $\qquad$ $\neg(\alpha \vee \beta) \longrightarrow \neg\alpha \wedge \neg\beta$
- $\neg\forall v.\, \alpha \longrightarrow \exists v.\, \neg\alpha$ $\qquad$ $\neg\exists v.\, \alpha \longrightarrow \forall v.\, \neg\alpha$
- $\neg\neg\alpha \longrightarrow \alpha$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$\neg(\forall x.\, (\neg p(x) \vee q(x))) \vee (\neg\forall y.\, p(y) \vee \forall z.\, q(z))$$

Step 3: Push negations inward as much as possible using the rewrites:

- $\neg(\alpha \wedge \beta) \longrightarrow \neg\alpha \vee \neg\beta$ $\quad$ $\neg(\alpha \vee \beta) \longrightarrow \neg\alpha \wedge \neg\beta$
- $\neg\forall v.\, \alpha \longrightarrow \exists v.\, \neg\alpha$ $\quad$ $\neg\exists v.\, \alpha \longrightarrow \forall v.\, \neg\alpha$
- $\neg\neg\alpha \longrightarrow \alpha$

$$\neg(\forall x.\, (\neg p(x) \vee q(x))) \vee (\neg\forall y.\, p(y) \vee \forall z.\, q(z)) \quad \longrightarrow \cdots \longrightarrow$$
$$\exists x.\, (p(x) \wedge \neg q(x)) \vee (\exists y.\, \neg p(y) \vee \forall z.\, q(z))$$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$\exists x.\, (p(x) \land \neg q(x)) \lor (\exists y.\, \neg p(y) \lor \forall z.\, q(z))$$

Step 4: Move all quantifiers outward (and so leftwards) using the rewrites:

- $\alpha \bowtie Qv.\beta \longrightarrow Qv.(\alpha \bowtie \beta)$ (ok because v does not occur free in α)
- $(Qv.\alpha) \bowtie \beta \longrightarrow Qv.(\alpha \bowtie \beta)$ (ok because v does not occur free in β)

where $Q \in \{\forall, \exists\}$ and $\bowtie \in \{\land, \lor\}$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$\exists x.\, (p(x) \land \neg q(x)) \lor (\exists y.\, \neg p(y) \lor \forall z.\, q(z))$$

Step 4: Move all quantifiers outward (and so leftwards) using the rewrites:

- $\alpha \bowtie Qv.\beta \longrightarrow Qv.(\alpha \bowtie \beta)$ (ok because v does not occur free in α)
- $(Qv.\alpha) \bowtie \beta \longrightarrow Qv.(\alpha \bowtie \beta)$ (ok because v does not occur free in β)

where $Q \in \{\forall, \exists\}$ and $\bowtie \in \{\land, \lor\}$

$$\begin{array}{ll}
\exists x.\, (p(x) \land \neg q(x)) \lor (\exists y.\, \neg p(y) \lor \forall z.\, q(z)) & \longrightarrow \\
\exists x.\, ((p(x) \land \neg q(x)) \lor (\exists y.\, \neg p(y) \lor \forall z.\, q(z))) & \longrightarrow \\
\exists x.\, ((p(x) \land \neg q(x)) \lor \forall z.\, (\exists y.\, \neg p(y) \lor q(z))) & \longrightarrow \\
\exists x.\, \forall z.\, ((p(x) \land \neg q(x)) \lor (\exists y.\, \neg p(y) \lor q(z))) & \longrightarrow \cdots \longrightarrow \\
\exists x.\, \forall z.\, \exists y.\, ((p(x) \land \neg q(x)) \lor (\neg p(y) \lor q(z))) &
\end{array}$$

I: Transforming into PNF

Any sentence can be transformed into a logically equivalent formula in PNF in 4 steps

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \land \neg q(x)) \lor (\neg p(y) \lor q(z)))$$

II: Transforming into PCNF

Transforming a PNF to a logically equivalent PCNF is straightforward

We apply the distributive laws from propositional logic

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \land \neg q(x)) \lor (\neg p(y) \lor q(z)))$$

becomes

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \lor \neg p(y) \lor q(z)) \land (\neg q(x) \lor \neg p(y) \lor q(z)))$$

This formula contains existentials and is therefore not yet in clausal form

II: Transforming into PCNF

Transforming a PNF to a logically equivalent PCNF is straightforward

We apply the distributive laws from propositional logic

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \land \neg q(x)) \lor (\neg p(y) \lor q(z)))$$

becomes

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \lor \neg p(y) \lor q(z)) \land (\neg q(x) \lor \neg p(y) \lor q(z)))$$

This formula contains existentials and is therefore not yet in clausal form

II: Transforming into PCNF

Transforming a PNF to a logically equivalent PCNF is straightforward

We apply the distributive laws from propositional logic

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \wedge \neg q(x)) \vee (\neg p(y) \vee q(z)))$$

becomes

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

This formula contains existentials and is therefore not yet in clausal form

III: Transforming into Clausal Form (Skolemization)

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

For every existential quantifier $\exists v$ in the PCNF, let $u_1, \ldots, u_n$ be the universally quantified variables preceding $\exists v$,

1. introduce a fresh function symbol f_v with arity n and $\langle \mathrm{sort}(u_1), \ldots, \mathrm{sort}(u_n), \mathrm{sort}(v) \rangle$
2. delete $\exists v$ and replace every occurrence of v by $f_v(u_1, \ldots, u_n)$

III: Transforming into Clausal Form (Skolemization)

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

For every existential quantifier $\exists v$ in the PCNF, let $u_1, \ldots, u_n$ be the universally quantified variables preceding $\exists v$,

1. introduce a fresh function symbol f_v with arity n and $\langle \mathrm{sort}(u_1), \ldots \mathrm{sort}(u_n), \mathrm{sort}(v) \rangle$
2. delete $\exists v$ and replace every occurrence of v by $f_v(u_1, \ldots, u_n)$

III: Transforming into Clausal Form (Skolemization)

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

For every existential quantifier $\exists v$ in the PCNF, let $u_1, \ldots, u_n$ be the universally quantified variables preceding $\exists v$,

1. introduce a fresh function symbol f_v with arity n and $\langle \mathrm{sort}(u_1), \ldots \mathrm{sort}(u_n), \mathrm{sort}(v) \rangle$
2. delete $\exists v$ and replace every occurrence of v by $f_v(u_1, \ldots, u_n)$

For the formula above, introduce nullary function (i.e., a constant) symbol f_x and unary function symbol f_y for $\exists x$ and $\exists y$, respectively

$$\forall z.\, ((p(f_x) \vee \neg p(f_y(z)) \vee q(z)) \wedge (\neg q(f_x) \vee \neg p(f_y(z)) \vee q(z)))$$

III: Transforming into Clausal Form (Skolemization)

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

For every existential quantifier $\exists v$ in the PCNF, let $u_1, \ldots, u_n$ be the universally quantified variables preceding $\exists v$,

1. introduce a fresh function symbol f_v with arity n and $\langle \mathrm{sort}(u_1), \ldots \mathrm{sort}(u_n), \mathrm{sort}(v) \rangle$
2. delete $\exists v$ and replace every occurrence of v by $f_v(u_1, \ldots, u_n)$

The functions f_v are called *Skolem functions* and the process of replacing existential quantifiers by functions is called *Skolemization*

III: Transforming into Clausal Form (Skolemization)

$$\exists x.\, \forall z.\, \exists y.\, ((p(x) \vee \neg p(y) \vee q(z)) \wedge (\neg q(x) \vee \neg p(y) \vee q(z)))$$

For every existential quantifier $\exists v$ in the PCNF, let $u_1, \ldots, u_n$ be the universally quantified variables preceding $\exists v$,

1. introduce a fresh function symbol f_v with arity n and $\langle \mathrm{sort}(u_1), \ldots \mathrm{sort}(u_n), \mathrm{sort}(v) \rangle$
2. delete $\exists v$ and replace every occurrence of v by $f_v(u_1, \ldots, u_n)$

The functions f_v are called *Skolem functions* and the process of replacing existential quantifiers by functions is called *Skolemization*

Note: Technically, the resulting formula is no longer a Σ-formula, but a Σ_E-formula, where $\Sigma_E^S = \Sigma^S$ and $\Sigma_E^F = \Sigma^F \cup \bigcup_v \{ f_v \}$

Clausal forms as clause sets

As with propositional logic, we can write a formula in clausal form unambiguously as a set of clauses

Example:

$$\forall z.\, ((p(f(z)) \lor \neg p(g(z)) \lor q(z)) \land (\neg q(f(z)) \lor \neg p(g(z)) \lor q(z))$$

can be written as

$$\Delta := \{\, \{p(f(z)), \neg p(g(z)), q(z)\}, \{\neg q(f(z)), \neg p(g(z)), q(z)\} \,\}$$

Traditionally, theorem provers for FOL use the latter version of the clausal form

Clausal forms as clause sets

As with propositional logic, we can write a formula in clausal form unambiguously as a set of clauses

Example:

$$\forall z.\, ((p(f(z)) \vee \neg p(g(z)) \vee q(z)) \wedge (\neg q(f(z)) \vee \neg p(g(z)) \vee q(z))$$

can be written as

$$\Delta := \{\, \{p(f(z)), \neg p(g(z)), q(z)\}, \{\neg q(f(z)), \neg p(g(z)), q(z)\} \,\}$$

Traditionally, theorem provers for FOL use the latter version of the clausal form

Clausal forms as clause sets

As with propositional logic, we can write a formula in clausal form unambiguously as a set of clauses

Example:

$$\forall z.\, ((p(f(z)) \vee \neg p(g(z)) \vee q(z)) \wedge (\neg q(f(z)) \vee \neg p(g(z)) \vee q(z))$$

can be written as

$$\Delta := \{\, \{p(f(z)), \neg p(g(z)), q(z)\}, \{\neg q(f(z)), \neg p(g(z)), q(z)\}\,\}$$

Traditionally, theorem provers for FOL use the latter version of the clausal form

A resolution-based proof system for PL

Recall: The satisfiability proof system consisting of the rules below is sound, complete and terminating for clause sets in PL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Can we extend this proof system to FOL?

A resolution-based proof system for PL

Recall: The satisfiability proof system consisting of the rules below is sound, complete and terminating for clause sets in PL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Can we extend this proof system to FOL?

A resolution-based proof system for FOL?

Resolve $$\frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

Clash $$\frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}}$$

Unsat $$\frac{\{\} \in \Delta}{\text{UNSAT}}$$

Sat $$\frac{\text{No other rules apply}}{\text{SAT}}$$

Consider the FOL clause set below where x, z are variables and a is a constant symbol

$$\Delta := \{\, \{\neg P(z), Q(z)\}, \{P(a)\}, \{\neg Q(x)\} \,\}$$

Note that Δ is equivalent to $\forall z.\, (P(z) \Rightarrow Q(z)) \wedge P(a) \wedge \forall x.\, \neg Q(x)$, which is unsatisfiable

However, no rules above apply to Δ

We need another rule to deal with variables

A resolution-based proof system for FOL?

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Consider the FOL clause set below where x, z are variables and a is a constant symbol

$$\Delta := \{\, \{\neg P(z), Q(z)\}, \{P(a)\}, \{\neg Q(x)\} \,\}$$

Note that Δ is equivalent to $\forall z.\, (P(z) \Rightarrow Q(z)) \wedge P(a) \wedge \forall x.\, \neg Q(x)$, which is unsatisfiable

However, no rules above apply to Δ

We need another rule to deal with variables

A resolution-based proof system for FOL?

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{C\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{C\} \quad \Phi := \Phi \cup \{C\}} \qquad \textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Consider the FOL clause set below where x, z are variables and a is a constant symbol

$$\Delta := \{\, \{\neg P(z), Q(z)\}, \{P(a)\}, \{\neg Q(x)\} \,\}$$

Note that Δ is equivalent to $\forall z.\, (P(z) \Rightarrow Q(z)) \wedge P(a) \wedge \forall x.\, \neg Q(x)$, which is unsatisfiable

However, no rules above apply to Δ

We need another rule to deal with variables

A resolution-based proof system for FOL?

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Consider the FOL clause set below where x, z are variables and a is a constant symbol

$$\Delta := \{\, \{\neg P(z), Q(z)\}, \{P(a)\}, \{\neg Q(x)\} \,\}$$

Note that Δ is equivalent to $\forall z.\, (P(z) \Rightarrow Q(z)) \wedge P(a) \wedge \forall x.\, \neg Q(x)$, which is unsatisfiable

However, no rules above apply to Δ

We need another rule to deal with variables

A resolution-based proof system for FOL?

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{C\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{C\} \quad \Phi := \Phi \cup \{C\}} \qquad \textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Consider the FOL clause set below where x, z are variables and a is a constant symbol

$$\Delta := \{\{\neg P(z), Q(z)\}, \{P(a)\}, \{\neg Q(x)\}\}$$

Note that Δ is equivalent to $\forall z.\ (P(z) \Rightarrow Q(z)) \wedge P(a) \wedge \forall x.\ \neg Q(x)$, which is unsatisfiable

However, no rules above apply to Δ

We need another rule to deal with variables

A resolution-based proof system for FOL

Resolve $$\frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

Clash $$\frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}}$$

Inst $$\frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

Unsat $$\frac{\{\} \in \Delta}{\text{UNSAT}}$$

Sat $$\frac{\text{No other rules apply}}{\text{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\}$ $C_2 : \{P(a)\}$ $C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3 \,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\} \,\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\} \,\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\} \,\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\} \,\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p\,\}) \cup (C_2 \setminus \{\, \neg p\,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C\,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C\,\} \quad \Phi := \Phi \cup \{\, C\,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\} \quad C_2 : \{P(a)\} \quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3\,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\}\,\}$	by Inst on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\}\,\}$	by Resolve on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\}\,\}$	by Inst on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\}\,\}$	by Resolve on C_5, C_6
	UNSAT	by Unsat on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\}$ $\quad C_2 : \{P(a)\}$ $\quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3 \,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\} \,\}$	by Inst on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\} \,\}$	by Resolve on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\} \,\}$	by Inst on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\} \,\}$	by Resolve on C_5, C_6
	UNSAT	by Unsat on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\} \quad C_2 : \{P(a)\} \quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3 \,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\} \,\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\} \,\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\} \,\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\} \,\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p\,\}) \cup (C_2 \setminus \{\, \neg p\,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C\,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C\,\} \quad \Phi := \Phi \cup \{\, C\,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\}$ $\quad C_2 : \{P(a)\}$ $\quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3\,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\}\,\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\}\,\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\}\,\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\}\,\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\}$ $\quad C_2 : \{P(a)\}$ $\quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3 \,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\} \,\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\} \,\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\} \,\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\} \,\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathrm{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathrm{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\} \quad C_2 : \{P(a)\} \quad C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{\, C_1, C_2, C_3 \,\}$	
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\} \,\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\} \,\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\} \,\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{\, C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\} \,\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{C\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{C\} \quad \Phi := \Phi \cup \{C\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Example: $C_1 : \{\neg P(z), Q(z)\}$ $C_2 : \{P(a)\}$ $C_3 : \{\neg Q(x)\}$

Φ	Δ	
$\{\ \}$	$\{C_1, C_2, C_3\}$	
$\{\ \}$	$\{C_1, C_2, C_3, C_4{:}\{\neg P(a), Q(a)\}\}$	by **Inst** on C_1 with $z \leftarrow a$
$\{\ \}$	$\{C_1, C_2, C_3, C_4, C_5{:}\{Q(a)\}\}$	by **Resolve** on C_2, C_4
$\{\ \}$	$\{C_1, C_2, C_3, C_4, C_5, C_6{:}\{\neg Q(a)\}\}$	by **Inst** on C_3 with $x \leftarrow a$
$\{\ \}$	$\{C_1, C_2, C_3, C_4, C_5, C_6, C_7{:}\{\}\}$	by **Resolve** on C_5, C_6
	UNSAT	by **Unsat** on C_7

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

This system is refutation-sound and complete for FOL clause sets without equality:

- If a clause set Δ_0 is unsatisfiable, there is a derivation of UNSAT from Δ_0

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

This system is refutation-sound and complete for FOL clause sets without equality:

- If a clause set Δ_0 is unsatisfiable, there is a derivation of UNSAT from Δ_0

The system is also solution-sound:

- There is a derivation of SAT from Δ_0 only if Δ_0 is satisfiable

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

This system is refutation-sound and complete for FOL clause sets without equality:

- If a clause set Δ_0 is unsatisfiable, there is a derivation of UNSAT from Δ_0

The system is not, and cannot be, terminating:

- if Δ_0 is satisfiable, it is possible for **Sat** to never apply

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Note: This proof system is challenging to implement efficiently because **Inst** is not constrained enough

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Automated theorem provers for FOL use instead a more sophisticated **Resolve** rule where two literals in different clauses are instantiated directly, and only as needed, to make them complementary (see ML Chap. 10)

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Automated theorem provers for FOL use instead a more sophisticated **Resolve** rule

where two literals in different clauses are instantiated directly, and only as needed, to make them complementary (see ML Chap. 10)

Example: $\{P(x, y), Q(a, f(y))\}$, $\{\neg Q(z, f(b)), R(g(z))\}$ resolve to $\{P(x, b), R(g(a))\}$

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Problem: How do we prove the unsatisfiability of these clause sets?

$$\{\, \{x \doteq y\}, \{\neg(y \doteq x)\} \,\} \quad \{\, \{x \doteq y\}, \{y \doteq z\}, \{\neg(x \doteq z)\} \,\} \quad \{\, \{x \doteq y\}, \{\neg(f(x) \doteq f(y))\} \,\}$$

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Problem: How do we prove the unsatisfiability of these clause sets?

$$\{\, \{x \doteq y\}, \{\neg(y \doteq x)\} \,\} \quad \{\, \{x \doteq y\}, \{y \doteq z\}, \{\neg(x \doteq z)\} \,\} \quad \{\, \{x \doteq y\}, \{\neg(f(x) \doteq f(y))\} \,\}$$

We need specialized rules for equality reasoning!

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\mathsf{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\mathsf{SAT}}$$

Another Problem: How to we prove the unsatisfiability of these clause sets?

$$\{\,\{x < x\}\,\} \quad \{\,\{x < y\}, \{y < z\}, \{\neg(x < z)\}\,\} \quad \{\,\{\neg(x{+}y \doteq y{+}x)\}\,\} \quad \{\,\{\neg(x{+}0 \doteq x)\}\,\}$$

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Another Problem: How to we prove the unsatisfiability of these clause sets?

$$\{\, \{x < x\} \,\} \quad \{\, \{x < y\}, \{y < z\}, \{\neg(x < z)\} \,\} \quad \{\, \{\neg(x+y \doteq y+x)\} \,\} \quad \{\, \{\neg(x+0 \doteq x)\} \,\}$$

The thing is: each of these clause set is actually satisfiable in FOL!

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{\, p \,\}) \cup (C_2 \setminus \{\, \neg p \,\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{\, C \,\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{\, C \,\} \quad \Phi := \Phi \cup \{\, C \,\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Another Problem: How to we prove the unsatisfiability of these clause sets?

$$\{\, \{x < x\} \,\} \quad \{\, \{x < y\}, \{y < z\}, \{\neg(x < z)\} \,\} \quad \{\, \{\neg(x+y \doteq y+x)\} \,\} \quad \{\, \{\neg(x+0 \doteq x)\} \,\}$$

The thing is: each of these clause set is actually satisfiable in FOL!

However, they are unsatisfiable in the theory of arithmetic

A resolution-based proof system for FOL

$$\textsc{Resolve}\ \frac{C_1, C_2 \in \Delta \quad p \in C_1 \quad \neg p \in C_2 \quad C = (C_1 \setminus \{p\}) \cup (C_2 \setminus \{\neg p\}) \quad C \notin \Delta \cup \Phi}{\Delta := \Delta \cup \{C\}}$$

$$\textsc{Clash}\ \frac{C \in \Delta \quad p, \neg p \in C}{\Delta := \Delta \setminus \{C\} \quad \Phi := \Phi \cup \{C\}} \qquad \textsc{Inst}\ \frac{C \in \Delta \quad v \in \mathcal{FV}(C) \quad \mathrm{sort}(t) = \mathrm{sort}(v)}{\Delta := \Delta \cup \{C[v \leftarrow t]\}}$$

$$\textsc{Unsat}\ \frac{\{\} \in \Delta}{\text{UNSAT}} \qquad \textsc{Sat}\ \frac{\text{No other rules apply}}{\text{SAT}}$$

Another Problem: How to we prove the unsatisfiability of these clause sets?

$$\{\{x < x\}\} \quad \{\{x < y\}, \{y < z\}, \{\neg(x < z)\}\} \quad \{\{\neg(x+y \doteq y+x)\}\} \quad \{\{\neg(x+0 \doteq x)\}\}$$

We need proof systems for satisfiability modulo theories