Resolución

Pedro Cabalar

Lógica
Grado en Inteligencia Artificial
Universidade da Coruña

1 de mayo de 2023

Resolución

- Permite decidir si una fórmula α es inconsistente.

Resolución

- Permite decidir si una fórmula α es inconsistente.
- Como primer paso, require transformar α en su Forma Normal Conjuntiva (CNF) equivalente, $\mathit{CNF}[\alpha]$.

Resolución

- La regla de resolución es una regla de inferencia definida como sigue:

$$\frac{\alpha \vee p \qquad\qquad \beta \vee \neg p}{\alpha \vee \beta}$$

La cláusula resultante $\alpha \vee \beta$ se denomina resolvente. Una disyunción vacía (o cláusula vacía) se corresponde con $\bot$.

Resolución

- La regla de resolución es una regla de inferencia definida como sigue:

$$\frac{\alpha \vee p \qquad\qquad \beta \vee \neg p}{\alpha \vee \beta}$$

  La cláusula resultante $\alpha \vee \beta$ se denomina resolvente. Una disyunción vacía (o cláusula vacía) se corresponde con $\bot$.

- Resultado principal [Robinson65]: α is insatisfactible sii existe una derivación de $\bot$ aplicando resolución a partir de $CNF[\alpha]$.

Resolución

- Ejemplo: probar que

$$\alpha : (\neg p \to q) \wedge (q \to r) \to (\neg r \to p)$$

es una tautología

Resolución

- Ejemplo: probar que

$$\alpha : (\neg p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (\neg r \rightarrow p)$$

  es una tautología

- Primero negamos la fórmula $\neg\alpha$ y luego obtenemos su CNF:

$$\begin{array}{rl} & \neg((\neg p \rightarrow q) \wedge (q \rightarrow r) \rightarrow (\neg r \rightarrow p)) \\ \equiv & \neg(\neg((\neg\neg p \vee q) \wedge (\neg q \vee r)) \vee (\neg\neg r \vee p)) \\ \equiv & ((p \vee q) \wedge (\neg q \vee r)) \wedge \neg(r \vee p) \\ \equiv & \underbrace{(p \vee q)}_{C_1} \wedge \underbrace{(\neg q \vee r)}_{C_2} \wedge \underbrace{\neg r}_{C_3} \wedge \underbrace{\neg p}_{C_4} \end{array}$$

Resolución

- Una posible aplicación de resolución sería:

$$
\begin{tikzcd}
C_1 : p \vee q \arrow[dr, dash] & C_2 : \neg q \vee r \arrow[d, dash] & & \\
& C_5 : p \vee r \arrow[dr, dash] & C_3 : \neg r \arrow[d, dash] & \\
& & C_6 : p \arrow[dr, dash] & C_4 : \neg p \arrow[d, dash] \\
& & & C_7 : \perp
\end{tikzcd}
$$

Resolución

Ejemplo: probar con los conjuntos de cláusulas

- $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c,\ \neg c\}$

Resolución

Ejemplo: probar con los conjuntos de cláusulas

- $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c,\ \neg c\}$
- ¿ Se puede probar de más de una forma?

Resolución

Ejemplo: probar con los conjuntos de cláusulas

- $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c,\ \neg c\}$
- ¿ Se puede probar de más de una forma?
- ¿Qué ocurre si tenemos el siguiente conjunto en su lugar? $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c\}$

Resolución

Ejemplo: probar con los conjuntos de cláusulas

- $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c,\ \neg c\}$
- ¿ Se puede probar de más de una forma?
- ¿Qué ocurre si tenemos el siguiente conjunto en su lugar? $\{a \vee b,\ \neg a \vee c,\ \neg b \vee c\}$
- Ejemplo: $\{a \vee b,\ \neg b,\ a \vee c,\ \neg a \vee d\}$ ¿es satisfactible?

- Problema: generar todos los posibles resolventes es inviable.

Resolución

- Problema: generar todos los posibles resolventes es inviable.
- Importante: usar heurísticas para explorar sólo una parte del árbol de búsqueda.

Resolución

- Problema: generar todos los posibles resolventes es inviable.
- Importante: usar heurísticas para explorar sólo una parte del árbol de búsqueda.
- Además, a veces se pueden aplicar algunas optimizaciones ...

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$
- Subsunción (subsumption): entendiendo la cláusulas como conjuntos de literales, si tenemos $C_1 \subseteq C_2$ decimos que C_1 subsume a C_2 y podemos eliminar esta última.

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$
- Subsunción (subsumption): entendiendo la cláusulas como conjuntos de literales, si tenemos $C_1 \subseteq C_2$ decimos que C_1 subsume a C_2 y podemos eliminar esta última.
- Literales puros: un literal is puro cuando su complentario no aparece en ninguna cláusula.

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$
- Subsunción (subsumption): entendiendo la cláusulas como conjuntos de literales, si tenemos $C_1 \subseteq C_2$ decimos que C_1 subsume a C_2 y podemos eliminar esta última.
- Literales puros: un literal is puro cuando su complentario no aparece en ninguna cláusula. Un literal puro no afecta a la satisfactibilidad: podemos fijar arbitrariamente un valor de verdad para su átomo asociado.

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$
- Subsunción (subsumption): entendiendo la cláusulas como conjuntos de literales, si tenemos $C_1 \subseteq C_2$ decimos que C_1 subsume a C_2 y podemos eliminar esta última.
- Literales puros: un literal is puro cuando su complentario no aparece en ninguna cláusula. Un literal puro no afecta a la satisfactibilidad: podemos fijar arbitrariamente un valor de verdad para su átomo asociado.
- Propagación unitaria (unit propagation): una cláusula unitaria es aquella que sólo contiene un literal.

Optimizaciones

- Eliminar tautologías: $p \vee \neg p \vee \alpha$
- Subsunción (subsumption): entendiendo la cláusulas como conjuntos de literales, si tenemos $C_1 \subseteq C_2$ decimos que C_1 subsume a C_2 y podemos eliminar esta última.
- Literales puros: un literal is puro cuando su complentario no aparece en ninguna cláusula. Un literal puro no afecta a la satisfactibilidad: podemos fijar arbitrariamente un valor de verdad para su átomo asociado.
- Propagación unitaria (unit propagation): una cláusula unitaria es aquella que sólo contiene un literal. Su átomo tiene un valor de certeza fijo y lo podemos simplificar en el resto de cláusulas.

Resolución

Algoritmos básicos:

- Davis-Putnam: elegir un átomo p y aplicar resolución en él de todos los modos posibles. El conjunto de cláusulas resolventes ya no contienen p (¡uno menos!).

Resolución

Algoritmos básicos:

- Davis-Putnam: elegir un átomo p y aplicar resolución en él de todos los modos posibles. El conjunto de cláusulas resolventes ya no contienen p (¡uno menos!).
- Davis-Putnam-Logemann-Loveland (DPLL): realizar backtracking sobre el valor de un átomo dado (suponerlo primero cierto y, si falla, suponerlo falso). Es utilizado por la mayoría de solucionadores SAT.

Resolución: estrategias

- Resolución Lineal: en cada aplicación, una de las cláusulas debe ser el último resolvente obtenido.

Resolución: estrategias

- Resolución Lineal: en cada aplicación, una de las cláusulas debe ser el último resolvente obtenido.
- Si α es insatisfactible, existe alguna cláusula C en $CNF(\alpha)$ tal que se puede derivar $\perp$ a partir de C usando exclusivamente resolución lineal.

Resolución: estrategias

- Resolución Lineal: en cada aplicación, una de las cláusulas debe ser el último resolvente obtenido.
- Si α es insatisfactible, existe alguna cláusula C en $CNF(\alpha)$ tal que se puede derivar $\perp$ a partir de C usando exclusivamente resolución lineal.
- Resolución de Entrada (input): se usa siempre una de las cláusulas de entrada originales que aparecen en $CNF(\alpha)$ (es un método incompleto).

Resolución: estrategias

- Resolución Lineal: en cada aplicación, una de las cláusulas debe ser el último resolvente obtenido.
- Si α es insatisfactible, existe alguna cláusula C en $CNF(\alpha)$ tal que se puede derivar $\perp$ a partir de C usando exclusivamente resolución lineal.
- Resolución de Entrada (input): se usa siempre una de las cláusulas de entrada originales que aparecen en $CNF(\alpha)$ (es un método incompleto).
- Resolución unitaria: se usa siempre una cláusula unitaria (es un método incompleto).

Resolución: estrategias

- Conjunto de soporte: detectar algún $S \subset CNF(\alpha)$ para el que sepamos que $CNF(\alpha) \setminus S$ es satisfactible.
- Ejemplo típico: queremos saber si γ se sigue de β, sabiendo que β es consistente. Esto es: ¿ $\beta \wedge \neg\gamma$ inconsistente?

Resolución: estrategias

- Conjunto de soporte: detectar algún $S \subset CNF(\alpha)$ para el que sepamos que $CNF(\alpha) \setminus S$ es satisfactible.
- Ejemplo típico: queremos saber si γ se sigue de β, sabiendo que β es consistente. Esto es: ¿ $\beta \wedge \neg\gamma$ inconsistente?
  Cuando lo negamos, nos queda $CNF(\beta) \cup \underbrace{CNF(\neg\gamma)}_{\text{support}}$.

Resolución: estrategias

- Conjunto de soporte: detectar algún $S \subset CNF(\alpha)$ para el que sepamos que $CNF(\alpha) \setminus S$ es satisfactible.
- Ejemplo típico: queremos saber si γ se sigue de β, sabiendo que β es consistente. Esto es: ¿ $\beta \wedge \neg\gamma$ inconsistente?
  Cuando lo negamos, nos queda $CNF(\beta) \cup \underbrace{CNF(\neg\gamma)}_{\text{support}}$.
- Resolución con soporte S: usamos siempre al menos una cláusula $C \notin S$. Es un método completo.

Resolución

- Cláusula de Horn: contiene como mucho un literal positivo.

Resolución

- Cláusula de Horn: contiene como mucho un literal positivo.
- Es la base de Prolog. Cada cláusula Horn se puede ver como una regla:

$$\begin{aligned} & p \vee \neg q_1 \vee \neg q_2 \vee \neg q_3 \\ \equiv\ & p \leftarrow q_1 \wedge q_2 \wedge q_3 \\ \equiv\ & p :- \ q_1, q_2, q_3. \quad \text{(una regla)} \end{aligned}$$

$$p \quad \text{(un hecho)}$$

$$\begin{aligned} & \neg q_1 \vee \neg q_2 \vee \neg q_3 \\ \equiv\ & \bot \leftarrow q_1 \wedge q_2 \wedge \neg q_3 \\ \equiv\ & ?- \ q_1, q_2, q_3. \quad \text{(la consulta)} \end{aligned}$$

Resolución

- Si $\mathit{CNF}(\alpha)$ sólo consta de cláusulas de Horn, decidir su satisfactibilidad (HORNSAT) es un problema **P**-completo.

Resolución

- Si $\mathit{CNF}(\alpha)$ sólo consta de cláusulas de Horn, decidir su satisfactibilidad (HORNSAT) es un problema **P**-completo.
- Las estrategias unitaria, de entrada y lineal se vuelven completas para cláusulas de Horn. Es más, el conjunto de cláusulas negativas (consultas) se puede usar como conjunto de soporte.

1. Resolución en Lógica Proposicional

2. Resolución en Lógica de Primer Orden

Resolución

- En general, el objetivo es reducir el problema a resolución proposicional

Resolución

- En general, el objetivo es reducir el problema a resolución proposicional
- Para ello, transformamos la fórmula de lógica de primer orden hasta lograr un conjunto de cláusulas CNF sin cuantificadores

Resolución

- Forma Normal Prénex: ningún cuantificador en el ámbito de una conectiva proposicional.

Resolución

- Forma Normal Prénex: ningún cuantificador en el ámbito de una conectiva proposicional. Esto es: todos los cuantificadores a la izquierda

- Forma Normal Prénex: ningún cuantificador en el ámbito de una conectiva proposicional. Esto es: todos los cuantificadores a la izquierda
- En Lógica clásica de Primer Órden, siempre se puede reducir a FN Prénex.

Resolución

- Forma Normal Prénex: ningún cuantificador en el ámbito de una conectiva proposicional. Esto es: todos los cuantificadores a la izquierda
- En Lógica clásica de Primer Órden, siempre se puede reducir a FN Prénex. Hay lógicas en que no (p.ej. intuicionista).

Resolución

- Forma Normal Prénex: ningún cuantificador en el ámbito de una conectiva proposicional. Esto es: todos los cuantificadores a la izquierda
- En Lógica clásica de Primer Órden, siempre se puede reducir a FN Prénex. Hay lógicas en que no (p.ej. intuicionista).
- Para obtener FN Prénex: primero reducir a forma normal negativa ($\neg$ sólo se aplica a átomos) como en caso proposicional, añadiendo:

$$\begin{aligned} \neg\exists x\ \alpha &\equiv \forall x\ \neg\alpha \\ \neg\forall x\ \alpha &\equiv \exists x\ \neg\alpha \end{aligned}$$

Forma Normal Prénex

- Luego, renombrar las variables para tener una diferente en cada cuantificador.

Forma Normal Prénex

- Luego, renombrar las variables para tener una diferente en cada cuantificador.
- Finalmente, aplicar:

$$
\begin{aligned}
\forall x\ \alpha(x) \wedge \beta &\equiv \forall x(\alpha(x) \wedge \beta) \\
\forall x\ \alpha(x) \vee \beta &\equiv \forall x(\alpha(x) \vee \beta) \\
\exists x\ \alpha(x) \wedge \beta &\equiv \exists x(\alpha(x) \wedge \beta) \\
\exists x\ \alpha(x) \vee \beta &\equiv \exists x(\alpha(x) \vee \beta)
\end{aligned}
$$

con x no libre en β.

Forma Normal Prénex

- Ejemplo:

$$
\begin{aligned}
& (P(y) \vee \exists x\, Q(x)) \rightarrow \forall y(R(y) \wedge \exists x\, A(x,y)) \\
\equiv\ & \neg P(y) \wedge \forall x\, \neg Q(x) \vee \forall y(R(y) \wedge \exists x\, A(x,y)) \\
\equiv\ & \neg P(y) \wedge \forall x\, \neg Q(x) \vee \forall z(R(z) \wedge \exists w\, A(w,z)) \\
\equiv\ & \underbrace{\forall x \forall z \exists w}_{\text{prefijo}} \underbrace{(\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(w,z)))}_{\text{matriz}}
\end{aligned}
$$

Forma Normal Prénex

- Luego, renombrar las variables para tener una diferente en cada cuantificador.

Forma Normal Prénex

- Luego, renombrar las variables para tener una diferente en cada cuantificador.
- Finalmente, aplicar:

$$
\begin{aligned}
\forall x\ \alpha(x) \wedge \beta &\equiv \forall x(\alpha(x) \wedge \beta)\\
\forall x\ \alpha(x) \vee \beta &\equiv \forall x(\alpha(x) \vee \beta)\\
\exists x\ \alpha(x) \wedge \beta &\equiv \exists x(\alpha(x) \wedge \beta)\\
\exists x\ \alpha(x) \vee \beta &\equiv \exists x(\alpha(x) \vee \beta)
\end{aligned}
$$

Forma Normal Prénex

- Luego, renombrar las variables para tener una diferente en cada cuantificador.
- Finalmente, aplicar:

$$\begin{array}{rcl}
\forall x\ \alpha(x) \wedge \beta & \equiv & \forall x(\alpha(x) \wedge \beta) \\
\forall x\ \alpha(x) \vee \beta & \equiv & \forall x(\alpha(x) \vee \beta) \\
\exists x\ \alpha(x) \wedge \beta & \equiv & \exists x(\alpha(x) \wedge \beta) \\
\exists x\ \alpha(x) \vee \beta & \equiv & \exists x(\alpha(x) \vee \beta)
\end{array}$$

con x no libre en β.

Forma Normal Prénex

Ejemplo:

$$\exists x\ (x < y \wedge \neg \exists x (z = x + y)) \rightarrow \forall y\ (y < sz) \vee \exists t\ (t = x + y)$$

Forma Normal Prénex

Ejemplo:

$$\begin{array}{rl} & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee \exists t\,(t = x + y) \\ \equiv & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \exists t\,(\,\forall y\,(y < sz) \vee (t = x + y)\,) \end{array}$$

Forma Normal Prénex

Ejemplo:

$$
\begin{aligned}
& \exists x\ (x < y \wedge \neg \exists x(z = x + y)) \rightarrow \forall y\ (y < sz) \vee \exists t\ (t = x + y) \\
\equiv\ & \exists x\ (x < y \wedge \neg \exists x(z = x + y)) \rightarrow \exists t\ (\ \forall y\ (y < sz) \vee (t = x + y)\ ) \\
\equiv\ & \exists t\ [\ \exists x\ (x < y \wedge \neg \exists x(z = x + y)) \rightarrow \forall y\ (y < sz) \vee (t = x + y)\ ]
\end{aligned}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
& \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee \exists t\,(t = x + y) \\
\equiv & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \exists t\,(\,\forall y\,(y < sz) \vee (t = x + y)\,) \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\,(u < sz) \vee (t = x + y)\,]
\end{array}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
& \exists x\ (x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\ (y < sz) \vee \exists t\ (t = x + y) \\
\equiv & \exists x\ (x < y \wedge \neg\exists x(z = x + y)) \rightarrow \exists t\ (\ \forall y\ (y < sz) \vee (t = x + y)\ ) \\
\equiv & \exists t\ [\ \exists x\ (x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\ (y < sz) \vee (t = x + y)\ ] \\
\equiv & \exists t\ [\ \exists x\ (x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\ (u < sz) \vee (t = x + y)\ ] \\
\equiv & \exists t\ [\ \exists x\ (x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\ (\ (u < sz) \vee (t = x + y)\ )\ ]
\end{array}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
 & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \forall y\,(y < sz) \vee \exists t\,(t = x + y) \\
\equiv & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \exists t\,(\,\forall y\,(y < sz) \vee (t = x + y)\,) \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \forall y\,(y < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \forall u\,(u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \forall u\,(\,(u < sz) \vee (t = x + y)\,)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \to \,(u < sz) \vee (t = x + y)\,]
\end{array}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
 & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee \exists t\,(t = x + y) \\
\equiv & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \exists t\,(\,\forall y\,(y < sz) \vee (t = x + y)\,) \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\,(u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\,(\,(u < sz) \vee (t = x + y)\,)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \ (u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \neg\exists w\,(z = w + y)) \rightarrow \ (u < sz) \vee (t = x + y)\,]
\end{array}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
 & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee \exists t\,(t = x + y) \\
\equiv & \exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \exists t\,(\,\forall y\,(y < sz) \vee (t = x + y)\,) \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall y\,(y < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\,(u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow \forall u\,(\,(u < sz) \vee (t = x + y)\,)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \neg\exists x(z = x + y)) \rightarrow\ (u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \neg\exists w\,(z = w + y)) \rightarrow\ (u < sz) \vee (t = x + y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x < y \wedge \forall w\,\neg(z = w + y)) \rightarrow\ (u < sz) \vee (t = x + y)\,]
\end{array}
$$

Ejemplo:

$$
\begin{array}{rl}
 & \exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall y\,(y<sz)\vee\exists t\,(t=x+y)\\
\equiv & \exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\exists t\,(\,\forall y\,(y<sz)\vee(t=x+y)\,)\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall y\,(y<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall u\,(u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall u\,(\,(u<sz)\vee(t=x+y)\,)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\neg\exists w\,(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\forall w\,\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,\forall w\,(x<y\wedge\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]
\end{array}
$$

Ejemplo:

$$
\begin{array}{ll}
 & \exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow \forall y\,(y<sz) \vee \exists t\,(t=x+y) \\
\equiv & \exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow \exists t\,(\,\forall y\,(y<sz) \vee (t=x+y)\,) \\
\equiv & \exists t\,[\,\exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow \forall y\,(y<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow \forall u\,(u<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,[\,\exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow \forall u\,(\,(u<sz) \vee (t=x+y)\,)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y \wedge \neg\exists x(z=x+y)) \rightarrow (u<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y \wedge \neg\exists w\,(z=w+y)) \rightarrow (u<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y \wedge \forall w\,\neg(z=w+y)) \rightarrow (u<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists x\,\forall w\,(x<y \wedge \neg(z=w+y)) \rightarrow (u<sz) \vee (t=x+y)\,] \\
\equiv & \exists t\,\forall u\,[\,\exists r\,\forall w\,(r<y \wedge \neg(z=w+y)) \rightarrow (u<sz) \vee (t=x+y)\,]
\end{array}
$$

Forma Normal Prénex

Ejemplo:

$$
\begin{array}{rl}
 & \exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall y\,(y<sz)\vee\exists t\,(t=x+y)\\
\equiv & \exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\exists t\,(\,\forall y\,(y<sz)\vee(t=x+y)\,)\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall y\,(y<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall u\,(u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\forall u\,(\,(u<sz)\vee(t=x+y)\,)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\neg\exists x(z=x+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\neg\exists w\,(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,(x<y\wedge\forall w\,\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists x\,\forall w\,(x<y\wedge\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,[\,\exists r\,\forall w\,(r<y\wedge\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]\\
\equiv & \exists t\,\forall u\,\forall r\,\exists w\,[\,(r<y\wedge\neg(z=w+y))\rightarrow\ (u<sz)\vee(t=x+y)\,]
\end{array}
$$

Forma Normal Prénex

- En general, la FN Prénex de una fórmula no es única. Podemos obtener distintos resultados, pero equivalentes

Forma Normal Prénex

- En general, la FN Prénex de una fórmula no es única. Podemos obtener distintos resultados, pero equivalentes
- Ejemplo:

$$\forall x\, P(x) \lor \exists y\, Q(y) \quad \equiv \quad \forall x\, \exists y\, \big(\, P(x) \lor Q(y)\, \big)$$

Forma Normal Prénex

- En general, la FN Prénex de una fórmula no es única. Podemos obtener distintos resultados, pero equivalentes
- Ejemplo:

$$\forall x\, P(x) \lor \exists y\, Q(y) \quad \equiv \quad \forall x\, \exists y\, (\, P(x) \lor Q(y)\,)$$

pero también

$$\forall x\, P(x) \lor \exists x\, Q(x) \quad \equiv \quad \exists y\, \forall x\, (\, P(x) \lor Q(y)\,)$$

Forma Normal Prénex

- En general, la FN Prénex de una fórmula no es única. Podemos obtener distintos resultados, pero equivalentes
- Ejemplo:

$$\forall x\, P(x) \vee \exists y\, Q(y) \quad \equiv \quad \forall x\, \exists y\, (\, P(x) \vee Q(y)\,)$$

pero también

$$\forall x\, P(x) \vee \exists x\, Q(x) \quad \equiv \quad \exists y\, \forall x\, (\, P(x) \vee Q(y)\,)$$

- Cuidado: el orden para sacar cuantificadores a la izquierda hay que respetarlo si afectan a la misma fórmula

$$\begin{aligned} \forall x\, P(x) \vee \exists y\, \forall z\, Q(y,z) \quad & \equiv \quad \forall x\, \exists y\, \forall z\, (\, P(x) \vee Q(y,z)\,) \\ & \not\equiv \quad \forall x\, \forall z\, \exists y\, (\, P(x) \vee Q(y,z)\,) \end{aligned}$$

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex.

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex. Se reemplazan for funciones Skolem.
- Eliminamos $\exists x$ del prefijo y cambiamos x en la matriz por $f(y_1, \ldots, y_n)$ donde:

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex. Se reemplazan for funciones Skolem.
- Eliminamos $\exists x$ del prefijo y cambiamos x en la matriz por $f(y_1, \ldots, y_n)$ donde:
  - f es un nuevo nombre de función que no aparecía antes

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex. Se reemplazan for funciones Skolem.
- Eliminamos $\exists x$ del prefijo y cambiamos x en la matriz por $f(y_1, \ldots, y_n)$ donde:
  - f es un nuevo nombre de función que no aparecía antes
  - $y_1, \ldots, y_n$ son todas las variables con $\forall y_i$ a la izquierda del $\exists x$ eliminado

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex. Se reemplazan for funciones Skolem.
- Eliminamos $\exists x$ del prefijo y cambiamos x en la matriz por $f(y_1, \dots, y_n)$ donde:
  - f es un nuevo nombre de función que no aparecía antes
  - $y_1, \dots, y_n$ son todas las variables con $\forall y_i$ a la izquierda del $\exists x$ eliminado
  - Si $n = 0$ usamos una constante de Skolem c (función 0-aria).

Skolemización

- Objetivo: eliminar los cuantificadores existenciales $\exists x$ de la FN Prénex. Se reemplazan for funciones Skolem.
- Eliminamos $\exists x$ del prefijo y cambiamos x en la matriz por $f(y_1, \ldots, y_n)$ donde:
  - f es un nuevo nombre de función que no aparecía antes
  - $y_1, \ldots, y_n$ son todas las variables con $\forall y_i$ a la izquierda del $\exists x$ eliminado
  - Si $n = 0$ usamos una constante de Skolem c (función 0-aria).
- En un ejemplo previo:

$$\forall x\, \forall z\, \exists w (\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(w, z))$$

se convierte en $$\forall x\, \forall z\, (\neg P(y) \wedge \neg Q(x) \vee (R(z) \wedge A(f(x, z), z))$$

Skolemización

- Un ejemplo más intuitivo: toda persona tiene una madre.

$$\begin{array}{rl} & \forall x(P(x) \to \exists y\; M(y,x)) \\ \equiv & \forall x \exists y(P(x) \to M(y,x)) \\ \text{se convierte en} & \forall x(P(x) \to M(madre(x),x)) \end{array}$$

Skolemización

- Un ejemplo más intuitivo: toda persona tiene una madre.

$$\begin{array}{rl} & \forall x(P(x) \rightarrow \exists y\ M(y,x)) \\ \equiv & \forall x \exists y(P(x) \rightarrow M(y,x)) \\ \text{se convierte en} & \forall x(P(x) \rightarrow M(madre(x),x)) \end{array}$$

- Atención: α y su skolemización ¡no son equivalentes!

Skolemización

- Un ejemplo más intuitivo: toda persona tiene una madre.

$$
\begin{array}{rl}
 & \forall x(P(x) \rightarrow \exists y\ M(y,x)) \\
\equiv & \forall x \exists y (P(x) \rightarrow M(y,x)) \\
\text{se convierte en} & \forall x(P(x) \rightarrow M(madre(x),x))
\end{array}
$$

- Atención: α y su skolemización ¡no son equivalentes!
  Sin embargo: α es satisfactible sii su skolemización es satisfactible.

Modelos de Herbrand

- Tomamos una fórmula α, la reducimos a su FN Prénex, skolemizamos y reducimos su matriz a CNF.

Modelos de Herbrand

- Tomamos una fórmula α, la reducimos a su FN Prénex, skolemizamos y reducimos su matriz a CNF.
- Pero entonces, ¿cuántos modelos primer orden tendríamos que considerar?

Modelos de Herbrand

- Tomamos una fórmula α, la reducimos a su FN Prénex, skolemizamos y reducimos su matriz a CNF.
- Pero entonces, ¿cuántos modelos primer orden tendríamos que considerar?
- Como sólo nos interesa satisfactibilidad, podemos restringirlo a modelos de Herbrand.

Modelos de Herbrand

- Tomamos una fórmula α, la reducimos a su FN Prénex, skolemizamos y reducimos su matriz a CNF.
- Pero entonces, ¿cuántos modelos primer orden tendríamos que considerar?
- Como sólo nos interesa satisfactibilidad, podemos restringirlo a modelos de Herbrand.
- Punto clave: en un modelo de Herbrand, cada constante C se interpreta como ella misma $C_I = C$.

Modelos de Herbrand

- Tomamos una fórmula α, la reducimos a su FN Prénex, skolemizamos y reducimos su matriz a CNF.
- Pero entonces, ¿cuántos modelos primer orden tendríamos que considerar?
- Como sólo nos interesa satisfactibilidad, podemos restringirlo a modelos de Herbrand.
- Punto clave: en un modelo de Herbrand, cada constante C se interpreta como ella misma $C_I = C$.
  Y lo mismo sucede para todo término sin variables
  $v_{I,\sigma}(f(t_1, \ldots, t_n)) = f(t_1, \ldots, t_n)$

Modelos de Herbrand

- El Dominio de Herbrand DH se define como el conjunto de todos los términos sin variables (ground) que se pueden formar.

Modelos de Herbrand

- El Dominio de Herbrand *DH* se define como el conjunto de todos los términos sin variables (ground) que se pueden formar.
- Ejemplo: si no tenemos funciones pero tenemos las constantes $Juan, Ann, Ana$, tenemos $DH = \{Juan, Ann, Ana\}$.

Modelos de Herbrand

- El Dominio de Herbrand *DH* se define como el conjunto de todos los términos sin variables (ground) que se pueden formar.
- Ejemplo: si no tenemos funciones pero tenemos las constantes $Juan, Ann, Ana$, tenemos $DH = \{Juan, Ann, Ana\}$.
- Ejemplo: si tenemos la función f y la constante 0, $DH = \{0, f(0), f(f(0)), f(f(f(0))), \dots\}$ es un conjunto infinito.

Modelos de Herbrand

- La Base de Herbrand BH se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.

Modelos de Herbrand

- La Base de Herbrand *BH* se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.

Modelos de Herbrand

- La Base de Herbrand *BH* se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.
- Una fórmula α es satisfactible sii el conjunto (posiblemente infinito) de cláusuluas que podemos formar a partir de su Skolemización es satisfactible (en sentido proposicional).

Modelos de Herbrand

- La Base de Herbrand *BH* se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.
- Una fórmula α es satisfactible sii el conjunto (posiblemente infinito) de cláusuluas que podemos formar a partir de su Skolemización es satisfactible (en sentido proposicional).
- Sin embargo, este conjunto suele ser infinito (basta con tener un símbolo de función).

Modelos de Herbrand

- La Base de Herbrand *BH* se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.
- Una fórmula α es satisfactible sii el conjunto (posiblemente infinito) de cláusuluas que podemos formar a partir de su Skolemización es satisfactible (en sentido proposicional).
- Sin embargo, este conjunto suele ser infinito (basta con tener un símbolo de función). Intentar generarlo para resolución no tiene sentido. Ejemplo:

$$\left[P(f(x)) \vee \neg Q(x)\right] \wedge Q(y) \wedge \neg P(z)$$

Modelos de Herbrand

- La Base de Herbrand *BH* se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.
- Una fórmula α es satisfactible sii el conjunto (posiblemente infinito) de cláusuluas que podemos formar a partir de su Skolemización es satisfactible (en sentido proposicional).
- Sin embargo, este conjunto suele ser infinito (basta con tener un símbolo de función). Intentar generarlo para resolución no tiene sentido. Ejemplo:

$$\left[P(f(x)) \vee \neg Q(x)\right] \wedge Q(y) \wedge \neg P(z)$$

- Idea: podemos mantener las variables y ver qué asignaciones generarían una resolución.

Modelos de Herbrand

- La Base de Herbrand BH se define como el conjunto de todos los átomos sin variables (ground) que se pueden formar.
- Podemos verlo como una signatura proposicional.
- Una fórmula α es satisfactible sii el conjunto (posiblemente infinito) de cláusuluas que podemos formar a partir de su Skolemización es satisfactible (en sentido proposicional).
- Sin embargo, este conjunto suele ser infinito (basta con tener un símbolo de función). Intentar generarlo para resolución no tiene sentido. Ejemplo:

$$\big[P(f(x)) \vee \neg Q(x)\big] \wedge Q(y) \wedge \neg P(z)$$

- Idea: podemos mantener las variables y ver qué asignaciones generarían una resolución. Ejemplo: tomando $z = f(x)$ e $y = x$ obtenemos cláusula vacía

Unificador más general (Robinson)

- Dado un conjunto de expresiones E, obtenemos un par de desacuerdo $\langle E_1, E_2 \rangle$ buscando de izquierda a derecha el primer símbolo distinto y tomando su subexpresión correspondiente.

Unificador más general (Robinson)

- Dado un conjunto de expresiones E, obtenemos un par de desacuerdo $\langle E_1, E_2 \rangle$ buscando de izquierda a derecha el primer símbolo distinto y tomando su subexpresión correspondiente.
- Ejemplo: dados $P(f(x), y)$ y $P(f(g(a, z), f(z))$ un par de desacuerdo sería $\langle x, g(a, z) \rangle$.

Unificador más general (Robinson)

- Dado un conjunto de expresiones E, obtenemos un par de desacuerdo $\langle E_1, E_2 \rangle$ buscando de izquierda a derecha el primer símbolo distinto y tomando su subexpresión correspondiente.
- Ejemplo: dados $P(f(x), y)$ y $P(f(g(a, z), f(z))$ un par de desacuerdo sería $\langle x, g(a, z) \rangle$.
- Si 2 literales se pueden unificar, existe un unificador más general que se calcula:

$\sigma := [\,]$;
while $|E| > 1$ {
  $D :=$ par de desacuerdo E;
  if D contiene algún x y un término t sin contener x {
    $E := E[x/t]$;
    $\sigma := \sigma \cdot [x/t]$; }
  else return 'no-unificable';
}