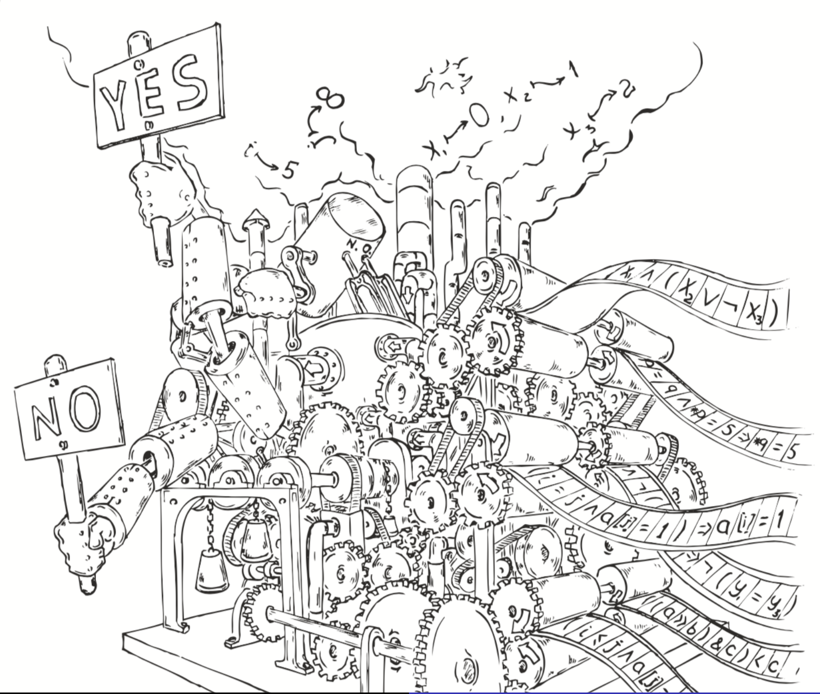

YES
NO
N.O.

Задачи разрешимости логических формул и приложения
Лекция 10. Формулы с кванторами

Роман Холин

Московский государственный университет

Москва, 2022

Определения

$$\forall x.\ \varphi \iff \neg \exists x.\ \neg \varphi$$

Определения

$$\forall x.\ \varphi \iff \neg\exists x.\ \neg\varphi$$

$$\forall x.\ \underbrace{((x<0) \land \exists y.\ \overbrace{(y > x \land (y \geq 0 \lor \exists x.\ \underbrace{(y = x+1)}_{\text{scope of } \exists x}))}^{\text{scope of } \exists y})}_{\text{scope of } \forall x}\ .$$

Определения

$$\forall x.\ \varphi \iff \neg\exists x.\ \neg\varphi$$

$$\forall x.\ \underbrace{((x < 0) \land \exists y.\ \overbrace{(y > x \land (y \geq 0 \lor \exists x.\ \underbrace{(y = x + 1)}_{\text{scope of } \exists x}))}^{\text{scope of } \exists y})}_{\text{scope of } \forall x}\ .$$

- Переменная называется свободной, если она не привязана к квантору

Определения

$$\forall x.\ \varphi \iff \neg\exists x.\ \neg\varphi$$

$$\forall x.\ \underbrace{\left((x < 0) \wedge \exists y.\ \overbrace{(y > x \wedge (y \geq 0 \vee \exists x.\ \underbrace{(y = x + 1)}_{\text{scope of } \exists x}))}^{\text{scope of } \exists y}\right)}_{\text{scope of } \forall x}\ .$$

- Переменная называется свободной, если она не привязана к квантору
- Формула замкнута, если не содержит свобожных переменных

QBF

$$\begin{aligned} \mathit{formula} : {} & \mathit{formula} \wedge \mathit{formula} \mid \neg \mathit{formula} \mid (\mathit{formula}) \mid \\ & \mathit{identifier} \mid \exists\, \mathit{identifier}.\ \mathit{formula} \end{aligned}$$

QBF

$$\begin{aligned} formula : \; & formula \wedge formula \mid \neg formula \mid (formula) \mid \\ & identifier \mid \exists\, identifier.\ formula \end{aligned}$$

Сложность - PSPACE

QDLA

$$
\begin{aligned}
formula &: formula \wedge formula \mid \neg formula \mid (formula) \mid \\
&\quad predicate \mid \forall\, identifier.\ formula \\
predicate &: \Sigma_i a_i x_i \leq c
\end{aligned}
$$

Пренексная нормальная форма

- Формула находится в Пренексной нормальной форме, если она имеет форму $Q[n]V[n]\ldots Q[1]V[1]. < quantifier - freeformula >$, где $Q[i]$ - квантор, $V[i]$ - переменная

Пренексная нормальная форма

- Формула находится в Пренексной нормальной форме, если она имеет форму $Q[n]V[n]\ldots Q[1]V[1]. < quantifier - freeformula >$, где $Q[i]$ - квантор, $V[i]$ - переменная
- Для каждой кванторной формулы существует эквивалентная ей формула в пренексной нормальной форме

Algorithm 9.2.1: PRENEX

Input: A quantified formula
Output: A formula in prenex normal form

1. Eliminate Boolean connectives other than $\vee$, $\wedge$, and $\neg$.
2. Push negations to the right across all quantifiers, using De Morgan's rules (see Sect. $\boxed{1.3}$) and ($\boxed{9.1}$).
3. If there are name conflicts across scopes, solve by renaming: give each variable in each scope a unique name.
4. Move quantifiers out by using equivalences such as

$$
\begin{aligned}
\phi_1 \wedge Qx.\ \phi_2(x) &\iff Qx.\ (\phi_1 \wedge \phi_2(x))\ , \\
\phi_1 \vee Qx.\ \phi_2(x) &\iff Qx.\ (\phi_1 \vee \phi_2(x))\ , \\
Q_1 y.\ \phi_1(y) \wedge Q_2 x.\ \phi_2(x) &\iff Q_1 y.\ Q_2 x.\ (\phi_1(y) \wedge \phi_2(x))\ , \\
Q_1 y.\ \phi_1(y) \vee Q_2 x.\ \phi_2(x) &\iff Q_1 y.\ Q_2 x.\ (\phi_1(y) \vee \phi_2(x))\ ,
\end{aligned}
$$

where $Q, Q_1, Q_2 \in \{\forall, \exists\}$ are quantifiers, $x \notin var(\phi_1)$, and $y \notin var(\phi_2)$.

Пример

$$\mathcal{Q} := \neg\exists x.\ \neg(\exists y.\ ((y \implies x) \wedge (\neg x \vee y)) \wedge \neg\forall y.\ ((y \wedge x) \vee (\neg x \wedge \neg y)))$$

Проекция

Проекция формулы $Q[n]V[n]\dots Q[2]V[2].\exists x.\phi$ называют формулу $Q[n]V[n]\dots Q[2]V[2].\phi$

Пример

$$\mathcal{Q} := \neg\exists x.\ \neg(\exists y.\ ((y \implies x) \wedge (\neg x \vee y)) \wedge \neg\forall y.\ ((y \wedge x) \vee (\neg x \wedge \neg y)))$$

Algorithm 9.2.2: QUANTIFIER-ELIMINATION

Input: A sentence $Q[n]V[n]\ldots Q[1]V[1].\ \phi$, where ϕ is quantifier-free

Output: A (quantifier-free) formula over constants ϕ', which is valid if and only if ϕ is valid

1. $\phi' := \phi$;
2. **for** $i := 1, \ldots, n$ **do**
3.   **if** $Q[i] = \exists$ **then**
4. $\phi' :=$ PROJECT($\phi', V[i]$);
5.   **else**
6. $\phi' := \neg$PROJECT($\neg\phi', V[i]$);
7. Return ϕ';

Элиминация кванторов для QBF

$$\exists y.\ \exists x.\ x \wedge \neg x \wedge y \qquad = \text{FALSE}\ ,$$
$$\exists y.\ \exists x.\ x \wedge y = \exists y.\ y = \text{TRUE}\ .$$

Элиминация кванторов для QBF

$$\exists y.\ \exists x.\ x \wedge \neg x \wedge y \quad = \mathrm{FALSE}\ ,$$
$$\exists y.\ \exists x.\ x \wedge y = \exists y.\ y = \mathrm{TRUE}\ .$$

$$\exists x.\ \bigvee_i \bigwedge_j l_{ij} \iff \bigvee_i \exists x.\ \bigwedge_j l_{ij}$$

Проекция с помощью бинарной резолюции

$$\exists y.\ \exists z.\ \exists x.\ (y \vee x) \wedge (z \vee \neg x) \wedge (y \vee \neg z \vee \neg x) \wedge (\neg y \vee z)$$

Проекция с помощью бинарной резолюции

$$\exists y.\ \exists z.\ \exists x.\ (y \vee x) \wedge (z \vee \neg x) \wedge (y \vee \neg z \vee \neg x) \wedge (\neg y \vee z)$$

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

Пример

$$\forall u_1.\ \forall u_2.\ \exists e_1.\ \forall u_3.\ \exists e_3.\ \exists e_2.$$
$$(u_1 \vee \neg e_1) \wedge (\neg u_1 \vee \neg e_2 \vee e_3) \wedge (u_2 \vee \neg u_3 \vee \neg e_1) \wedge (e_1 \vee e_2) \wedge (e_1 \vee \neg e_3)$$

Элиминация кванторов с помощью расширения Шенона

$$\exists x.\ \varphi = \varphi|_{x=0} \vee \varphi|_{x=1}$$
$$\forall x.\ \varphi = \varphi|_{x=0} \wedge \varphi|_{x=1}$$

Пример

$$\exists y.\ \forall z.\ \exists x.\ (y \vee (x \wedge z))$$

Элиминация кванторов для QDLA

$$\sum_{i=1}^{n-1} a'_i \cdot x_i < x_n < \sum_{i=1}^{n-1} a_i \cdot x_i$$

$$\sum_{i=1}^{n-1} a'_i \cdot x_i < \sum_{i=1}^{n-1} a_i \cdot x_i$$

Пример

$$\forall x.\ \exists y.\ \exists z.\ (y+1 \leq x \quad \wedge \quad z+1 \leq y \quad \wedge \quad 2x+1 \leq z)$$

Наивный алгоритм разрешения QBF

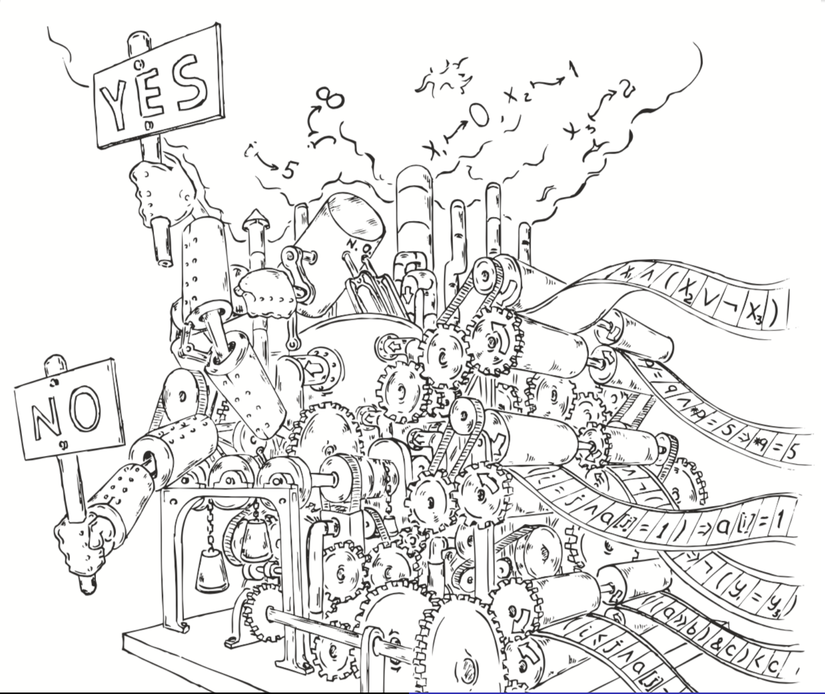
Наивный алгоритм разрешения QBF

Algorithm 9.3.1: SEARCH-BASED-DECISION-OF-QBF

Input: A QBF $\mathcal{Q}$ in PNF $Q[n]V[n] \ldots\ Q[1]V[1].\ \phi$, where ϕ is in CNF
Output: "Valid" if $\mathcal{Q}$ is valid, and "Not valid" otherwise

function MAIN(QBF formula $\mathcal{Q}$)
    if QBF$(\mathcal{Q}, \emptyset, n)$ **then return** "Valid";
    else return "Not valid";

function QBF($\mathcal{Q}$, assignment set $\hat{v}$, $level \in \mathbb{N}_0$)
    if $(\phi|_{\hat{v}}$ simplifies to FALSE) **then return** FALSE;
    if $(level = 0)$ **then return** TRUE;
    if $(Q[level] = \forall)$ **then**
        return $\begin{pmatrix} \mathrm{QBF}(\mathcal{Q}, \hat{v} \cup \neg V[level], level - 1) \wedge \\ \mathrm{QBF}(\mathcal{Q}, \hat{v} \cup V[level], level - 1) \end{pmatrix}$;
    else
        return $\begin{pmatrix} \mathrm{QBF}(\mathcal{Q}, \hat{v} \cup \neg V[level], level - 1) \vee \\ \mathrm{QBF}(\mathcal{Q}, \hat{v} \cup V[level], level - 1) \end{pmatrix}$;


YES
NO