

4.4. ОБЩ МЕТОД НА РЕЗОЛЮЦИЯТА

1. Унифициращ алгоритъм

Общият метод на резолюцията се прилага върху клаузна форма на правилно построена формула на предикатната логика без да се търсят базови екземпляри на клаузите, т.е. при наличие в клаузите на индивидуални променливи.

Дефиниции:

Субституция – крайно множество от подредени двойки от вида

$$\alpha = \{\langle v_1, t_1 \rangle, \langle v_2, t_2 \rangle, \dots, \langle v_i, t_i \rangle, \dots, \langle v_n, t_n \rangle\},$$

където v_i са различни индивидуални променливи; t_i - терми, различни от v_i .

Унификатор на дизюнкция от атомни форми от вида

$$A_1 \vee A_2 \vee \dots \vee A_n$$

- субституция, при която е възможно заместването на всички променливи с терми до получаване на равенство на отделните атомни форми в дизюнкцията:

$$A'_1 \leftrightarrow A'_2 \leftrightarrow \dots \leftrightarrow A'_n,$$

където A'_i се получава от A_i след заместване на двойките $\langle v_i, t_i \rangle$.

Унифицируема дизюнкция - такава дизюнкция, за която съществува унификатор.

Най-общ унификатор – унификатор β , за който всеки друг унификатор α за същата дизюнкция я превръща в базов екземпляр на унифицируемата с унификатор β дизюнкция

Пример:

Дадена е ППФ в клаузна форма:

$$\forall x \forall y \forall z \forall y_1 (p(x, f(x)) \wedge q(y, g(y)) \wedge (\overline{p(a, y_1)} \vee \overline{q(y_1, z)}))$$

със следните клаузи:

$$c_1 : p(x, f(x))$$

$$c_2 : q(y, g(y))$$

$$c_3 : \overline{p(a, y_1)} \vee \overline{q(y_1, z)}$$

където x, y, y_1, z са индивидуалните променливи; f, g - аргументните функции; p, q - двуместните предикати.

В множеството предикати се търсят предикати с еднакво име и брой на аргументите. След това същите се съпоставят:

$$\begin{array}{ll} p(x, f(x)) & \text{от } c_1; \\ \overline{p(a, y_1)} & \text{от } c_3. \end{array}$$

Ако съществуват литерали, участващи с права и инверсна форма се прави субституция, която да ги изравни:

Субституцията $\{\langle x, a \rangle, \langle y_1, f(a) \rangle\}$ изравнява предикатите. Резолвентата на c_1 и c_3 има вида:

$$c_4 : \overline{q(f(a), z)}.$$

Съпоставят се литералите на c_2 и c_4 . Тези литерали са с еднакво име и местност, поради което се прави опит за изравняване:

$$c_2 : q(y, g(y));$$

$$c_4 : \overline{q(f(a), z)}.$$

Субституцията $\{\langle y, f(a) \rangle, \langle z, g(f(a)) \rangle\}$ изравнява двата литерала до:

$$q(f(a), g(f(a))),$$

при което в резултат на приложението на общия метод на резолюцията се извежда празна клауза.

Условията, при които е изведена неудовлетворимостта на ППФ са:

$$x = a; \quad y = y_1 = f(a) \quad z = g(f(a))$$

Най-общият идентификатор има вида: $\beta = \{\langle y, a \rangle, \langle z, f(a) \rangle\}$

2. Структура на унифициращия алгоритъм

Алгоритъмът започва с празната субституция $\alpha = \{ \}$ и построява постъпково най-общия унификатор β , ако съществува.

На всяка $k \geq 1$ стъпка алгоритъмът работи по следния начин:

1. Прилага текущата субституция.
2. Ако всички атомни формули в дизюнкцията са идентични, алгоритъмът спира, обявява унифицируемост, а текущата субституция α_k за най-общ унификатор.
3. Ако атомните формули не са идентични, определя множеството на несъвпадение, разглеждайки всяка атомарна форма като низ от символи, които не съвпадат.
4. Проверява има ли в множеството на несъвпадение поне една променлива. Ако няма – спира като обявява неунифицируемост.
5. Разширява α_k с нова двойка, която да направи поне два члена от множеството на несъвпадение идентични.
6. Увеличава k и преминава към т. 1.

3. Унифициращ алгоритъм в езика за логическо програмиране Prolog

В Prolog унифициращият алгоритъм се изпълнява на всяка стъпка от действието на машината за логически извод. Той се осъществява върху два обекта от базата знания посредством процедурата, написана на езика, реализиращ интерпретатора на Prolog.

Същността на действието на алгоритъма се реализира чрез съпоставяне на параметрите на двата обекта, подлежащи на унификация, като следствие от действието му се връща логическа стойност *true* при унифицируемост или *false* при невъзможност за унифициране.

Съпоставянето се извършва на базата на следната таблица:

	<i>Atom</i> b	<i>Variable</i> X_b	<i>Term</i> t_b
<i>atom</i> a	<i>if</i> $a = b$ <i>true</i> <i>else false</i> $u = \{ \}$	<i>true</i> $u = \{ \langle X_b, a \rangle \}$	<i>false</i> $u = \{ \}$
<i>variable</i> X_a	<i>true</i> $u = \{ \langle X_a, b \rangle \}$	<i>true</i> $u = \{ \langle X_a, X_b \rangle \}$	<i>true</i> $u = \{ \langle X_a, t_b \rangle \}$
<i>term</i> t_a	<i>false</i> $u = \{ \}$	<i>true</i> $u = \{ \langle X_b, t_a \rangle \}$	

При съпоставяне на термите t_a и t_b в таблицата се извършва следното:

- 1) Проверява се за еднаквост на имената, като се осъществява проверка на синтактичното съответствие на името на t_a с името на t_b . При несъвпадение унификацията е неуспешна.
- 2) Проверява се за еднаква местност (брой на аргументите на термите). Ако не съвпадат унификацията е неуспешна.
- 3) Прави се цикъл по всички аргументи на термите отляво надясно като за всеки аргумент се прилага същото съпоставяне по дадената таблица. Ако има възможност за изравняване на всички аргументни двойки, тогава унификацията е успешна.

Обикновено обектите в Prolog-интерпретатора се представят посредством указател към обекта. Ефективен достъп и бързо изпълнение на алгоритъма се постига като се установи възможно най-бързо типът на обекта.