Змістовий модуль 4. Метод резолюцій

Тема 9. Метод резолюцій у пропозиційній логіці. Підстановка та уніфікація

План лекції.

- Метод резолюцій у пропозиційній логіці
- Основна теорема резолюції для пропозиційної логіки
- Підстановка та уніфікація
- Алгоритм уніфікації.
  - Опис алгоритму уніфікації
  - Обґрунтування алгоритму уніфікації

Існують комп'ютерні програми, котрі розроблено для автоматизації міркувань, виконуваних за допомогою доведення теорем. У багатьох із цих програм використано правило виведення резолюція. На основі цього правила Дж. Робінсон (G. Robinson) 1965 р. запропонував *метод резолюцій* автоматичного доведення теорем.

4.1. Метод резолюцій у пропозиційній логіці

Нагадаємо, що літералом ми називаємо атом або його заперечення, диз'юнктом (або елементарною диз'юнкцією) – диз'юнкцію літералів. Диз'юнкт може складатися й з одного літералу. Нам знадобиться ще один диз'юнкт – порожній, який не містить літералів. Його позначають $\square$, він є фальшивим у будь-якій інтерпретації. Це означає, що формула $A\wedge\square$ еквівалентна $\square$, а формула $A\vee\square$ еквівалентна A. Порожній диз'юнкт фактично є тим самим, що і тотожно фальшива формула **F**, але в контексті методу резолюцій прийнято використовувати позначення $\square$.

Нагадаємо, що літерали p і $\neg p$ називають *контрарними*, а пару $\{p, \neg p\}$ – контрарною парою.

Метод резолюцій ґрунтується на *правилі резолюції*. Нагадаємо його:

$$\frac{p \vee A, \neg p \vee B}{A \vee B}.$$

Зазначимо, що диз'юнкти можуть містити будь-яке число літералів. Нехай диз'юнкти D_1 і D_2 такі, що D_1 містить літерал p, контрарний літералу $\neg p$ із D_2. Викреслимо p з D_1 і $\neg p$ з D_2, та побудуємо диз'юнкцію D решти літералів цих диз'юнктів. Отриманий диз'юнкт D є резольвентою диз'юнктів D_1 і D_2. Оформимо це у вигляді такого твердження.

Теорема 4.1. Нехай дано два диз'юнкти $D_1 = p\vee A$ і $D_2 = \neg p\vee B$. Тоді резольвента $D = A\vee B$ диз'юнктів D_1 і D_2 є логічним наслідком D_1 і D_2.

Доведення. Припустимо, що диз'юнкти D_1 і D_2 істинні при інтерпретації $\boldsymbol{I}$. Доведемо, що резольвента D диз'юнктів D_1 і D_2 також істинна при інтерпретації $\boldsymbol{I}$. Зазначимо, що або p, або $\neg p$ фальшиве при $\boldsymbol{I}$. Припустимо, що p фальшиве при $\boldsymbol{I}$. Тоді диз'юнкт D_1 не може складатися з одного літералу, бо тоді б він був фальшивим при інтерпретації $\boldsymbol{I}$. Отже, A має бути істинним при $\boldsymbol{I}$; тоді і резольвента $D = A\vee B$ буде істинною при $\boldsymbol{I}$. Аналогічно, коли фальшиве $\neg p$ при $\boldsymbol{I}$, то B має бути істинним, отже, і резольвента $D = A\vee B$ буде істинною при $\boldsymbol{I}$.

Приклад 4.1. З диз'юнктів $\neg p \vee q \vee r$ і $p \vee t$ виводиться диз'юнкт $q \vee r \vee t$ (резольвента).

Нехай S – множина диз'юнктів. *Резолютивним виведенням із S* називають послідовність диз'юнктів

$$D_1, D_2, \ldots, D_m$$

таку, що кожний диз'юнкт цієї послідовності або належить S, або випливає з попередніх диз'юнктів цієї послідовності за правилом резолюції.

Виведення порожнього диз'юнкту $\square$ із S називають *доведенням невиконуваності S*, або *спростуванням S*.

Диз'юнкт D називають *вивідним* із S, якщо існує виведення з S, останнім диз'юнктом якого є D.

Приклад 4.2. Нехай $S = \{\neg p \vee q \vee r, \neg q \vee t, p\}$. Тоді послідовність $D_1 = \neg p \vee q \vee r$, $D_2 = \neg q \vee t$, $D_3 = \neg p \vee r \vee t$, $D_4 = p$, $D_5 = r \vee t$ є виведенням із S. Диз'юнкт $r \vee t$ вивідний з S.

Важливе значення для методу резолюцій має теорема 1.8. Нагадаємо її.

Нехай дано формули $A_1, A_2, \ldots, A_k$ і формулу B. Формула B є логічним наслідком $A_1, A_2, \ldots, A_k$ тоді й тільки тоді, коли множина формул $\{A_1, A_2, \ldots, A_k, \neg B\}$ невиконувана.

Для доведення того, що формула B – логічний наслідок формул $A_1, A_2, \ldots, A_k$, метод резолюцій застосовують так. Спочатку утворюють множину формул $\{A_1, \ldots, A_k, \neg B\}$. Після цього кожну з цих формул приводять до кон'юнктивної нормальної форми і в отриманих формулах закреслюють знаки кон'юнкції. Отримують множину диз'юнктів $S = \{D_1, D_2, \ldots, D_m\}$. Якщо порожній диз'юнкт $\square$ вивідний із S, то формула B є логічним наслідком формул $A_1, A_2, \ldots, A_k$. Якщо з S неможливо вивести $\square$, то B не є логічним наслідком формул $A_1, A_2, \ldots, A_k$.

Приклад 4.3. За допомогою методу резолюцій доведемо, що множина $S = \{\neg p \vee q, \neg q, p\}$ невиконувана. Запишемо кожну елементарну диз'юнкцію в окремому рядку:

(1) $\neg p \vee q$;
(2) $\neg q$;
(3) p;
(4) $\neg p$.

Резольвенту (4) отримано з елементарних диз'юнкцій (1) та (2). Далі,

(5) $\square$.

Резольвента (5) – порожній диз'юнкт, його отримано з диз'юнкту (3) та резольвенти (4). Виведення порожнього диз'юнкту $\square$ із множини S доводить її невиконуваність.

Приклад 4.4 Методом резолюцій доведемо, що формула $r \vee s$ є логічним наслідком формул $p \vee q$, $p \to r$, $q \to s$. Запишемо гіпотези у формі диз'юнктів і випишемо кожний із них в окремому рядку:

(1) $p \vee q$;
(2) $\neg p \vee r$;
(3) $\neg q \vee s$.

Заперечення висновку $\neg(r \vee s) \equiv \neg r \wedge \neg s$ утворює два диз'юнкти $\neg r$ та $\neg s$, кожний із яких складається з одного літералу, *ці диз'юнкти теж випишемо в окремих рядках*:

(4) $\neg r$;
(5) $\neg s$.

Послідовно побудувавши резольвенти за правилом резолюції, виведемо порожній диз'юнкт. Біля кожної резольвенти записано номери диз'юнктів, із яких цю резольвенту отримано:

(6) $\neg p$ (2), (4);
(7) q (6), (1);
(8) $\neg q$ (3), (5);
(9) $\square$ (7), (8).

Отримано порожній диз'юнкт. Отже, множина S невиконувана, тобто формула $r \vee s$ – логічний наслідок формул $p \vee q$, $p \to r$ і $q \to s$.

4.2. Основна теорема резолюції для пропозиційної логіки

Доведемо повноту методу резолюцій. Тут зручно ввести поняття резолютивного замикання $RC(S)$ множини диз'юнктів S. *Резолютивне замикання* $RC(S)$ множини диз'юнктів S – це множина всіх диз'юнктів, які можна отримати шляхом повторного застосування правила резолюції до диз'юнктів з множини S або до одержуваних диз'юнктів.

Теорема 4.2 (основна теорема резолюції для пропозиційної логіки). У пропозиційній логіці множина диз'юнктів S невиконувана тоді і тільки тоді, коли її резолютивне замикання $RC(S)$ містить порожній диз'юнкт $\square$. *Іншими словами, множина диз'юнктів* S *невиконувана тоді і тільки тоді, коли існує резолютивне виведення з* S *порожнього диз'юнкту* $\square$.

Доведення. Необхідність. Дано: множина диз'юнктів S пропозиційної логіки невиконувана. Потрібно довести: множина $RC(S)$ містить порожній диз'юнкт $\square$. Використаємо закон контрапозиції $A \to B \equiv \neg B \to \neg A$: доведемо, що коли множина $RC(S)$ *не містить* порожнього диз'юнкту $\square$, то множина S *виконувана*. Справді, у такому припущенні для множини S можна створити інтерпретацію з підходящими істинісними значеннями для *атомів* (пропозиційної логіки) $p_1, \dots, p_k$, яка задовольняє цю множину. Алгоритм побудови цієї інтерпретації такий.

Для i від 1 до k:

- якщо в множині $RC(S)$ є диз'юнкт, який містить *літерал* $\neg p_i$, такий, що всі попередні *літерали* цього диз'юнкту мають значення F при вже присвоєних значеннях для *атомів* $p_1, \dots, p_{i-1}$, то *атому* p_i присвоїти значення F;
- інакше *атому* p_i присвоїти значення T.

Це присвоювання значень *атомам* $p_1, \dots, p_k$ формує *інтерпретацію* для множини диз'юнктів S, яка задовольняє цю множину – це означає, що при цих присвоєних *атомам* $p_1, \dots, p_k$ логічних значеннях всі диз'юнкти з множини S набувають значення T. Щоб у цьому переконатися, припустимо протилежне – нехай (при якомусь i в послідовності) присвоювання значення *атому* p_i спричиняє для якогось диз'юнкту D значення F. Як тільки таке трапиться, всі інші *літерали* в диз'юнкті D повинні мати значення F для вже присвоєних значень *атомів* $p_1, \dots, p_{i-1}$. (Інакше цей диз'юнкт D не міг би мати значення F.) Тоді D має виглядати як $(\mathbf{F} \vee \mathbf{F} \vee \dots \vee \mathbf{F} \vee p_i)$ або як $(\mathbf{F} \vee \mathbf{F} \vee \dots \vee \mathbf{F} \vee \neg p_i)$. Якщо тільки один із цих двох диз'юнктів є в $RC(S)$, то алгоритм присвоїть відповідне істиннісне значення для p_i таке, що зробить диз'юнкт D істинним. Отже, диз'юнкту D можна надати значення F тільки якщо обидва наведені диз'юнкти є в $RC(S)$. Тепер, оскільки множина $RC(S)$ замкнена щодо резолюції, то вона має містити резольвенту цих двох диз'юнктів, а всі *літерали* цієї резольвенти мають значення F для вибраних значень *атомів* $p_1, \dots, p_{i-1}$. Це суперечить припущенню, що перший випадок отримання диз'юнктом значення F відбувся на кроці i. Отже, ми довели, що в інтерпретації, побудованій за цим алгоритмом, жодний диз'юнкт із $RC(S)$ не отримає значення F. Ми побудували інтерпретацію, яка задовольняє $RC(S)$, а, отже, задовольняє і S, бо S є підмножиною $RC(S)$.

Достатність. Дано: множина $RC(S)$ містить порожній диз'юнкт $\square$. Потрібно довести: множина S невиконувана. Від протилежного. Припустимо, що множина S виконувана і нехай $\boldsymbol{I}$ – інтерпретація, яка задовольняє S (тобто задовольняє всі диз'юнкти з цієї множини). Але коли інтерпретація задовольняє диз'юнкти, скажімо, D_1 і D_2, то вона має задовольняти також будь-яку їхню резольвенту (теорема 4.1). Отже, інтерпретація $\boldsymbol{I}$ задовольняє всі диз'юнкти в $RC(S)$. Але це неможливо, бо один з цих диз'юнктів є $\square$. Тому множина S невиконувана, що й потрібно було довести.

4.3. Підстановка та уніфікація

Перейдемо до логіки першого порядку. <u>Щодо змінних у диз'юнктах уважатимемо, що вони зв'язані кванторами загальності, але самі квантори не писатимемо. Звідси випливає, що дві однакові змінні в різних диз'юнктах можна вважати різними.</u>

Передусім зазначимо, що в логіці першого порядку правило резолюції у розглянутій раніше формі вже не підходить. Справді, множина диз'юнктів $S = \{P(x), \neg P(a)\}$

невиконувана, оскільки, як припускається, змінна x зв'язана квантором загальності. Разом з тим, якщо використовувати правило резолюції для пропозиційної логіки, то із S порожнього диз'юнкта неможливо отримати. Але, за змістом зрозуміло, що в цьому випадку потрібно зробити. Оскільки диз'юнкт $P(x)$ змістовно означає, що для будь-якого x істинне $P(x)$, то $P(x)$ буде істинним і для конкретного значення a. Якщо виконати підстановку a замість x, то отримаємо множину диз'юнктів $S' = \{P(a), \neg P(a)\}$. Множини S і S' одночасно виконувані (або невиконувані). Але невиконуваніcть S' за допомогою відомого правила резолюції виводиться тривіально. Цей приклад підказує, що в логіці першого порядку правило резолюції потрібно доповнити можливістю робити підстановку.

Підстановкою (t_i замість x_i) називають множину

$$\sigma = \{t_1 / x_1, t_2 / x_2, \ldots, t_n / x_n\},$$

де $x_1, x_2, \ldots, x_n$ – попарно різні змінні, $t_1, t_2, \ldots, t_n$ – терми, причому *терм* t_i *не містить змінної* x_i $(1 \le i \le n)$.

Якщо $\sigma = \{t_1 / x_1, \ldots, t_n / x_n\}$ – підстановка, а E – вираз, то через $E\sigma$ позначатимемо вираз, отриманий із E одночасною заміною x_1 на t_1, …, x_n на t_n. Сам вираз $E\sigma$ називають *прикладом виразу* E.

У якості виразу може бути терм, атом, літерал, диз'юнкт. Зазначимо, що означення прикладу узгоджується з означенням основного прикладу диз'юнкта, яке подано в пункті 3.4, а саме: основний приклад диз'юнкта не має предметних змінних; приклад диз'юнкта може мати предметні змінні.

Приклад 4.5. Нехай $\sigma = \{f(x_2)/x_1,\ \ c/x_2,\ \ g(x_4)/x_3\}$, $E = R(x_1, x_2, x_3) \vee \neg P(f(x_2))$. Тут виразом є диз'юнкт. Тоді $E\sigma = R(f(x_2), c, g(x_4)) \vee \neg P(f(c))$.

Для зручності вводять ще *порожню підстановку* – підстановку, яка не містить елементів. Таку підстановку позначатимемо як ε.

В процедурі доведення за методом резолюцій, для того, щоб знаходити контрарні пари літералів, ми дуже часто повинні уніфікувати (склеювати) два або більше виразів, тобто повинні знаходити підстановку, яка може зробити декілька виразів однаковими. У зв'язку з цим сформулюємо таке означення.

Нехай $\{E_1, \ldots, E_k\}$ – множина літералів або термів. Підстановку σ називають *уніфікатором* цієї множини, якщо $E_1\sigma = \ldots = E_k\sigma$. Множина *уніфікується*, якщо існує уніфікатор цієї множини.

Приклад 4.6. Множина $\{Q(a, x, f(x)), Q(u, y, z)\}$ уніфікується підстановкою $\{a/u, y/x, f(y)/z\}$. Множина $\{R(x, f(x)), R(u, u)\}$ не уніфікується. Справді, якщо замінити x на u, то отримаємо множину $\{R(u, f(u)), R(u, u)\}$. Робити заміну u на $f(u)$ заборонено за означенням підстановки, та й не має сенсу, бо вона приводить до формул $R(f(u), f(f(u)))$ та $R(f(u), f(u))$, які також різні.

Якщо множина уніфікується, то як правило, існує декілька уніфікаторів цієї множини. Серед усіх уніфікаторів даної множини виділяють так званий найзагальніший уніфікатор.

Уведемо поняття найзагальнішого уніфікатора множини літералів або термів.

Нехай $\theta = \{t_1/x_1,\ t_2/x_2,\ \ldots,\ t_k/x_k\}$ і $\lambda = \{s_1/y_1,\ s_2/y_2,\ \ldots,\ s_m/y_m\}$. Тоді *добутком* (або *композицією*) підстановок θ і λ називають підстановку, яку отримують із множини

$$\{t_1\lambda/x_1,\ t_2\lambda/x_2,\ \ldots,\ t_k\lambda/x_k,\ s_1/y_1,\ s_2/y_2,\ \ldots,\ s_m/y_m\}$$

викреслюванням всіх елементів $t_i\lambda/x_i$ $(i = 1, \ldots, k)$, для яких $t_i\lambda = x_i$, а також всіх елементів s_j/y_j $(j = 1, \ldots, m)$, для яких $y_j \in \{x_1, \ldots, x_k\}$.

Зауваження. У множині $\{t_1\lambda/x_1,\ t_2\lambda/x_2,\ \ldots,\ t_k\lambda/x_k,\ s_1/y_1,\ s_2/y_2,\ \ldots,\ s_m/y_m\}$ запис $t_i\lambda$ означає результат застосування підстановки λ до терму t_i $(i = 1, \ldots, k)$.

Добуток підстановок θ і λ позначатимемо як $\theta \circ \lambda$.

Приклад 4.7. Нехай

$\theta = \{t_1/x_1,\ t_2/x_2,\ t_3/x_3\} = \{f(y)/x,\ y/z,\ g(d)/u\}$,

$\lambda = \{s_1/y_1,\ s_2/y_2\} = \{c/x,\ z/y\}$.

Тоді множина із означення добутку підстановок виглядатиме так:

$\{t_1\lambda/x_1,\ t_2\lambda/x_2,\ t_3\lambda/x_3,\ s_1/y_1,\ s_2/y_2\} =$

$= \{f(y)\lambda/x,\ y\lambda/z,\ g(d)\lambda/u,\ c/x,\ z/y\} = \{f(z)/x,\ z/z,\ g(d)/u,\ c/x,\ z/y\}$.

Але оскільки $t_2\lambda = x_2$, то $t_2\lambda/x_2$ (тобто z/z) викреслюємо з множини. Крім того, оскільки y_1 є серед $y_1 = x \in \{x, z, u\} = \{x_1, x_2, x_3\}$, то s_1/y_1 (тобто c/x) також потрібно викреслити.

Як результат отримаємо добуток

$$\theta \circ \lambda = \{f(z)/x,\ g(d)/u,\ z/y\}.$$

Приклад 4.8. Нехай

$\theta = \{t_1/x_1,\ t_2/x_2\} = \{f(y)/x,\ z/y\}$

$\lambda = \{s_1/y_1,\ s_2/y_2,\ s_3/y_3\} = \{a/x,\ b/y,\ y/z\}$.

Множина

$\{t_1\lambda/x_1,\ t_2\lambda/x_2,\ s_1/y_1,\ s_2/y_2, s_3/y_3\} =$

$= \{f(y)\lambda/x,\ z\lambda/y,\ a/x,\ b/y,\ y/z\} = \{f(b)/x,\ y/y,\ a/x,\ b/y,\ y/z\}$.

Із цієї множини вилучаємо елемент y/y і елементи a/x і b/y, бо $y_1 = x \in \{x, y\} = \{x_1, x_2\}$ і $y_2 = y \in \{x, y\} = \{x_1, x_2\}$.

Композиція $\theta \circ \lambda = \{f(b)/x,\ y/z\}$.

Зауваження.

1. $(\theta \circ \lambda) \circ \sigma = \theta \circ (\lambda \circ \sigma)$ {добуток підстановок асоціативний}

2. $\sigma \circ \varepsilon = \varepsilon \circ \sigma = \sigma$ {порожня підстановка є нейтральним елементом відносно множення}

3. $E(\theta \circ \lambda) = (E\theta)\lambda$.

Приклад 4.9. $E = P(x, f(y), g(x, z))$. Підстановки θ і λ візьмемо з прикладу 4.8.

$E(\theta \circ \lambda) = P(x, f(y), g(x, z))(\theta \circ \lambda) = P(f(b), f(y), g(f(b), y))$;

$(E\theta)\lambda = (P(x, f(y), g(x, z))\theta)\lambda = P(f(y), f(z), g(f(y), z))\lambda =$

$= P(f(b), f(y), g(f(b), y))$.

Уніфікатор σ множини літералів і термів називають *найзагальнішим уніфікатором*, якщо для будь-якого уніфікатора θ цієї множини існує підстановка λ така, що $\theta = \sigma \circ \lambda$.

Приклад 4.10. Для множини $\{P(x, f(a), g(z)),\ P(f(b), y, v)\}$ найзагальнішим уніфікатором є підстановка $\sigma = \{f(b)/x,\ f(a)/y,\ g(z)/v\}$. Якщо взяти інший уніфікатор, скажімо, $\theta = \{f(b)/x,\ f(a)/y,\ c/z,\ g(c)/v\}$, то для нього $\lambda = \{c/z\}$.

Якщо множина літералів або термів уніфікується, то найзагальніший уніфікатор існує. Це твердження ми доведемо нижче. Тепер подамо алгоритм знаходження найзагальнішого уніфікатора. Цей алгоритм називають *алгоритмом уніфікації*. Для викладення алгоритму нам потрібно ввести поняття множини неузгодженостей.

Нехай *M* – множина літералів або термів. Виділимо першу зліва позицію, у якій не для всіх літералів є один і той самий символ. Після цього для кожного літералу випишемо вираз, який починається з цього символу. (Таким виразом може бути сам літерал, атом або терм.) Множину отриманих виразів називають *множиною неузгодженостей в M*.

Приклад 4.11.

а) Нехай $M = \{P(x, f(y), a), P(x, u, g(y)), P(x, c, v)\}$. Перша зліва позиція, у якій не всі літерали мають один і той самий символ – це п'ята позиція. Справді, перші чотири позиції

$P\left(x,\right.$ є однаковими у всіх літералах. Отже, тут множина неузгодженостей складається з термів $f(y)$, u, c.

б) Нехай $M = \{P(x, y), \neg P(a, g(z))\}$, тоді множина неузгодженостей – це сама множина M.

в) Якщо $M = \{\neg P(x, y), \neg Q(a, v)\}$, то множина неузгодженостей є $\{P(x, y), Q(a, v)\}$.

4.4. Алгоритм уніфікації

4.3.1. Опис алгоритму уніфікації

Розглянемо алгоритм уніфікації та приклади його застосування.

Алгоритм уніфікації

Крок 1. Присвоїти $k := 0$, $M_k := M$, $\sigma_k := \varepsilon$.

Крок 2. Якщо множина M_k складається з одного літерала, то видати σ_k як найзагальніший уніфікатор і завершити роботу. Інакше знайти множину N_k неузгодженостей в M_k.

Крок 3. Якщо в множині N_k існує змінна v_k і терм t_k, який **не** містить v_k, то перейти до кроку 4. Інакше видати повідомлення, що множину M неможливо уніфікувати та завершити роботу.

Крок 4. Обчислити $\sigma_{k+1} = \sigma_k \circ \{t_k / v_k\}$, тобто ця підстановка отримується із σ_k заміною v_k на t_k і, можливо, додаванням елемента t_k / v_k. У множині M_k виконати заміну v_k на t_k і отриману множину літералів взяти як M_{k+1}. (Зазначимо, що $M_{k+1} = M\sigma_{k+1}$.)

Крок 5. Присвоїти $k := k + 1$ і перейти до кроку 2.

Приклад 4.12. Нехай $M = \{P(x, f(y)), P(a, u)\}$. Під час першої ітерації алгоритму буде знайдено підстановку $\sigma_1 = \{a / x\}$, а також множину неузгодженостей $N_0 = \{x, a\}$. Тоді множина $M_1 = \{P(a, f(y)), P(a, u)\}$. Під час другої ітерації алгоритму буде обчислено підстановку $\sigma_2 = \{a / x, f(y)/u\}$ і отримано множину $M_2 = \{P(a, f(u))\}$. Оскільки M_2 складається з одного літерала, то алгоритм завершить роботу і видасть σ_2.

Приклад 4.13. Нехай $M = \{P(x, f(y)), P(a, b)\}$. Під час першої ітерації алгоритму буде знайдено підстановку $\sigma_1 = \{a / x\}$ і $M_1 = \{P(a, f(y)), P(a, b)\}$. На третьому кроці другої ітерації буде видано повідомлення про неможливість уніфікації множини M, оскільки множина неузгодженостей $N_1 = \{f(y), a\}$ не містить змінної.

4.3.2. Обґрунтування алгоритму уніфікації

Зазначимо, що під час виконання кроку 4 із множини M_k вилучається одна із змінних (змінна v_k), а нової змінної не виникає. Це означає, що алгоритм уніфікації завжди завершує роботу, оскільки крок 4 не може виконуватися нескінченно. Достатньо очевидно, що коли алгоритм завершує роботу на кроці 3, то множину M уніфікувати неможливо. Також зрозуміло, що коли алгоритм завершує роботу на кроці 2, то σ_k – уніфікатор множини M. Але те, що σ_k – найзагальніший уніфікатор, довести складніше.

Теорема 4.2. Нехай M – скінченна непорожня множина літералів. Якщо множину M можна уніфікувати, то алгоритм уніфікації завершує роботу на кроці 2, і підстановка σ_k, яку видає алгоритм, є найзагальнішим уніфікатором множини M.

Доведення. Нехай θ – якийсь уніфікатор множини M. Індукцією за k доведемо існування підстановки λ_k такої, що $\theta = \sigma_k \circ \lambda_k$ (тут σ_k – результат роботи алгоритму).

<u>База індукції.</u> Якщо $k = 0$, то $\sigma_k = \varepsilon$. Тоді як λ_k візьмемо θ.

<u>Крок індукції.</u> Припустимо, що для всіх значень k, які задовольняють нерівності $1 \le k \le l$, існує підстановка λ_k така, що $\theta = \sigma_k \circ \lambda_k$.

Якщо $M\sigma_l$ містить один літерал, то під час наступної ітерації алгоритм зупиниться на кроці 2. Тоді σ_l буде найзагальнішим уніфікатором, бо $\theta = \sigma_l \circ \lambda_l$.

Нехай $M\sigma_l$ містить більше одного літерала. Тоді алгоритм знайде множину неузгодженостей N_l. Підстановка λ_l має уніфікувати множину N_l, бо $\theta = \sigma_l \circ \lambda_l$ – уніфікатор множини M. Оскільки множину неузгодженостей N_l можна уніфікувати, то вона містить (хоча б одну) змінну v.

Нехай t – терм із N_l, відмінний від v. Множина N_l уніфікується підстановкою λ_l, тому $v\lambda_l = t\lambda_l$. Звідси випливає, що терм t не містить змінної v. Можна вважати, що на кроці 4 алгоритму для отримання σ_{l+1} використано заміну v на t, тобто $\sigma_{l+1} = \sigma_l \circ \{t / v\}$. Із рівності $v\lambda_l = t\lambda_l$ випливає, що λ_l містить елемент $t\lambda_l / v$.

Нехай $\lambda_{l+1} = \lambda_l - \{t\lambda_l / v\}$. Тоді $t\lambda_{l+1} = t\lambda_l$, бо терм t не містить змінної v. Далі маємо такі рівності:

$$\{t / v\} \circ \lambda_{l+1} = \lambda_{l+1} \cup \{t\lambda_{l+1} / v\} = \lambda_{l+1} \cup \{t\lambda_l / v\} = \lambda_l .$$

Це означає, що $\lambda_l = \{t / v\} \circ \lambda_{l+1}$. Отже,

$$\theta = \sigma_l \circ \lambda_l = \sigma_l \circ \{t / v\} \circ \lambda_{l+1} = \lambda_{l+1} \circ \sigma_{l+1} .$$

Звідси випливає, що для довільного k існує підстановка λ_k така, що $\theta = \sigma_k \circ \lambda_k$. Оскільки множину M можна уніфікувати, то алгоритм має закінчити роботу на кроці 2. Тоді остання підстановка σ_k буде уніфікатором множини M (див. текст перед формулюванням теореми). Але оскільки для довільного уніфікатора θ існує така підстановка λ_k, що $\theta = \sigma_k \circ \lambda_k$, то σ_k – найзагальніший уніфікатор множини M. Теорему доведено.