К лаб.14 Унификация 2-х термов

Унификация двух термов – это основной шаг доказательства. В процессе работы система выполняет большое число унификаций.

**Унификация –** операция, которая позволяет формализовать процесс логического вывода (наряду с правилом резолюции). С практической точки зрения – это основной вычислительный шаг работы программы.

Унификация - попытка "увидеть одинаковость", может завершаться успехом или тупиковой ситуацией (неудачей) , тогда включается механизм отката к предыдущему шагу.

Процесс унификации запускается автоматически, но пользователь имеет право запустить его вручную с помощью утверждения T1 = T2 (это, в угоду пользователям, но только внешнее отступление от синтаксиса Prolog =(Т1, Т2) ).

**Определения:**

Терм S называется **более общим** чем терм Т, если T является примером S, а S не является примером T.

S называется наиболее **общим примером** T1 и T2, если S такой их общий пример, который является более общим по отношению к любому другому их примеру. Например:

Пусть есть терм T=f(Х1 Х2 Хn). Если одну переменную конкретизировать значением, то это будет пример: f(a, Х2, Хn), но более общий чем пример: f(a, b, Хn).

**Унификатор двух термов** – подстановка, которая будучи применена к двум термам даст одинаковый результат.

**Наиболее общий унификатор двух термов** - унификатор, соответствующий наиболее общему их примеру.

**Теорема**: если два терма унифицируемы, то существует единственный, с точностью до переименования переменных, наиболее общий унификатор.

**Алгоритм унификации**

Алгоритм встроен в систему и для своей работы **использует три области** памяти:

**стек**: хранит равенства, унификацию которых надо доказать;

**рабочее поле**: равенство вытащенное из стека, которое сейчас доказываем;

**результирующая ячейка** памяти: накапливает наиболее общий унификатор.

**и флаг** - переменная **'неудача'**: равна **1**, если возникла тупиковая ситуация иначе **0.**

шаг:

**начало:** Сопоставляем Т1=Т2

занести в стек **Т1=Т2**

положить **неудача=0**

пока стек не пуст – **цикл**:

1. **считать** из стека **в рабочую область** очередное равенство S=Т
2. **обработать считанное** по правилам:

а) если S и Т несовпадающие константы, то неудача=1, и выход из цикла

б) если одинаковые константы то *следующий шаг цикла*

в) если S переменная и Т терм содержащий S, то неудача=1, и выход из цикла

г) если S переменная и Т терм НЕ содержащий S, то *отыскать в стеке и* в *результирующей ячейке* все вхождения S и *заменить* на Т. *Добавить в результирующую* *ячейку равенство S=Т. Следующий шаг цикла*

д) если S и Т составные термы с разными функторами или разными арностями, то неудача=1, выход из цикла

е) если S и Т составные термы с одинаковыми функторами и арностью: S=f(s1 s2 .. sm); T=f(t1 t2 ... tm), то *занести в стек* равенство S1=T1, S2=T2 ... Sm=Tm.

1. **очистить рабочее поле**

– **конец цикла**

**если** неудача = 1 то *унификация невозможна*

**если** неудача = 0 то *унификация успешна, результирующая ячейка содержит наиболее общий унификатор*

**конец.**

Пример работы алгоритма унификации двух термов:

t(X, p(X,Y)) = t(q(W), p(q(a), b))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| шаг унифи ка ции | результирующая ячейка | рабочее поле | пункт алгоритма | стек |
| 0 |  |  | 1. | t(X, p(X,Y)) =  t(q(W), p(q(a), b)) |
| 1 |  | t(X, p(X,Y)) = t(q(W), p(q(a), b)) ----> | е) | X=q(W)  p(X,Y)=p(q(a),b) |
| 2 | X=q(W) | <--------- X=q(W) ---------> | г) | p(q(W),Y)=p(q(a),b) |
| 3 | X=q(W) | p(q(W),Y)=p(q(a),b) ------> | е) | q(W)=q(a)  Y=b |
| 4 | X=q(W) | q(W)=q(a) ------> | е) | W=a  Y=b |
| 5 | X=q(а), W=a | <------- W=a | г) | Y=b |
| 6 | X=q(а), W=a, Y=b | <----- Y=b | г) |  |
| Вы  вод: | **подстановка** | Т.к. стек пуст – **успех** и  в рез. ячейке подстановка | | |

Фактически, формально, работа алгоритма унификации заключается в по парном сопоставлении термов и попытке построить для них общий пример, для чего и нужна подстановка. Например, для поиска ответа на вопрос система должна найти подходящее знание. А знание зафиксировано в заголовке правила. Т.е. система должна подобрать подходящее правило (подходящий заголовок) (назначение алгоритма унификации). И система должна «понять» формально, что заголовок подходит. Для этого она строит унификатор – подстановку (побочный эффект работы алгоритма унификации). В результате применения подстановки некоторые переменные конкретизируются значениями, которые (значения) могут и будут далее использованы при доказательстве истинности тела выбранного правила (что тоже выполняется с использованием алгоритма унификации). Т.е. значения переменных переходят на следующий шаг доказательства. Таким образом, с помощью алгоритма унификации происходит двунаправленная передача параметров процедурам. Например, из внешнего мира в программу для дальнейшего использования или из программы во внешний мир – значения интересующего нас параметра.

!!!! В лабораторных работах будем использовать только конъюнкцию (термы в теле разделяются запятыми). И простой вопрос.