Таблица к заданию

Вопрос: *sum([1, 2, 3], Result)*

№ шага	Сравнение термы, результат, подстановка, если есть	Дальнейшие действия, прямой ход или откат (к чему приводит?)
0		Состояние резольвенты: sum([1, 2, 3], Result)
1	Сравнение: sum([1, 2, 3], Result) == lenRec([], Res, CurRes) Унификация: неуспешна (несовпадение функторов)	Прямой ход Переход к следующему предложению
2-5		
6	Сравнение: sum([1, 2, 3], Result) == sum(List, Result) Унификация: успешна Подстановка: {List = [1, 2, 3], Result = Result}	Образование новой резольвенты: 1. Редукция верхней подцели: замена sum([1, 2, 3], Result) телом найденного правила Получена конъюнкция целей: sumRec(List, Result, 0) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: sumRec([1, 2, 3], Result, 0)
7	Сравнение: sumRec([1, 2, 3], Result, 0) == lenRec([], Res, CurRes)	Прямой ход Переход к следующему предложению
	Унификация: неуспешна (несовпадение функторов)	
8-10		
11	Сравнение: sumRec([1, 2, 3], Result, 0) == sumRec([H T], Res, CurRes) Унификация: успешна Подстановка: {H = 1, T = [2, 3], Res = Result, CurRes = 0}	Образование новой резольвенты: 1. Редукция верхней подцели: замена sumRec([1, 2, 3], Result, 0) телом найденного правила Получена конъюнкция целей: TmpRes = CurRes + H, sumRec(T, Res, TmpRes) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: TmpRes = 0 + 1, sumRec([2, 3], Result, TmpRes)

{Res = 6} Встречен системный предикат отсечения,	13-18 19-24	Сравнение:	
Сравнение: SumRec([], Result, 6) == sumRec([], Res, CurRes) Унификация: успешна Подстановка: (Res = Result, CurRes = 6) 29	25	sumRec([], Result, 6) == lenRec([], Res, CurRes)	
30 1. Редукция верхней подцели: замена зит мес ([], Res.ult, 6) телом найденного правила 1. Редукция верхней подцели: замена зит мес ([], Res.ult, 6) телом найденного правила 1. Подучена конъюнкция целей: Res = Result, CurRes = 6} 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: Res = 6, ! 29 Сравнение: Res = 6 Результат: успех 1. Редукция верхней подцели: замена Res = 6, так как результат – истина 1. Подстановка: (Res = 6) 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: / выстранный видено: формирование подстановки к полученной конъюнкции целей. Новое состояние резольвенты: / выстранный видено: формирование подстановки в качестве побочного эффект (Res = 6) 30 1 Решение найдено: формирование подстановки в качестве побочного эффект (Res = 6) Встречен системный предикат отсечения, откат с отсечения, откат с отсечения остаточных предложен процедуры относительно шага 28 31 Сравнение: миже ([], Result, 6) == sum(List, Result) Прямой ход Переход к следующему предложению	26-27		
Res = 6 1. Редукция верхней подцели: замена Res = 6, так как результат – истина Подстановка: {Res = 6} 2. Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты: ! 1. Решение найдено: формирование подстановки в качестве побочного эффект {Res = 6} Встречен системный предикат отсечения, откат с отсечением остаточных предложен процедуры относительно шага 28 31 Сравнение: sumRec([], Result, 6) == sum(List, Result)	28	sumRec([], Result, 6) == sumRec([], Res, CurRes) Унификация: успешна Подстановка:	 Редукция верхней подцели: замена sumRec([], Result, 6) телом найденного правила Получена конъюнкция целей: Res = CurRes, ! Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты:
подстановки в качестве побочного эффект {Res = 6} Встречен системный предикат отсечения, откат с отсечением остаточных предложен процедуры относительно шага 28 Сравнение: SumRec([], Result, 6) == sum(List, Result) Прямой ход Переход к следующему предложению	29	Res = 6 Результат: успех Подстановка:	 Редукция верхней подцели: замена Res = 6, так как результат – истина Применение подстановки к полученной конъюнкции целей. Новое состояние резольвенты:
sumRec([], Result, 6) == sum(List, Result)Переход к следующему предложению	30	!	подстановки в качестве побочного эффекта: {Res = 6} Встречен системный предикат отсечения, откат с отсечением остаточных предложений
	31	sumRec([], Result, 6) == sum(List, Result)	
32-35	32-35		

36		Конец БЗ Обратная трассировка: 1) Отмена крайней редукции (на шаге 24) 2) Восстановление предыдущего состояния резольвенты (с шага 23): sumRec([3], Res, 3) 3) Реконкретизация переменных с шага 24: {H = 3, T = [], Res = Result, CurRes = 6}	
		Переход к следующему предложению относительно шага 24	
Решения далее найдены не будут, в итоге система через несколько восстановлений восстановит резольвенту до шага 0			
51		Конец Б3 Обратная трассировка: 1) Отмена крайней редукции (на шаге 0) 2) Восстановление предыдущего состояния резольвенты: резольвента пуста	
		Завершение работы На вопрос удалось ответить утвердительно 1 подстановка были возвращены в качестве побочного эффекта	