

	Date Page
11/123	20h8 - Proof by Unification:
	by Unification:
	Aim to implement with
	Am so implement unification in First ordu logice
	import re
	dy get Athibute (engression):
	any thin = energing and + (" a ") 5
	engression = "(" goln (engression)
	e grama solut (1)
	engrenien=")" gin (engrenien)
	attributes = enpression splet (5")
	relum aldribulis
	I you a story on the house and a second
	dy get Initial Predicale (enpression): zeten engrania: split ("(")[")
	geteen engrassion split ("(") (")
	the state of the s
	dy is (onstand (char):
	Return chen is upper and liver (chan) ==1.
	dy is Variable (char): Julian Char is lower () and lan (char) = = 1.
	rum car so source on a will char
	dly reglan Afribules (enp, old, new):
	attribute = get Athibutes (eng)
	predicate = get Initial Predicate (enp).
	for intelex val en enumerate (altribute):
	Arul = = old.
	athir de (al. 1) - ner
	return predicale + "(" + ", " poin (attributes) +")"
	- Junior J
	And American Commencer American American American
	ter apple (eng, substitutions):
	Les apple (eng, substitutions): for substitution in substitution:



	Date Page
	eng = replace Attribute Centraldraw
	Adun enp.
	dy cheek auro (rang enp):
	hod(var) = -1
	Estern Jo Falm
	rutum True.
	dy get First Part (engression):
	activities = get Attribute (engression)
	situr attributes (6)
	dy get Remaining Past (engrapion):
	predicale = get InitialPredicate (engrania)
	new Engression - predicate + "" " . join
	(allributes (1:)) + ")"
1	Euteur new Engression.
	dy very (onp), enp):
1	yenr1 enp
	Refer ()
	the state of the s
	Is is Conjunt (very 1) and is Constant (congr):
	if cny 1 1 = eng 2.
	point (f" { comps and fenges on the
	Constant (a de la 120/1)
	Retrem (annot bramped')
	- CATANA PRO
	if is (andant (conf 1).
	notion (Com a a series)
	return ((engr 2, engr)) ignut chech lungs (cong 3, engr) clu ()
	ig is Constant (enp.):
	- James (ong).
- 1	



Date Page
return ((engr, engr)).
is es Variable (enr 2):
gettin [(enr 2 enr 1)] is not dead
is es Variable (eng 2): setten [(eng 2, eng 1)) is not deck check Quers (enp, eng 1) else ()
"if get Initial Predicals Congs) 1 = get Initial Pre
die fa (e)
print ("(annot be umpred as the predicates. do not march!") relan ().
do not march (')
retun ().
attribute (auts = lin (get Attribute's (engy))
attitute (aut = len (get Attribute (enge))
is altribute (out 1 = Attribute Cout 2:
print (f" Length of allribute factorisele Cast
I) and factorimete (and 2)
do not masch (annot be unfied')
rition ()
head = get First Part (eng +)
head = get First Part (engr).
initial substitution = unify (head) heads)
if not inital substitution:
return ().
il and to Cartain I'm
If attribute (aut 1 = = 1'. nturn initial Substitution
man man sansporter
Quil (Reade 1)
tuil 1 = get Remaining part (enp1)
suit = get Remainingfast (engs)
if mitial Engelitation.) = ():
puil 2 = copply (buil j initial substitution) puil 2 = copply (buil 2 initial substitution
Jull 1 = apply (Jull 2 , milled & so the with

remaining substitution = Unify (suit 1, kut) reten inival Substitulis + romainy Substitu if - nam == " _ main -- ", print ("the the first engression") es = inject() print (Enter the seemed engreenin") er=input() rubo titutin = unify (e1, e1) print ("The oubstitulion au: ") Print (['/'. joh (substitution) for outstitute in). out 1: Inte the first enpression Know (f(n), y) Inta the second engravion Anus (J, roha) The yobstitulions an: (T/fen), Tohn/y Rule the first engrapion lik (A,y) guter the second expression like (k, g(n)) A and k au constants (annot be unjud The substitutions au.



	Date_Page
	Entu the first engression.
	Enter the second engression
	Cannot be sended
	my as the predicales de
	(unnot be unfred as the predicates de The substitutions au:
	().
	out out: Intu your first engression
	knows (f(x), y)
	Satu the
	Intu the second empronion. Anous (I, John)
	The out it for
	the outstitutions au: ('J /f(n) , 'John / y').
	John Jy J.
	8.4. 41. 61.
	Sintu the first engression
	like (A, y)
1	enty the second supremin.
+	lite(k, g(n))
-	A and k au constants - (annot be unified.
-	The substitution are
	L_J.
	Entu the first engrusion.
	Shedent (n)
	gutu the occord ingression
	Kaulu (Kon)
	Cannot be unified on the predicates do not
	mark.
	The substitutions are:



	of lgorithm:
	of eggrinn.
	partial Unity (x,y,0) where a substitution to ware n and y identitical. a constant, list, or compand,
	of and it identifical
	mputs: n, a variable, a constant, list, or compound,
A	enpression.
	y, a variable, constant, sun or companie
	of the subsetitulin buch her myen.
	(aftignal defeuets to emply).
	11 value - Julius Hen Julian gracion
	eln y n-y hen return O
	In it viviala ((k) min runt anity or city)
	ela ej variable l'(g) then sety in Onigy Var (y, u, o)
	un ej compound ((n) and compound?(y) then new Unique (n, mgs y 1 mgs, Vnipy (2, op, y, op, o))
	eln ij (15 T? (N) and (15 T? (Y) then.
	return Unity (4, KEST, y, KEST, Unity (4, FIRST Y, FIRST
	main only (4, assign)
	elu Ketter n Julius.
	The result of the second of th
	parelin Unity - Var (var, 2,0) return a sabolitulin
	is 5 var Aval & co then return Unity (val, x, o).
	eln es & n/ val 1 & a the return Unity Com note
	eln if Oun - Chech? (Var, n) the return fulling
	eln ej & n/vul & E o then return Unity (val, x, o). eln ej & n/vul & E o then return Unity Cvar vulo eln ej Ouw - Chech? (Var, n) then return fuelou. eln return edd & vul/ n & to o
	A MAN A MANAGEMENT AND A
	Annual Control of the