1 –

insere três fatos

q(a,b) - insere na database;

q(1,2) - insere no final da database;

q(foo,blug) - insere no começo da database.

Banco de dados :

q(foo, blug).

q(a, b).

q(1, 2).

Retira o q(1,2) e adiciona no final p(X) :- h(X)

Banco de dados:

q(foo, blug).

q(a, b).

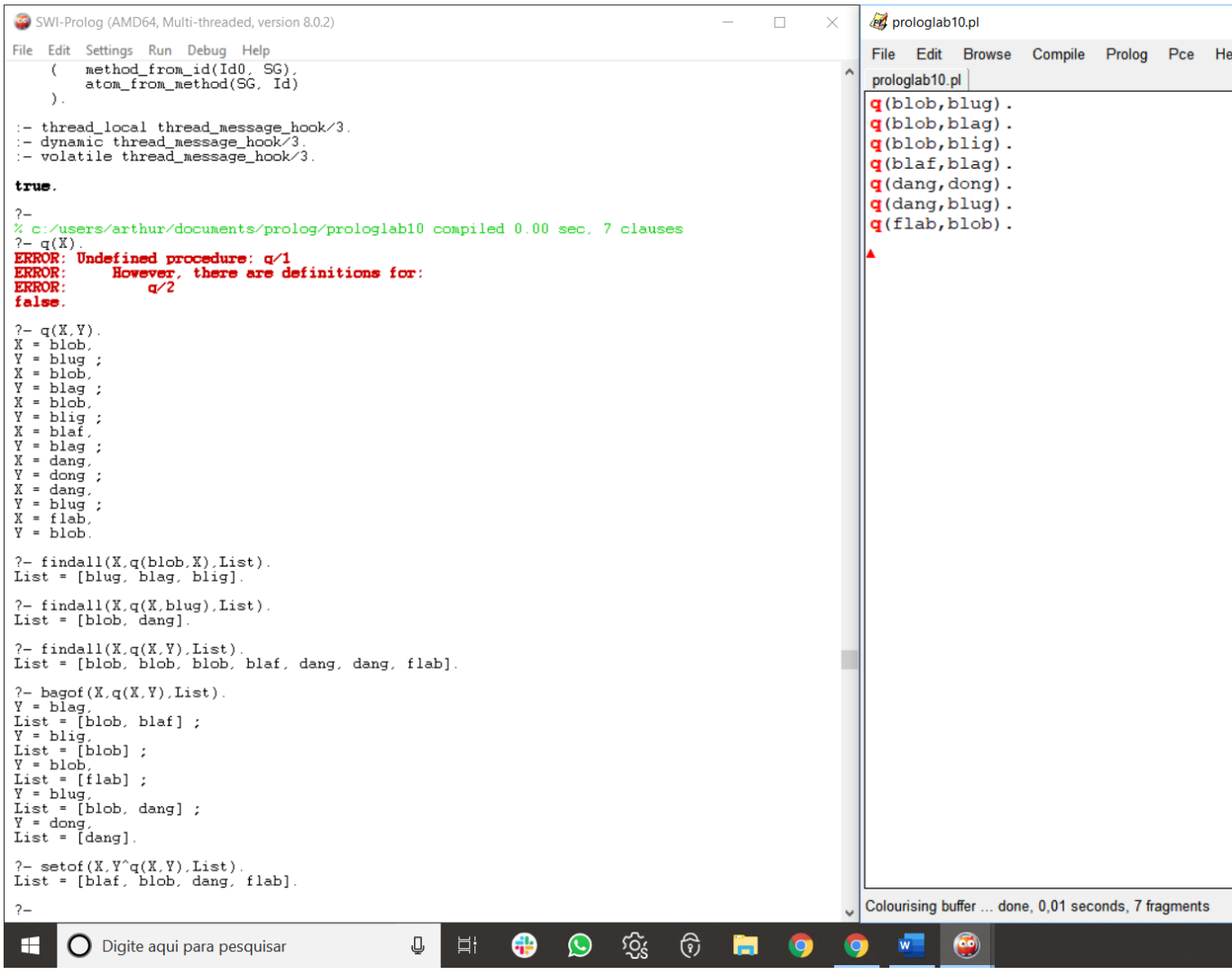
p(A) :- h(A).

Retira todos os fatos com q(_).

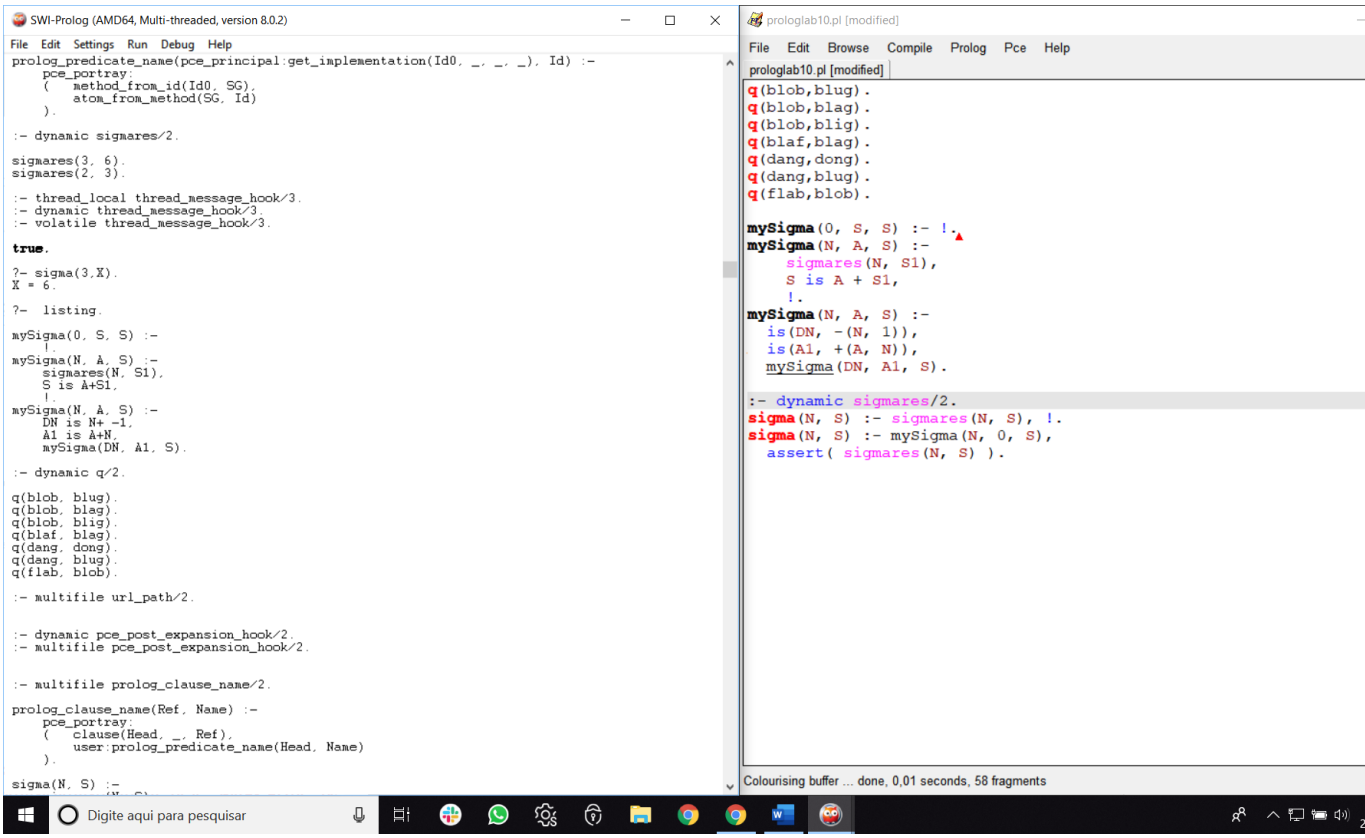
Banco de dados:

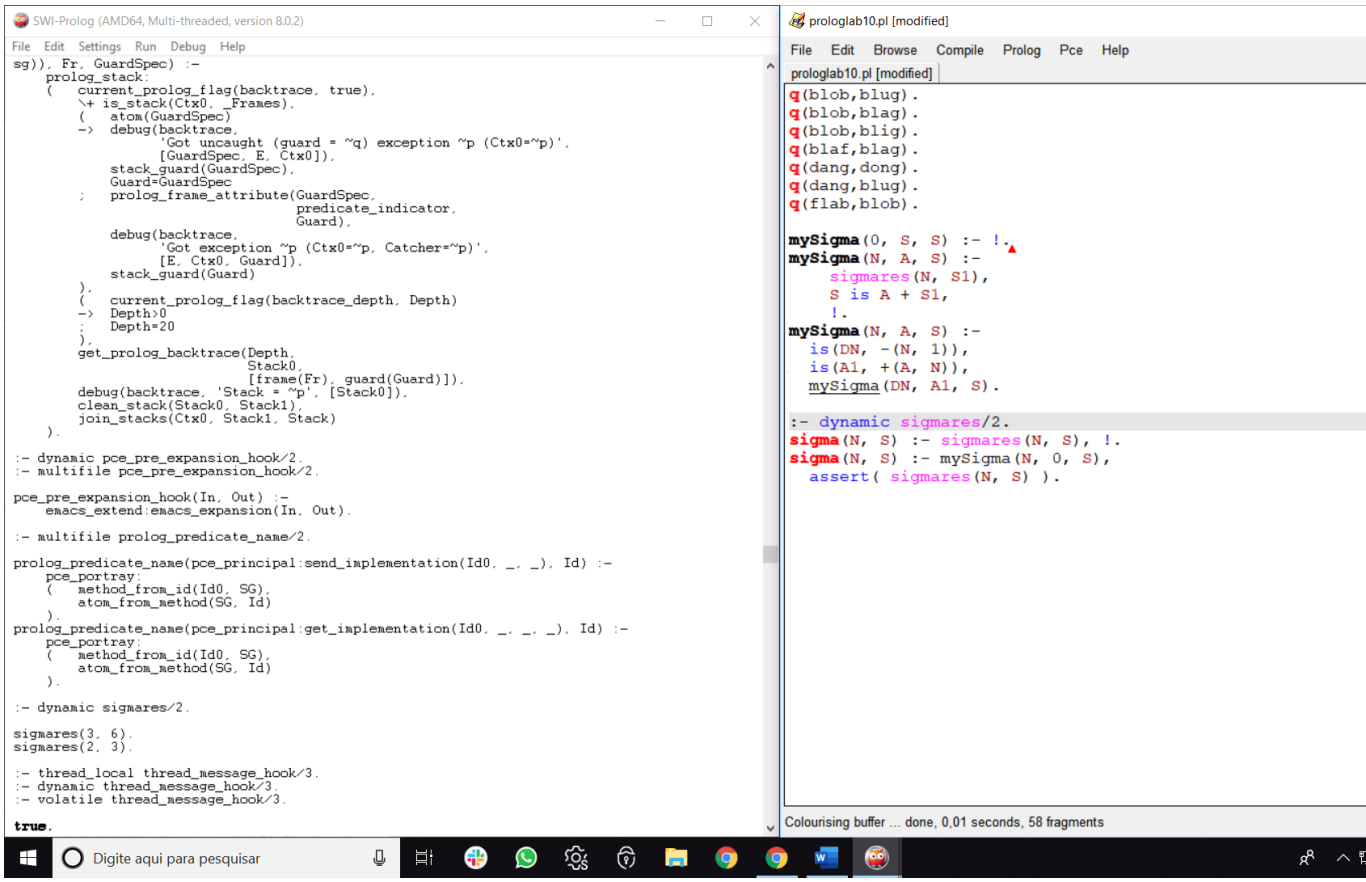
p(A) :- h(A).

2-


SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
(method_from_id(Id0, SG),
atom_from_method(SG, Id)
).
:- thread_local thread_message_hook/3.
:- dynamic thread_message_hook/3.
:- volatile thread_message_hook/3.
true.
?-
% c:/users/arthur/documents/prolog/prologlab10 compiled 0.00 sec, 7 clauses
?- q(X).
ERROR: Undefined procedure: q/1
ERROR: However, there are definitions for:
ERROR: q/2
false.
?- q(X,Y).
X = blob,
Y = blug ;
X = blob,
Y = blag ;
X = blob,
Y = blig ;
X = blaf,
Y = blag ;
X = dang,
Y = dong ;
X = dang,
Y = blug ;
X = flab,
Y = blob.
?- findall(X,q(blob,X),List).
List = [blug, blag, blig].
?- findall(X,q(X,blug),List).
List = [blob, dang].
?- findall(X,q(X,Y),List).
List = [blob, blob, blob, blaf, dang, dang, flab].
?- bagof(X,q(X,Y),List).
Y = blag,
List = [blob, blaf] ;
Y = blig,
List = [blob] ;
Y = blob,
List = [flab] ;
Y = blug,
List = [blob, dang] ;
Y = dong,
List = [dang].
?- setof(X,Y^q(X,Y),List).
List = [blaf, blob, dang, flab].
?-
prologlab10.pl
File Edit Browse Compile Prolog Pce
prologlab10.pl
q(blob,blug).
q(blob,blag).
q(blob,blig).
q(blaf,blag).
q(dang,dong).
q(dang,blug).
q(flab,blob).
Colourising buffer ... done, 0,01 seconds, 7 fragments
Digite aqui para pesquisar

3 –


SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
prolog_predicate_name(pce_principal:get_implementation(Id0, _, _, _), Id) :-
pce_portray:
(method_from_id(Id0, SG),
atom_from_method(SG, Id)
).
:- dynamic sigmares/2.
sigmares(3, 6).
sigmares(2, 3).
:- thread_local thread_message_hook/3.
:- dynamic thread_message_hook/3.
:- volatile thread_message_hook/3.
true.
?- sigma(3,X).
X = 6.
?- listing.
mySigma(0, S, S) :-
!.
mySigma(N, A, S) :-
sigmares(N, S1),
S is A+S1,
!.
mySigma(N, A, S) :-
DN is N+ -1,
A1 is A+N,
mySigma(DN, A1, S).
:- dynamic q/2.
q(blob, blug).
q(blob, blag).
q(blob, blig).
q(blaf, blag).
q(dang, dong).
q(dang, blug).
q(flab, blob).
:- multifile url_path/2.
:- dynamic pce_post_expansion_hook/2.
:- multifile pce_post_expansion_hook/2.
:- multifile prolog_clause_name/2.
prolog_clause_name(Ref, Name) :-
pce_portray:
(clause(Head, _, Ref),
user:prolog_predicate_name(Head, Name)
).
sigma(N, S) :-
prologlab10.pl [modified]
File Edit Browse Compile Prolog Pce Help
prologlab10.pl [modified]
q(blob,blug).
q(blob,blag).
q(blob,blig).
q(blaf,blag).
q(dang,dong).
q(dang,blug).
q(flab,blob).
mySigma(0, S, S) :- !.
mySigma(N, A, S) :-
sigmares(N, S1),
S is A + S1,
!.
mySigma(N, A, S) :-
is(DN, -(N, 1)),
is(A1, +(A, N)),
mySigma(DN, A1, S).
:- dynamic sigmares/2.
sigma(N, S) :- sigmares(N, S), !.
sigma(N, S) :- mySigma(N, 0, S),
assert(sigmares(N, S)).
Colourising buffer ... done, 0,01 seconds, 58 fragments
Digite aqui para pesquisar


SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
sg)), Fr, GuardSpec) :-
prolog_stack:
(current_prolog_flag(backtrace, true),
\+ is_stack(Ctx0, _Frames),
(atom(GuardSpec)
-> debug(backtrace,
'Got uncaught (guard = ~q) exception ~p (Ctx0=~p)',
[GuardSpec, E, Ctx0]),
stack_guard(GuardSpec),
Guard=GuardSpec
; prolog_frame_attribute(GuardSpec,
predicate_indicator,
Guard),
debug(backtrace,
'Got exception ~p (Ctx0=~p, Catcher=~p)',
[E, Ctx0, Guard]),
stack_guard(Guard)
),
(current_prolog_flag(backtrace_depth, Depth)
-> Depth>0
; Depth=20
),
get_prolog_backtrace(Depth,
Stack0,
[frame(Fr), guard(Guard)]),
debug(backtrace, 'Stack = ~p', [Stack0]),
clean_stack(Stack0, Stack1),
join_stacks(Ctx0, Stack1, Stack)
).
:- dynamic pce_pre_expansion_hook/2.
:- multifile pce_pre_expansion_hook/2.
pce_pre_expansion_hook(In, Out) :-
emacs_extend:emacs_expansion(In, Out).
:- multifile prolog_predicate_name/2.
prolog_predicate_name(pce_principal:send_implementation(Id0, _, _), Id) :-
pce_portray:
(method_from_id(Id0, SG),
atom_from_method(SG, Id)
).
prolog_predicate_name(pce_principal:get_implementation(Id0, _, _, _), Id) :-
pce_portray:
(method_from_id(Id0, SG),
atom_from_method(SG, Id)
).
:- dynamic sigmares/2.
sigmares(3, 6).
sigmares(2, 3).
:- thread_local thread_message_hook/3.
:- dynamic thread_message_hook/3.
:- volatile thread_message_hook/3.
true.
prologlab10.pl [modified]
File Edit Browse Compile Prolog Pce Help
prologlab10.pl [modified]
q(blob,blug).
q(blob,blag).
q(blob,blig).
q(blaf,blag).
q(dang,dong).
q(dang,blug).
q(flab,blob).
mySigma(0, S, S) :- !.
mySigma(N, A, S) :-
sigmares(N, S1),
S is A + S1,
!.
mySigma(N, A, S) :-
is(DN, -(N, 1)),
is(A1, +(A, N)),
mySigma(DN, A1, S).
:- dynamic sigmares/2.
sigma(N, S) :- sigmares(N, S), !.
sigma(N, S) :- mySigma(N, 0, S),
assert(sigmares(N, S)).
Colourising buffer ... done, 0,01 seconds, 58 fragments
Digite aqui para pesquisar