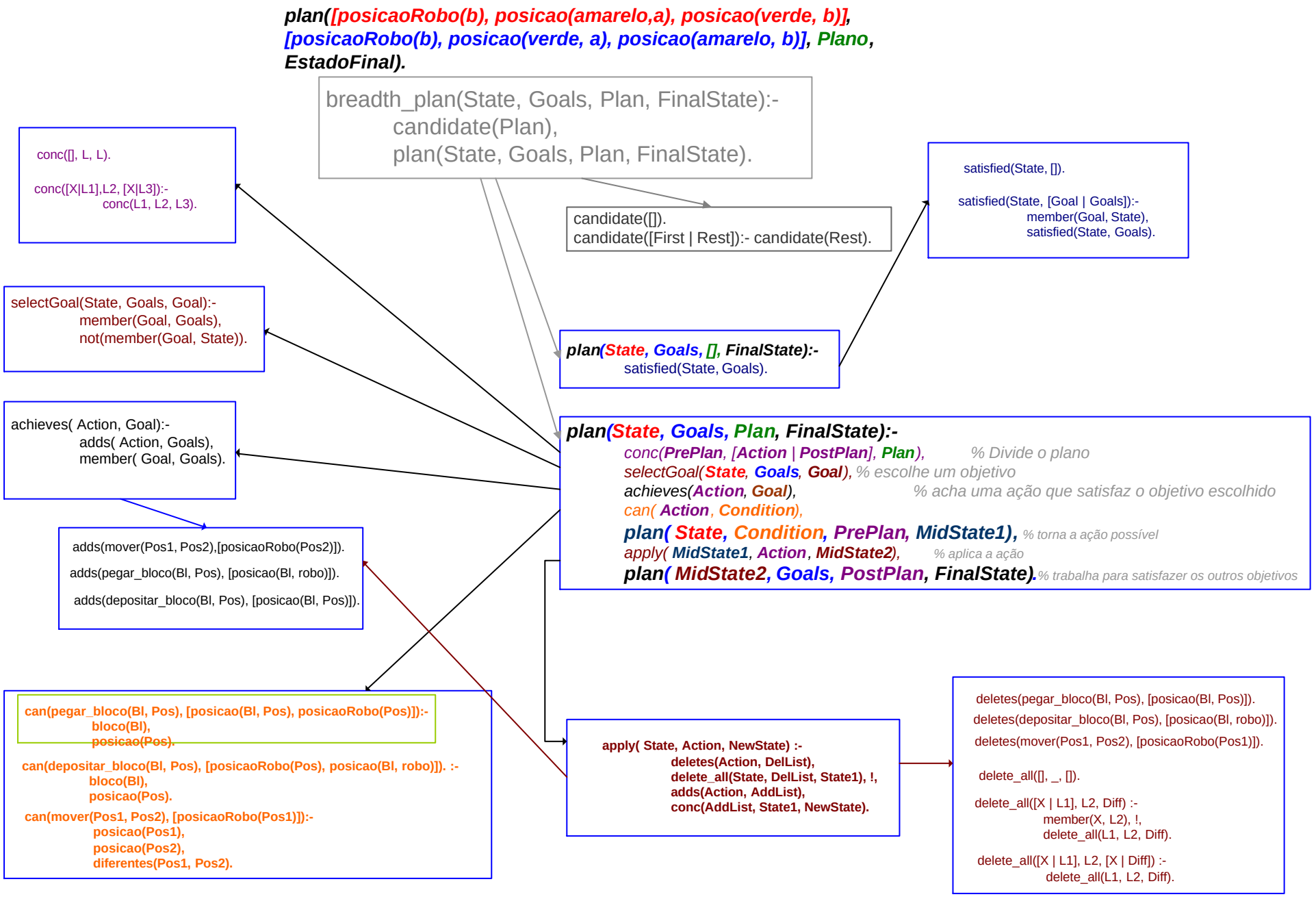
plan([posicaoRobo(b), posicao(amarelo,a), posicao(verde, b)], [posicaoRobo(b), posicao(verde, a), posicao(amarelo, b)], Plano, EstadoFinal).
breadth_plan(State, Goals, Plan, FinalState):-
candidate(Plan),
plan(State, Goals, Plan, FinalState).
conc([], L, L).
conc([X|L1],L2, [X|L3]):-
conc(L1, L2, L3).
candidate([]).
candidate([First | Rest]):- candidate(Rest).
satisfied(State, []).
satisfied(State, [Goal | Goals]):-
member(Goal, State),
satisfied(State, Goals).
selectGoal(State, Goals, Goal):-
member(Goal, Goals),
not(member(Goal, State)).
plan(State, Goals, [], FinalState):-
satisfied(State, Goals).
achieves(Action, Goal):-
adds(Action, Goals),
member(Goal, Goals).
plan(State, Goals, Plan, FinalState):-
conc(PrePlan, [Action | PostPlan], Plan), % Divide o plano
selectGoal(State, Goals, Goal), % escolhe um objetivo
achieves(Action, Goal), % acha uma ação que satisfaz o objetivo escolhido
can(Action, Condition),
plan(State, Condition, PrePlan, MidState1), % torna a ação possível
apply(MidState1, Action, MidState2), % aplica a ação
plan(MidState2, Goals, PostPlan, FinalState). % trabalha para satisfazer os outros objetivos
adds(mover(Pos1, Pos2),[posicaoRobo(Pos2)]).
adds(pegar_bloco(Bl, Pos), [posicao(Bl, robo)]).
adds(depositar_bloco(Bl, Pos), [posicao(Bl, Pos)]).
can(pegar_bloco(Bl, Pos), [posicao(Bl, Pos), posicaoRobo(Pos)]):-
bloco(Bl),
posicao(Pos).
can(depositar_bloco(Bl, Pos), [posicaoRobo(Pos), posicao(Bl, robo)]). :-
bloco(Bl),
posicao(Pos).
can(mover(Pos1, Pos2), [posicaoRobo(Pos1)]):-
posicao(Pos1),
posicao(Pos2),
diferentes(Pos1, Pos2).
apply(State, Action, NewState) :-
deletes(Action, DelList),
delete_all(State, DelList, State1), !,
adds(Action, AddList),
conc(AddList, State1, NewState).
deletes(pegar_bloco(Bl, Pos), [posicao(Bl, Pos)]).
deletes(depositar_bloco(Bl, Pos), [posicao(Bl, robo)]).
deletes(mover(Pos1, Pos2), [posicaoRobo(Pos1)]).
delete_all([], _, []).
delete_all([X | L1], L2, Diff) :-
member(X, L2), !,
delete_all(L1, L2, Diff).
delete_all([X | L1], L2, [X | Diff]) :-
delete_all(L1, L2, Diff).