

# Um Ambiente para Desenvolver Programação em Lógica Baseado no Paradigma de Estilo Literário

VII Semana de Iniciação Científica

Aluna: Flávia Peligrinelli Ribeiro

Orientadora: Mariza A. S. Bigonha



# Objetivo

Implementar um sistema de projeções para o sistema HyperPro Básico



# Hyperpro

- Ambiente de documentação e desenvolvimento de programas em lógica com restrições.
- Utiliza a ferramenta Thot



# Thot

- Editor de texto estruturado
- WYSIWYG (*what you see is what you get*)
- Facilidades de hypertexto



# Linguagens do Thot

- S - define a estrutura do documento
- P - define a apresentação do documento
- T - traduz de documentos Thot para outros formalismos
- A - define aplicações



# Programação Literária

Programas devem ser escritos não somente para os computadores executarem mas também para os seres humanos entenderem



# Programação Literária

Em um único arquivo:

- documentação
- código

# Programação Literária

## 1 Introduction

This example shows the methodology of writing a program, using hyper logic programming. The problem is to place  $n$  queens on an  $n \times n$  chessboard so that there is no couple of queens threatening each other.

## 2 Solution in Prolog

`solution/2`<sup>[cpr]</sup>

`Versions: Version1`<sup>[\*]</sup>

`solution-1` is a list of integers in  $[1..solution-2]$ , of length `solution-2`, denoting the positions of `solution-2` reciprocally non-attacking queens on the chessboard.

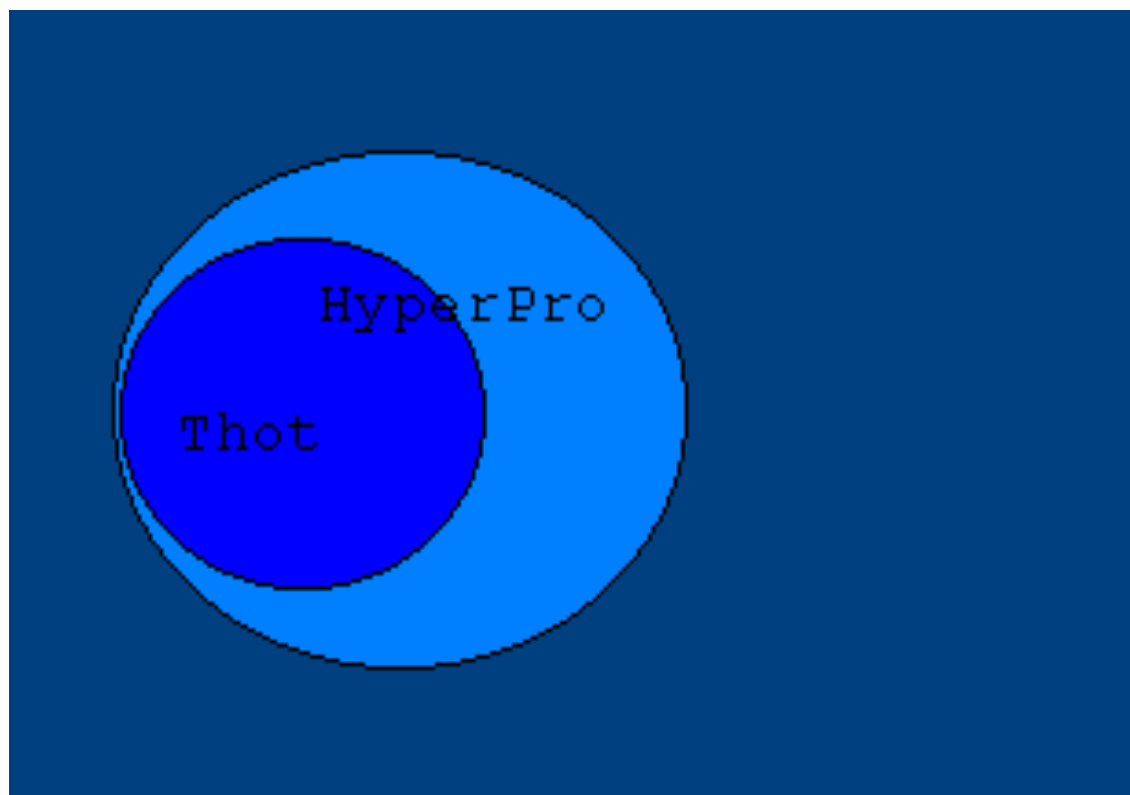
*`solution-2` is a positive integer (size of the chessboard). Type: `solution-1` is a variable or a list of elements, each of them variable or integer.*

*For all  $x, y, u, v$ , ( $y$  in `solution-1` and  $v$  in `solution-1` and  $rank-of(y, x)$  and  $rank-of(v, u)$  and  $not\ x=u$ ) implies ( $not\ y=v$  and  $not\ x-u = y-v$  and  $not\ x-u = v-y$ )*

```
solution(S, N) :-
    ( gen-list-int(L, 1, N) [def]
    , ( permutation(L, S) [def]
      , safe(S) [def]
      )
    ).
```



# Hyperpro





# Facilidades propostas no HyperPro

- Diferentes visões do documento
- Exportações para latex, ASCII e HTML
- Diferentes índices



# Vistas do HyperPro

- Vista de Comentário
- Vista de Asserções
- Vista de Tipos
- Vista de Programas



# Projeções

Vistas do documento com partes selecionadas pelo usuário.



# Especificação das Projeções

- Projeção Recursiva
- Projeção Manual
- Projeção Automática
- Projeção Baseada em Índices

# Projeção Recursiva

## 2.1 List and integer package

In This somme relations on lists, which are independent from the nature of the elements, and integers are defined.

```
permutation/2
```

```
Versions: Version1[*]
```

*permutation-1 is a list of integers.*

*list(permutation-1) and list(permutation-2).*

*length(permutation-1) = length(permutation-2).*

*forall x, x in permutation-1 implies x in permutation-2).*

*empty(permutation-1) implies empty(permutation-2).*

*Forall n,L, integer(n) and length(L,n) implies perm(L,L).*

```
permutation([], []).
```

```
permutation(L1, [X|L2]) :-  
    ( del(X, L1, L3)  
    , permutation(L3, L2)  
    ).
```



# Projeção Recursiva

```
permutation([], []).  
permutation(L1, [X|L2]) :-  
    ( del(X, L1, L3)  
    , permutation(L3, L2)  
    ).
```

# Projeção Manual

## 1 Introduction

This example shows the methodology of writing a program, using hyper logic programming  
The problem is to place  $n$  queens on an  $n \times n$  chessboard so that there is no couple of queens threatening each other.

## 2 Solution in Prolog

**solution/2**<sup>[cpr]</sup>  
Versions: **Version1**<sup>[\*]</sup>

`solution-1` is a list of integers in  $[1..solution-2]$ , of length `solution-2`, denoting the positions of `solution-2` reciprocally non-attacking queens on the chessboard.

*solution-2 is a positive integer (size of the chessboard). Type: solution-1 is a variable or a list of elements, each of them variable or integer.*

*For all  $x, y, u, v$ , ( $y$  in `solution-1` and  $v$  in `solution-1` and  $rank-of(y, x)$  and  $rank-of(v, u)$  and not  $x=u$ ) implies (not  $y=v$  and not  $x-u = y-v$  and not  $x-u = v-y$ )*

```
solution(S, N):-  
    ( gen-list-int(L, 1, N)[def]  
    , ( permutation(L, S)[def]  
      , safe(S)[def]
```





# Projeção Manual

This example shows the methodology of writing a program, using hyper logic programming

The problem is to place  $n$  queens on an  $n \times n$  chessboard so that there is no couple of queens threatening each other.



# Trabalhos Futuros

- Implementação de uma biblioteca de Interface
- Continuação da implementação das Projeções
- Manuais do sistema e do usuário