Aluno: Marcello Henrique Soares Cipriano

Matrícula: 22153898

Atividade da Disciplina: "Inteligência Artificial" – Período 2024/2

Especificação:

Executar o programa de aprendizado *hyper* (disponível no **Livro do Bratko – capítulo 21**) para todos as bases de conhecimento especificadas como exemplos no mesmo capítulo do Livro, em complementação à atividade feita anteriormente que tratou da execução do *mini-hyper*.

Relatório:

O **capítulo 21** do Livro do Bratko (Prolog Programming For Artificial Intelligence), que trata do assunto: "Inductive Logic Programming – ILP", uma abordagem em aprendizado de máquina, traz exemplos de programas que podem promover o aprendizado de máquina a partir de uma base de conhecimento e exemplos positivos e negativos. Os dois programas apresentados foram: mini-hyper e hyper.

Os testes com o mini-hyper foram objeto de uma primeira parte desse trabalho, apresentada no dia 25/10/2024 com sua inclusão no repositório acessado pelo link do github: https://github.com/MHCipriano/Atividade_ILP_mini_hyper.git. E nesta primeira parte, conferiu-se o resultado do mini-hyper com o Background Knowledge (BK) de relações familiares especificado em um programa Prolog descrito nesse citado livro do Bratko e disponível no material do professor sob o nome "family_relations_ILP_example.pl".

Porém, o outro teste pedido com o mini-hyper em relação ao programa da página 514 do Livro, que trata do aprendizado para predicados a fim de determinar quando uma lista teria um número par ou ímpar de elementos (**odd_even**) não foi exitoso, tendo sido percebido que havia predicados nesse segunda BK com 3 parâmetros, como **backliteral**, por exemplo, sendo que o mini_hyper fornecido somente operava sobre esse predicado com dois parâmetros.

Tendo sido confirmada essa restrição pelo professor em sala de aula, foi então conferido que para tal base de conhecimento somente seria possível operar a indução ou aprendizado das regras de formação do predicado usando o programa *hyper* apresentado no mesmo capítulo do livro, razão pela qual foi proposta essa segunda parte do trabalho a fim de que fossem executadas todas as bases de exemplos dadas no capítulo usando dessa vez o *hyper*.

Nessas condições, foram montadas cópias dos programas a serem usados, cujas especificações foram fornecidas no livro: o **hyper.pl**, programa completo contendo as regras e procedimentos de indução das hipóteses de formação dos predicados a partir das bases de conhecimento e exemplos dados; e os programas Prolog dessas bases: **member.pl; odd_even.pl; path.pl; insertion_sort.pl;** e **arch.pl**.

Antes de mostrar exemplos dos resultados obtidos com o hyper.pl sobre cada uma dessas bases, importante consignar que esses novos programas, bem como este relatório da segunda parte do trabalho estarão colacionados ao mesmo repositório acima indicado onde foi aposta a primeira parte (acesso pelo mesmo link do github), porém em um folder específico chamado hyper.

Por fim, apresentamos abaixo telas demonstrativas dos resultados obtidos em cada um dos testes realizados, conforme proposta da atividade, enfatizando que um dos testes, o do **arch.pl** não executou, por razões que não foi possível identificar; e o teste para a base **path.pl**, embora tenha iniciado uma execução, não conseguiu encontrar um ponto de parada. Os testes para as demais bases foram bem-sucedidos e trouxeram os seguintes resultados:

MEMBER:

ODD_EVEN:

```
- [hyper].
true.
 ?- [odd_even].
true.
 ?- induce(H), show_hyp(H).
Hypotheses generated: 31
Hypotheses refined: 0
To de refined: 22
Refining hypo with cost72:
 _3944.
_4076.
Hypotheses generated: 35
Hypotheses refined: 1
To de refined: 22
Refining hypo with cost72:
 _4432.
_4564.
Hypotheses generated: 39
Hypotheses refined: 2
To de refined: 23
Refining hypo with cost63:
 _5080.
_5212.
Hypotheses generated: 43
Hypotheses refined: 3
To de refined: 23
Refining hypo with cost73:
  _5602.
_5734.
```

... (intervalo de resultados)

PATH:

```
- [hyper].
true.
 ?- [path].
true.
?- induce(H), show_hyp(H).
Hypotheses generated: 5
Hypotheses refined: 0
To de refined: 4
Refining hypo with cost93:
  1626.
Hypotheses generated: 12
Hypotheses refined: 1
To de refined: 4
Refining hypo with cost84:
  2018.
Hypotheses generated: 23
Hypotheses refined: 2
To de refined: 4
Refining hypo with cost63:
  2386.
Hypotheses generated: 30
Hypotheses refined: 3
To de refined: 3
Refining hypo with cost106:
   2656.
... (intervalo de resultados)
Hypotheses generated: 6915
Hypotheses refined: 489
To de refined: 1229
Refining hypo with cost66:
 _595576.
_595816.
Hypotheses generated: 6930
Hypotheses refined: 490
To de refined: 1230
Refining hypo with cost65:
 _603756.
_603984.
Hypotheses generated: 6941
Hypotheses refined: 491
To de refined: 1230
Refining hypo with cost66:
  606594.
  606834.
```

INSERTION_SORT:

```
- [hyper].
true.
  ?- [insertion_sort].
true.
 ?- induce(H), show_hyp(H).
Hypotheses generated: 5
Hypotheses refined: 0
To de refined: 4
Refining hypo with cost72:
  1696.
Hypotheses generated: 12
Hypotheses refined: 1
To de refined: 3
Refining hypo with cost84:
 _1952.
_2098.
Hypotheses generated: 19
Hypotheses refined: 2
To de refined: 7
Refining hypo with cost83:
Hypotheses generated: 26
Hypotheses refined: 3
To de refined: 10
Refining hypo with cost64:
  4192.
4356.
```

... (intervalo de resultados)

```
Hypotheses generated: 5083
Hypotheses refined: 351
To de refined: 772
Refining hypo with cost86:

_360278.
_360492.

Hypotheses generated: 5108
Hypotheses refined: 352
To de refined: 783
Refining hypo with cost55:
_372944.
_373146.

Hypotheses generated: 5126
Hypotheses refined: 353
To de refined: 787
_374466.
_374668.
H = [[sort([], [])]/[], [sort([_A|_B], _C), sort(_B, _D), insert_sorted(_A, _D, _C)]/[_A:item, _B:list, _D:list, _C:list]] |
```

ARCH:

```
- [hyper].
true.
?- [arch].
true.
?- induce(H), show_hyp(H).
Hypotheses generated: 5
Hypotheses refined: 0
To de refined: 4
Refining hypo with cost63:
 1906.
    Exception: (22) term(object, _1672, _2088) ? |
```