Resolução Exame Tipo

* 1. O algoritmo “subir a colina” é um algoritmo de pesquisa que utiliza informação somente local. Parte de um estado inicial que poderá ser aleatório e na sua versão mais simples o nó sucessor será um nó simplesmente melhor que o seu pai, já na variante “subida íngreme” o nó sucessor será aquele que mais próximo da solução (isto é maior valor) estiver. O algoritmo termina quando o nó é solução ou não tem sucessores. Este algoritmo nunca volta atrás ou seja os nós por onde passar irão fazer parte do caminho a percorrer.  
       
     Simples - a, b, e, j  
     Subida íngreme - a, b, f
  2. inicio(a).  
     filho(a, b).  
     filho(a, d).  
     filho(a, c).  
     filho(b, e).  
     filho(b, f).  
     filho(b, g).  
     filho(c, h).  
     filho(d, i).  
     filho(e, j).  
     estimativa(b, 4).  
     estimativa(d, 6).  
     estimativa(c, 5).  
     estimativa(e, 3).  
     estimativa(f, 2).  
     estimativa(g, 7).  
     estimativa(j, 2).  
     estimativa(h, 6).  
     estimativa(i, 8).  
       
     % Subir colina (Estado Atual, Estado Final)  
     scolina(E, Ef) :-  
      findall(F, filho(E, F), S),  
      S \= [],  
      melhorSucessor(S, M), !,  
      scolina(M, Ef).  
     scolina(Ef, Ef).  
       
     melhorSucessor([M], M).  
     melhorSucessor([X, Y | N], M) :-  
      estimativa(X, Ex), estimativa(Y, Ey),  
      ( Ex < Ey, melhorSucessor([X | N], M);  
      melhorSucessor([Y | N], M) ).
  3. **Forma clausal:**∀x adv(x) → rico(x)  
     ∀x rico(x) → grande(carro(x))  
     ∀x grande(carro(x)) → gasta(carro(x))  
       
     **O universo é constituído por:**  
     1. adv(eva)  
     2. ~adv(X) U rico(X)  
     3. ~rico(X) U grande(carro(X))  
     4. ~grande(carro(X)) U gasta(carro(X))
  4. 5. ~gasta(carro(eva))  
       
     Pela regra 4 e 5  
     6. ~grande(carro(eva))  
       
     Pela regra 6 e 3  
     7. ~rico(eva)  
       
     Pela regra 7 e 2  
     8. ~adv(eva)  
       
     Inconsistência (contradição) entre a regra 8 e 1 logo o carro de eva gasta muita gasolina.

1. equipa(franca).  
   equipa(brasil).  
   equipa(portugal).  
   equipa(alemanha).  
     
   pertencer(portugal, grupoA).  
   pertencer(alemanha, grupoA).  
     
   jogar(franca, europeu).  
   jogar(portugal, europeu).  
   jogar(alemanha, europeu).  
     
   ser(portugal, latino).  
   ser(franca, latino).  
   ser(espanha, latino).  
   ser(italia, latino).  
     
   determinante(s-f) → [a]; [‘A’].  
   determinante(s-m) → [o]; [‘O’].  
   determinante(p-f) → [as]; [‘As’].  
   determinante(p-m) → [os]; [‘Os’].  
     
   preposicao(s-m, ao) → [ao].  
   preposicao(s-m, no) → [no].  
     
   pronomeInterrogativo(p-f, qt) → [‘Quantas’]; quantas.  
   pronomeInterrogativo(p-\_,qs) → [‘Quais’]; quais.  
     
   nome(s-f, franca) → [‘Franca’].  
   nome(s-m, brasil) → [‘Brasil’].  
   nome(s-f, alemanha) → [‘Alemanha’].  
   nome(s-m, portugal) → [‘Portugal’].  
   nome(s-m, campeonato) → [campeonato].  
   nome(s-m, grupoA) → [grupoA].  
     
   adjetivo(s-m, europeu) → [europeu].  
   adjetivo(p-f, latino) → [latinas].  
   adjetivo(s-f, latino) → [latina].  
   adjetivo(p-m, latino) → [latinos].  
   adjetivo(s-m, latino) → [latino].  
     
   verbo(s, Sujeito, jogar) → [joga].  
   verbo(p, Sujeito, jogar) → [jogam].  
   verbo(p, Sujeito, pertencer) → [pertencem], {equipa(Sujeito); assert(erro(semantico)), fail}.  
   verbo(s, Sujeito, pertencer) → [pertence], {equipa(Sujeito); assert(erro(semantico)), fail}.  
     
   frase(A, S, O) → sintagma\_nominal(N, S), sintagma\_verbal(N, A, S, O).  
   frase(A, S, O) → sintagma\_interrogativo(Q, N, Sujeito), sintagma\_verbal(N, Sujeito  
   1. prob(aceite) = 4 / 10 prob(~aceite) = 6 / 10  
        
      prob(nota\_16) = 3 / 10 prob(nota\_15) = 3 / 10  
      prob(nota\_14) = 4 / 10  
        
      prob(top\_10) = 4 / 10 prob(top\_20) = 3 / 10  
      prob(top\_30) = 3 / 10  
        
      p(boa) = 6 / 10 prob(normal) = 4 / 10  
        
      prob(aceite | boa) = 4 / 6 prob(~aceite | boa) = 2 / 6  
      prob(aceite | normal) = 1 / 4 prob(~aceite | normal) = 3 / 4  
        
      prob(aceite | top\_10) = 3 / 4 prob(~aceite | top\_10) = 1 / 4  
      prob(aceite | top\_20) = 2 / 3 prob(~aceite | top\_20) = 1 / 3  
      prob(aceite | top\_30) = 0 / 3 prob(~aceite | top\_30) = 3 / 3  
        
      prob(aceite | nota\_16) = 3 / 3 prob(~aceite | nota\_16) = 0 / 3  
      prob(aceite | nota\_15) = 2 / 3 prob(~aceite | nota\_15) = 1 / 3  
      prob(aceite | nota\_14) = 0 / 4 prob(~aceite | nota\_14) = 4 / 4  
        
      prob(nota\_14 | top\_10) = 1 / 4 prob(~nota\_14 | top\_10) = 3 / 4  
      prob(nota\_14 | top\_20) = 1 / 4 prob(~nota\_14 | top\_20) = 3 / 4  
      prob(nota\_14 | top\_30) = 2 / 4 prob(~nota\_14 | top\_30) = 2 / 4  
        
      prob(nota\_15 | top\_10) = 1 / 3 prob(~nota\_15 | top\_10) = 2 / 3  
      prob(nota\_15 | top\_20) = 1 / 3 prob(~nota\_15 | top\_20) = 2 / 3  
      prob(nota\_15 | top\_30) = 1 / 3 prob(~nota\_15 | top\_30) = 2 / 3  
        
      prob(nota\_16 | top\_10) = 2 / 3 prob(~nota\_16 | top\_10) = 1 / 3  
      prob(nota\_16 | top\_20) = 1 / 3 prob(~nota\_16 | top\_20) = 2 / 3  
      prob(nota\_16 | top\_30) = 0 / 3 prob(~nota\_16 | top\_30) = 3 / 3  
        
      **H(aceite | nota)**  
        
      H(aceite | nota) =   
      p(nota\_16) \*   
       [ -p(aceite | nota\_16) \* log2(p(aceite | nota\_16)) - p(~aceite | nota\_16) \*   
       log2(p(~aceite | nota\_16)) ]   
      + p(nota\_15) \*  
       [ -p(aceite | nota\_15) \* log2(p(aceite | nota\_15)) - p(~aceite | nota\_15) \*  
       log2(p(~aceite | nota\_15)) ]  
      + p(nota\_14) \*  
       [ -p(aceite | nota\_14) \* log2(p(aceite | nota\_14)) - p(~aceite | nota\_14) \*  
       log2(p(~aceite | nota\_14)) ]  
        
      (=)  
        
      H(aceite | nota) =   
       3/10 \* [ -3/3 \* log2(3/3) - 0/3 \* log2(0/3) ] +  
       3/10 \* [ -2/3 \* log2(2/3) - 1/3 \* log2(1/3) ] +  
       4/10 \* [ -0/4 \* log2(0/4) - 4/4 \* log2(4/4) ]  
        
      (=)  
        
      H(aceite | nota) = 0.3 \* 0 + 0.3 \* [ 0.4 + 0.52 ] + 0.4 \* 0  
       = 0.3 \* 0.92  
       = 0.276  
        
      **H(aceite | universidade)**  
        
      H(aceite | universidade) =  
      p(top\_10) \*  
       [-p(aceite | top\_10) \* log2(p(aceite | top\_10)) - p(~aceite | top\_10) \*  
       log2(p(~aceite | top\_10)) ]  
      + p(top\_20) \*  
       [-p(aceite | top\_20) \* log2(p(aceite | top\_20)) - p(~aceite | top\_20) \*  
       log2(p(~aceite | top\_20)) ]  
      + p(top\_30) \*  
       [-p(aceite | top\_30) \* log2(p(aceite | top\_30)) - p(~aceite | top\_30) \*  
       log2(p(~aceite | top\_30)) ]  
        
      (=)  
        
      H(aceite | universidade) =  
       4/10 \* [ -3/4 \* log2(3/4) - 1/4 \* log2(1/4) ] +  
       3/10 \* [ -2/3 \* log2(2/3) - 1/3 \* log2(1/3) ] +  
       3/10 \* [ -0/3 \* log2(0/3) - 3/3 \* log2(3/3) ]  
        
      (=)  
        
      H(aceite | universidade) = 0.4 \* [ 0.31 + 0.5 ] + 0.3 \* [ 0.4 + 0.52 ] + 0.3 \* 0  
       = 0.4 \* 0.81 + 0.3 \* 0.92  
       = 0.6  
        
      **H(aceite | recomendação)**  
        
      H(aceite | recomendação) =   
      p(boa) \*   
       [ -p(aceite | boa) \* log2(p(aceite | boa)) - p(~aceite | boa) \*   
       log2(p(~aceite | boa)) ]   
      + p(normal) \*  
       [ -p(aceite | normal) \* log2(p(aceite | normal)) - p(~aceite | normal) \*  
       log2(p(~aceite | normal)) ]  
        
      (=)  
        
      H(aceite | recomendação) =  
       6/10 \* [ -4/6 \* log2(4/6) - 2/6 \* log2(2/6) ] +  
       4/10 \* [ -1/4 \* log2(1/4) - 1/3 \* log2(3/4) ]  
        
      (=)  
        
      H(aceite | recomendação) = 0.6 \* [ 0.4 + 0.52 ] + 0.4 \* [ 0.5 + 0.31 ]  
       = 0.6 \* 0.92 + 0.4 \* 0.81  
       = 0.876  
        
      O atributo de menor entropia é a **nota** (raiz da árvore).  
        
      **Caso 1: Nota 14**  
      **H(aceite | universidade ∩ nota\_14)**H(aceite | universidade ∩ nota\_14) =  
      p(top\_10 ∩ nota\_14) \*  
       [-p(aceite | top\_10 ∩ nota\_14) \* log2(p(aceite | top\_10 ∩ nota\_14)) - p(~aceite | top\_10 ∩ nota\_14) \* log2(p(~aceite | top\_10 ∩ nota\_14)) ]  
      + p(top\_20 ∩ nota\_14) \*  
       [-p(aceite | top\_20 ∩ nota\_14) \* log2(p(aceite | top\_20 ∩ nota\_14)) - p(~aceite | top\_20 ∩ nota\_14) \* log2(p(~aceite | top\_20 ∩ nota\_14)) ]  
      + p(top\_30 ∩ nota\_14) \*  
       [-p(aceite | top\_30 ∩ nota\_14) \* log2(p(aceite | top\_30 ∩ nota\_14)) - p(~aceite | top\_30 ∩ nota\_14) \* log2(p(~aceite | top\_30 ∩ nota\_14)) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_14) =  
       1/4 \* [ -0/1 \* log2(0/1) - 1/1 \* log2(1/1) ] + 1/4 \* [ -0/1 \* log2(0/1) - 1/1 \* log2(1/1) ] +  
       2/4 \* [ -0/2 \* log2(0/2) - 2/2 \* log2(2/2) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_14) = 0.25 \* 0 + 0.25 \* 0 + 0.50 \* 0  
       = 0  
        
      O atributo universidade possui valor de entropia nulo.  
      Caso 1.1: nota=14 e universidade=top\_10 => Não aceite  
      Caso 1.2: nota=14 e universidade=top\_20 => Não aceite  
      Caso 1.3: nota=14 e universidade=top\_30 => Não aceite  
        
      **Caso 2: Nota 15**  
      **H(aceite | universidade ∩ nota\_15)**H(aceite | universidade ∩ nota\_15) =  
      p(top\_10 ∩ nota\_15) \*  
       [-p(aceite | top\_10 ∩ nota\_15) \* log2(p(aceite | top\_10 ∩ nota\_15)) - p(~aceite | top\_10 ∩ nota\_15) \* log2(p(~aceite | top\_10 ∩ nota\_15)) ]  
      + p(top\_20 ∩ nota\_15) \*  
       [-p(aceite | top\_20 ∩ nota\_15) \* log2(p(aceite | top\_20 ∩ nota\_15)) - p(~aceite | top\_20 ∩ nota\_15) \* log2(p(~aceite | top\_20 ∩ nota\_15)) ]  
      + p(top\_30 ∩ nota\_15) \*  
       [-p(aceite | top\_30 ∩ nota\_15) \* log2(p(aceite | top\_30 ∩ nota\_15)) - p(~aceite | top\_30 ∩ nota\_15) \* log2(p(~aceite | top\_30 ∩ nota\_15)) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_15) =  
       1/3 \* [ -1/1 \* log2(1/1) - 0/1 \* log2(0/1) ] +  
       1/3 \* [ -1/1 \* log2(1/1) - 0/1 \* log2(0/1) ] +  
       1/3 \* [ -0/1 \* log2(0/1) - 1/1 \* log2(1/1) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_15) = 0.33 \* 0 + 0.33 \* 0 + 0.33 \* 0  
       = 0  
        
      O atributo universidade possui valor de entropia nulo.  
      Caso 2.1: nota=15 e universidade=top\_10 => Aceite  
      Caso 2.2: nota=15 e universidade=top\_20 => Aceite  
      Caso 2.3: nota=15 e universidade=top\_30 => Não aceite  
        
      **Caso 2: Nota 16**  
      **H(aceite | universidade ∩ nota\_16)**H(aceite | universidade ∩ nota\_16) =  
      p(top\_10 ∩ nota\_16) \*  
       [-p(aceite | top\_10 ∩ nota\_16) \* log2(p(aceite | top\_10 ∩ nota\_16)) - p(~aceite | top\_10 ∩ nota\_16) \* log2(p(~aceite | top\_10 ∩ nota\_16)) ]  
      + p(top\_20 ∩ nota\_16) \*  
       [-p(aceite | top\_20 ∩ nota\_16) \* log2(p(aceite | top\_20 ∩ nota\_16)) - p(~aceite | top\_20 ∩ nota\_16) \* log2(p(~aceite | top\_20 ∩ nota\_16)) ]  
      + p(top\_30 ∩ nota\_16) \*  
       [-p(aceite | top\_30 ∩ nota\_16) \* log2(p(aceite | top\_30 ∩ nota\_16)) - p(~aceite | top\_30 ∩ nota\_16) \* log2(p(~aceite | top\_30 ∩ nota\_16)) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_16) =  
       2/3 \* [ -2/2 \* log2(2/2) - 0/2 \* log2(0/2) ] +  
       1/3 \* [ -1/1 \* log2(1/1) - 0/1 \* log2(0/1) ] +  
       0/3 \* [ -0 \* log2(0) - 0 \* log2(0) ]  
        
      (=)  
        
      H(aceite | universidade ∩ nota\_16) = 0.66 \* 0 + 0.33 \* 0 + 0  
       = 0  
        
      O atributo universidade possui valor de entropia nulo.  
      Caso 3.1: nota=16 e universidade=top\_10 => Aceite  
      Caso 3.2: nota=16 e universidade=top\_20 => Aceite  
        
      Uma vez que a entropia da universidade para todos os casos possíveis da nota é igual a zero pode-se concluir que com a **nota** e a **universidade** é possível calcular todos os resultados possíveis.  
      1. SE nota = 14 E universidade = top\_10 ENTÃO Não aceite
      2. SE nota = 14 E universidade = top\_20 ENTÃO Não aceite
      3. SE nota = 14 E universidade = top\_30 ENTÃO Não aceite
      4. SE nota = 15 E universidade = top\_10 ENTÃO Aceite
      5. SE nota = 15 E universidade = top\_20 ENTÃO Aceite
      6. SE nota = 15 E universidade = top\_30 ENTÃO Não aceite
      7. SE nota = 16 E universidade = top\_10 ENTÃO Aceite
      8. SE nota = 16 E universidade = top\_20 ENTÃO Aceite