Princípio da casa dos pombos

Alexandre Rademaker

FGV/EMAp, Brazil

MD 2019.2

O princípio

Imagine que eu tenho 3 casas de pombos. Se eu possuo 4 pombos, então certamente em alguma casa haverá mais de um pombo. Se a quantidade de pombos é maior que a quantidade de casas, haverá certamente alguma casa com mais de um pombo. Foi pensando em exemplos como este que o princípio é chamado de Princípio da Casa dos Pombos.

https://www.youtube.com/watch?v=pPuvnD4PYNE&t=200s
https://www.youtube.com/watch?v=2-mxYrCNX60

aplicação I

Em uma gaveta, há 4 meias pretas, 2 meias brancas, 8 meias cinzas. Qual é a quantidade mínima de meias que preciso retirar desta gaveta para garantir que terei pelo menos duas meias de cores diferentes?

Vamos pensar na pior das hipóteses? Se estou querendo retirar duas meias de cores diferentes, o azar é pegar várias meias da mesma cor. Eu começo a pegar meias cinzas (porque é a que tem maior quantidade). Sou tão azarado que pego 8 meias cinzas consecutivamente.

Depois que pego 8 meias cinzas, não tem como escapar. A próxima meia tem que ser de outra cor. Portanto, 9 meias é a quantidade mínima de meias para garantir que teremos pelo menos duas meias de cores distintas. Pode até ser que das 9 meias eu tenha mais de duas meias com cores diferentes, mas isso é sorte e não certeza.

aplicação II

Quantas pessoas precisa haver em um auditório para ter certeza de que pelo menos duas delas fazem aniversário no mesmo dia?

Estamos procurando ter certeza. E havendo 2 pessoas no auditório nunca poderíamos ter certeza de que ambas nasceram no mesmo dia. Com 365 pessoas no auditório, ainda não estaríamos em condições de assegurar que duas delas fazem aniversário no mesmo dia.

No entanto, existe um argumento categórico: se houver 367 pessoas no auditório, não há como fugir: pelo menos duas têm de fazer aniversário no mesmo dia.

aplicação III

estagiários: problema

Um grupo de 6 estagiários foi designado para rever 50 processos e cada processo deveria ser revisto por apenas um dos estagiários. No final do trabalho, todos os estagiários trabalharam e todos os processos foram revistos. É correto afirmar que:

1. um dos estagiários reviu 10 processos;
2. todos os estagiários reviram, cada um, pelo menos 5 processos;
3. um dos estagiários só reviu 2 processos;
4. quatro estagiários reviram 7 processos e dois estagiários reviram 6 processos;
5. pelo menos um dos estagiários reviu 9 processos ou mais.

aplicação III

estagiários: solução

(A) um dos estagiários reviu 10 processos. Seria possível 5 estagiários analisando 1 processo cada um e o sexto analisando 45 processos. Muitas outras situações tornam a alternativa A falsa.

(B) todos os estagiários reviram, cada um, pelo menos 5 processos; Não podemos garantir. Basta raciocinar da mesma maneira que a letra A. Com o mesmo raciocínio percebemos que as alternativas C e D são falsas.

(E) pelo menos um dos estagiários reviu 9 processos ou mais. Neste caso a pior das hipóteses é colocar cada estagiário para trabalhar com no máximo 8 processos. Como são 6, $6 \times 8 = 48$. Alguém terá que trabalhar com mais de 8 processos. Portanto, pelo menos um dos estagiários reviu 9 processos ou mais.

formalização em LP

math.stackexchange.com

Todo pombo está em uma casa (A):

$$\bigwedge_{i=1}^{n+1} \bigvee_{j=1}^{n} p_{i,j},$$

Em alguma casa temos dois pombos distintos (B)

$$\bigvee_{j=1}^{n} \bigvee_{1 \leq i < k \leq n+1} (p_{i,j} \wedge p_{k,j}).$$

O principio é uma implicação $A \to B$. Porém, algumas pessoas entendem que existe uma condição extra de que cada pombo está em **apenas** uma casa! É necessário?

$$\bigwedge_{i=1}^{n+1} \bigwedge_{1 \leq j < k \leq n} \neg(p_{i,j} \wedge p_{i,k}).$$

Instância PHP_3

formalização SNARK

PHP_2 é trivial! Então vamos olhar para PHP_3

```
(ql:quickload :snark)
(in-package :snark-user)

(initialize)
(use-resolution t)

(prove
 '(implies
    (and (or P11 P12) (or P21 P22) (or P31 P32))
    (or (or (and P11 P21) (and P11 P31) (and P21 P31))
        (or (and P12 P22) (and P22 P32) (and P12 P32)))))
```

Instância PHP_3

Sequent Calculus and ND

http://logitext.mit.edu/logitext.fcgi/tutorial
http://logitext.mit.edu/main
https://en.wikipedia.org/wiki/Natural_deduction

ND intro $\rightarrow$ left rules
ND eliminação $\rightarrow$ right rules

```
((P11 \/ P12) /\ (P21 \/ P22) /\ (P31 \/ P32)) ->
((P11 /\ P21) \/ (P21 /\ P31) \/ (P11 /\ P31) \/
 (P12 /\ P22) \/ (P22 /\ P32) \/ (P12 /\ P32))
```

Instância PHP_3

Lean

Como? Trabalhoso mas não difícil!