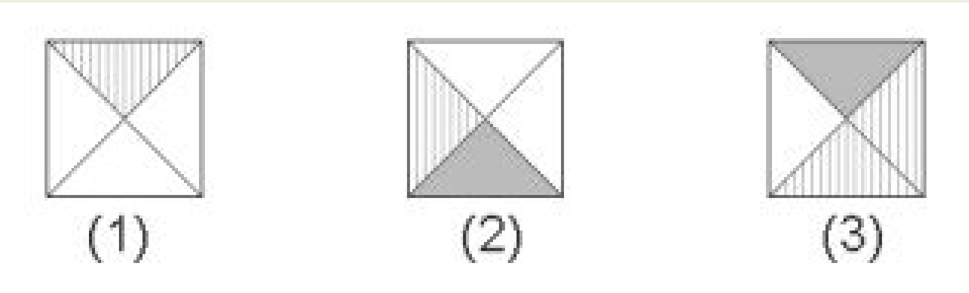
IBM1088 Linguagens Formais e Teoria da Computação

Aspectos práticos sobre a limitação da Computação

Evandro E. S. Ruiz
evandro@usp.br

Primeiro problema

- Projeto em Computação numa fábrica de ladrilhos
- 3 tipos de ladrilhos

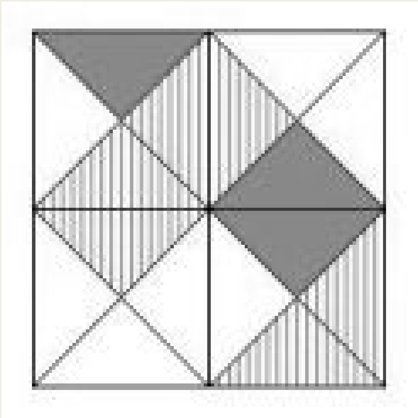


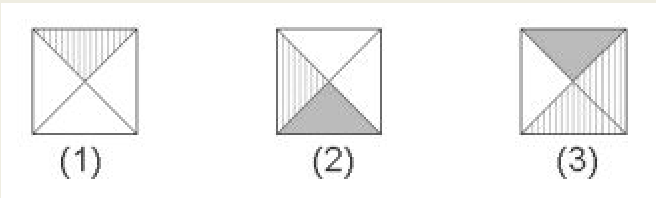
O pedido

- Faça um programa que visualize um cômodo (`nxn`) coberto com os ladrilhos
- Só podem ser usados estes 3 tipos
- Qualquer quantidade de cada um
- Requisitos
  - Casamento de cores nas bordas
  - Os ladrilhos não podem ser rotacionados

Uma solução

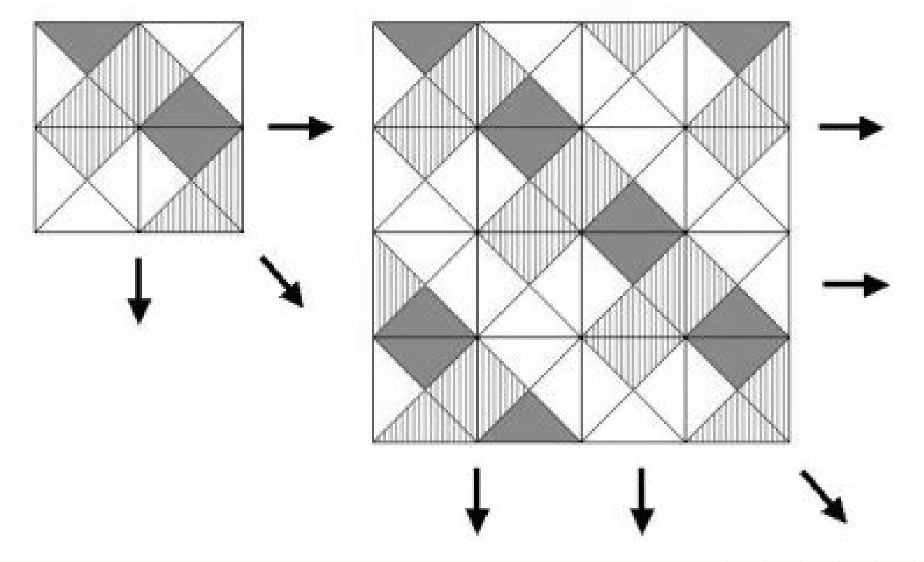
- Teste 2x2
- Funciona!





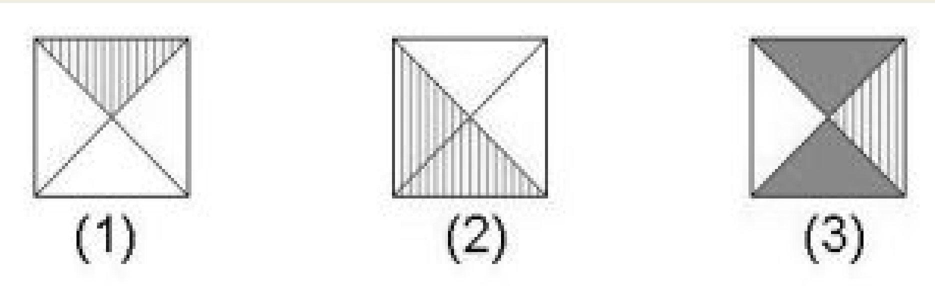
Tem escalabilidade...

- Teste 4x4
- Também funciona! Vamos programar

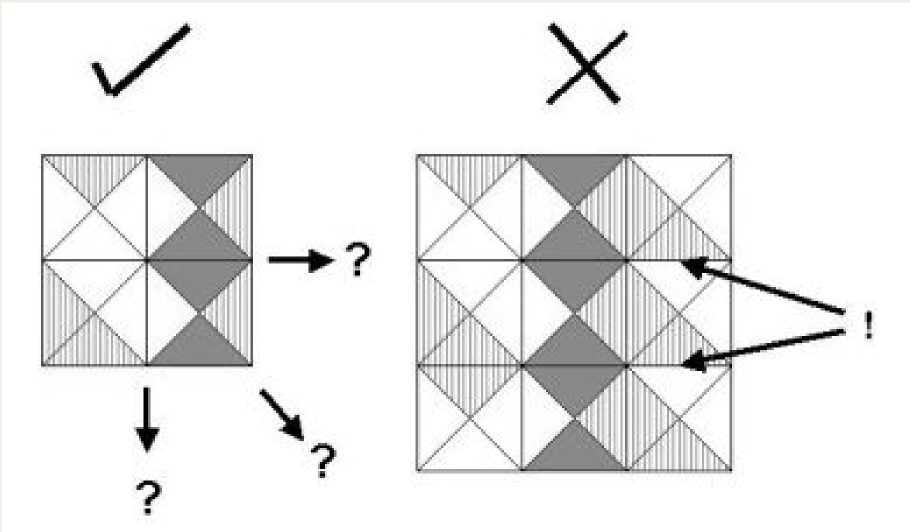


Enquanto isso...

- O patrão trocou os modelos dos ladrilhos

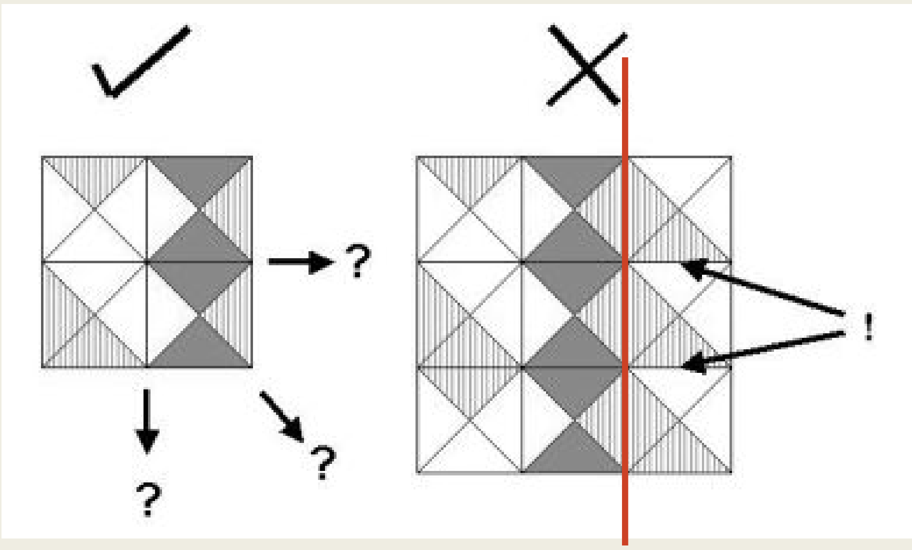


E os cômodos...



- O mosaico 2x2 funciona, mas...
- Não é extensível

E os cômodos...



- O mosaico 2x2 funciona, mas
- Não é extensível

Problema geral do mosaico

Dado um conjunto T de ladrilhos, estes ladrilhos podem ser usados para cobrir **qualquer** área `NxN`?

Problema geral do mosaico

Dado um conjunto T de ladrilhos, estes ladrilhos podem ser usados para cobrir **qualquer** área `NxN`?

- O problema não pode ser generalizado
- $\nexists$ algoritmo que resolva o problema
- Problema **não computável**, insolúvel

Problema geral do mosaico

Dado um conjunto T de ladrilhos, estes ladrilhos podem ser usados para cobrir **qualquer** área `NxN`?

- Ao mesmo tempo é um problema **indicidível**
- "Existe um algoritmo que resolva o problema do azulejo (ladrilho)?". Não!

Problema geral do mosaico

- Problema não é a infinitude
- É não poder detectar uma propriedade X qualquer numa solução
- Não tem algoritmo
- Problema não computável

Teorema de Rice

Para qualquer propriedade não trivial, não existe um algoritmo para encontrar tal propriedade.

- Ou seja,
- Não existe um algoritmo para decidir sobre uma propriedade não trivial da Computação
- Existem problemas para os quais nada podemos fazer, nem analisar uma propriedade deste e fazer uma afirmação

Teorema de Rice

Para qualquer propriedade não trivial, não existe um algoritmo para encontrar tal propriedade.

- As propriedades semânticas são um problema indecidível

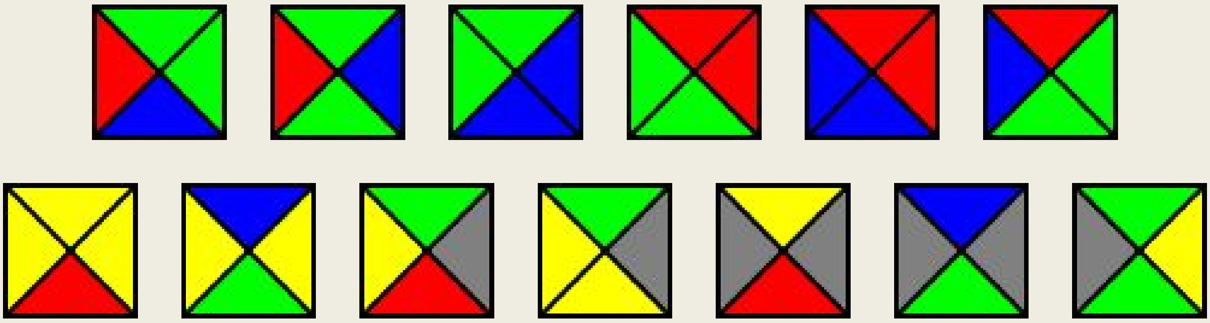
Rice em outras palavras

Seja **S** um conjunto de linguagens que são não triviais, o que significa:

- Existe uma MT que reconhece uma linguagem em **S**
- Existe uma MT que reconhece uma linguagem que não está em **S**

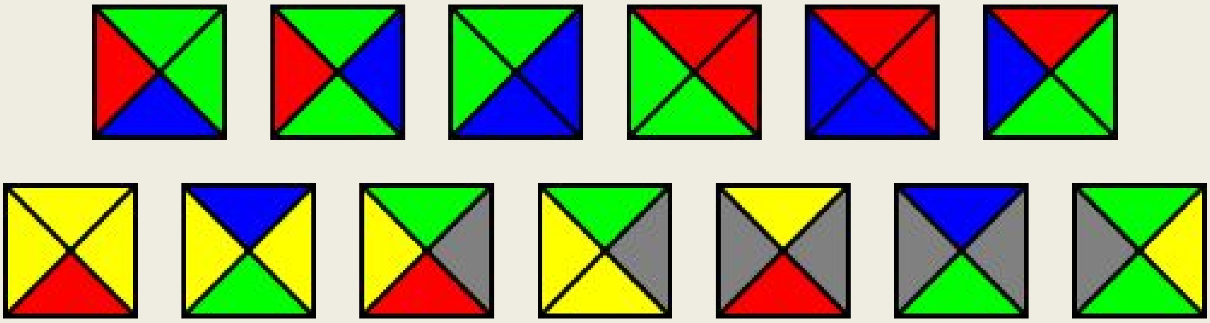
Então, é indecidível determinar se a linguagem reconhecida por uma MT arbitrária encontra-se em **S**. (similar ao problema da parada)

Origem: Mosaico de Wang



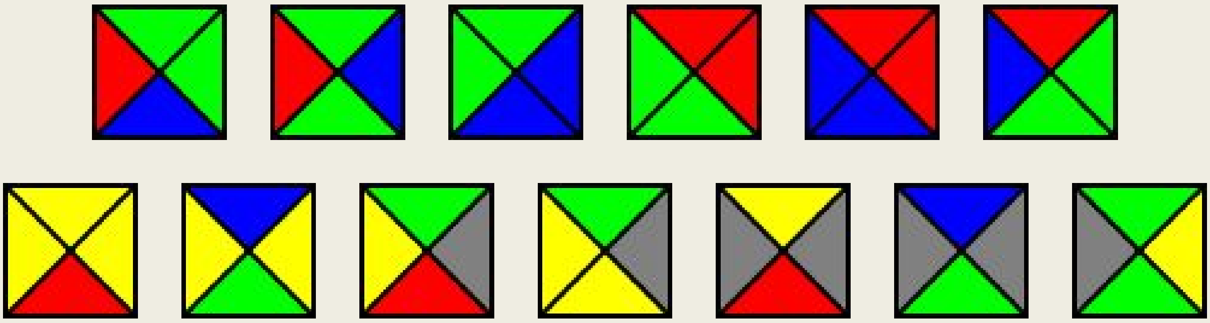
- Proposto por Hao Wang, em 1961
- 13 ladrilhos
- Conjectura: se podemos ladrilhar o finito, existe o mosaico periódico

Mosaico de Wang



- Proposto por Hao Wang, em 1961
- 13 ladrilhos
- Conjectura: se podemos ladrilhar o finito, existe o mosaico periódico (∄ uma MT que resolva?)

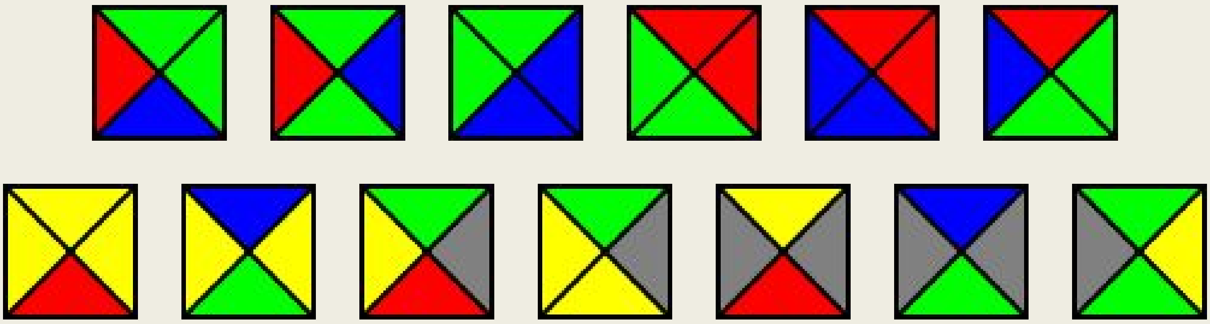
Mosaico de Wang



- Proposto por Hao Wang, em 1961
- 13 ladrilhos
- Conjectura: se podemos ladrilhar o finito, existe o <u>mosaico periódico</u> (∄ uma MT que resolva?)

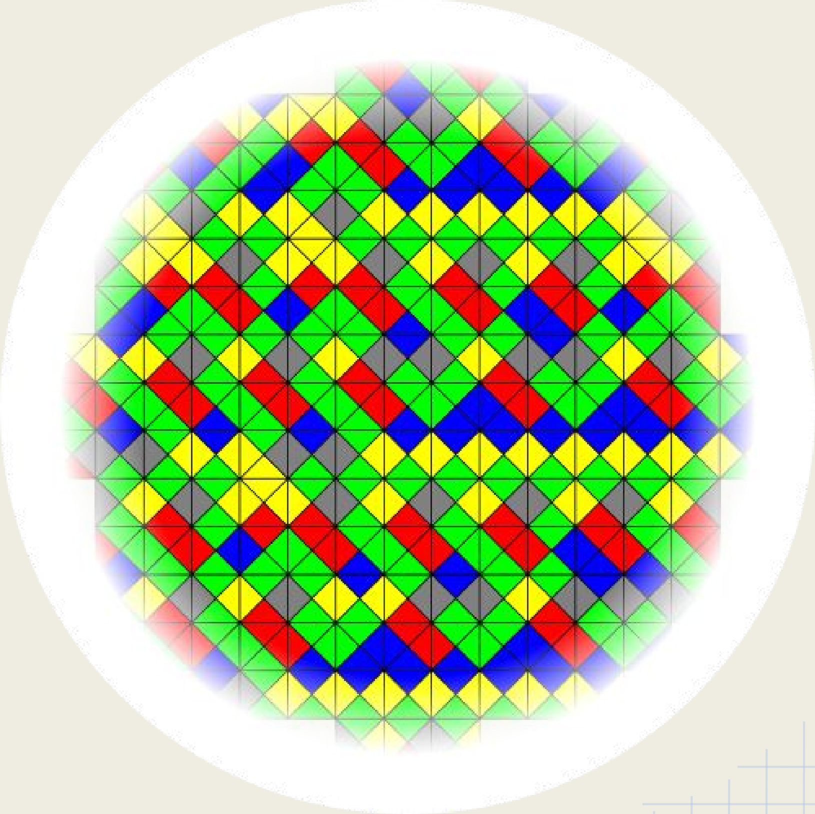
Não

Mosaico de Wang



- Proposto por Hao Wang, em 1961
- 13 ladrilhos
- Conjectura: se podemos ladrilhar o finito, existe o mosaico periódico É um problema indecidível!

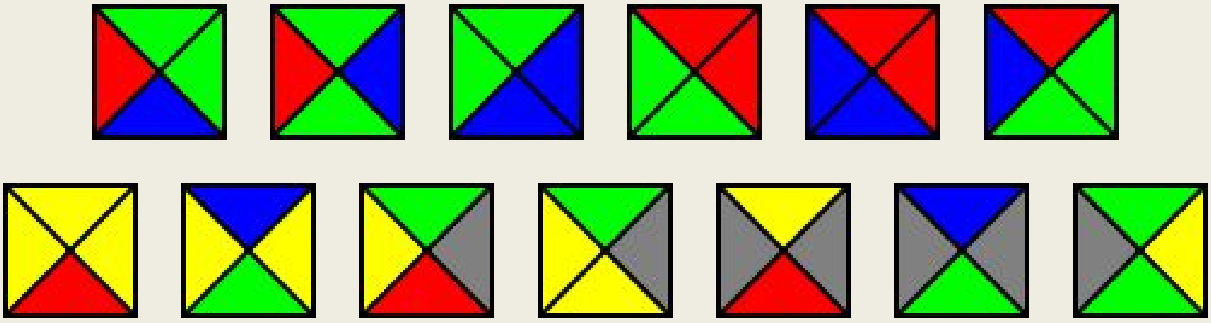
Mosaico com 13 ladrilhos



Wang na contra-mão

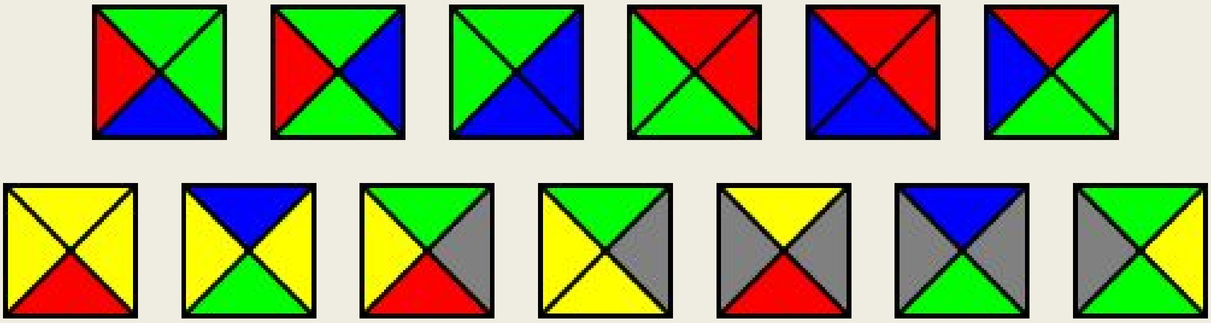
- O problema do mosaico é indecidível pois o “espaço” para verificação é infinito
- ∃ várias possiblidades para serem testadas
- Infinitas possibilidades
- Como afirmar que uma MT com conjunto finito de estados **não pode** solucionar um problema de infinitas possibilidades?

Problema do dominó

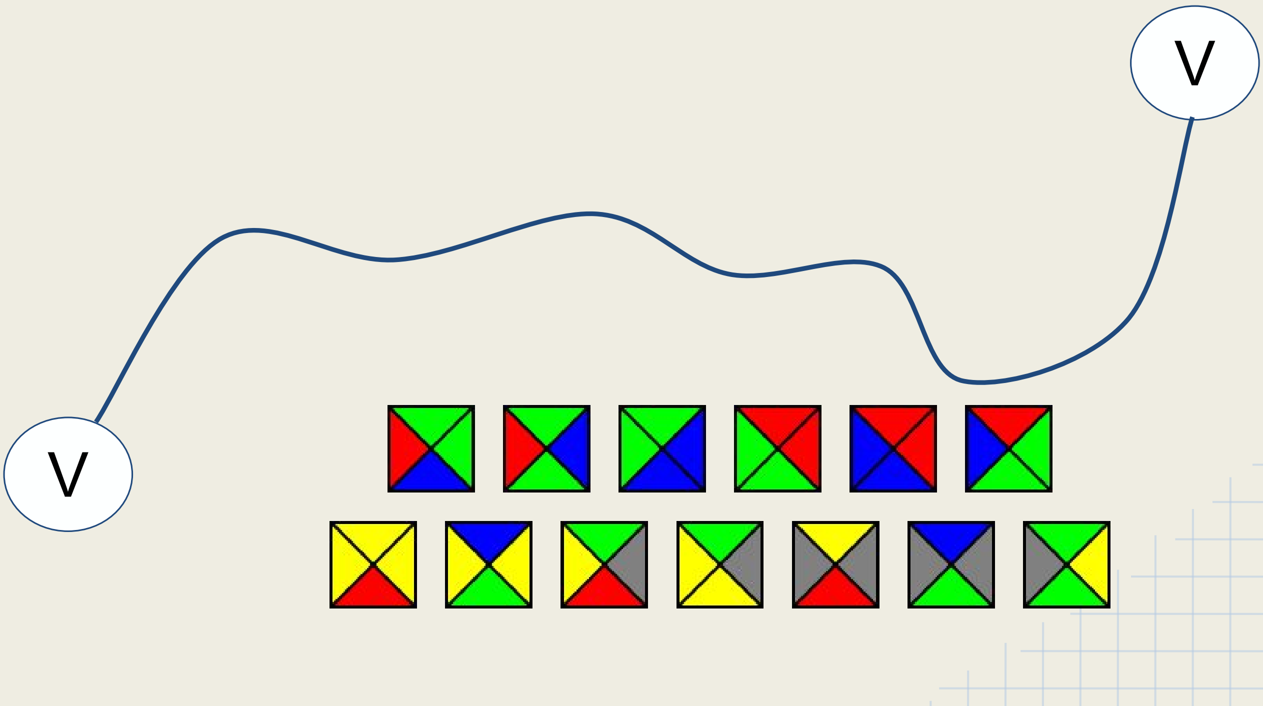


- Semelhante ao problema do mosaico de Wang
- Conjunto **T** finito de peças
- Dois pontos: V e W

O problema:



Podemos conectar V a W usando as peças do dominó e mantendo as restrições de casamento de cores?


V
V

Dominó e limites

- Diferente do mosaico
- No dominó, V e W são limitantes
- O problema **não é** infinito

Dominó e limites

- Diferente do mosaico
- No dominó, V e W são limitantes
- O problema não é infinito

- Se o cordão de dominós pode ir a qualquer lugar, o problema é **decidível**

Surpreendente!

Dominó e limites

- Diferente do mosaico
- No dominó, V e W são limitantes
- O problema não é infinito

- Se o cordão de dominós pode ir a qualquer lugar, o problema é **decidível**

- Se limitamos o espaço do cordão, o problema é **indecidível**

Dominó e limites

- Diferente do mosaico
- No dominó, V e W são limitantes
- O problema não é infinito

- Se o cordão de dominós pode ir a qualquer lugar, o problema é **decidível**

- Se limitamos o espaço do cordão, o problema é **indecidível**

Surpreendente!

Dominó

- Alegação de que infinitas possibilidades torna o problema indecidível é falsa
- O que o torna indecidível a a inexistência de uma propriedade que pode ser analisada, testada

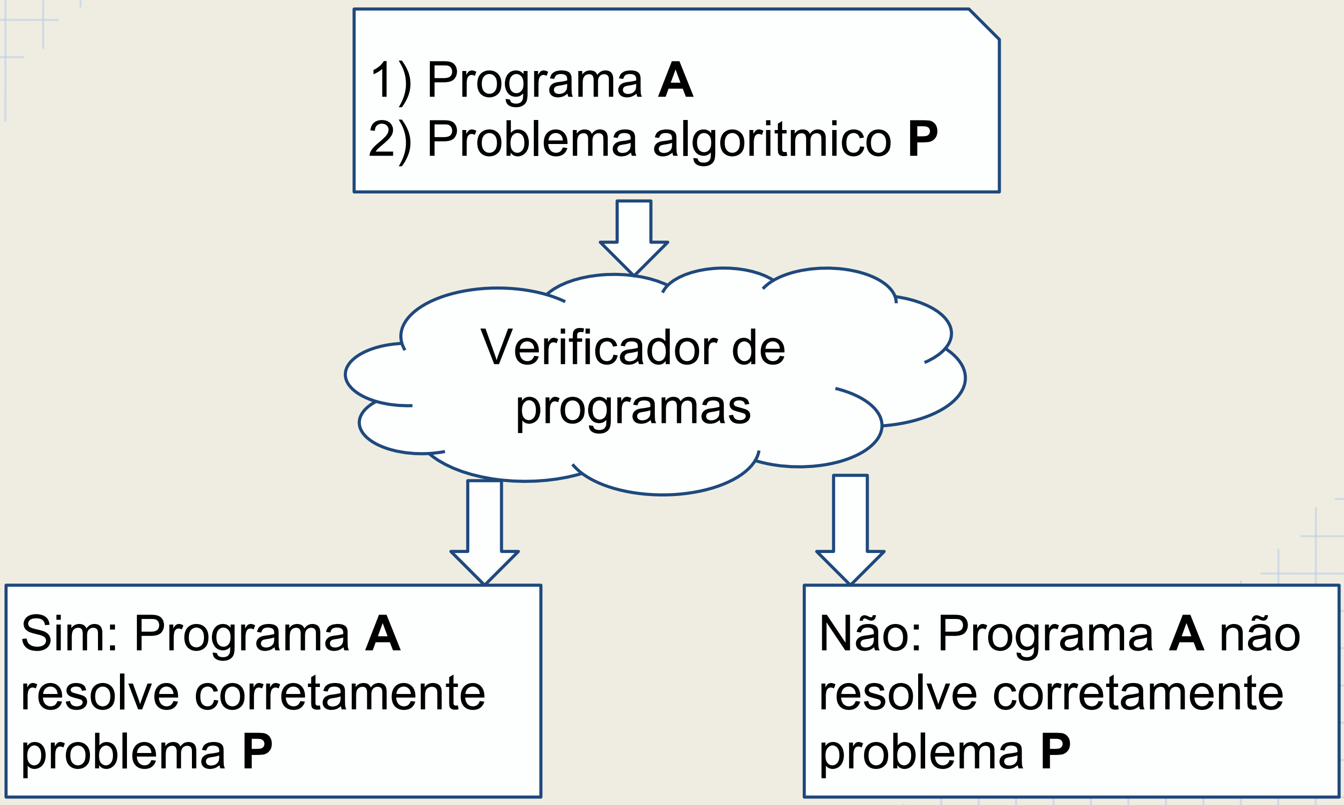
Programas

- Esperar por programas corretos é a exigência mínima de um programador
- Exigir programas corretos é obrigação

Programas

- Esperar por programas corretos é a exigência mínima de um programador
- Exigir programas corretos é obrigação
- Difícil é saber se o programa termina ou não
- Difícil é saber se ele existe

Ideal mas...


1) Programa **A**
2) Problema algoritmico **P**
Verificador de programas
Sim: Programa **A** resolve corretamente problema **P**
Não: Programa **A** não resolve corretamente problema **P**

Ideal mas é indecidível

1) Programa **A**
2) Problema algoritmico **P**

Verificador de programas

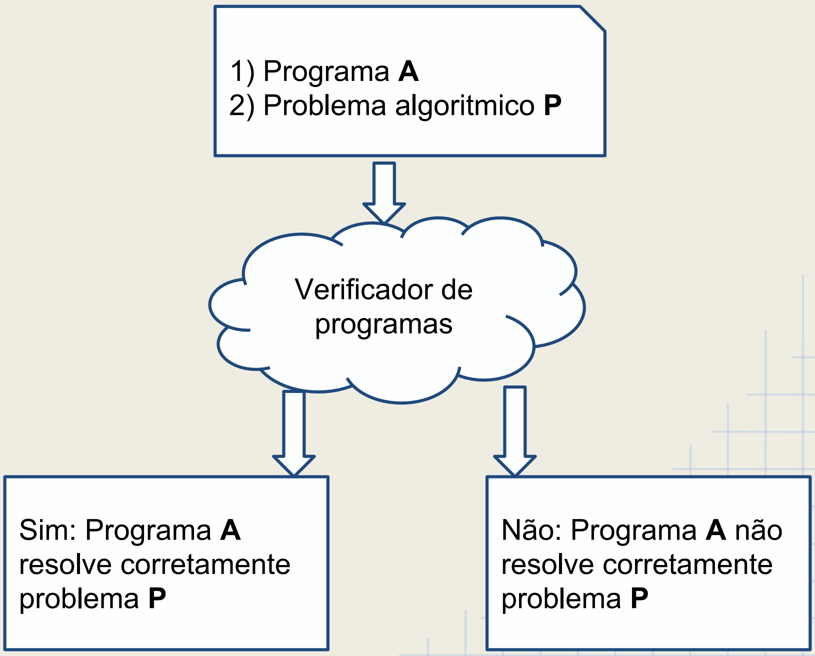
Sim: Programa **A** resolve corretamente problema **P**

Não: Programa **A** não resolve corretamente problema **P**

Problema indecidível

Ideal... nem tanto

- Mesmo adotando em A uma linguagem bem definida
- Sempre existirão programas que não poderemos verificar



Ideal... resolver o Problema da Parada

Dada uma descrição de um programa e uma entrada finita, decida se o programa terminará de executar ou executará indefinidamente dada esta entrada

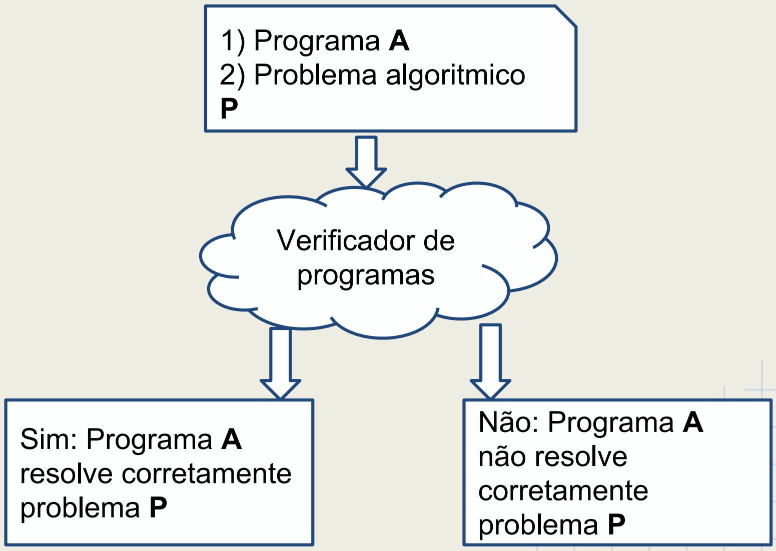
Ideal... resolver o Problema da Parada

Dada uma descrição de um programa e uma entrada finita, decida se o programa terminará de executar ou executará indefinidamente dada esta entrada

- Alan Turing mostrou que $\nexists$ um algoritmo genérico para resolver este problema
- Um dos primeiros problemas indecidíveis

Ideal... mas é o Problema da Parada

- A_{MT} = {<**M**,*w*> | **M** é uma MT, **M** aceita *w*?}
- A_{MT} é indecidível



"Tipos de programas"

- Estritamente computacionais, aritméticos, *number crunching*
- Recuperação de informação (termos)
- Otimização (menor caminho)
- Tomada de decisão
- Mistos

A realidade da Computação

- Não sabemos se um programa termina dado um conjunto de dados de entrada
- Nem sabemos se termina dado um conjunto **específico** de dados de entrada

Tipos de problemas algoritmicos

Exemplo

- **Entrada** (4 parâmetros):
  Programa P e um número X além de uma saída Y + um valor K

  Saída:
  = 2K se P faz $Y=X^2$
  = 3K caso contrário

- Misto: decisão e computação

Verificação do programa e da entrada de dados

```
k=9
while (k != 1):
    k=k-2
print "ok"
```

- imprime ok para **k** ímpar

```
k=10
while (k != 1):
    k=k-2
print "ok"
```

- Não para se **k**=par

Verificação do programa e da entrada de dados

```
k=9
while (k != 1):
    k=k-2
print "ok"
```

- imprime ok para **k** ímpar

```
k=10
while (k != 1):
    k=k-2
print "ok"
```

- Não para se **k**=par

Não existe uma MT que analise o algoritmo para decidir se o programa para.

Problema 3**x**+1

- O problema
- Tome um número **x** Natural
- Se **x**=par, divida **x**/2 e armazene o resultado em x
- Se **x**=ímpar, faça 3*****x**+1 e armazene em **x**
- Repita o processo indefinidamente

Problema 3x+1

- O problema
- Tome um número **x** Natural
- Se **x**=par, divida **x**/2 e armazene o resultado em x
- Se **x**=ímpar, faça 3***x**+1 e armazene em **x**
- Repita o processo indefinidamente

- A conjectura
- Não importa o número inicial, o processamento **sempre termina com 1**

Problema 3**x**+1

```
x=7
while (x != 1):
    if (x%2 == 0):
        x=x/2
    else:
        x=3*x+1
    print x,
```

- Saída: 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.
- e PARA!
- Teste para x=8

Conjectura de Collatz

- Proposta por Lothar Collatz em 1937
- Dr. Honoris Cause pela USP 5/set/1955
- Seqüência não apresenta periodicidade
- Fractais de Collatz
- Testada para **x**=$5*2^{60}$

Conjectura de Collatz

- Proposta por Lothar Collatz em 1937
- Dr. Honoris Cause pela USP 5/set/1955
- Seqüência não apresenta periodicidade
- Fractais de Collatz
- Testada para $\mathbf{x}=5*2^{60}$

- Um dos problemas não resolvidos da Matemática
- *"A seqüência de Collatz, a partir de qualquer n>0, sempre chega a 1?"*

Problemas difíceis e piores

- Existem problemas que são difíceis
- Mas existem problemas que são ainda piores de serem resolvidos
- Vejamos o caso do Mosaico

Problema geral do mosaico

Dado um conjunto T de ladrilhos, estes ladrilhos podem ser usados para cobrir **qualquer** área NxN?

- Problema indecidível e **não computável**
- Não há algoritmo que responda **sim** ou **não** a este problema
- Mudando este problema...

Problema limitado do mosaico

Dado um conjunto T de ladrilhos, estes ladrilhos podem ser usados para cobrir **uma** área NxN **específica**?

- Problema é decidível
- Problema NP
- $O(n^2)$

Equivalência

- Problema do mosaico limitado, é computável, porém é NP
- Problema do mosaico ilimitado é não computável

Equivalência

- Problema do mosaico limitado, é computável, porém é NP
- Problema do mosaico ilimitado é não computável

- Problema do dominó limitado, é indecidível, não computável
- Problema do dominó ilimitado é decidível, computável

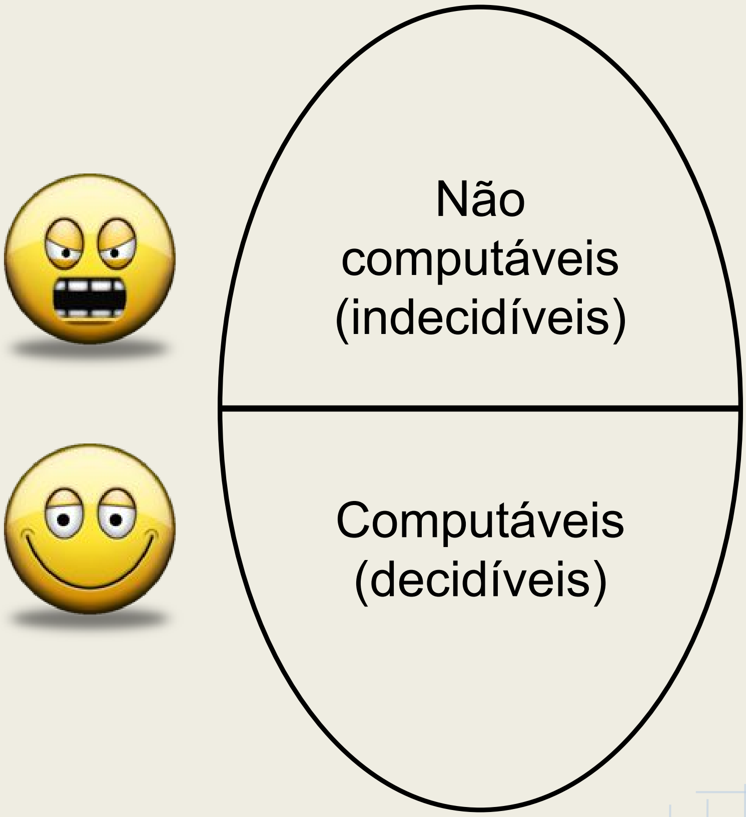
Equivalência ainda

- Dominó, mosaico e o problema da parada (melhor, sua negação) são **computacionalmente equivalentes**
- Podem ser reduzidos a um único

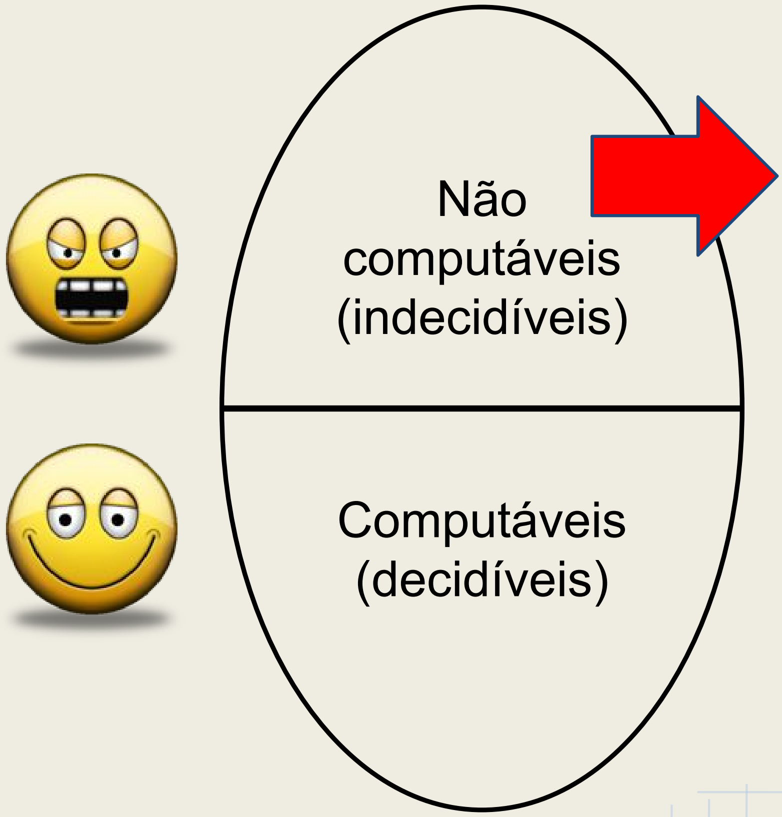
Equivalência ainda

- Dominó, mosaico e o problema da parada (melhor, sua negação) são **computacionalmente equivalentes**
- Podem ser **reduzidos** a um único
- Único = programa "guru" (sabemos que não é possível, imaginem)
- Guru decide se pode haver uma solução ou não

Problemas algoritmicos



Problemas algoritmicos

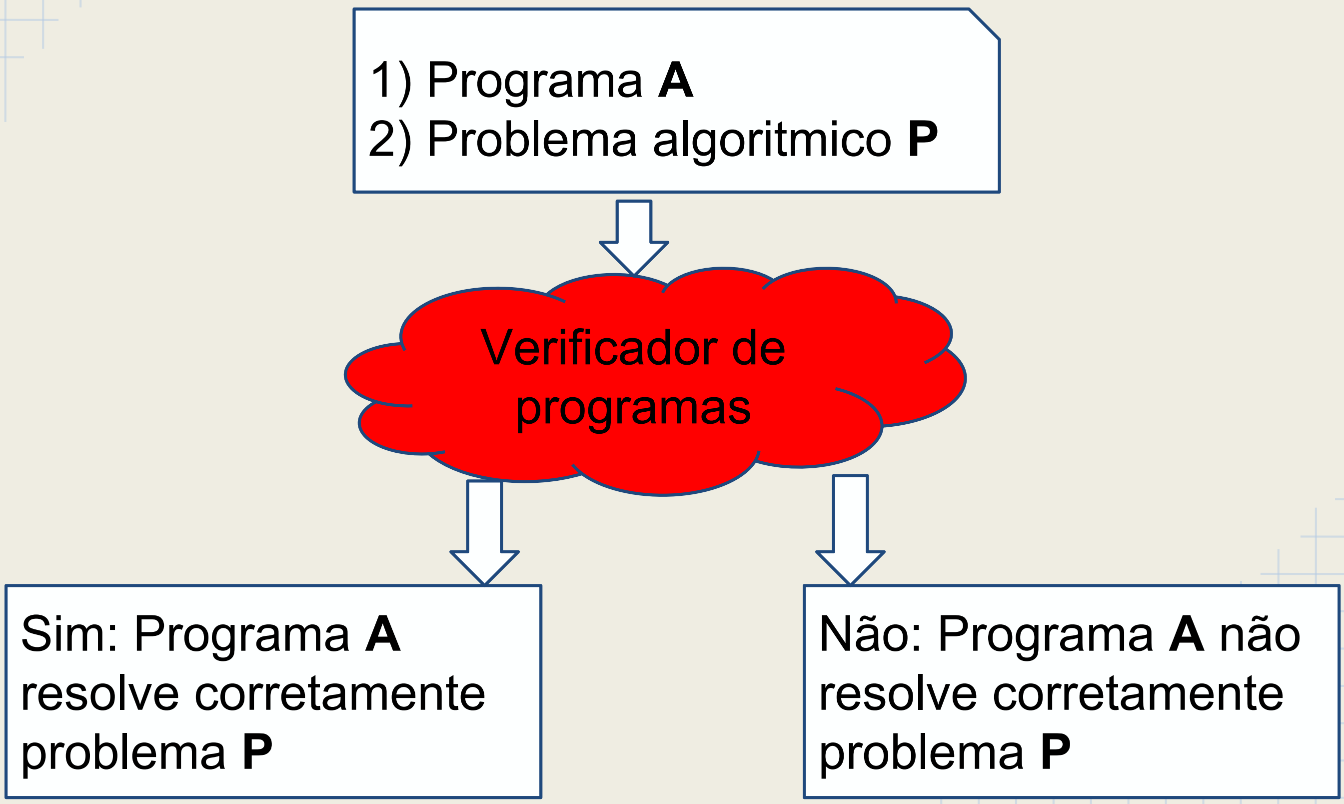


Mas ainda existem problemas ainda mais complexos

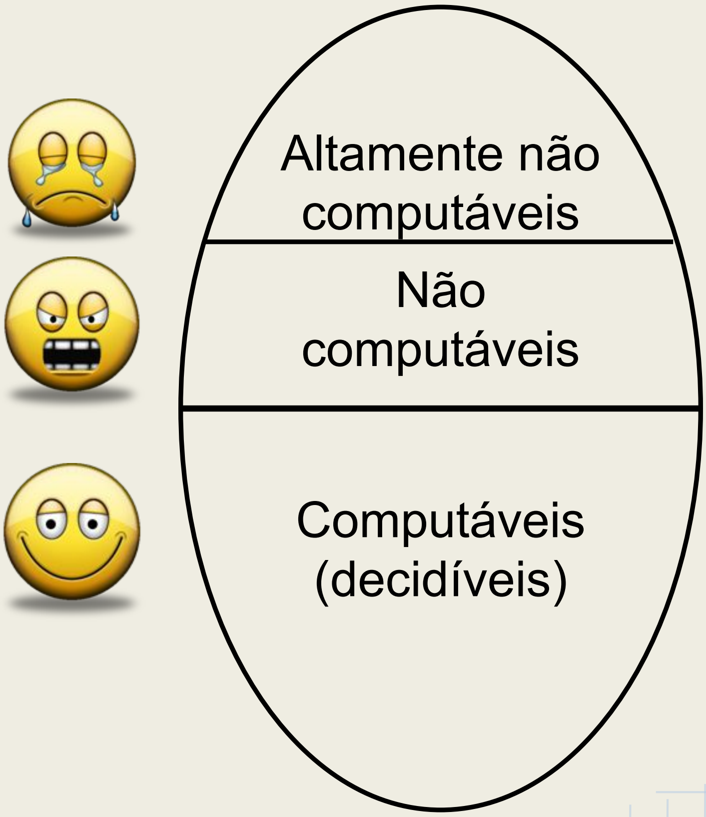
Ainda piores...

- O que pode ser pior que ser indecidível?
- O problema da parada pode ser **reduzido** ao problema da *verificação de programas*
- Sabemos que o **problema da verificação de programas** é ainda mais difícil
- "Menos indecidível" ainda

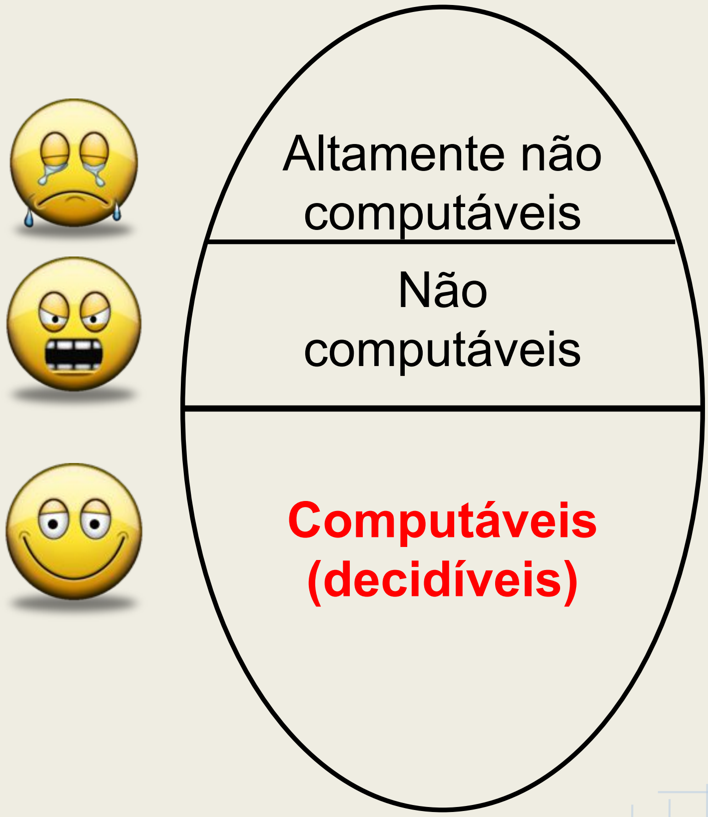
Ideal mas o mais difícil



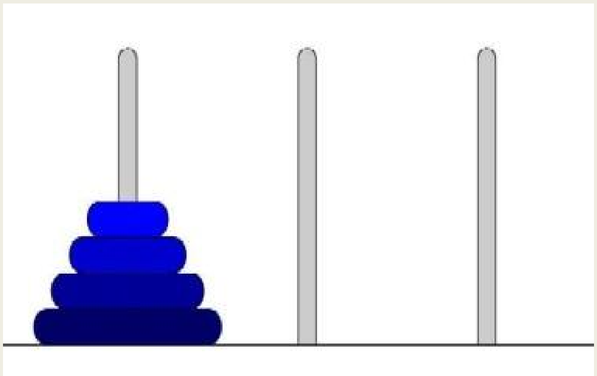
Problemas algoritmicos



Foco nos computáveis

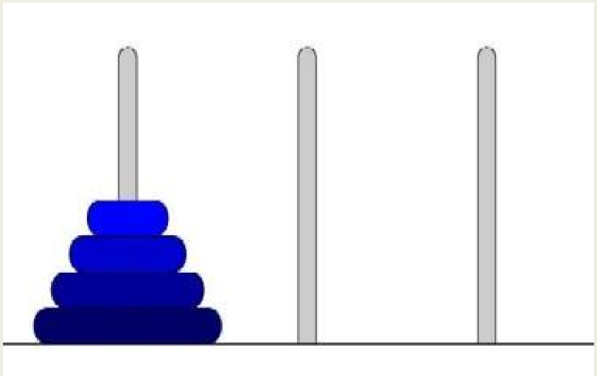


Torre de Hanoi



- $O(2^N - 1)$
- Isso é grande?
- É pior que $O(N^2)$?

Torre de Hanoi



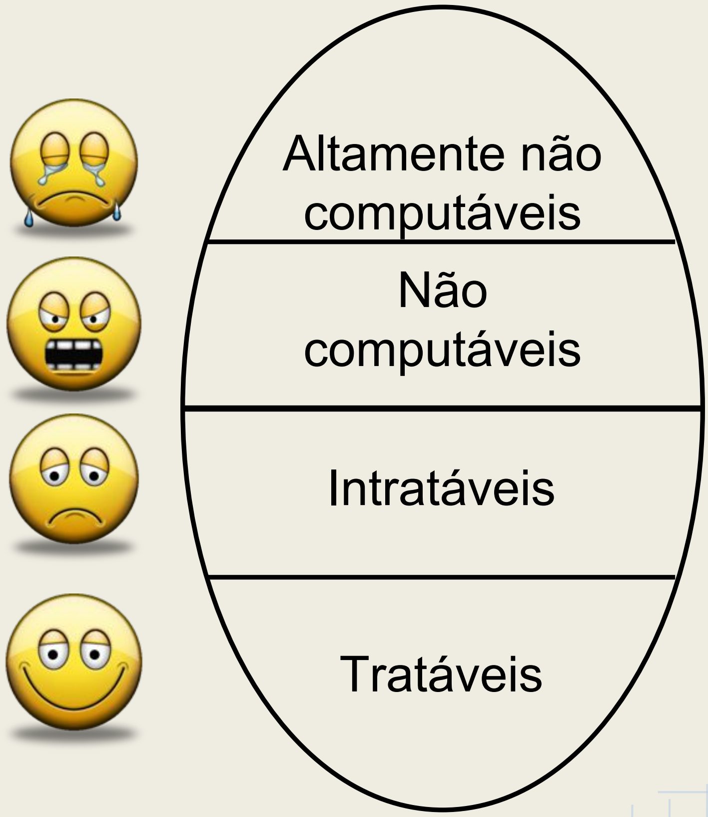
- $O(2^N - 1)$
- N=100
- N^2=10.000
- $2^N - 1$ > quantidade de microsegundos desde o Big Bang (>13 bilhões anos)

Bons e os ruins

- **Bons** = polinomiais
- **Ruins** = super-polinomiais

- **Bons** = logaritmicos, lineares, quadráticos ou (N log N)
- **Ruins** = 2^N, N!, N^N

Os intratáveis



Quando é intratável

- Problemas intratáveis requerem prova
- Prova que não existe algoritmos tratável
- E que não existirá algoritmo tratável
- Prova de O(exponencial) no limite inferior
- Ou seja, nos melhores casos

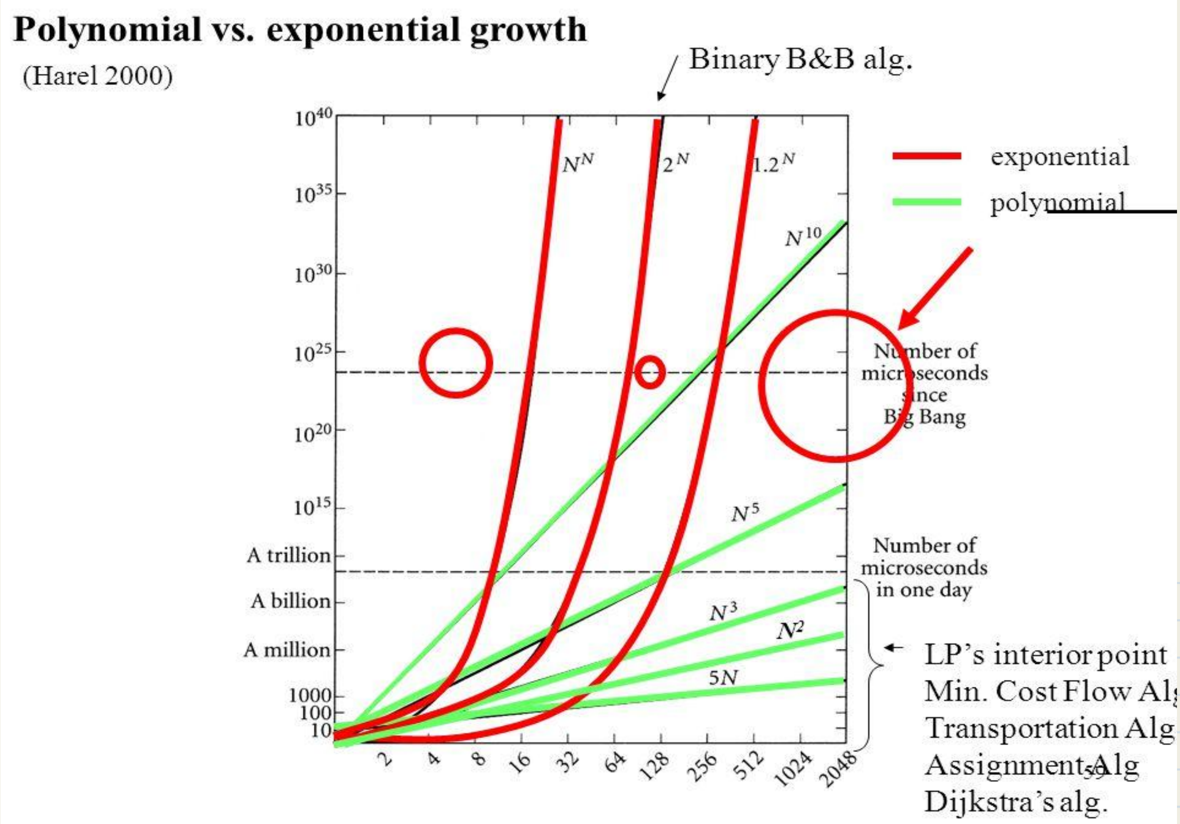
Aritmética de Presburger

- Dadas: P,Q asserções
- Considere o formalismo:
  "Se P é verdade, então Q é falso"

- Exemplo:
  (3<x) e (x+2y<=6) então (y<0)

- Aritmética de Presburger é uma Teoria de Primeira Ordem que só contém **adições** (**OU lógico**)
- Útil para demonstrar teoremas
- Útil para processamento de **Ontologias** (conhecimento)

Recordando Complexidade



Problemas NP

- Wikipedia lista > 3 mil problemas NP
- São decidíveis mas intratáveis
- Talvez com computadores qüânticos...

Problemas NP

- Wikipedia lista > 3 mil problemas NP
- São decidíveis mas intratáveis
- Talvez com computadores qüânticos...

- Não existe prova que eles precisam de algoritmos exponenciais
- Ninguém encontra um algoritmo polinomial para estes problemas

Problemas NP

- Caminhos em grafos
- Árvores espalhadas mínimas
- Roteamento
- Problemas de fluxo
- Mosaico (cobertura)
- Base de dados (armazenamento e compressão)
- Seqüênciamento de tarefas
- Ordenação de tarefas em multiprocessamento
- Lógica
- Geração de código
- ...

NP: problema fatal

- Ou são todos intratáveis
- Ou nenhum é
- Isso que significa NP-**completo**

- Resolva um NP e todos serão resolvidos
- **1 polinomial = todos polinomiais**

- Ou
- **se o caso mais simples de 1 for exponencial, todos serão**

Problema maior

P = NP?

- Maior mistério da Computação
- Programação Linear (otimização)
- Não é NP-completo
- Método Simplex, 1947

Obrigado!

evandro@usp.br