Lema do Bombeamento

Lema do Bombeamento
Motivação

Considere a linguagem

$$L_1 = 01^* = \{0, 01, 011, 0111, \ldots \}$$

O string $0\underline{1}1$ é dito ***bombeável*** em L_1 porque podemos tomar a porção sublinhada e bombeá-la (repeti-la) tantas vezes quanto se queira, obtendo *sempre* strings em L_1.

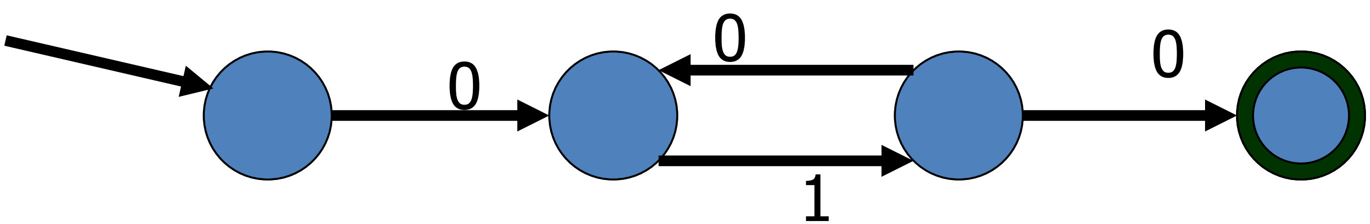
Q: Quais dos seguintes strings são bombeáveis?

1. 01111
2. 01
3. 0

Lema do Bombeamento
Motivação

1. Bombeável: 011$\underline{1}$1, 0$\underline{1}$111, 0$\underline{111}$1, 0$\underline{1111}$, etc.
2. Bombeável: 0$\underline{1}$
3. 0 *não* bombeável

Seja L_2 a linguagem definida pelo autômato:

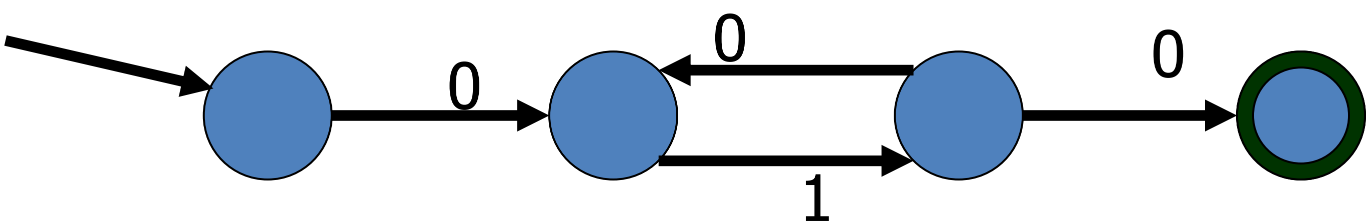


Q: 01010 é bombeável?

Lema do Bombeamento Motivação

A: Bombeável: 0$\underline{10}$10, 01$\underline{01}$0. Os substrings sublinados correspondem a ciclos no AF!

Ciclos do AF podem ser repetidos um número arbitrário de vezes: bombeamento.



Seja $L_3 = \{011, 11010, 000, \varepsilon\}$

Q: Que strings são bombeáveis?

Lema do Bombeamento Motivação

A:Nenhum! Quando um string pode ser bombeado (de modo não trivial), é sempre possível obter infinitos possíveis strings por meio desse bombeamento. Portanto, linguagens finitas não satisfazem a propriedade de bombeamento.

O Lema do Bombeamento provê um critério para quando strings podem ser bombeados:

Lema do Bombeamento

THM: Dada uma linguagem regular L, existe um número p (***número de bombeamento***) tal que, para qualquer string em L de comp. $\geq p$ é bombeável nos seus p primeiros simbolos. Em outras palavras, todo $u \in L$, tal que $|u| \geq p$, podemos escrever:

- $u = xyz$ (*x* é um prefixo, *z* é um sufixo)
- $|y| \geq 1$ (a parte do meio *y* é não vazia)
- $|xy| \leq p$ (bomb. nos *p* primeiros simbolos)
- $xy^iz \in L$ para todo $i \geq 0$ (a parte *y* pode ser bombeada)

Lema do Bombeamento: Prova

EX: Mostre que **pal**=$\{x \in \Sigma^* \mid x = x^R\}$ não é regular.

1. Suponha **pal** regular
2. Então ela tem um n^{o}. de bombeamento p
3. Mas… considere o string 0^p10^p. Esse string pode ser bombeado nos p primeiros símbolos? A resposta é NÃO, porque qualquer aumento da primeira porção - 0^p- resulta em um string que não é um palindromo
4. (2)$\rightarrow\leftarrow$(3) <contradição!> Portanto, nossa suposição (1) está errada e podemos concluir que **pal** *não* é uma linguagem regular

Lema do Bombeamento: Modelo

De modo geral, para provar que *L não* é regular:

1. Suponha *L* regular
2. Então *L* tem um no. de bombeamento *p*
3. *Encontre um padrão de string envolvendo p, que pertença a L, e que não possa ser bombeado.* ***Essa é a parte difícil*.**
4. (2)→←(3) <contradição!> Portanto, a nossa suposição em (1) está errada e podemos concuir que *L não* é regular.

Lema do Bombeamento: Exemplo

Como as partes 1, 2 e 4 são idênticas para qualquer prova usando o lema do bombeamento, os exemplos a seguir mostram apenas a parte 3 da prova.

Lema do Bombeamento: Exemplo

EX: Mostre que L={$a^n b^n$| n = 0,1,2, … } não é regular.

Parte 3) Considere $a^p b^p$. Por hipótese, a parte a ser bombeada deve estar contida nos p primeiros símbolos do string. Nesse caso, obteríamos um string com mais *a*'s do que *b*'s, que não corresponde ao padrão de strings de L.

Lema do Bombeamento: Exemplo

Algumas vezes pode ser útil bombear para *menos* e não para mais, ou seja, simplesmente remover a parte y do padrão do string. Isso corresponde a tomar $i = 0$ no lema do bombeamento:

EX: Mostre que $\{a^m b^n \mid m > n\}$ não é regular.

Part 3) Considere $a^{p+1} b^p$. Por hipótese, podemos bombear p/ *menos* um substring das p primeiras letras desse string. Como y é não vazio, isso resulta em um decréscimo do número de a's no padrão, significando que o no. de a's é menor ou igual ao no. de b's. Portanto, o string resultante não pertence à linguagem!

Lema do Bombeamento: Exemplo

Algumas vezes precisamos examinar o resultado do bombeamento com mais cuidado:

EX: Mostre que $\{1^n \mid n$ é primo$\}$ não é regular.

Part 3) Dado p, escolhemos um número primo n maior que p. Considere 1^n. Por hipótese, podemos bombear um substring dos p primeiros simbolos de 1^n. Seja m o comprimento da parte bombeada. Bombeando *i vezes*, obtemos o string $1^{(n-m)+im} = 1^{n+(i-1)m}$.

Q: Determine i de modo que o expoente não seja um número primo.

Lema do Bombeamento: Exemplo

A: Tome $i = n + 1$. Então o string resultante do bombeamento é

$$1^{n+(i-1)m} = 1^{n+(n+1-1)m} = 1^{n+nm} = 1^{n(1+m)}$$

Portanto, o expoente não é um número primo, significando que o padrão não pertence à linguagem.

Lema do Bombeamento: Prova

Considere um grafo com n vértices. Suponha que você ande pelo grafo, visitando um certo número de vértices.

Q: Quantos vértices você pode visitar, antes de ser forçado a visitar um mesmo vértice uma segunda vez?

Lema do Bombeamento: Prova

A: Se você visita n+1 vértices, necessariamente terá que visitar um deles mais de uma vez.

Q: Porque?

Lema do Bombeamento: Princípio da Casa dos Pombos

R: Princípio da Casa dos Pombos.

Mais precisamente. Sua visita a n+1 vértices define a seguinte função:

$$f : \{1, 2, 3, \ldots , n+1\} \rightarrow \{\text{conj. card. } n \}$$

$$f(i) = i\text{-ésimo vértice visitado}$$

Como o domínio é maior que o contra-domínio, a função não pode ser injetora.

Lema do Bombeamento: Prova

Considere agora o string aceito *u*. Como *L* é regular por hipótese, seja *M* o AF que aceita *L*. Seja $p = |Q|$ = no. de estados de *M*. Suponha $|u| \geq p$. O caminho rotulado por *u* visita p+1 estados nos primeiros *p* símbolos. Então *u* deve visitar algum estado mais de uma vez. O sub-caminho de *u* conectando a primeira e a segunda visita a esse vértice é um *loop*, e nos dá a parte *y* que pode ser bombeada (contida nos *p* primeiros símbolos*)*