

# BCC242

## Alfabeto, Strings, Linguagens

Registro aqui o agradecimento à Profa. Lucília por ceder slides que fazem parte deste material.

# Exemplo: Máquina de Venda

- A máquina de venda retorna uma coca-cola por \$0.45
- Ela aceita apenas moedas de \$0.25 (quarter) ou de \$0.10 (dime)
- Come o seu dinheiro se você não inserir o valor correto 😊

Pode ser modelado pelo pseudo-código:

# Exemplo: Máquina de Venda

```
cocaVend() {  
    int total = 0, coin;  
    while (total != 45) {  
        receive(coin);  
        if ((coin==10 && total==40)  
            || (coin==25 && total>=25))  
            reject(coin);  
        else  
            total += coin;  
    }  
    return new Coca();  
}
```

# Exemplo: Máquina de Venda

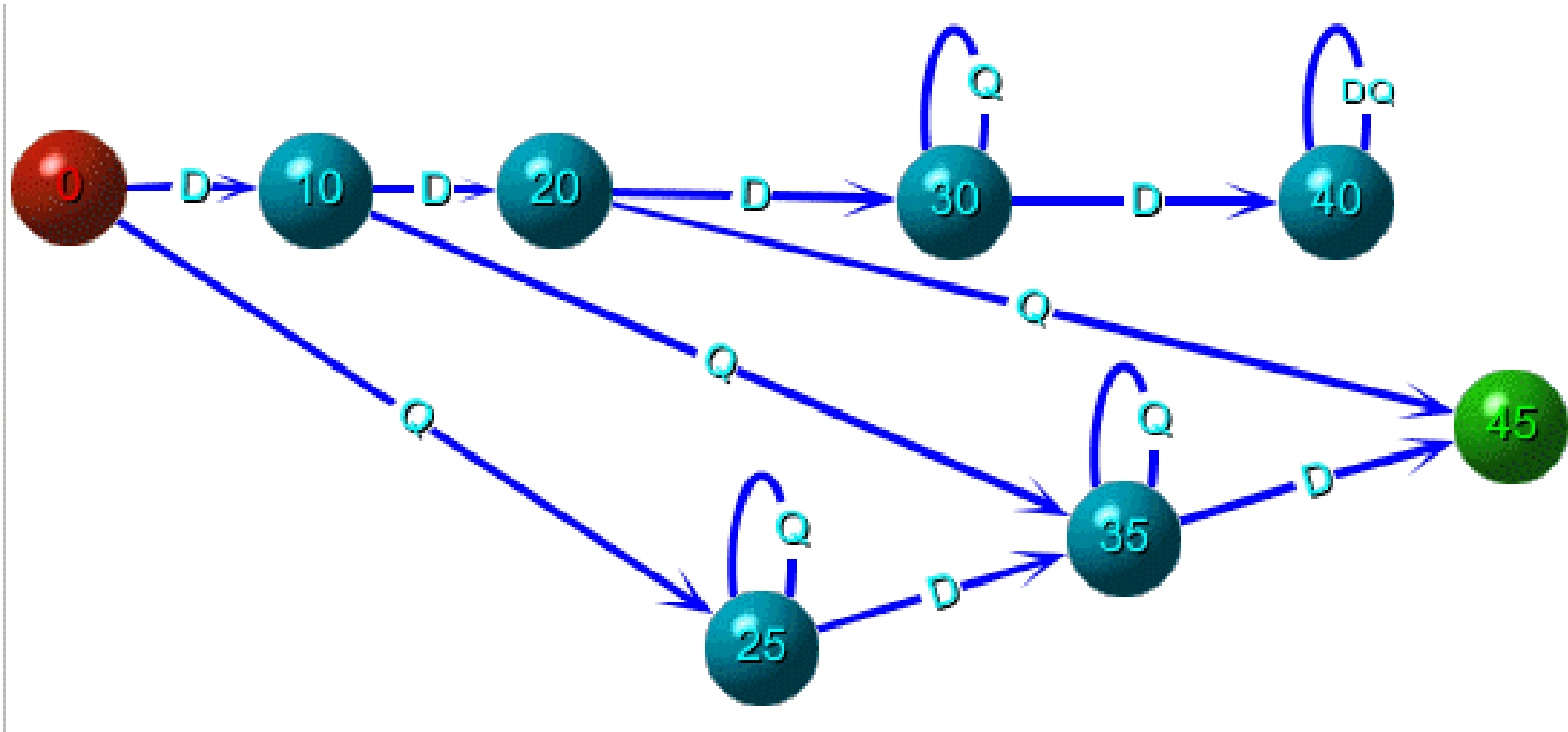
Porque esse modelo é demasiadamente complicado?

# Exemplo: Máquina de Venda

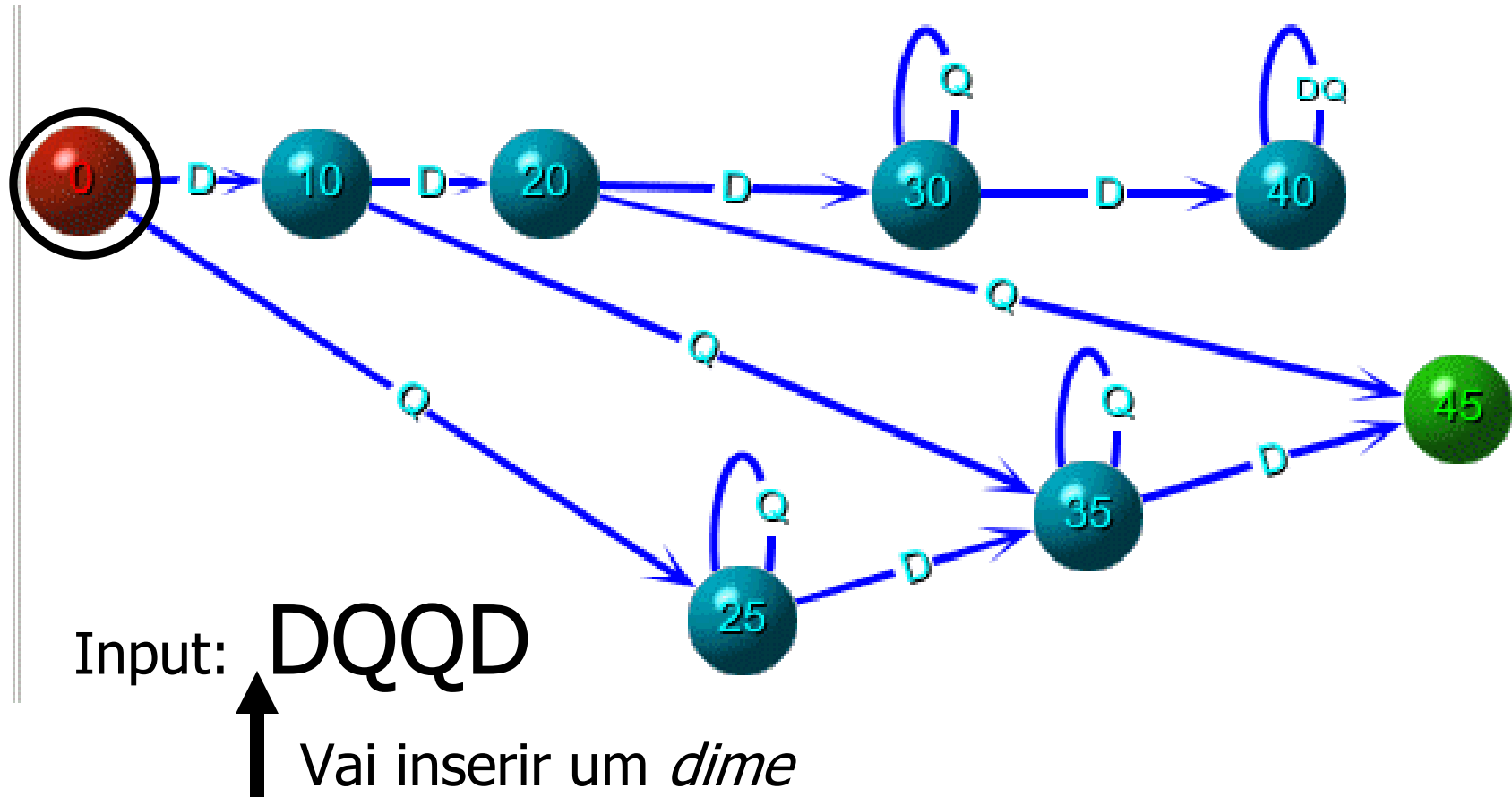
Porque esse modelo é complicado demais?

- 1) Máquinas de venda já existiam muito antes dos primeiros computadores, ou de Java!
- 2) Não precisa de fato de `int`'s. Cada `int` introduz  $2^{32}$  possibilidades multiplicativamente!
- 3) Não é necessário saber como somar inteiros para modelar uma máquina de venda  
(`total += coin`)
- 4) `if/else`, gramática da linguagem Java ... são artifícios que apenas complicam a essência do problema

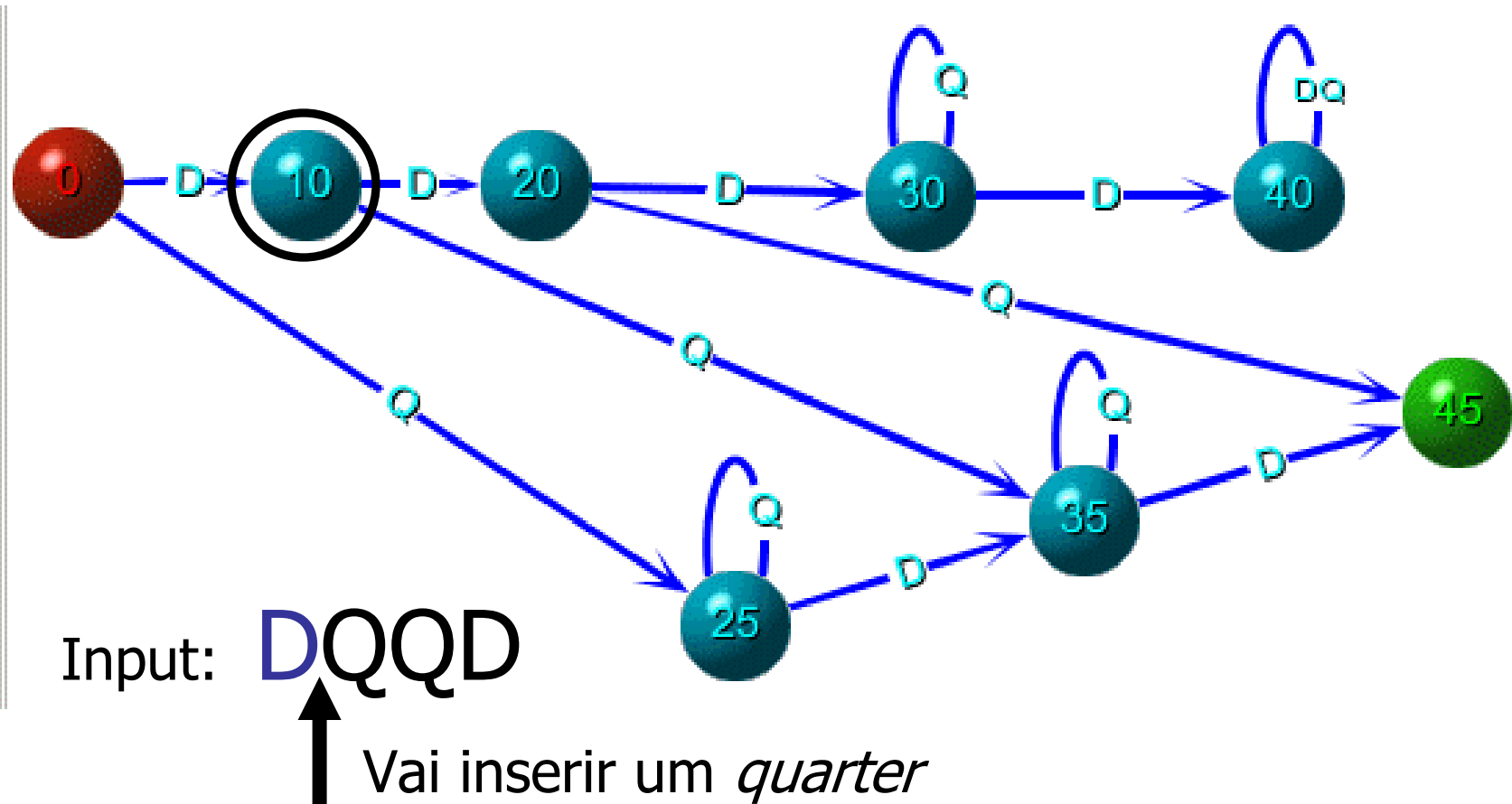
# Exemplo: Máquina de Venda



# Exemplo: Máquina de Venda

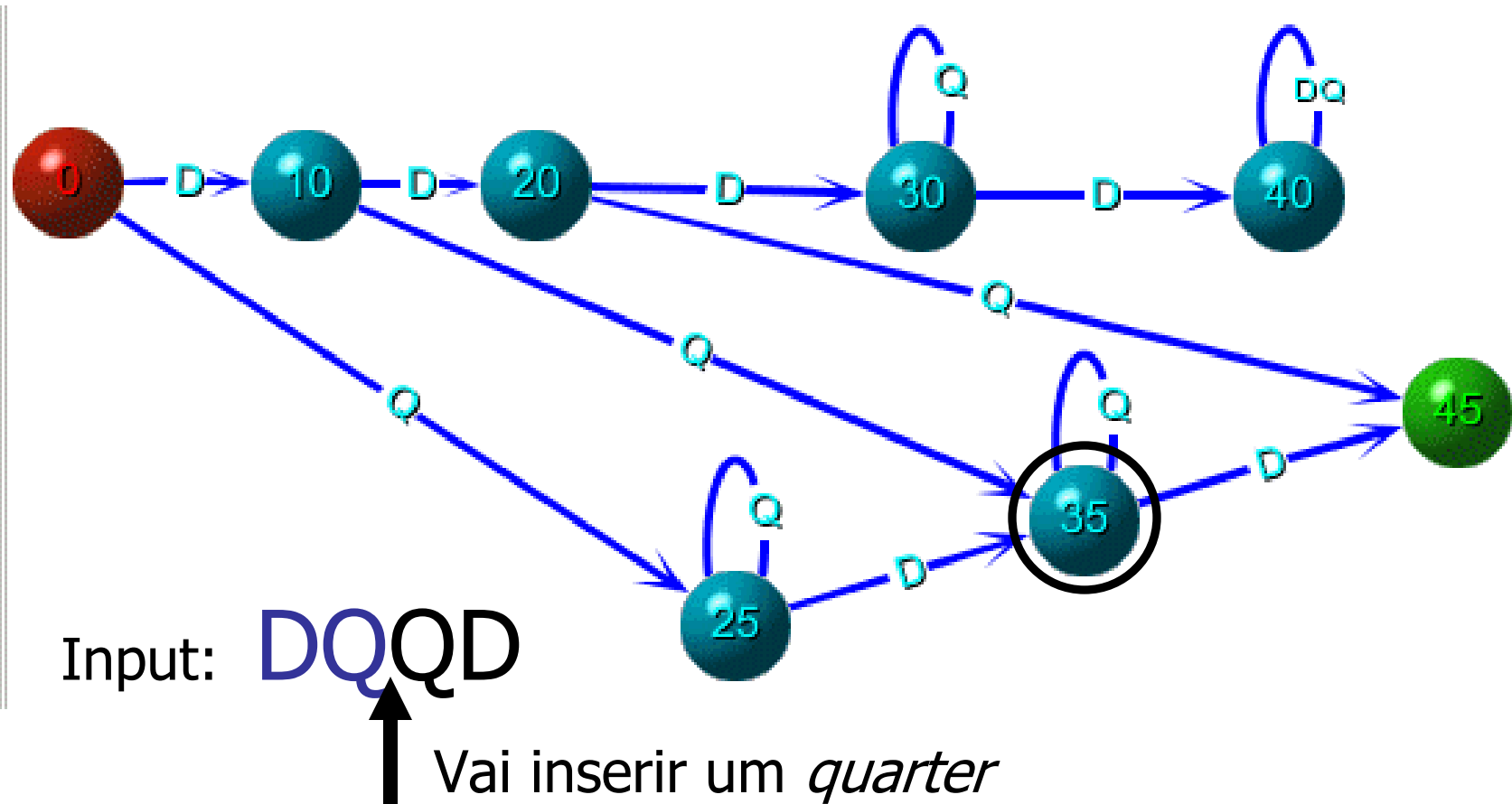


# Exemplo: Máquina de Venda

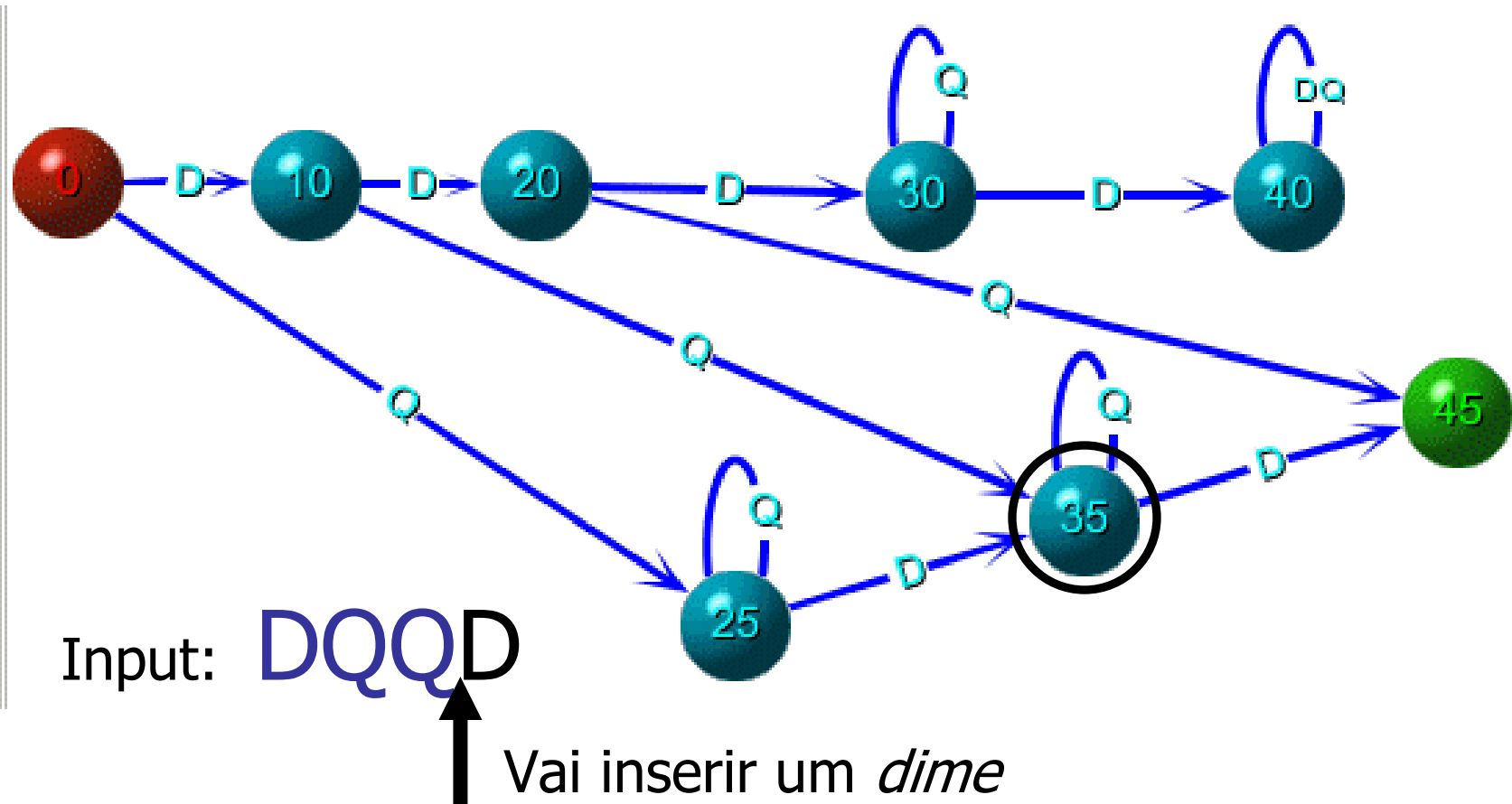




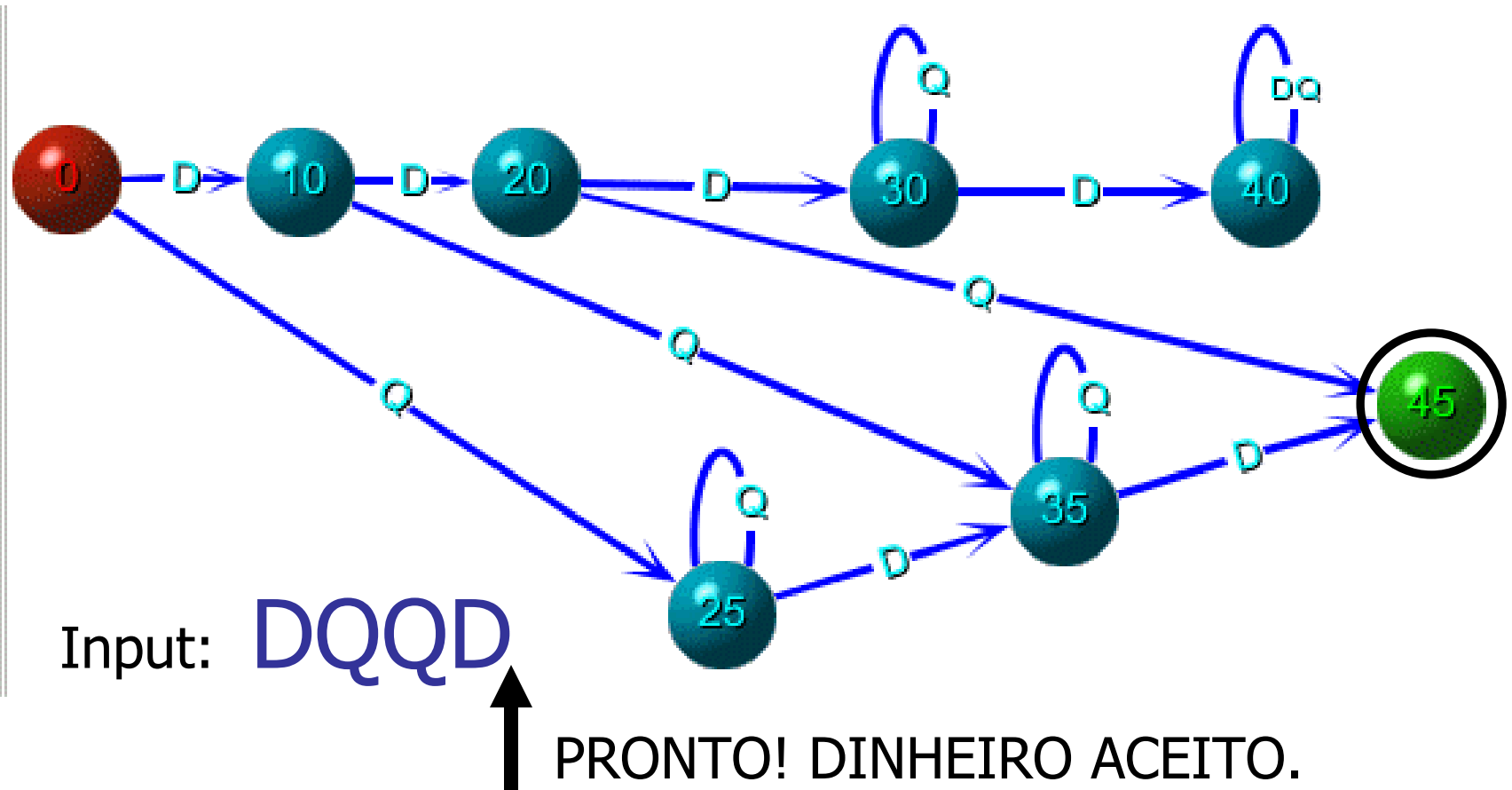
# Exemplo: Máquina de Venda



# Exemplo: Máquina de Venda



# Exemplo: Máquina de Venda



# Exemplo: Máquina de Venda

O que tornou esse exemplo mais simples do que o pseudocódigo Java?

# Vending Machine Example

1. São necessários apenas de 2 tipos de moedas “D” and “Q” – *símbolos/letras do alfabeto*
2. São necessários apenas 7 possíveis valores totais correntes – *estados/nodos/vertices*
3. Muito mais claro e esteticamente agradável

Podemos agora generalizar e abstrair...

# Alfabetos, Strings, Linguagens

DEF: Um **alfabeto**  $\Sigma$  é um conjunto de símbolos (caracteres, letras). Um **string** (ou palavra) sobre  $\Sigma$  é uma sequência de símbolos. O **string vazio** é um string que não contém nenhum símbolo, e é denotado por  $\varepsilon$ .

Q1: O que é  $\Sigma$  no exemplo da máquina de venda?

Q2: Quais strings são aceitos/não aceitos pela máquina de venda do exemplo?

Q3: O que significa  $\varepsilon$  no nosso exemplo?

# Alfabetos, Strings, Linguagens

R1:  $\Sigma = \{ D, Q \}$

R2:

Aceito: QDD, DQD, DDQ, QQQQDD, etc.

Não aceito: Q, D, DD, etc.

DDD ... você se ferrou!

R3:  $\epsilon$  significa tentar obter uma coca sem pagar nada (Não inserindo qq. moeda).

# Alfabetos, Strings, Linguagens

DEF: O ***comprimento*** de um string é o número de símbolos que ele contém (repetições são permitidas).

EX: Os comprimentos dos strings (QDD, DQD, DDQ, QQQQDD) são: 3, 3, 3, 6

Q: Qual é o comprimento de  $\epsilon$  ?



# Alfabetos, Strings, Linguagens

R:  $|\epsilon| = 0$

# Alfabetos, Strings, Linguagens

DEF: A **concatenação** de dois strings é o string resultante de colocar o segundo depois (à direita) do primeiro. Dados os strings  $u$  e  $v$ , denotamos sua concatenação por  $u \circ v$ , ou simplesmente  $uv$ .

EX:  $aba \circ cate = abacate$ ,  $QQ \circ DD = QQDD$ ,  
 $DDD \circ u$  não é aceito, qualquer que seja  $u$ !

Q1: Porque se pode afirmar isso?

Q2: Defina concatenação recursivamente.

Q3: Obtenha uma fórmula para  $|u \circ v|$ .

# Alfabetos, Strings, Linguagens

R1: Você já perdeu o seu dinheiro, não importa que outras moedas vá inserir.

$$R2: \varepsilon \circ v = v$$

$$au \circ v = a(u \circ v)$$

$$R3: |u \circ v| = |u| + |v|$$

# Alfabetos, Strings, Linguagens

DEF: O **reverso** de um string  $u$  é denotado por  $u^R$ .

EX:  $(\text{banana})^R = \text{ananab}$

DEF: Se  $\Sigma$  é um alfabeto,  $\Sigma^*$  denota o conjunto de todos os strings sobre  $\Sigma$ .  
Uma **linguagem sobre  $\Sigma$**  é um subconjunto de  $\Sigma^*$ , i.e. um conjunto de strings *cada um* sendo uma seqüência de símbolos de  $\Sigma$ .

# Alfabetos, Strings, Linguagens

EX:  $\Sigma = \{ D, Q \}$

$\Sigma^* = \{ \varepsilon,$   
D, Q,  
DD, DQ, QD, QQ,  
DDD, DDQ, DQD, DQQ, QDD, QDQ, QQD, QQQ,  
DDDD, DDDQ, ... }

Defina  $L = \{ u \in \Sigma^* \mid u \text{ resulta em uma venda} \}$

Exercício: Quais são os strings de  $L$  com comprimento 1? E comprimento 2? 3? 4? 5?

# Linguagens formais

- Tem uma sintaxe bem definida
- Tem uma semântica precisa
- Exemplos:
  - Java, C, Pascal, HTML

# Exemplos

- Seja  $\Sigma = \{0, 1\}$ . São exemplos de linguagens:
  - $\emptyset$
  - $\{\epsilon\}$
  - $\{0\}$
  - $\{\epsilon, 0\}$
  - $\{w \in \Sigma^* \mid 1 \leq |w| \leq 5\}$
  - $\{0^n 1^n \mid n \in \mathbf{N}\}$

# União, interseção e diferença

- Como linguagem é um conjunto podemos usar as operações de conjunto para definir linguagens.
- Sejam  $L_1$  e  $L_2$  linguagens sobre  $\Sigma_1$  e  $\Sigma_2$ .
  - $L_1 \cup L_2$ , uma linguagem sobre  $\Sigma_1 \cup \Sigma_2$
  - $L_1 \cap L_2$ , uma linguagem sobre  $\Sigma_1 \cap \Sigma_2$
  - $L_1 - L_2$ , uma linguagem sobre  $\Sigma_1$



# Prefixo, sufixo e substring

- Seja uma palavra  $w=xzy$ , em que  $x$ ,  $y$ , e  $z$  podem ser  $\varepsilon$ .
  - $z$  é uma substring de  $w$
  - $x$  é um prefixo de  $w$
  - $y$  é um sufixo de  $w$
- Quais são os prefixos, sufixos e substrings de  $w=abc$ ?

# Concatenação de linguagens

- A concatenação de duas linguagens  $L_1$  e  $L_2$  é dada por:
  - $L_1L_2 = \{xy \mid x \in L_1 \text{ e } y \in L_2\}$
  - $\{0, 01\}\{00, 11, \varepsilon\} =$
- Sejam as linguagens  $L_1 = \{w \in \{0, 1\}^* \mid |w|=5\}$  e  $L_2 = \{0y \mid y \in \{0, 1\}^*\}$ .
  - $L_1L_1 =$
  - $L_1L_2 =$
  - $L_2L_1 =$
  - $L_2L_2 =$

# Fecho de Kleene (Kleene star)

- Seja  $L$  uma linguagem.  $L^n$  será usada para denotar  $L$  concatenada com  $L$   $n$  vezes.
  - $L^0 = \{\varepsilon\}$
  - $L^n = L^{n-1} L$ , para  $N > 0$
- Fecho de Kleene de uma linguagem  $L$ ,  $L^*$ , pode ser definida por:
  - $\varepsilon \in L^*$
  - Se  $x \in L^*$  e  $y \in L$ , então  $xy \in L^*$

# Fecho positivo de Kleene (Kleene positivo)

- $L^+ = LL^*$
- $L^* = L^+ \cup \{ \varepsilon \}$
- Ejemplos:
  - $\emptyset^* =$
  - $\emptyset^+ =$
  - $\{\varepsilon\}^* =$
  - $\{\varepsilon\}^+ =$
  - $\{0\}^* =$
  - $\{0\}^+ =$
  - $\{\varepsilon, 00, 11\}^* =$

# Exercício

- Usando operadores de conjunto e linguagem defina sobre  $\Sigma=\{0,1\}$ :
  - a) A linguagem dos strings que começam com 0.
  - b) A linguagem dos strings que possuem a substring 00.
  - c) A linguagem dos strings que possuem 00 ou 11.
  - d) A linguagem dos strings que possuem tamanho par.
  - e) A linguagem dos strings que possuem tamanho ímpar.
  - f) A linguagem dos strings que terminam com 0 seguido por um número ímpar de 1's consecutivos.
  - g) A linguagem dos strings de tamanho par que começam ou terminam com 0.

# Exercício

- Forneça definições recursivas para:
  - a)  $\{0\}^* \{1\}^*$
  - b)  $\{0^n 1^n \mid n \in \mathbf{N}\}$
  - c)  $\{w \in \{0,1\}^* \mid w \text{ contém } 00\}$
  - d)  $\{w \in \{0,1\}^* \mid |w| \text{ é par}\}$