

Aula 2

Conteúdo

1	Modelação de Sistemas Concorrentes	1
2	Sistemas de Transição	2
3	Não-determinismo	4
4	Tipo de Sistemas de Transição	6
5	Modelação de Processos Concorrentes	7

1 Modelação de Sistemas Concorrentes

Máquina de Vendas



Acções de comunicação (*observáveis*)

- Pressionar um botão
- Inserir moedas
- Injectar uma bebida/comida

Acções internas

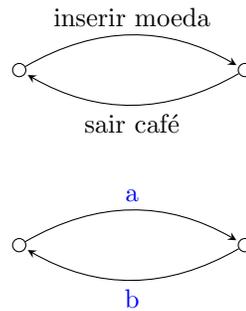
(*não observáveis*)

- Escolher a bebida
- Mecanismo para injectar o produto certo
- Tirar o dinheiro do bolso

- Verificar as moedas inseridas
- Pegar na bebida

Modelo abstracto de um processo

- Um processo é descrito usando um conjunto de ações, *alfabeto do processo*
- *Com*: conjunto de ações de comunicação (actividades observáveis) a, b, c, \dots
- *Int*: conjunto de acções internas (computações locais) $[a], [b], [c], \dots$
- $Act = Com \cup Int$ conjunto de todas as acções
- A execução de uma *ação* muda o estado do processo:



Usar só letras para simplificar

2 Sistemas de Transição

Constituintes de um processo

- Um conjunto de estados
- Um conjunto de transições entre estados
- Um estado inicial
- Cada transição é etiquetada por uma ação que acciona a mudança de estado.

Estados



- Cor actual de um semáforo de trânsito
- Valor corrente das variáveis de um programa e do contador de programa
- A avião a voar
- Valor da conta bancária

Transições

- Passagem de um semáforo de trânsito de vermelho para verde
- Execução de um comando num programa
- Aterragem de um avião
- Depositar dinheiro numa conta bancária

Sistemas de transição

Definição 2.1. *Um sistema etiquetado de transições (LTS) sobre Act é um triplo $(S, \longrightarrow, s_0)$*

- S conjunto de estados
- $\longrightarrow \subseteq S \times Act \times S$ a relação de transição
- $s_0 \in S$ estado inicial

Um Fósforo...



- $S = \{UNUSED, BURNING, EXTINGUISHED\}$
- $\longrightarrow = \{(UNUSED, strike, BURNING), (BURNING, extinguish, EXTINGUISHED)\}$
- $s_0 = UNUSED$

Sucessores

Em vez de $(s, \alpha, s') \in \longrightarrow$ escrevemos $s \xrightarrow{\alpha} s'$ e dizemos que s' é um *sucessor* de s .

- $Post(s, \alpha) = \{ s' \in S \mid s \xrightarrow{\alpha} s' \}$
- $Post(s, A) = \bigcup_{\alpha \in A} Post(s, \alpha)$, $A \subseteq Act$
- $Post(s) = Post(s, Act)$
- $Post(C, \alpha) = \bigcup_{s \in C} Post(s, \alpha)$, $C \subseteq S$
- $Post(C, A) = \bigcup_{\alpha \in A} Post(C, \alpha)$
- Ações que ocorrem no estado s

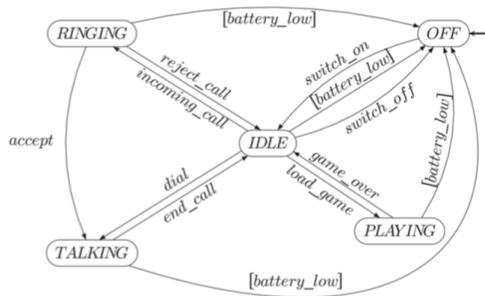
$$Act(s) = \{ \alpha \in Act \mid \exists s' : s \xrightarrow{\alpha} s' \}$$

- Ações que podem ser observadas no estado s

$$Com(s) = \{ \alpha \in Com \mid \exists s' : s \xrightarrow{\alpha} s' \}$$

- Análogo para $Int(s)$

Um Telefone



- Descrever o sistema de transições $(S, \longrightarrow, s_0)$ e $Act = Com \cup Int$.
- $Post(RINGING, Act) = \{TALKING, IDLE, OFF\}$
- $Post(Post(RINGING, Act), Int) = \{OFF\}$
- $Com(RINGING) = \{accept, reject_call\}$

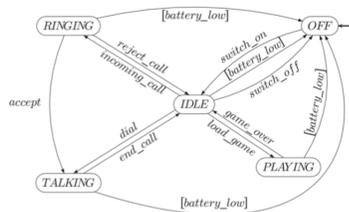
Estados atingíveis

- Seja $TS = (S, \longrightarrow, s_0)$ um LTS.
- Um estado $s' \in S$ é *atingível* de s se $s' = s$ ou
- $\exists n \geq 0$ e estados s_1, s_2, \dots, s_n tal que $s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \cdots \xrightarrow{\alpha_n} s_n$ e $s_n = s'$.
- $Reach(s)$ conjunto de estados atingíveis de s
- $Reach(TS) = Reach(s_0)$

3 Não-determinismo

Não-determinismo

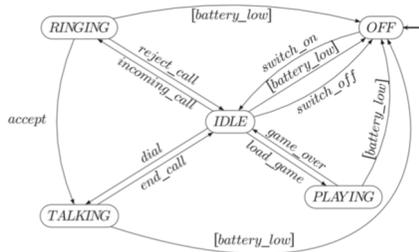
- Permite descrever o comportamento de sistemas reais
- Abstração de detalhes dos sistemas
 - não é necessária uma descrição completa
 - os sistemas podem ser muito complexos
 - alguns parâmetros são desconhecidos
- existem várias maneiras de interagir
- a noção é mais geral que em Linguagens Formais



Não-determinismo

- Seja $TS = (S, \longrightarrow, s_0)$ um LTS.
- TS é *determinístico* se e só se para todo $s \in S$
- $|Post(s)| \leq 1$ e $|Act(s)| \leq 1$
- senão é *não-determinístico*

Um sistema é não determinístico se tem um estado com duas ou mais transições mas não sabemos qual irá acontecer.

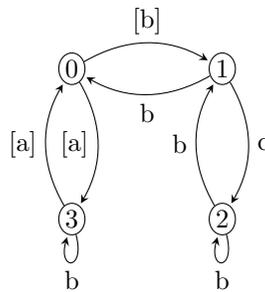


Não-determinismo interno e externo

Um estado s é

- não-determinístico *externo* sse $|Com(s)| > 1$
- não-determinístico *interno* sse $|Post(s, Int)| > 1$ ou $|Post(s, a)| > 1$ para algum $a \in Com(s)$

Não-determinismo interno e externo



- 1 é não-determinístico externo
- 0 e 2 são não-determinísticos interno.
- 3 é não-determinístico mas nem interno nem externo.

4 Tipo de Sistemas de Transição

Tipo de Sistemas de Transição

Seja $TS = (S, \longrightarrow, s_0)$ com ações em Act

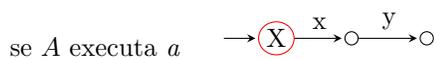
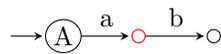
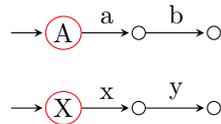
- os LTS são isomorfos como grafos (Isomorfismo)
- ou os LTS tenham os mesmos caminhos (Equivalência por traços (linguagens))
- ou ...
- iremos deixar isto para mais tarde

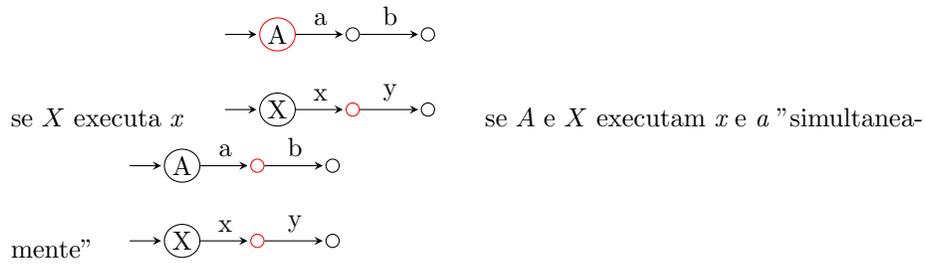
5 Modelação de Processos Concorrentes

Processos Concorrentes

- Cada processo é representado por um sistema de transições (LTS)
- Neste caso, o tempo avança quando se muda de estado por uma transição (execução de um programa sequencial)
- O que acontece se executam concorrentemente?
- Assumimos que:
- O tempo só é considerado de forma relativa (um processo p ocorre antes do processo q)
- A ações são atômicas e instantâneas
- Os processos concorrentes executam independentemente excepto se explicitamente comunicarem (coordenação)

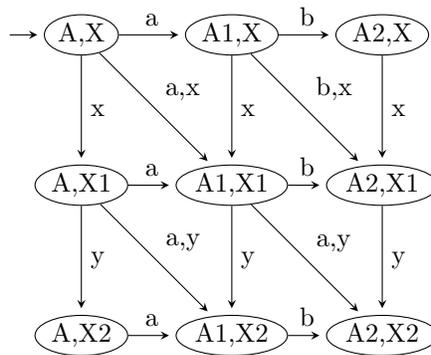
Processos concorrentes A e X





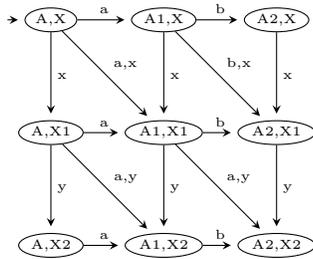
Produto de LTS

A execução de dois processos é um novo processo (LTS)!



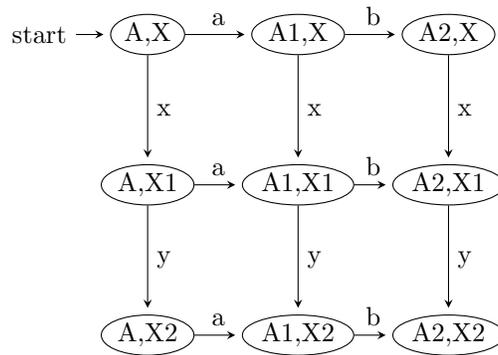
Simultaneidade

- As diagonais têm conjuntos de ações (!)
- Mas como observar que duas ações independentes ocorrem ao mesmo tempo?
- O seu efeito corresponde a ser primeiro uma e depois outra não interessando a ordem
- Sendo não determinístico, podemos ignorar essas transições.
- Não são observáveis



Produto de LTS: sem *diagonais*

Os estados atingíveis são os mesmos. As ações podem ocorrer por qualquer ordem (não determinismo).



Modelo de Concorrência

- Dados dois processos independentes a concorrência é a
- Intercalagem (*interleaving/shuffle*) não-determinística de todas as ações dos dois processos
- Diamante de intercalagem

