

Grafcet

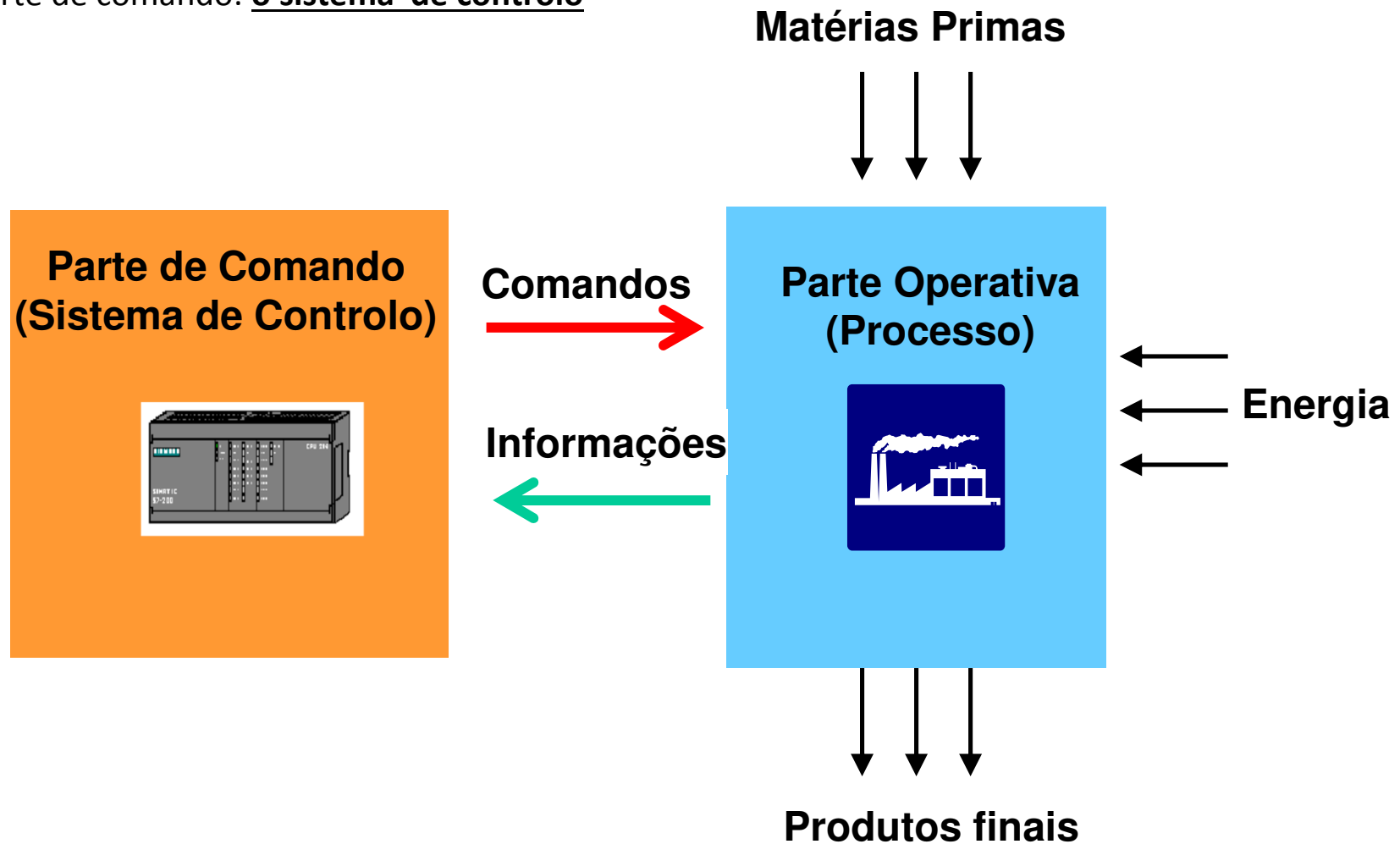
*Graphe Fonctionnel de Commande, Étapes
Transitions*

Sumário

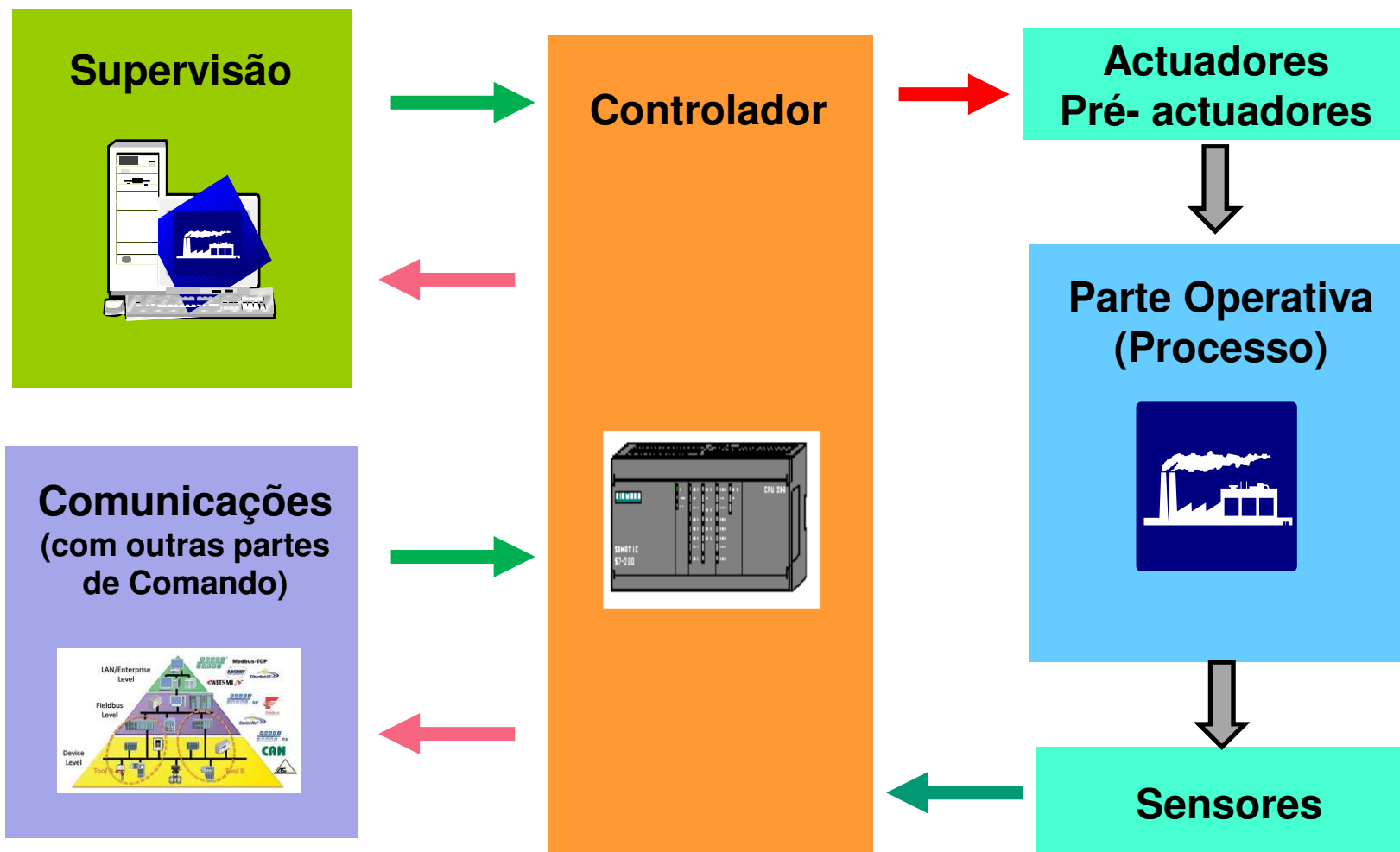
- Introdução
- Estruturas típicas
 - Sequência
 - Concorrência
 - Sincronização/partilha de recursos
- Hierarquia
- Validação / verificação
- Implementação em plataformas genéricas

Estrutura de um Sistema Automatizado

- Parte operativa: o processo
- Parte de comando: o sistema de controlo



Esquema Funcional de um Sistema Automatizado



Especificação do controlador

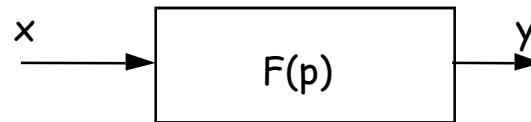
- Como especificar o funcionamento do controlador do processo ?



- Depende do tipo de processo a ser controlado:
 - Processos contínuos
 - Processos discretos

Controlo de processos contínuos

- O estado do processo pode ser descrito através de uma variável analógica (temperatura, tensão, velocidade, etc.);
- Entre a entrada e a saída do processo existe uma relação bem estabelecida que pode ser descrita através de um modelo contínuo (função de transferência);

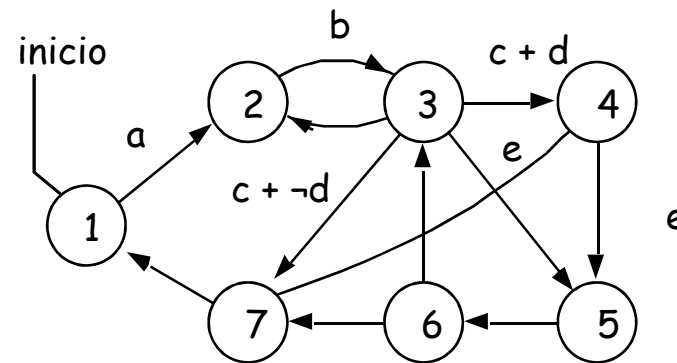


- A tarefa principal do sistema de controlo neste tipo de processo diz respeito à sua **regulação**, ie. manter o estado da saída num determinado nível ou valor.
- Típico das indústrias de processos (i.e. químicas, farmacêuticas, etc.)

Controlo de processos discretos

- A evolução do processo é definida por um conjunto de estados. As transições entre estados são definidas por eventos:

- Definidos por condições (ex. $A > 1$)
- Tempo ($t = 0.1s$)



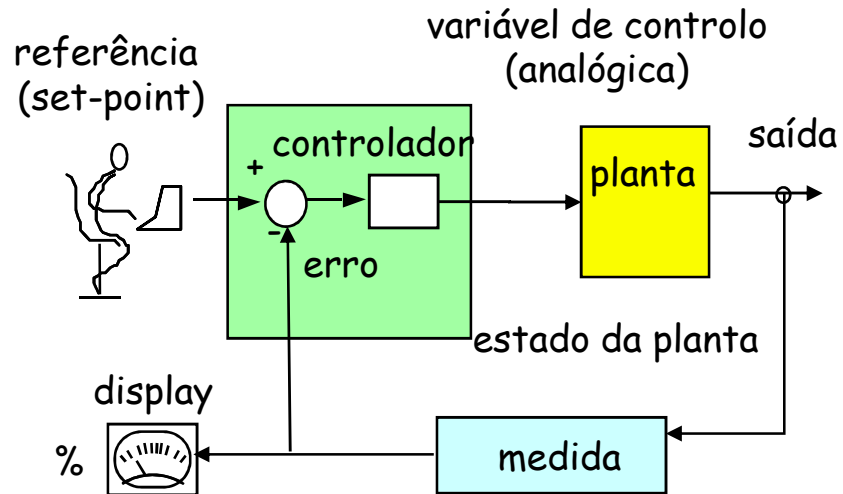
- Descritos através de:

- Sistemas combinacionais
- Sistemas sequenciais
- Máquinas de estados
- Grafcet
- Redes de Petri, etc.

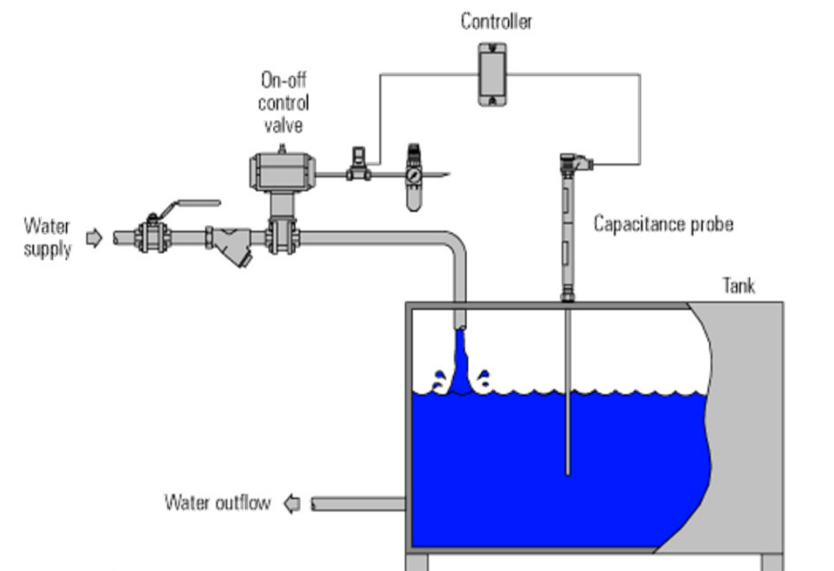
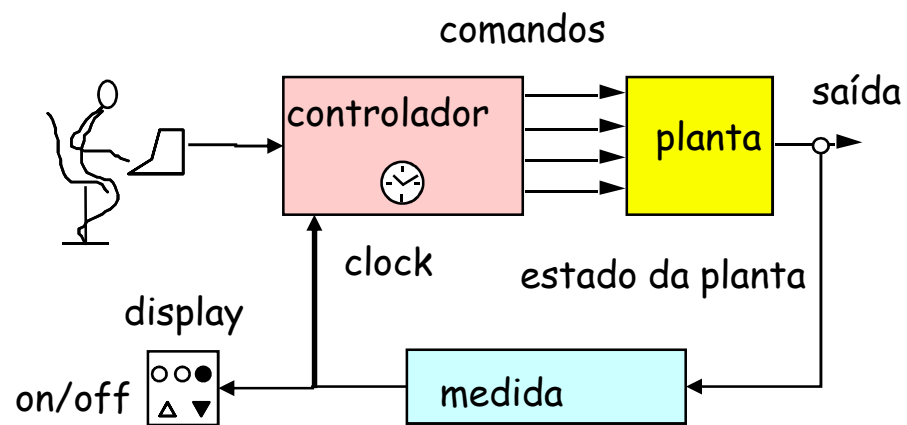
- A tarefa principal do sistema de controlo neste tipo de processos diz respeito ao seu **comando**, i.e. envio de comandos para o processo.
- Típico das indústrias de manufatura (ex. manufatura de itens individuais ou lotes)

Exemplo: controlo de um tanque

— Processo contínuo : regular o nível do tanque



— Processo discreto: encher ou esvaziar o tanque



O Grafcet

- O **Grafcet** é um formalismo desenvolvido para modelar **especificação funcional de controladores para sistemas de eventos discretos** (processos discretos)
 - Sistemas que são caracterizados por um conjunto de estados discretos e cujas transições entre estados (i.e. o comportamento ao longo do tempo) são governadas por eventos assíncronos (i.e. não sincronizados com um relógio)
- Principais vantagens:
 - Notação compacta, não ambígua e formal
 - Formalismo gráfico e de fácil interpretação
 - Metodologia estruturada, abordagem *top-down*
 - Modelação de funções lógicas, comportamentos sequenciais e concorrentes
 - Normalizada : IEC 60848 Ed. 2.0 - GRAFCET Specification Language for Sequential Function Charts
 - Facilidade de implementação em plataformas computacionais:
 - Autómatos : uma 'linguagem' específica - SFC (Sequential Function Charts)
 - Outras plataformas computacionais: existem métodos (simples) para implementar (i.e. programar) o Grafcet utilizando linguagens de programação correntes (ex. C, Java, etc.)
 - Fácil comunicação entre profissionais de várias áreas envolvidas no processo de automatização.

Elementos básicos

— O Grafect é um gráfico composto por:

Etapas: que representam o estado do sistema:

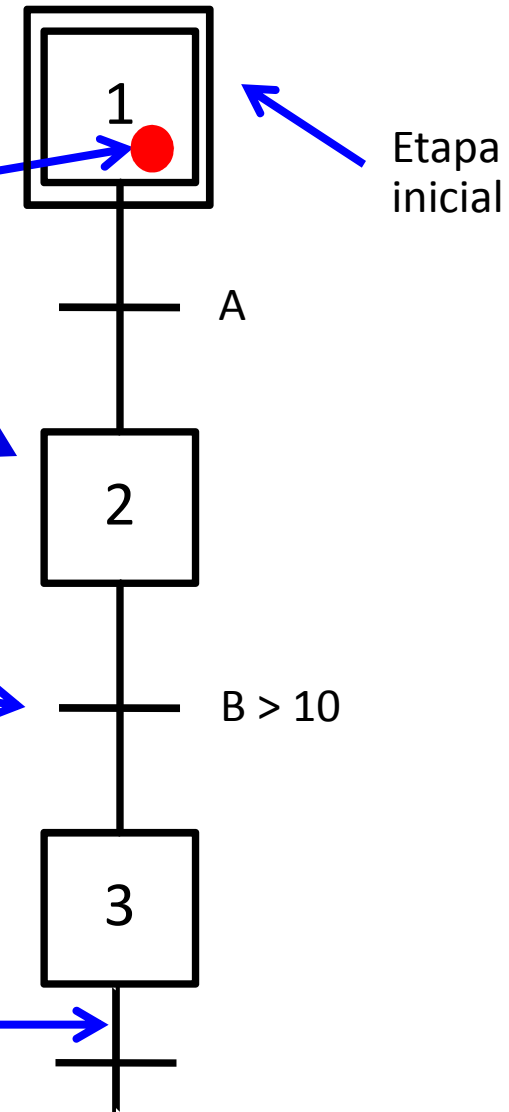
- activas: estado actual do sistema
- inactivas

Transições: as condições (eventos) que levam à mudança de estado

- definida por uma condição booleana (predicado) que pode incluir quaisquer tipo de variáveis

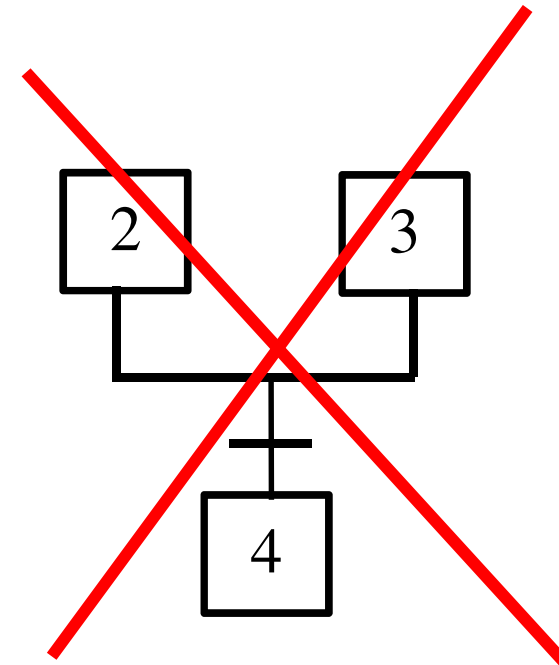
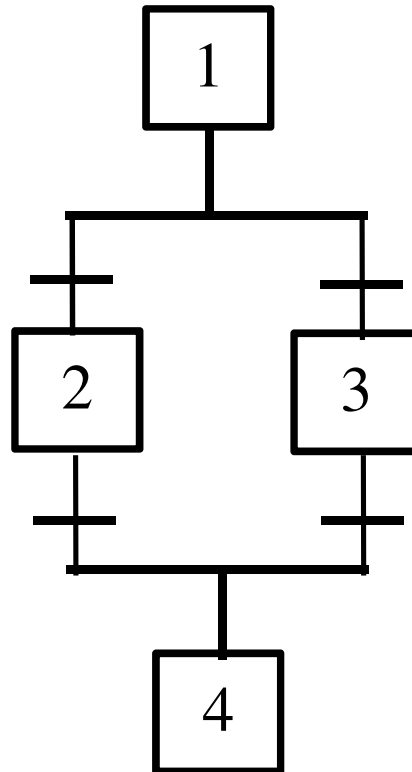
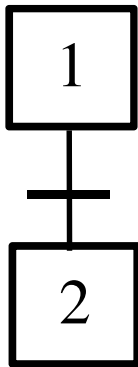
Ligações orientadas ('arcos'): descrevem o 'fluxo' da evolução do sistema

- ligações etapa → transição e transição → etapa



Regras básicas da estrutura

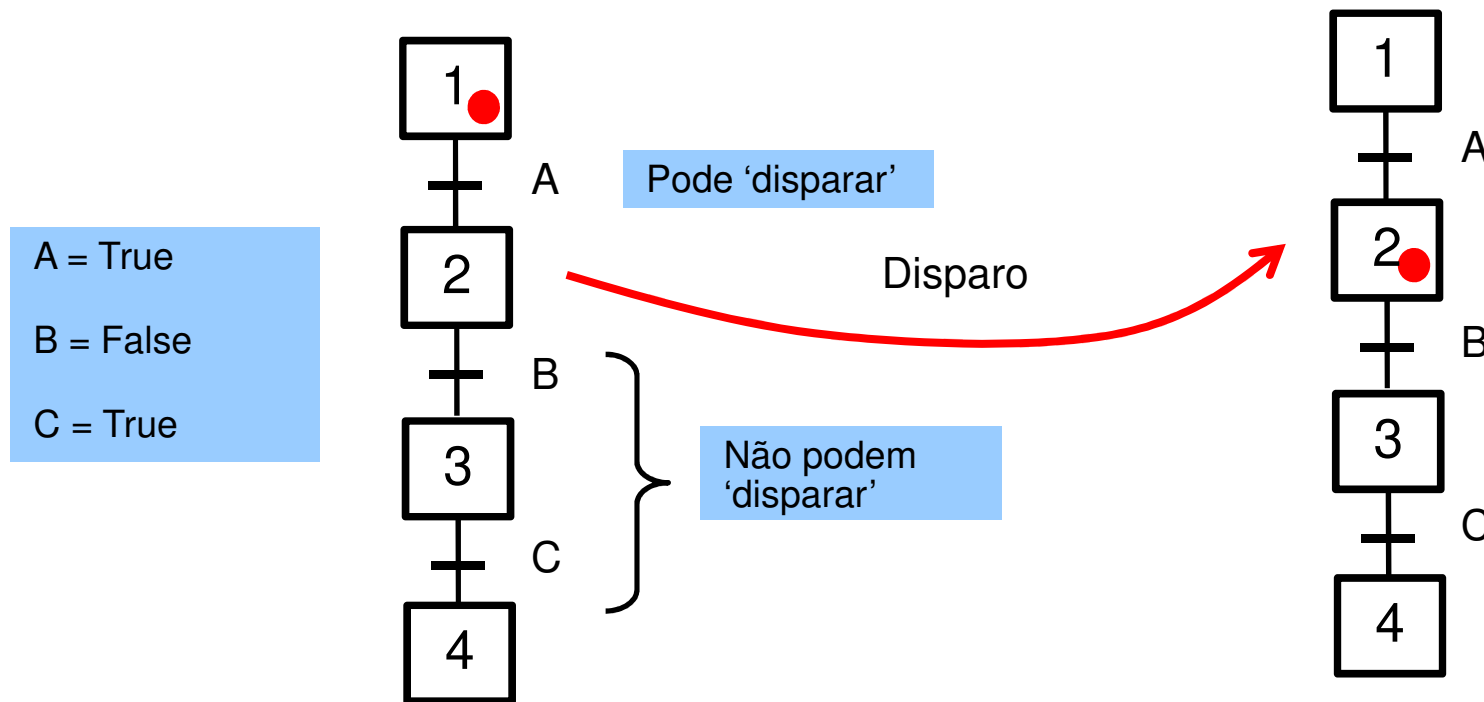
- Entre 2 etapas consecutivas é necessário existir sempre uma transição, e vice-versa.
- Um 'arco' só deve ligar uma etapa a uma transição e vice-versa



Embora formalmente não esteja errado, não se recomenda esta representação

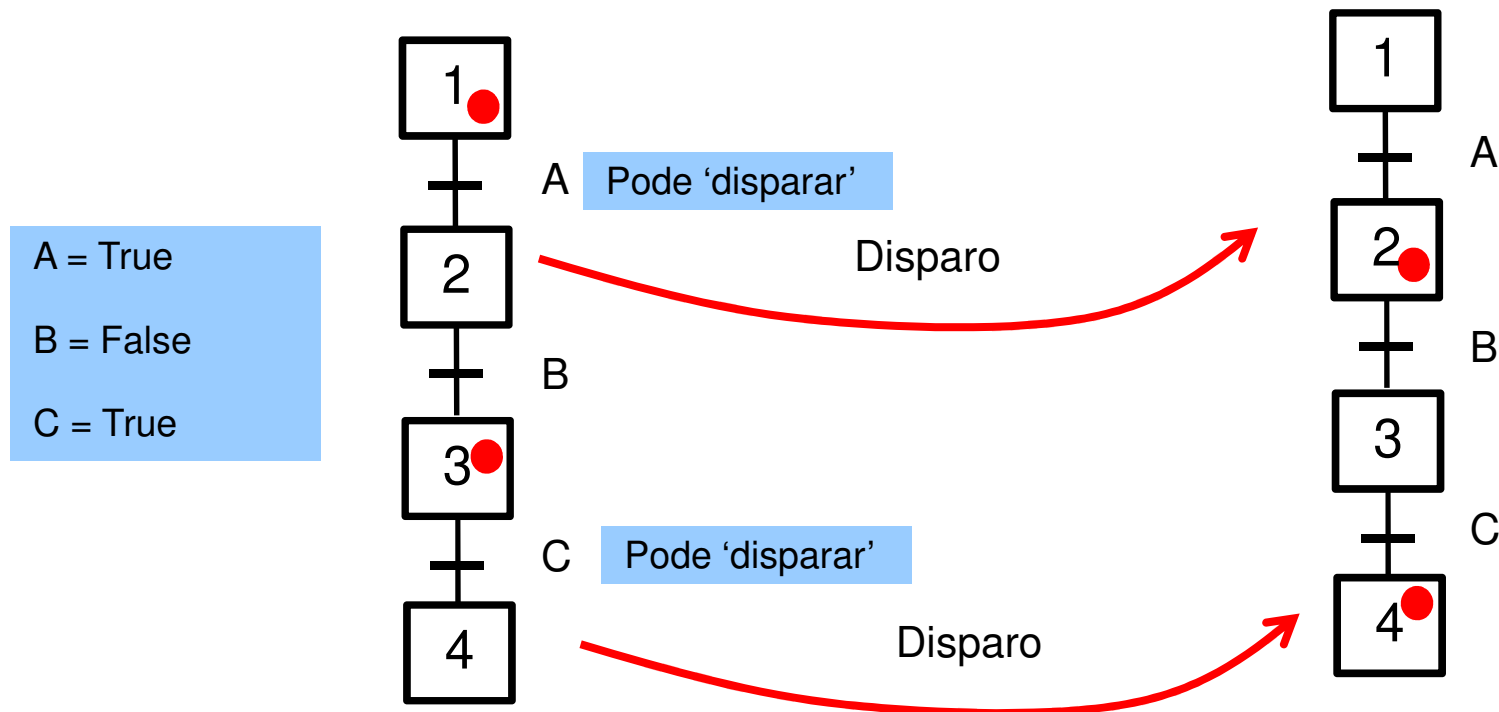
Regras de evolução

- Uma transição só pode ‘disparar’ quando todas as etapas anteriores estão activas e quando a condição lógica associada à transição é verdadeira.
- Quando uma transição dispara:
 - Todas as etapas anteriores são desactivadas e todas as etapas posteriores são activadas;
 - O ‘disparo’ é instantâneo (i.e tempo nulo).



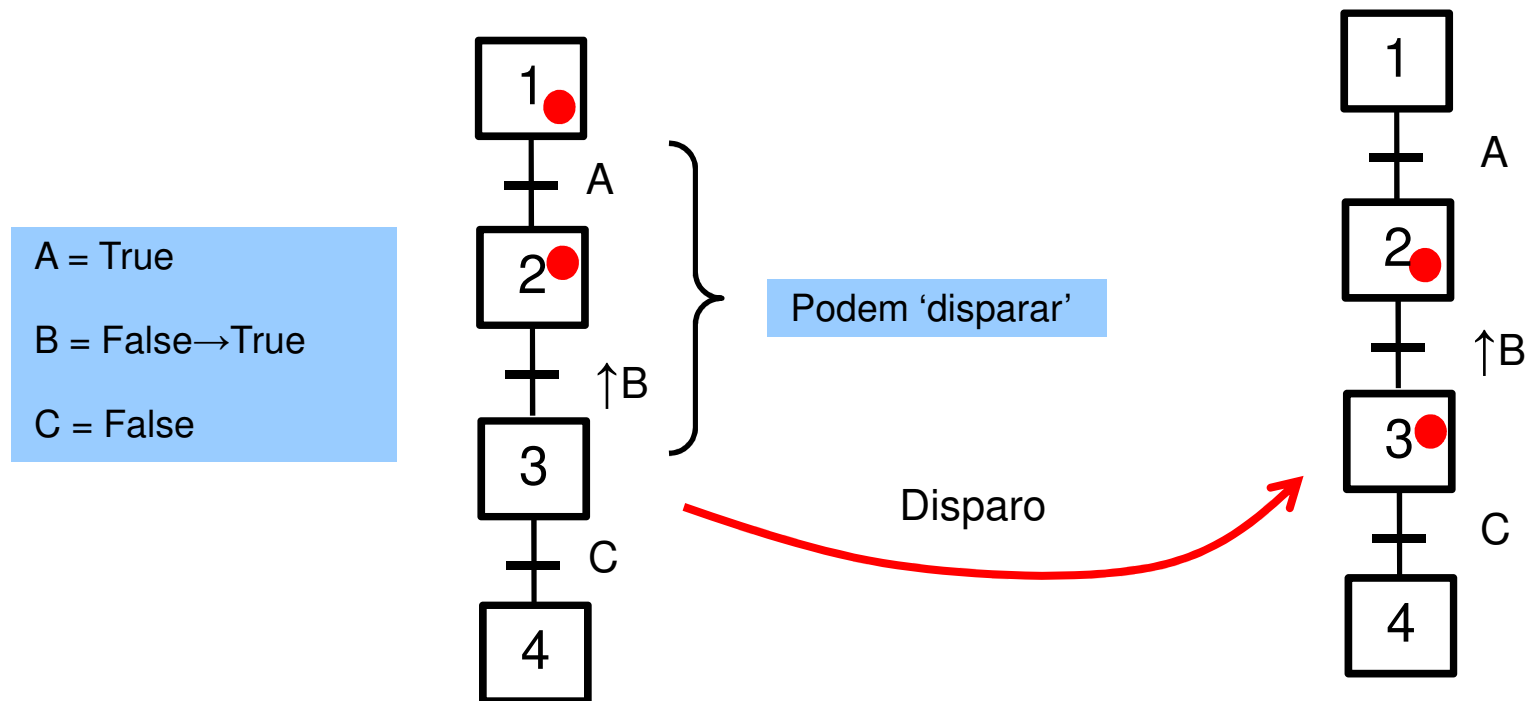
Regras de evolução (2)

- Se existirem várias transições que podem ‘disparar’ no mesmo instante, então são ‘disparadas’ no mesmo instante:
 - Não existe uma ordem pela qual os ‘disparos’ são realizados (decorrem em paralelo)



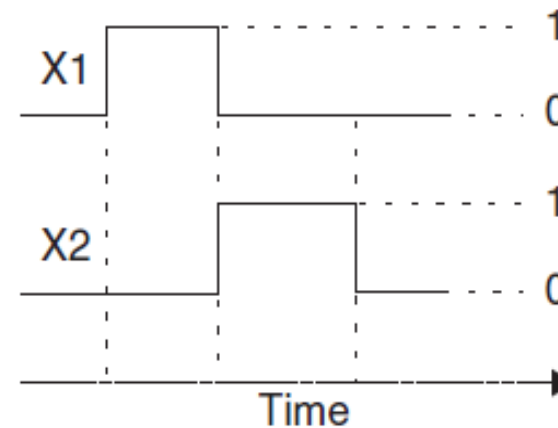
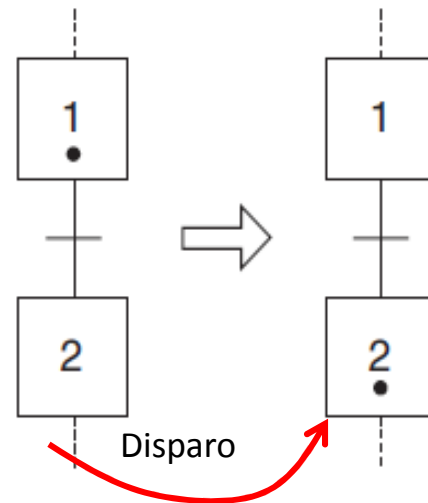
Regras de evolução (3)

- Se uma etapa é activada e desactivada no mesmo instante, então permanece activada

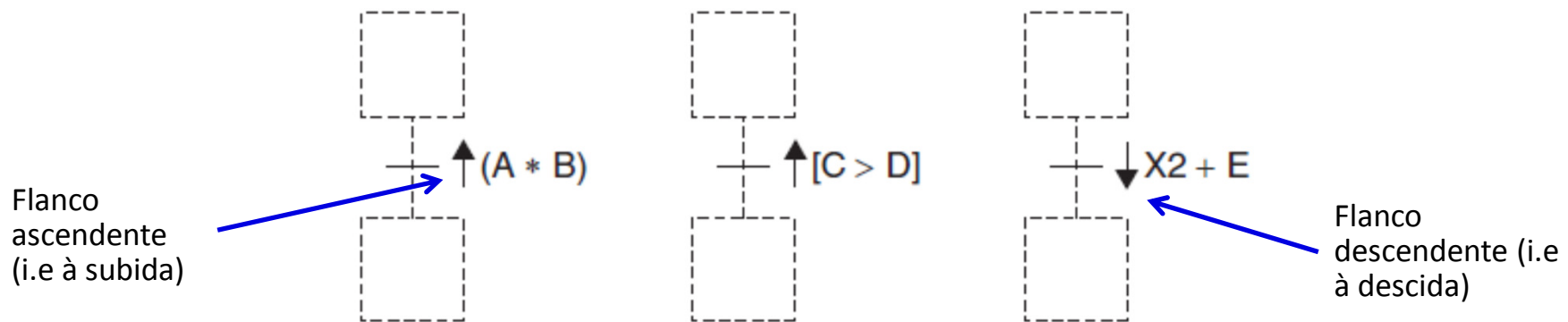


Variáveis internas & eventos

- Associada a cada etapa i existe uma variável booleana X_i que indica o estado da etapa (activa: $X_i=1$, ou inactiva: $X_i=0$).

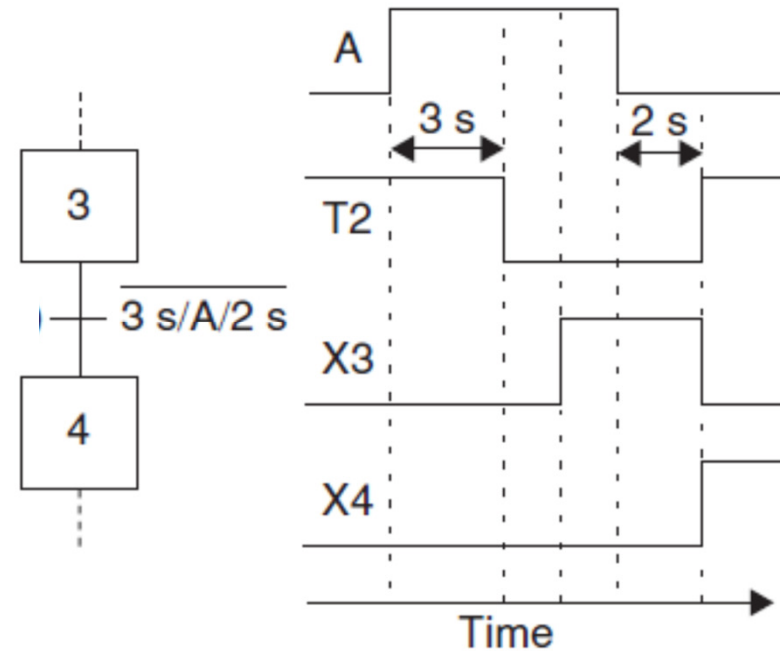
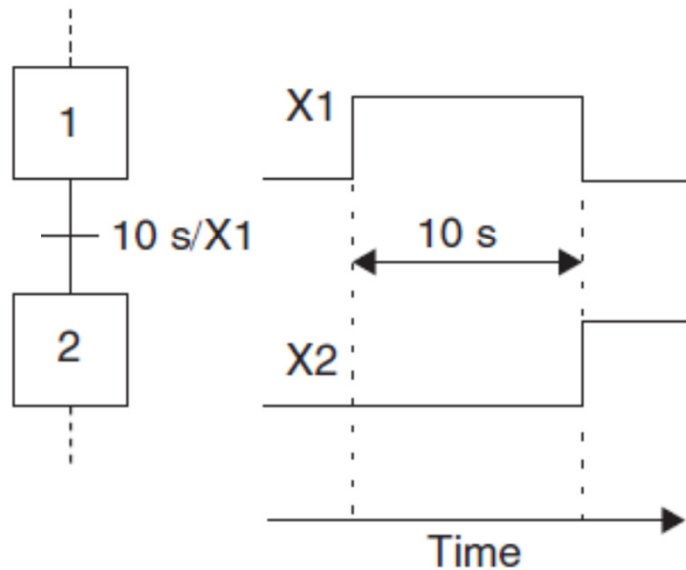


- É possível definir eventos associados às transições ($0 \rightarrow 1$ e $1 \rightarrow 0$) de variáveis booleanas (ou de condições lógicas).



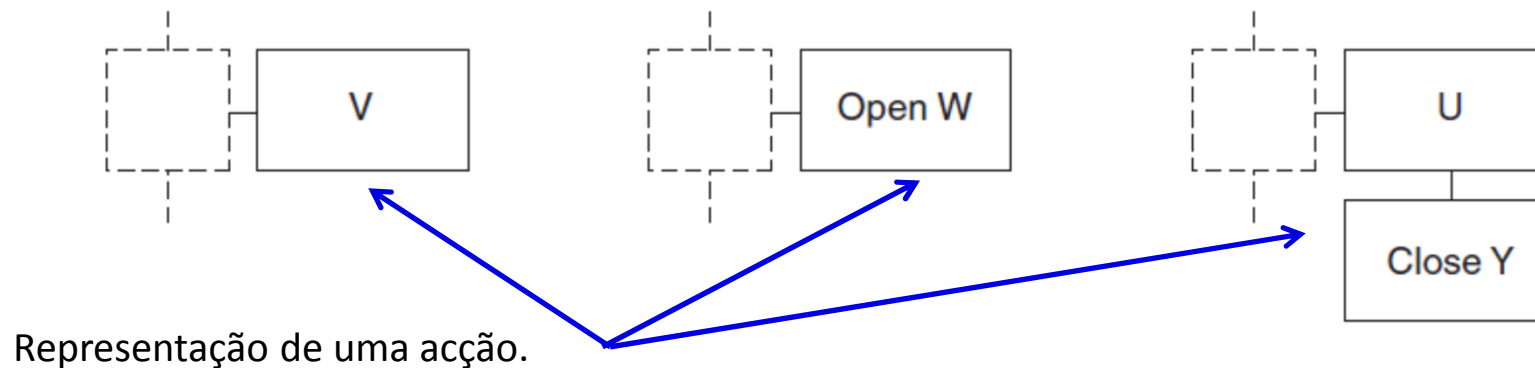
Representação do tempo

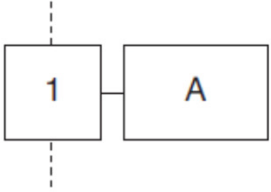
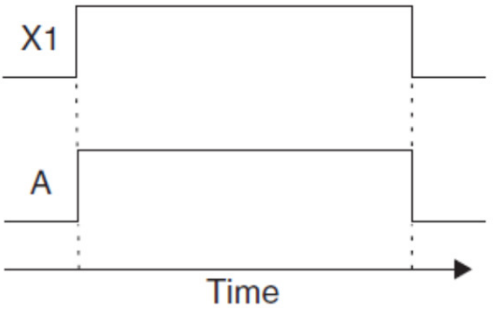
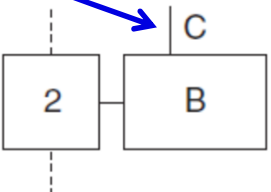
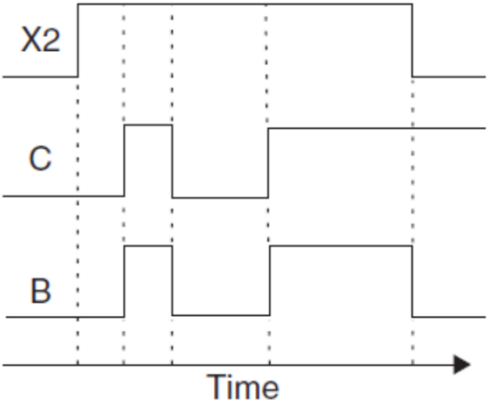
- É possível definir condições (i.e. eventos) que dependem do tempo.
- A sintaxe utilizada é a seguinte: $T1 / C / T2$, em que:
 - $T1$ e $T2$ são tempos, e C é uma variável booleana (ou uma condição lógica).
 - $T1$: tempo após activação de C
 - $T2$: tempo após desactivação de C



Acções

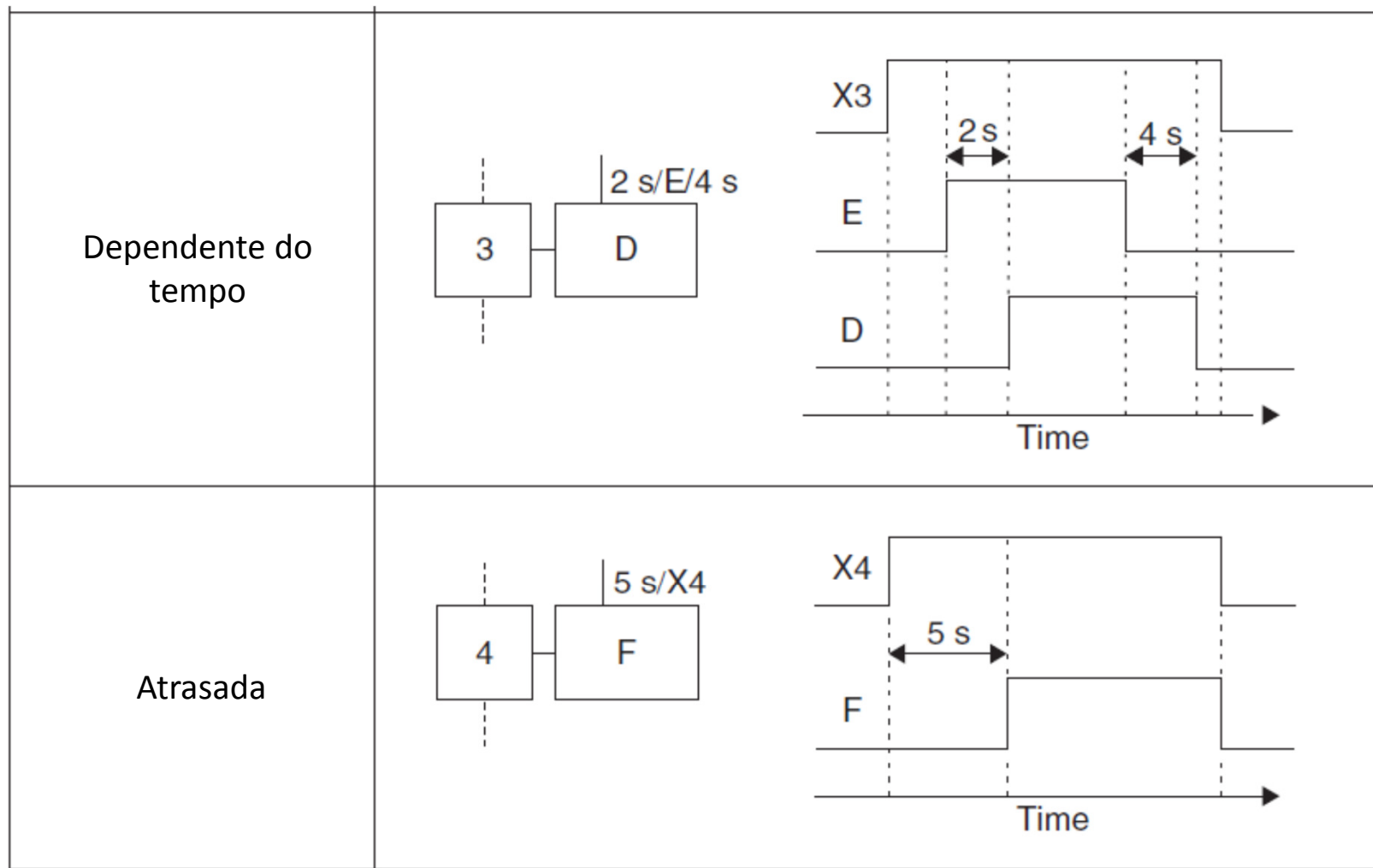
- A cada etapa podem ser associadas acções.
- Em termos simples as acções representam os comandos que são enviados para o sistema automatizado.
- Existem 2 tipos de acções:
 - Contínuas: executadas continuamente enquanto uma etapa está activa;
 - É possível definir condições adicionais para a sua execução (ver adiante)
 - Memorizadas/impulsionais ('stored'): executadas apenas quando ocorre um evento



Continuous actions types	Example	
Básica		
Condicionada		

Representação de condições adicionais: a acção só é executada quando, a etapa estiver activa e a condição for verdadeira

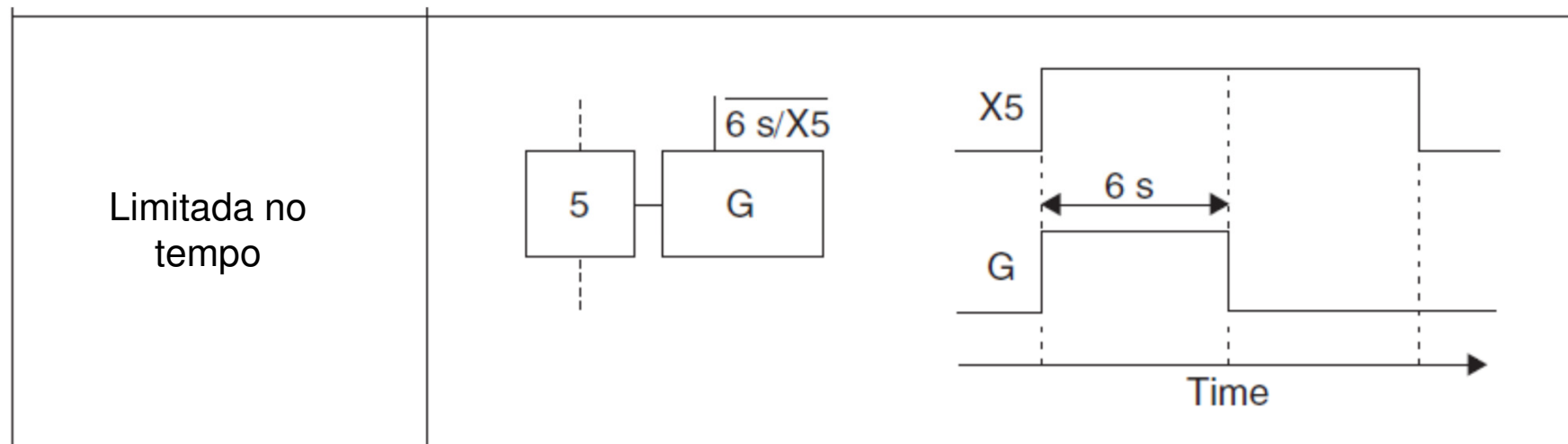
Acções contínuas (2)



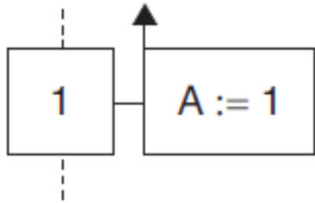
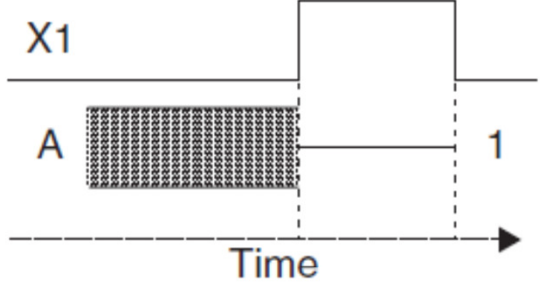
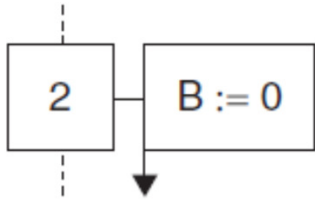
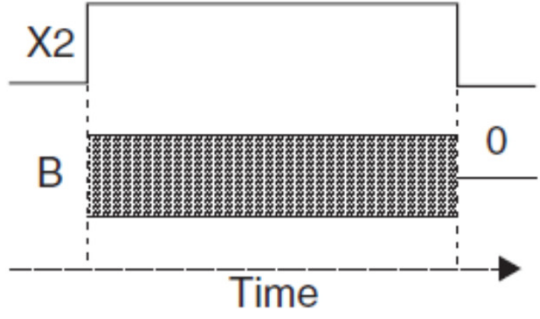
As condições associadas às acções podem ser qualquer expressão lógica (incluindo variáveis internas e tempo)

Acções contínuas (3)

- Considera-se que uma variável (booleana) está desactivada (i.e. OFF) quando não é referida em nenhuma acção contínua (que esteja a ser executada)
 - Para ‘cancelar uma acção’ basta não a referir
- Uma acção contínua não é executada quando uma etapa é activada e desactivada no mesmo instante (não teria lógica este comportamento)

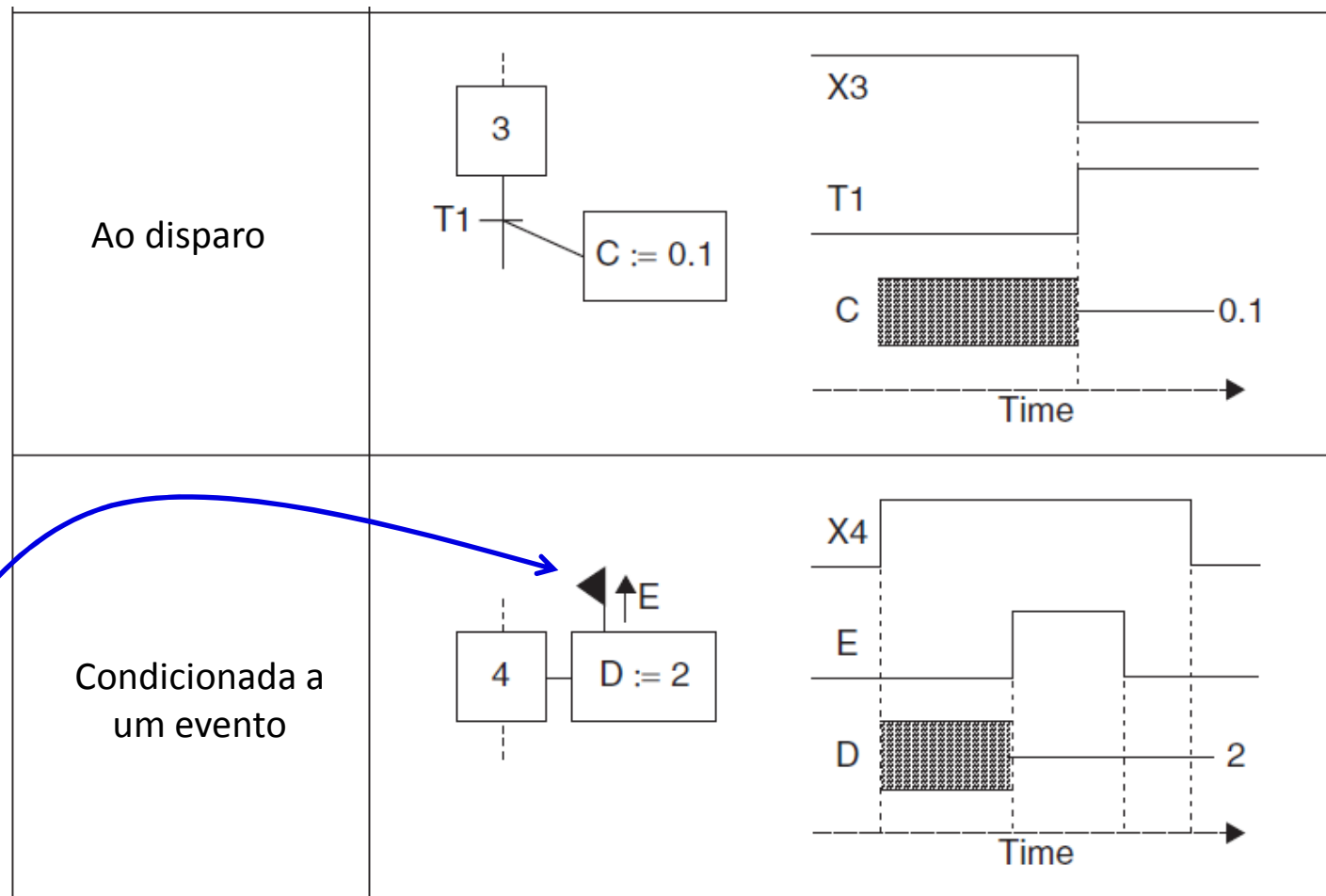


Acções impulsionaís/armazenadas

Stored actions types	Example
À activação	 
À desactivação	 

O valor da variável é modificado apenas ocorre o evento, e mantém este valor até que ocorra uma nova modificação (i.e. não depende da etapa ficar inactiva)

Acções impulsivas/armazenadas (2)



É possível incluir eventos nas condições de execução de uma acção

Uma acção impulsional é executada quando uma etapa é activada e desactivada no mesmo instante