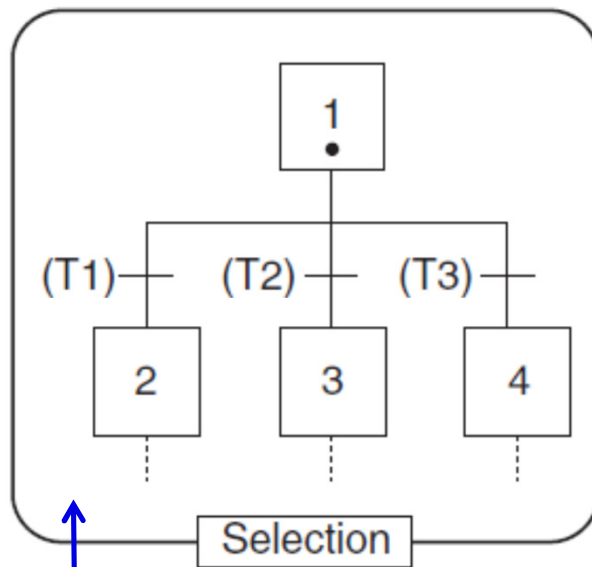


Sumário

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

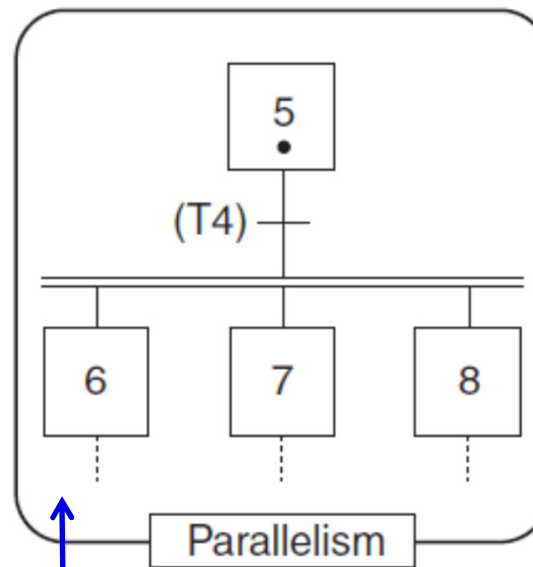
Sequências de execução

- Existem estruturas específicas para representar os comportamento mais comuns.



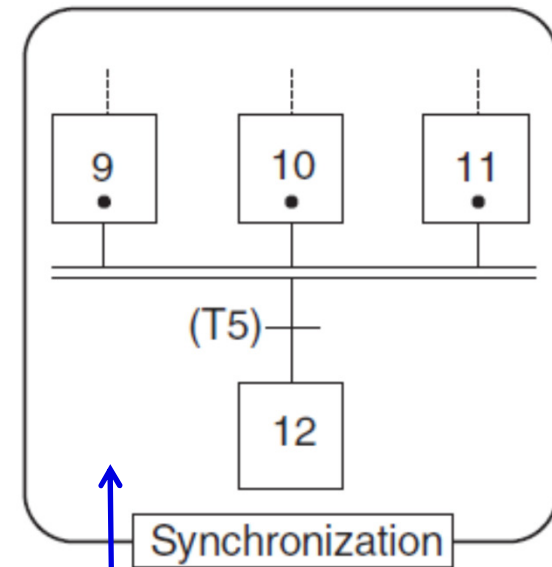
Seleccionar uma alternativa entre várias (OU).

nota: nada impede que todas/algumas possam disparar simultaneamente



Activar várias etapas simultaneamente

quando T4 dispara, as etapas 6, 7 e 8 são activadas simultaneamente (e etapa 5 é desactivada)



O sistema só evolui quando várias condições estão satisfeitas simultaneamente

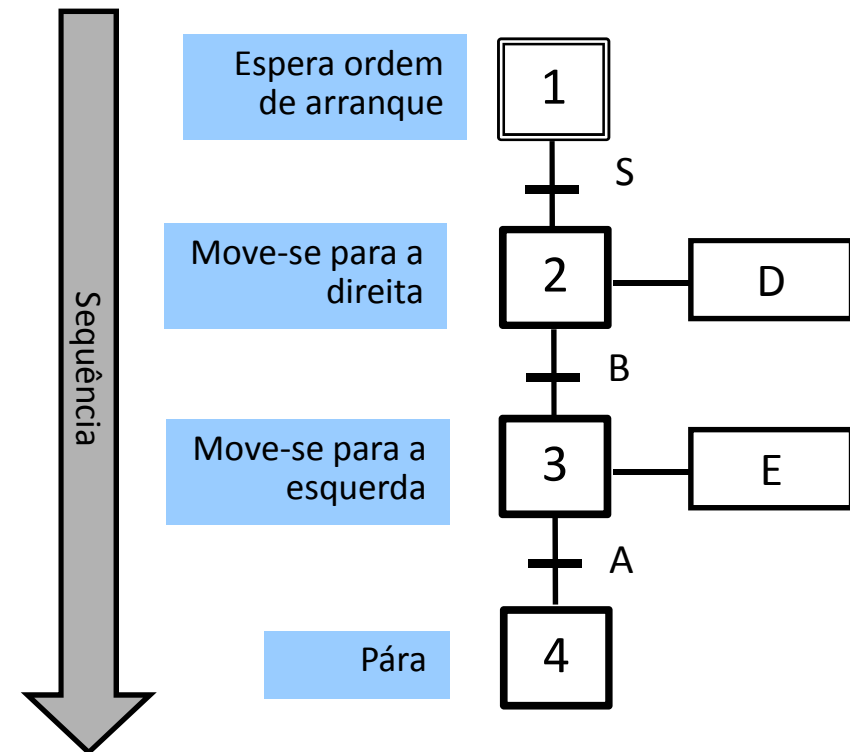
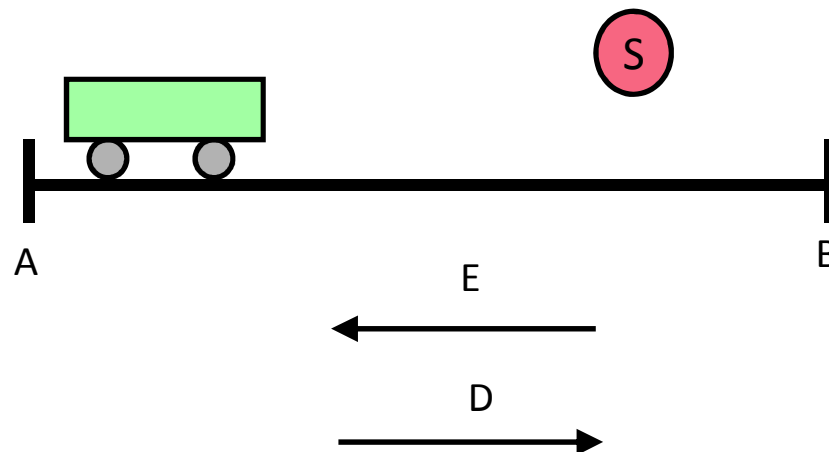
T5 só pode disparar quando as etapas 9, 10 e 11 estão activas. O disparo desactiva estas etapas e activa a etapa 12

Sequência

- A situação mais comum, é um comportamento sequencial:
 - I.e. executar um conjunto de acções de forma sequencial

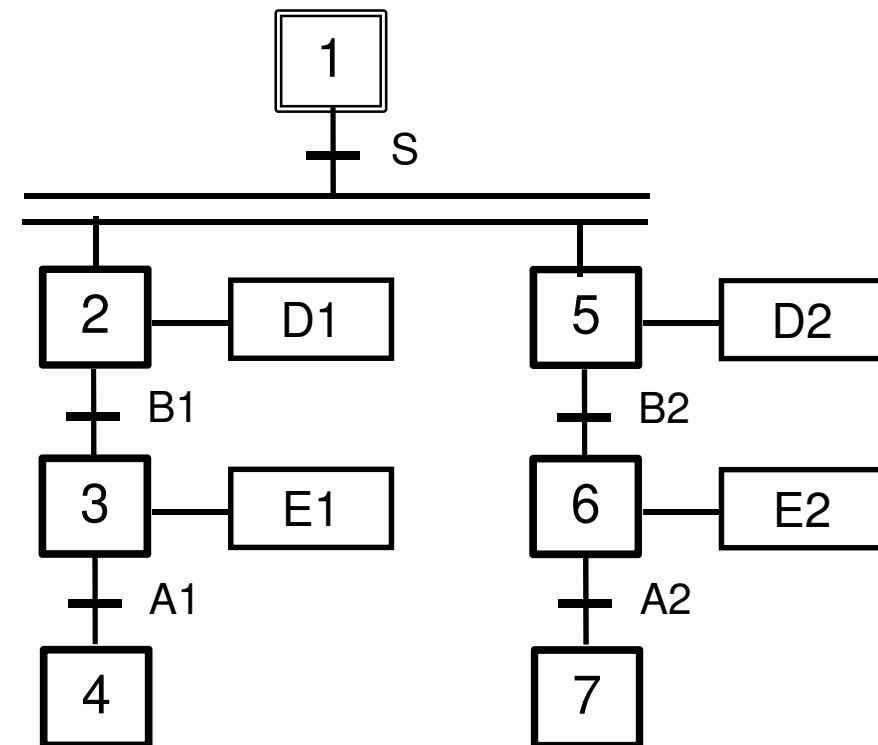
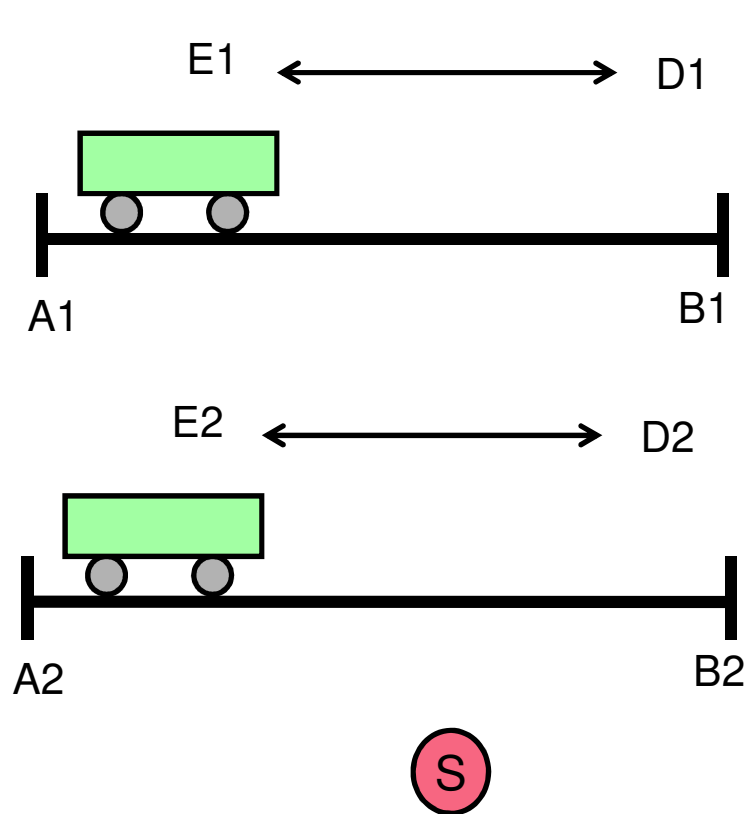
- Exemplo:

- Um veículo desloca-se num carril entre 2 posições (A e B) através dos comandos direita (D) e esquerda (E). O movimento inicia-se quando há uma ordem de arranque (S).



Concorrência

- Quando se pretende executar diversas acções em paralelo / concorrentemente / em simultâneo
- Exemplo: temos agora 2 veículos que queremos controlar em simultâneo.

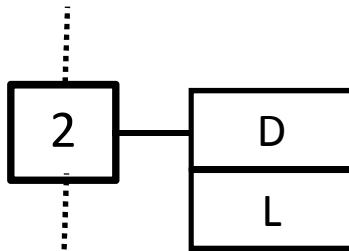


Concorrência (2)

— É importante distinguir entre...

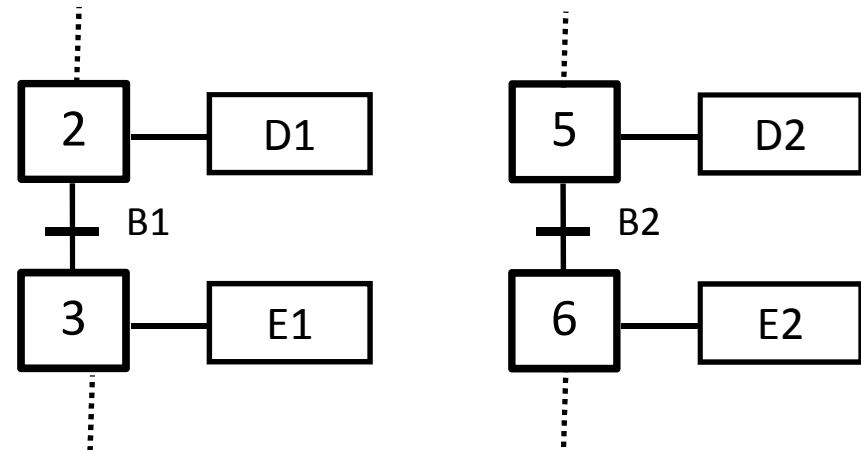
- Acções concorrentes

...quando o veículo está em movimento deve ser ligada uma lâmpada (L)



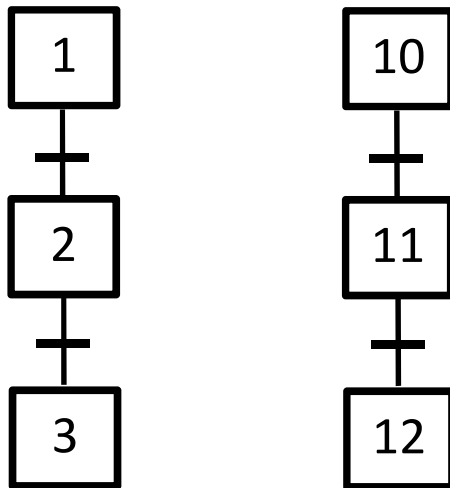
- Sequências de etapas concorrentes

...os veículos movimenta-se em simultâneo

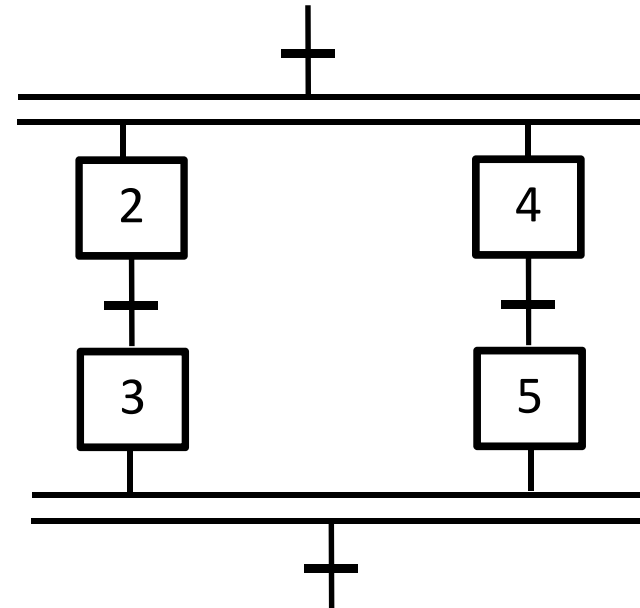


Concorrência (3)

— Formas de modelar etapas concorrentes



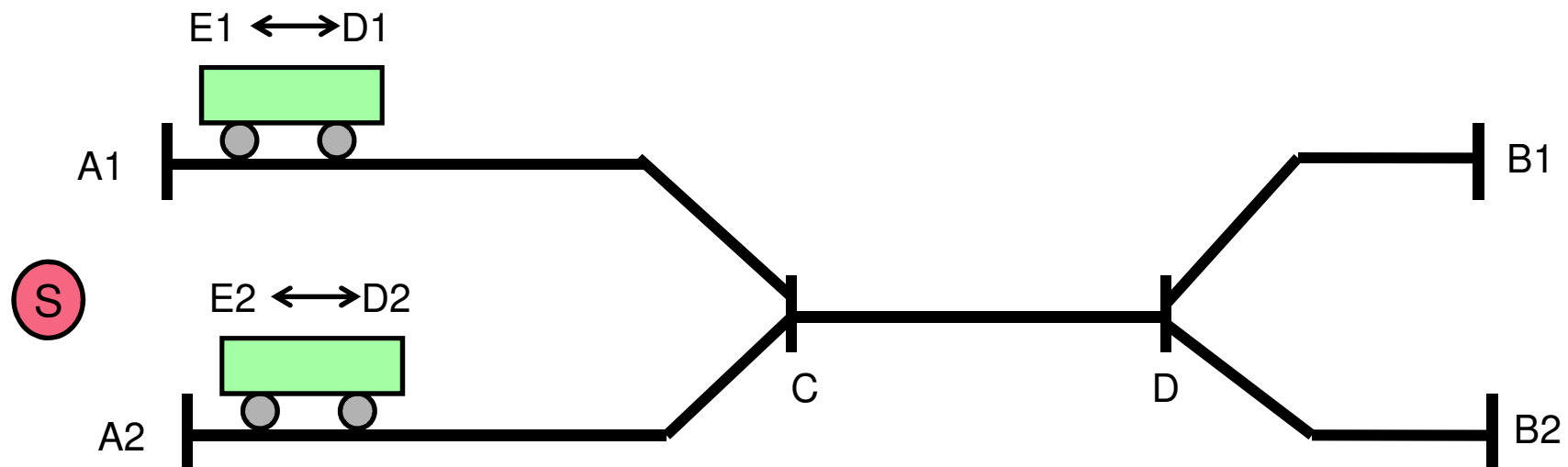
Grafcets paralelos



Paralelismo dentro do
mesmo Grafcet

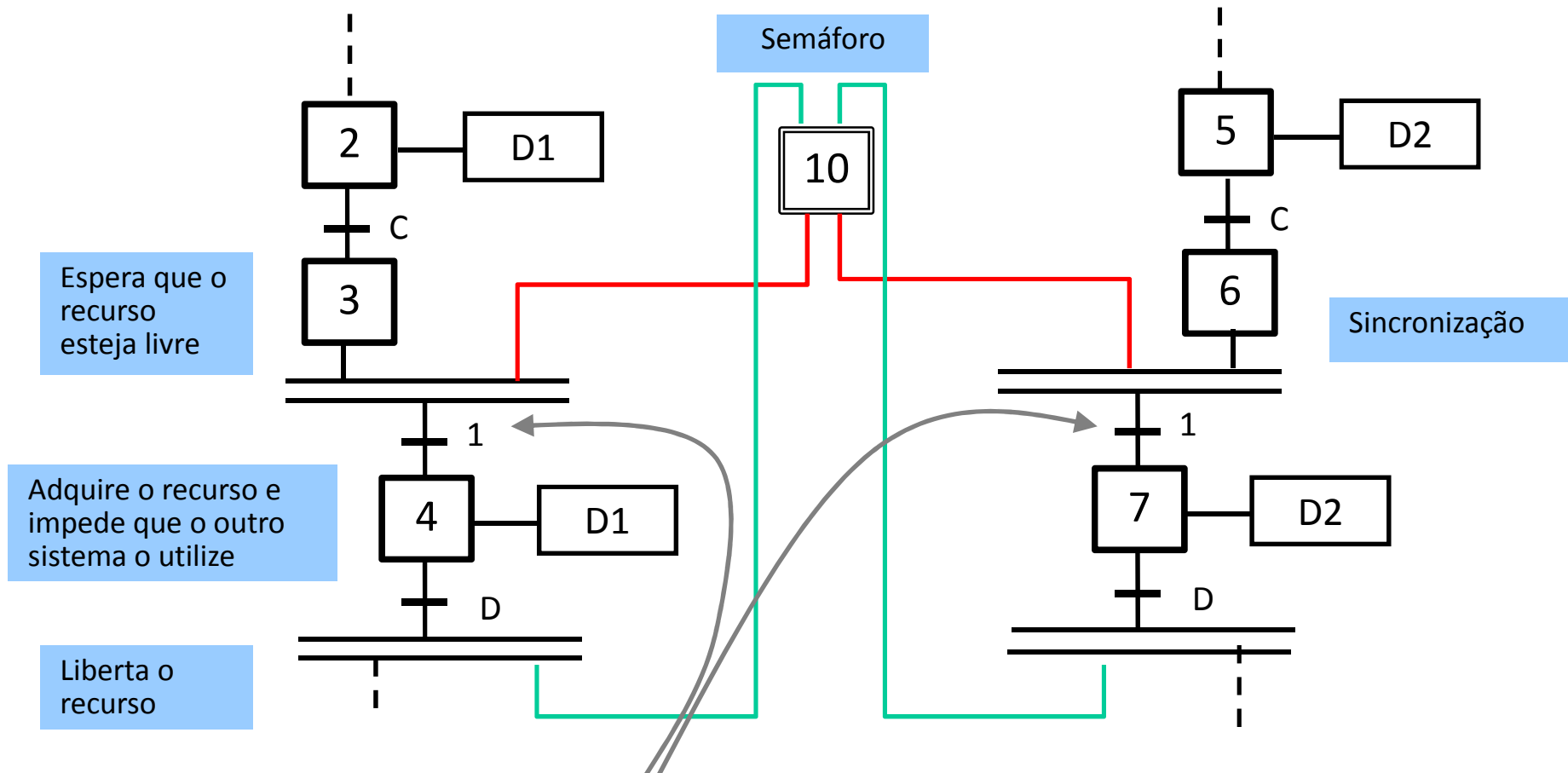
Partilha de recursos/sincronização

- Por vezes é necessário partilhar recursos
 - I.e. um elemento que é partilhado entre vários sistemas, e só pode ser utilizado por um sistema de cada vez (acesso mutuamente exclusivo)
- e também sincronizar o acesso aos mesmos
 - Exemplo: os dois veículos partilham um troço comum (C-D). Se o troço estiver ocupado, os veículos devem esperar. (assume-se que quando se envia um comando de movimento, este acciona também as 'agulhas' que ligam os carris respectivos)



Modelação da partilha de recursos/sincronização

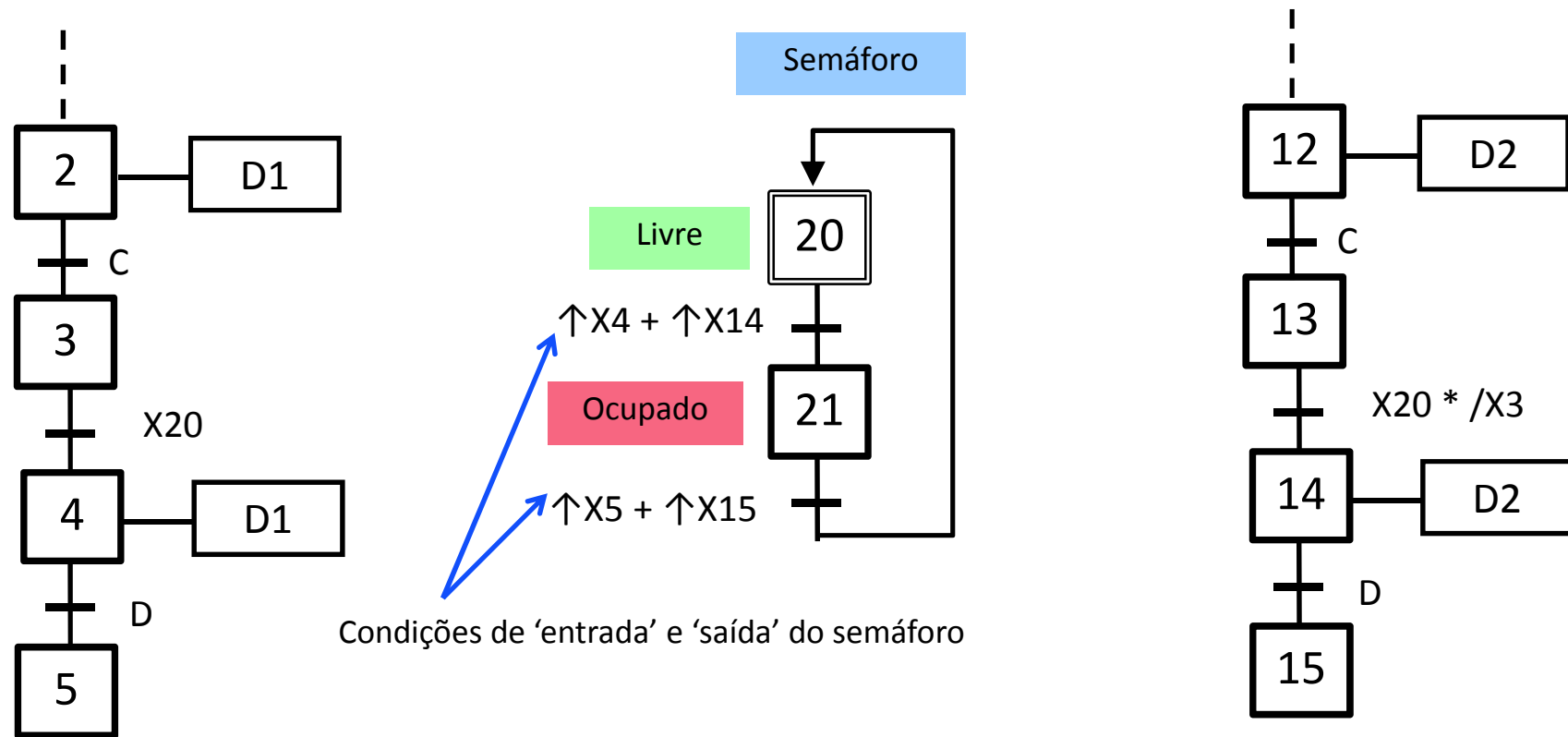
- Semáforo na estrutura do Grafcet
 - Uma etapa indica se o recurso está livre (etapa activada) ou não (desactivada)
 - Simples de implementar e de validar.



Atenção à possibilidade de ambos utilizarem o recurso ao mesmo tempo...(erro)

Modelação da partilha de recursos/sincronização (2)

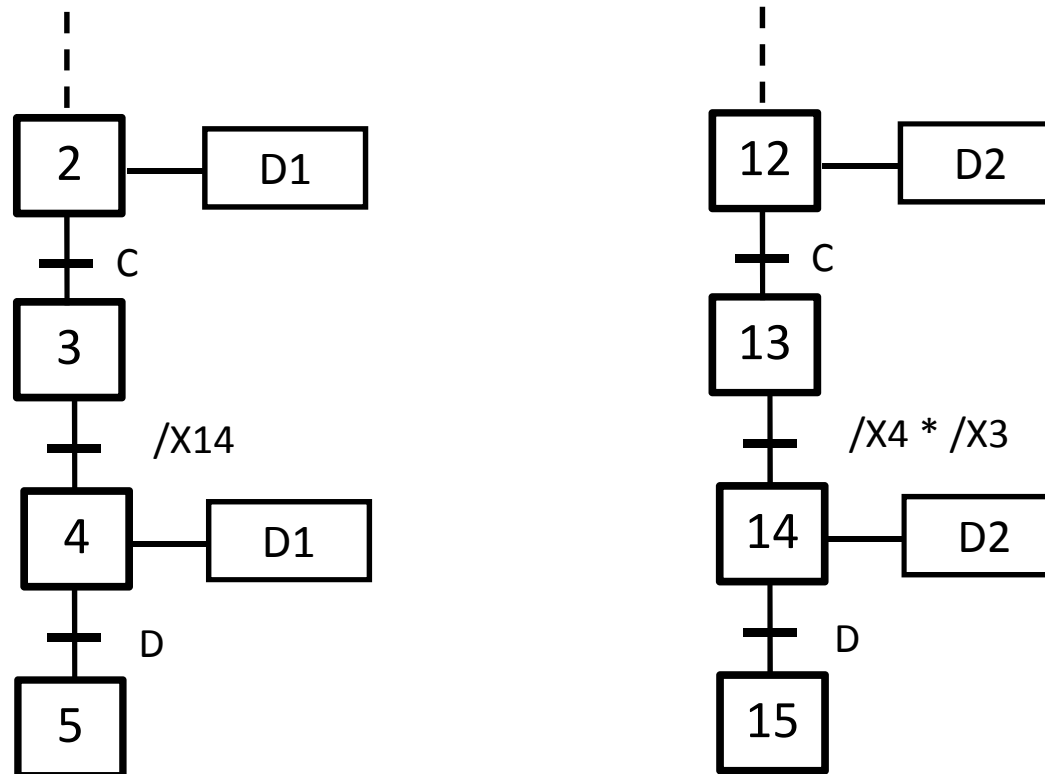
- Semáforo baseado na interpretação interna
 - Utilizar uma etapa para indicar se o recurso está livre, e variáveis internas (X_i) para sincronizar o acesso ao mesmo
 - É necessário ter cuidado em definir as condições do semáforo



Impede que ambos utilizem o recurso ao mesmo tempo...

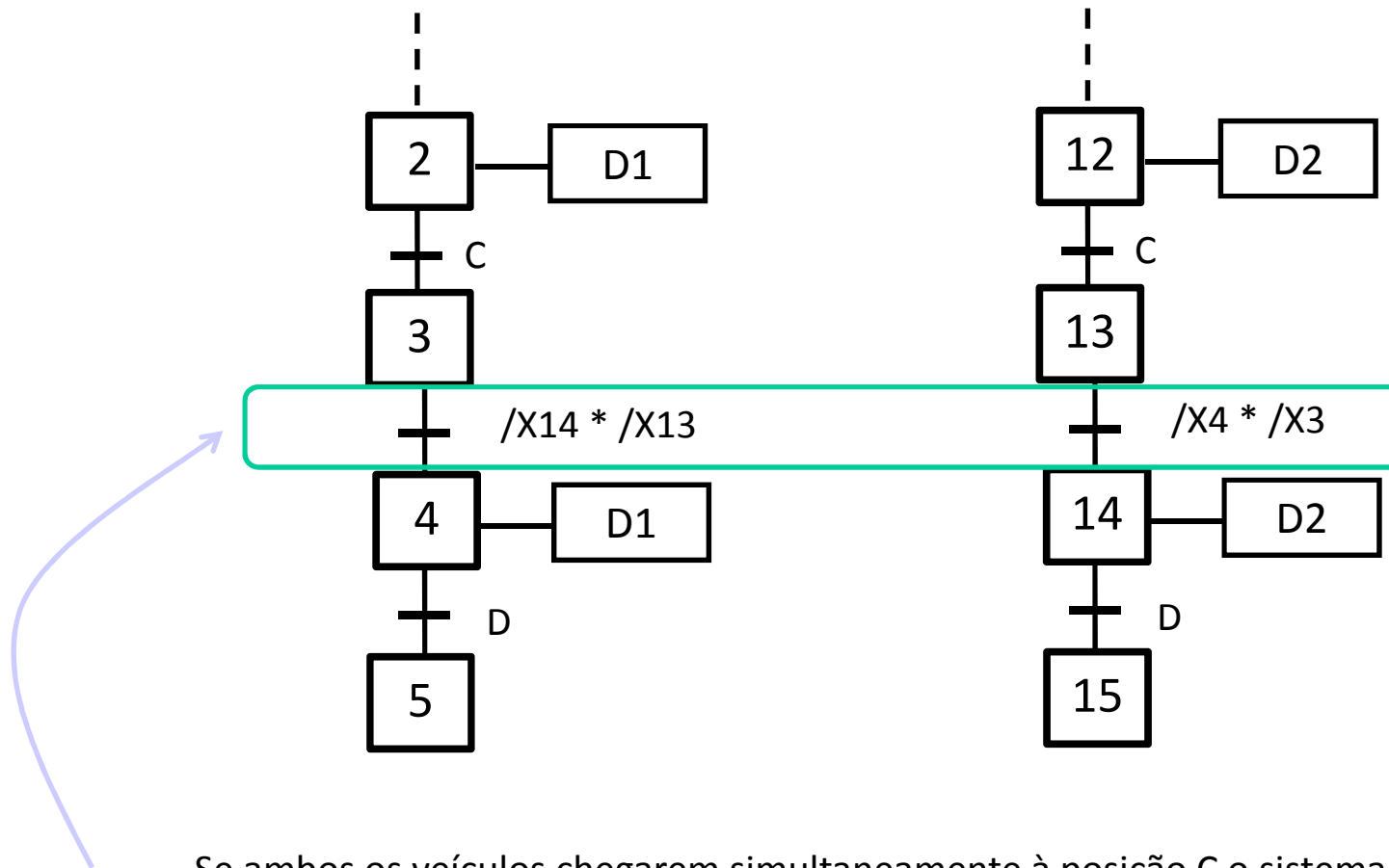
Modelação da partilha de recursos/sincronização (3)

- Semáforo distribuído:
 - Utilizar variáveis internas (X_i) para sincronizar o acesso ao recurso partilhado
 - Mais complexo de utilizar (e validar), especialmente quando o 'recurso' implica a utilização de várias etapas:



Bloqueio mútuo

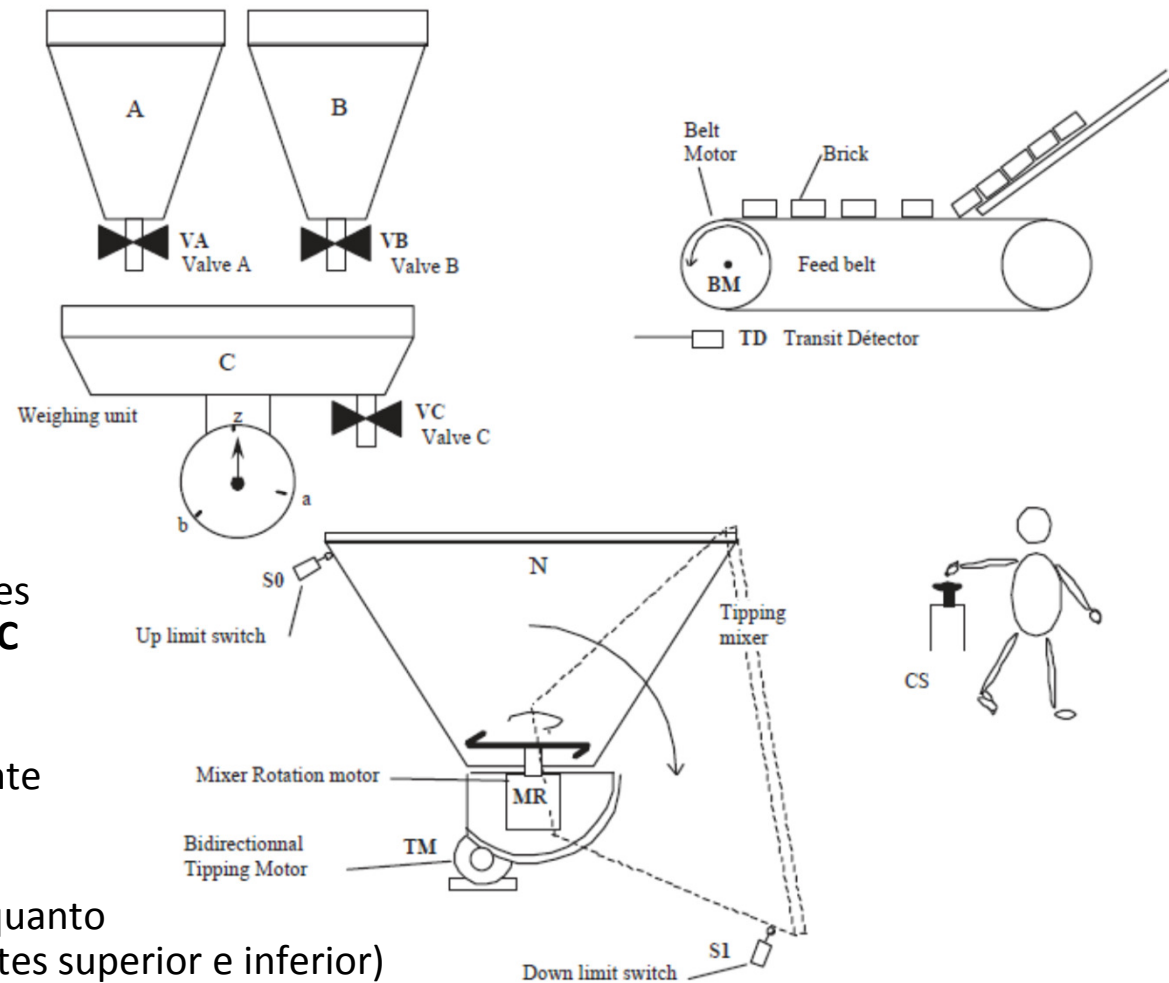
- Quando se utilizam recursos partilhados existe o perigo de bloqueio mútuo (denominado **deadlock**)
- Isto acontece devido a erros na modelação

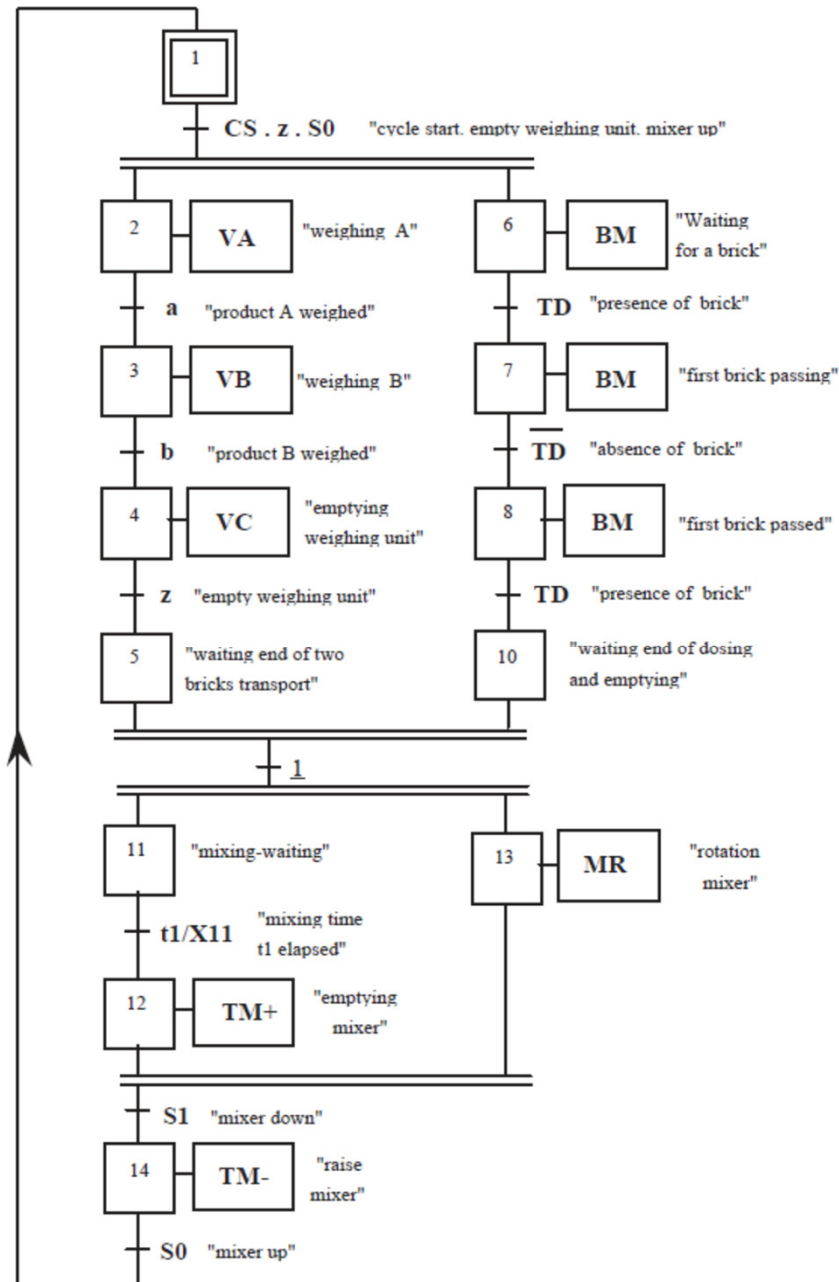


Se ambos os veículos chegarem simultaneamente à posição C o sistema bloqueia !

Exemplo: Misturador

- Mistura (em **N**) de 3 produtos: A e B, previamente pesados em **C**, juntamente com blocos que chegam ao sistema através de um tapete de transporte (**BM**).
- A atuação no botão **CS** inicia o ciclo de mistura:
 1. O produto **A** é pesado (em **C**) até o sensor **a** ser ativado
 2. O produto **B** é pesado/adicionado (em **C**) até o sensor **b** ser ativado
 3. Os produtos são descarregados no misturador (válvulas **VA**, **VB** e **VC**)
 4. Em paralelo com as ações anteriores dois blocos são descarregados em **C** (deteção em **TD**)
 5. O misturador (**MR**) é ativado durante t_1 segundos
 6. O misturador continua ativado enquanto é descarregado (**TM**) (**S0** e **S1**, limites superior e inferior)





Solução que utiliza apenas ações contínuas

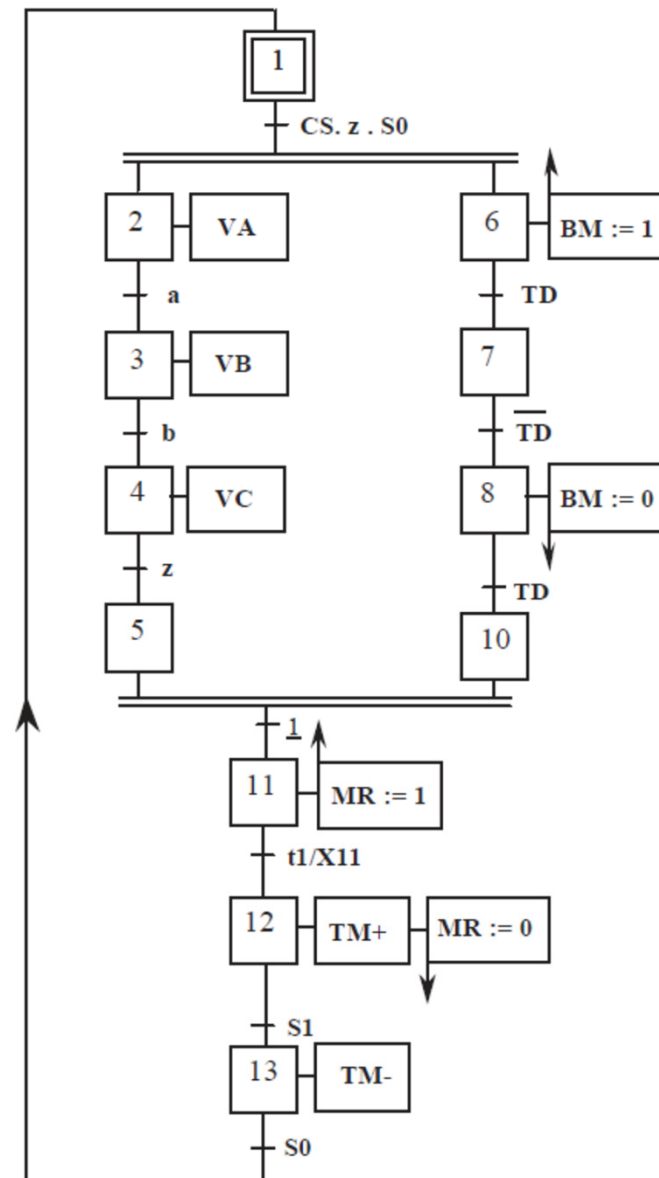
Inputs

CS	Cycle.Start
TD	Transit Detector
a	Fluid weigh A reached
b	Fluid weigh A + B reached
z	Empty weighing unit
S0	mixer up
S1	mixer down

Outputs

BM	Belt Motor
MR	Mixer Rotation Motor
TM+	Tipping Motor (down)
TM-	Tipping Motor (up)
VA	opening Valve A
VB	opening Valve B
VC	opening Valve C

Solução que utiliza ações contínuas e impulsionais



Sumário

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

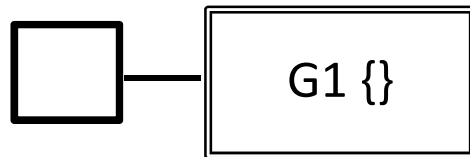
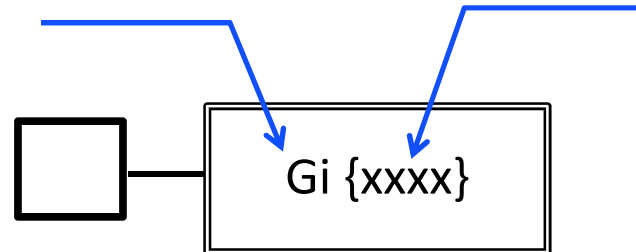
Hierarquia

- Em muitas situações é necessário que um Grafcet ‘comande’ a evolução de outro Grafcet.
 - Neste caso diz-se que há uma hierarquia entre os Grafcet.
- Casos típicos:
 - Obrigar um Grafcet a evoluir para um estado específico;
 - Impedir que um Grafcet evolua;
 - Desactivar todas as acções que um Grafcet está a executar.
- O suporte deste tipo de comportamentos é realizado por intermédio de **macro-acções** (não confundir com macro-etapas...ver adiante):
 - Uma macro-acção é tem as características de uma acção normal (i.e. contínua ou impulsional)
 - A execução das macro-acções têm prioridade sobre as regras de evolução do Grafcet, i.e. Forçam a evolução dos Grafcet controlados.

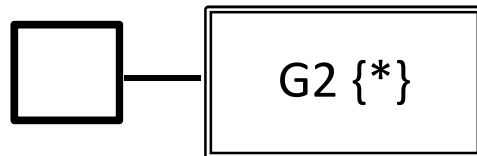
Sintaxe das macro-acções

Nome do Grafcet que se pretende comandar/controlar

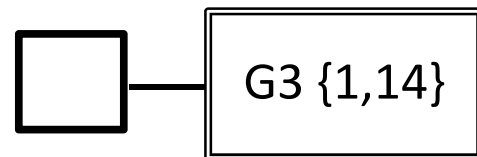
Tipo de macro-acção que se pretende executar



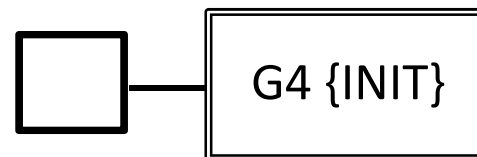
Desactiva todas as etapas de G1



'Congela' G2, i.e. as etapas que estão activas no momento continuam activas



Força G3 a ter as etapas 1 e 14 activas



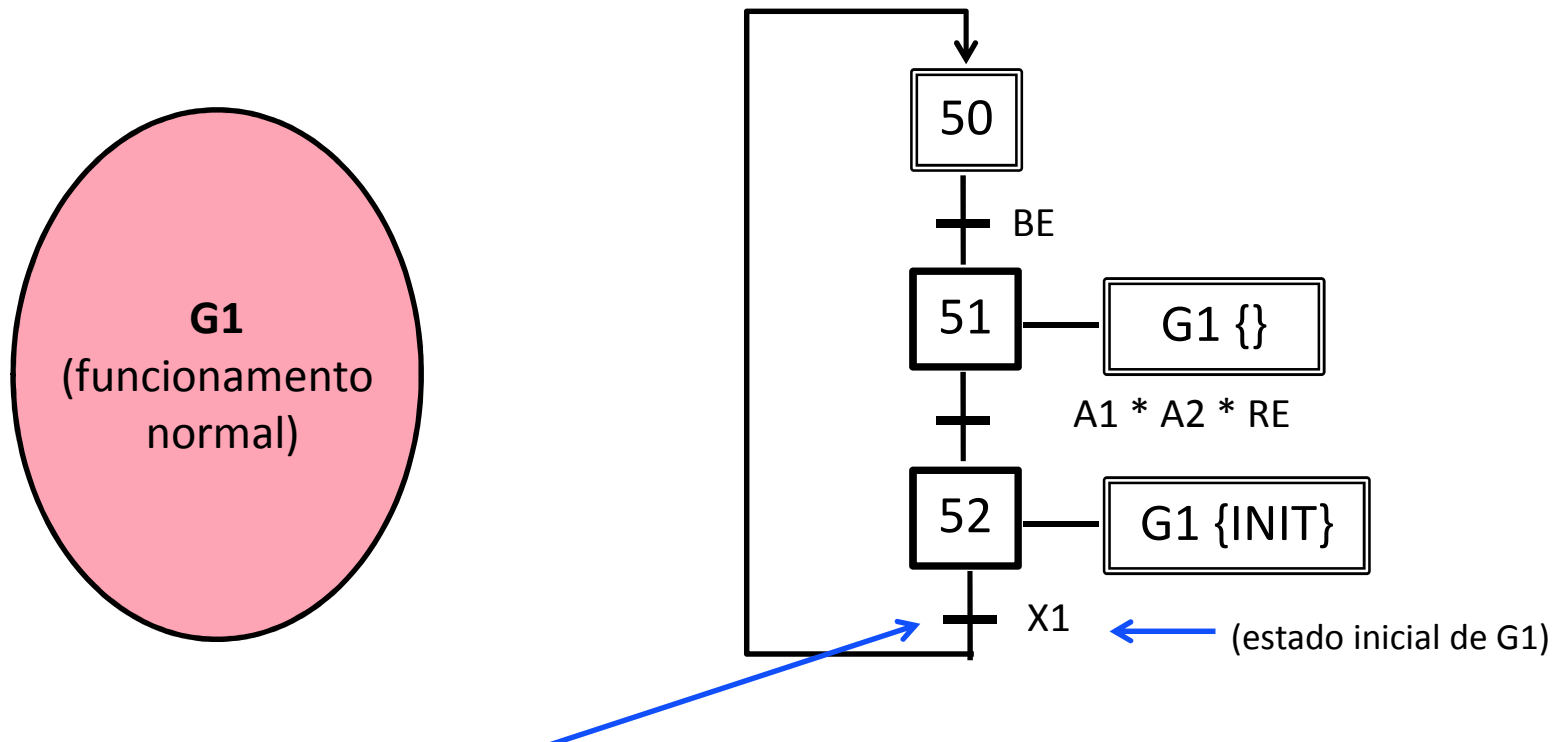
Força G4 para o estado inicial

Os Grafcet que são forçados não evoluem

Ou seja, as respectivas transições não disparam !

Macro-acções : exemplo

- Voltando ao exemplo do veículo:
 - ... se o botão de paragem de emergência (BE) for activado os veículos devem parar imediatamente. Após esta acção assume-se que os veículos foram movimentados manualmente para as posições iniciais (A1 e A2). O sistema pode retomar o funcionamento normal quando o botão de resume (RE) for activado.

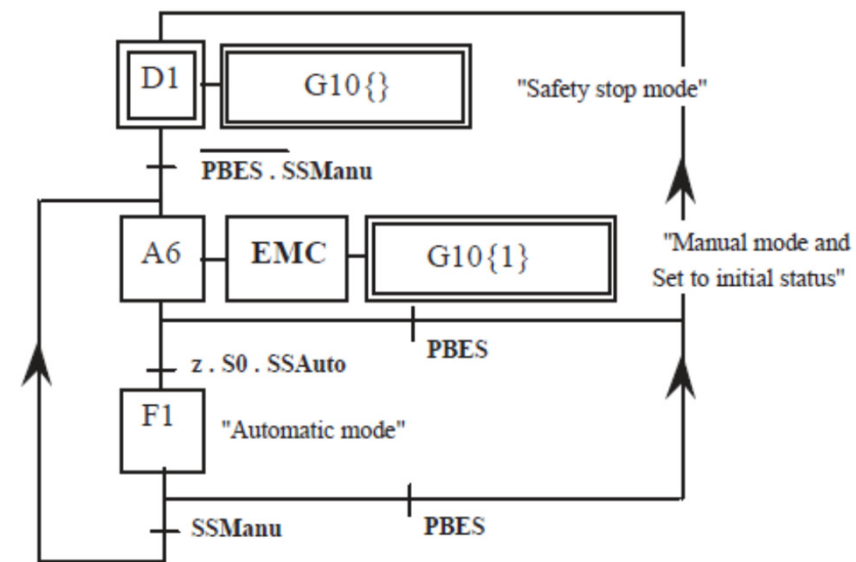
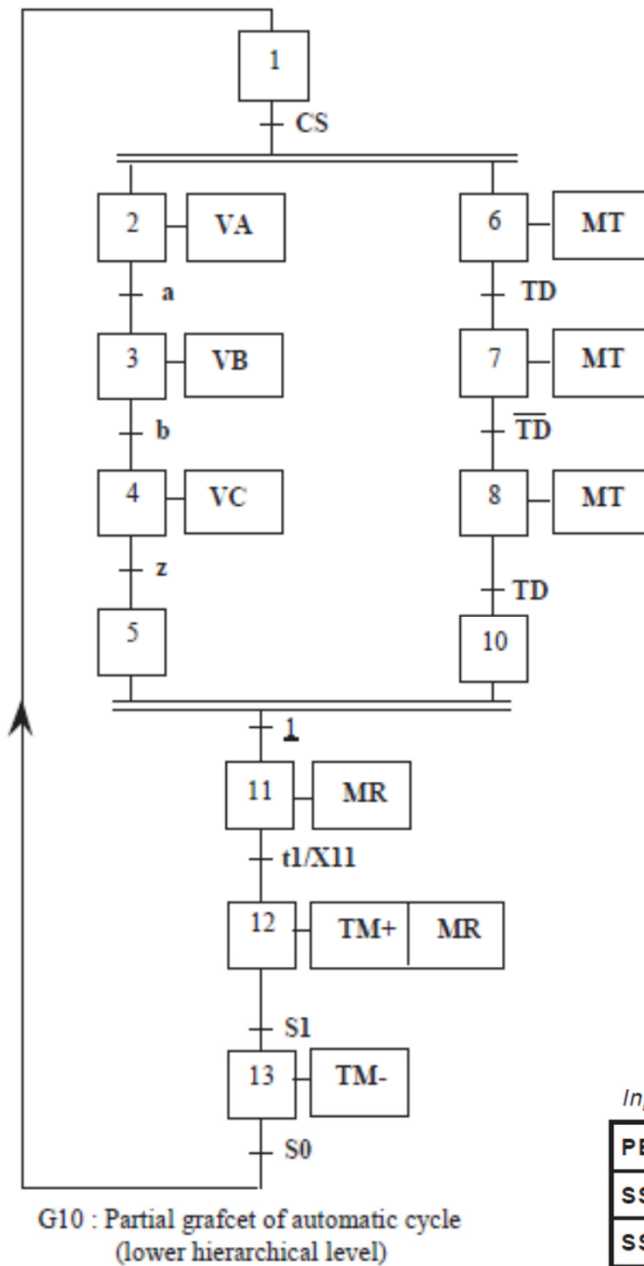


Necessário para garantir que a macro-acção deixa de ser realizada quando **G1** é inicializado (caso contrário **G1** ficaria sempre no estado inicial)

Solução do **Misturador** que utiliza macro-ações

Considerando

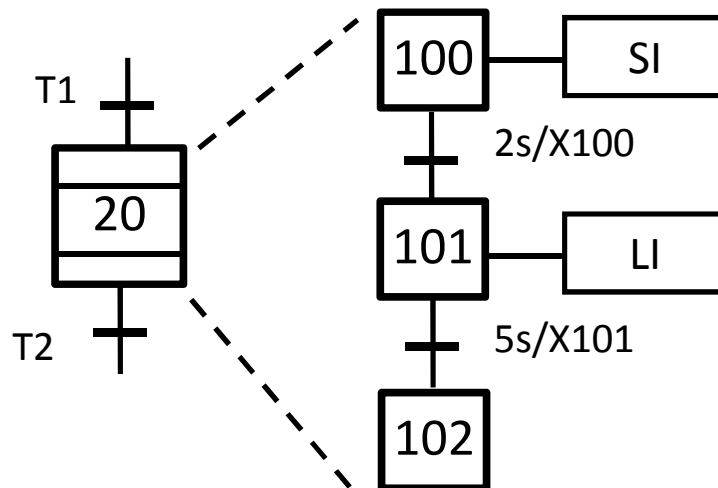
- A ativação do botão de emergência pára o sistema
- Um seletor que indica o modo de funcionamento (Manual / Automático)



Inputs		Outputs	
PBES	Push-Button Emergency Stop	EMC	Enabled Manual Controls
SSAuto	Selector-Switch on Auto mode		
SSManu	Selector-Switch on Manu mode		

Macro-etapas

- Existem situações em que é necessário executar repetidamente a mesma sequência de acções;
- Estas sequências podem ser representadas por macro-etapas (comportamento semelhante a uma subrotina);
- Exemplo: ...quando o veículo circula no troço partilhado (C-D), é activada uma sirene (SI) durante 2s, seguida de uma luz intermitente (LI) durante 5s



Quando a etapa 20 é activada, o Grafcet 'salta' para a etapa 100 e executa a sequência 100-102

Só quando a etapa 102 ficar activa é que a transição T2 pode disparar

É preciso ter atenção para que a mesma macro-etapa não seja invocada em simultâneo no mesmo Grafcet – deve ser encarada como um recurso partilhado

Solução do **Misturador** que utiliza macro-etapas:

- Grafcet principal com a descrição global
- Uma macro-etapa modela o funcionamento de cada subsistema

