NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Departamento de Engenharia Electrotécnica e de Computadores

**Secção de Robótica e Manufactura Integrada**

| Description of Lab Work nº 3 | |
|---|---|
| Course | **Real-Time systems / *Sistemas de Tempo Real*** |
| Year | 2024/2025 |
| Aim | Development of real-time applications using graphical languages |
| Classes | 2 classes x 3 hours + 6 extra hours (outside classes) |
| Delivery Date | **10/12/2024** |

Concrete Objectives:

1. Utilization of graphical languages to model real-time systems
    a. Modeling of concurrent/simultaneous behaviors
    b. Modeling with Petri Nets

2. Application of PN modeling in a real problem.

3. Implement the real problem requirements.

4. Fill in a questionnaire and submit the work (according to the procedures) in Moodle.

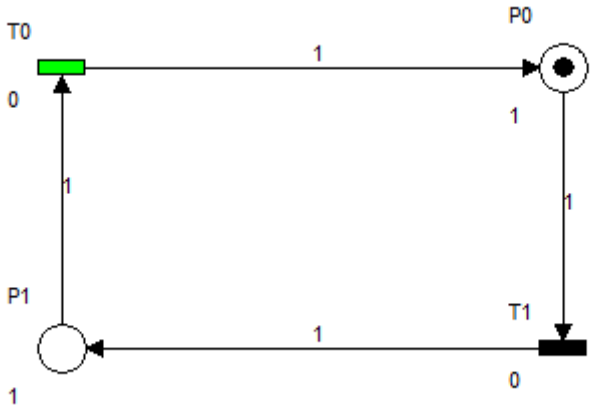
**Alunos**:

Rafael Silva 62966

Rita Ribeiro 63147

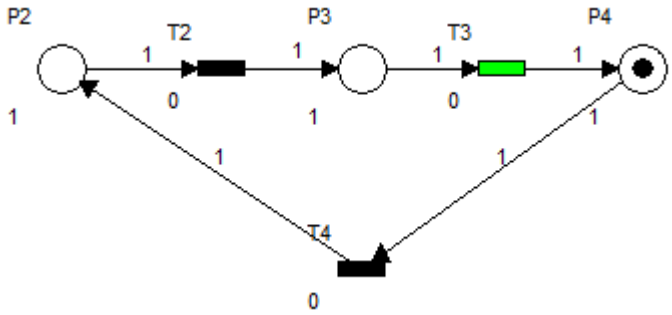
Tomás Pratas 63372

# R1 - Model and simulate a secure PN




Uma Rede de Petri Limitada (bounded PN) é aquela em que o número máximo de tokens permitidos em qualquer lugar (place) da rede é limitado a um valor específico. Quando este limite é igual a 1 (k = 1), a rede é designada como Rede de Petri Segura (Safe PN) ou 1-bound. Isto significa que cada place da rede pode conter, no máximo, apenas um token de cada vez. Esta propriedade de limitação específica caracteriza as Redes de Petri Seguras como um caso particular das Redes de Petri Limitadas.

# R2 - Model and simulate a conservative PN



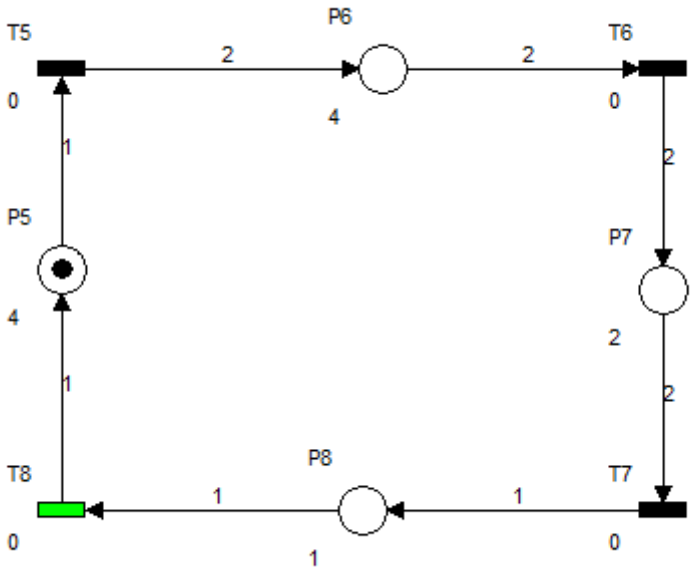

Uma Rede de Petri (PN) é considerada estritamente conservativa quando o número total de tokens na rede se mantém constante ao longo da sua execução. Nesta situação, não existe qualquer perda ou ganho de tokens, uma vez que estes são apenas transferidos de um lugar (place) para outro através da ativação das transições.
Assim, para cada token que é removido de um place, outro é adicionado a um place diferente, garantindo a conservação total dos tokens na rede.

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Departamento de Engenharia Electrotécnica e de Computadores

Secção de Robótica e Manufactura Integrada

# R3 - Model and simulate a PN with capacity places and weighted arcs




Uma Rede de Petri (PN) que inclui lugares (places) com capacidade específica n e arcos ponderados com peso m possui características particulares relacionadas à gestão de tokens na rede.

Os lugares com capacidade n podem armazenar, no máximo, n tokens em simultâneo. Por outro lado, um arco com peso m determina que, durante a ativação de uma transição, são consumidos exatamente m tokens.

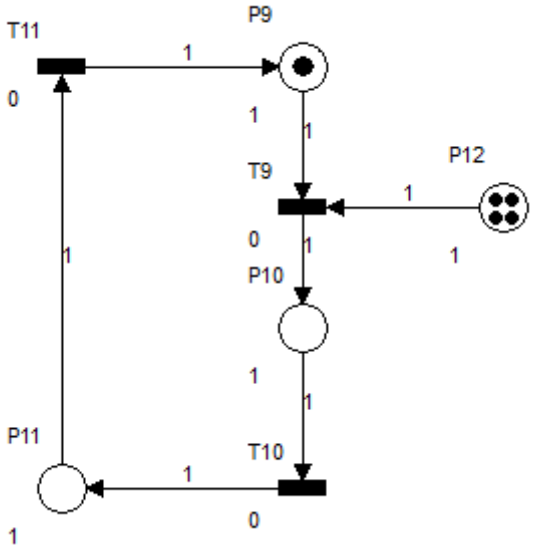
Assim, a capacidade dos lugares e o peso dos arcos influenciam diretamente a dinâmica de armazenamento e transferência de tokens na rede.

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

Departamento de Engenharia Electrotécnica e de Computadores

Secção de Robótica e Manufactura Integrada

# R4 - Model and simulate a PN which starts working and enters in a deadlock state after a few iterations




O exemplo apresentado ilustra uma Rede de Petri (PN) que, após iniciar a sua execução, acaba por entrar num estado de impasse, conhecido como "deadlock".

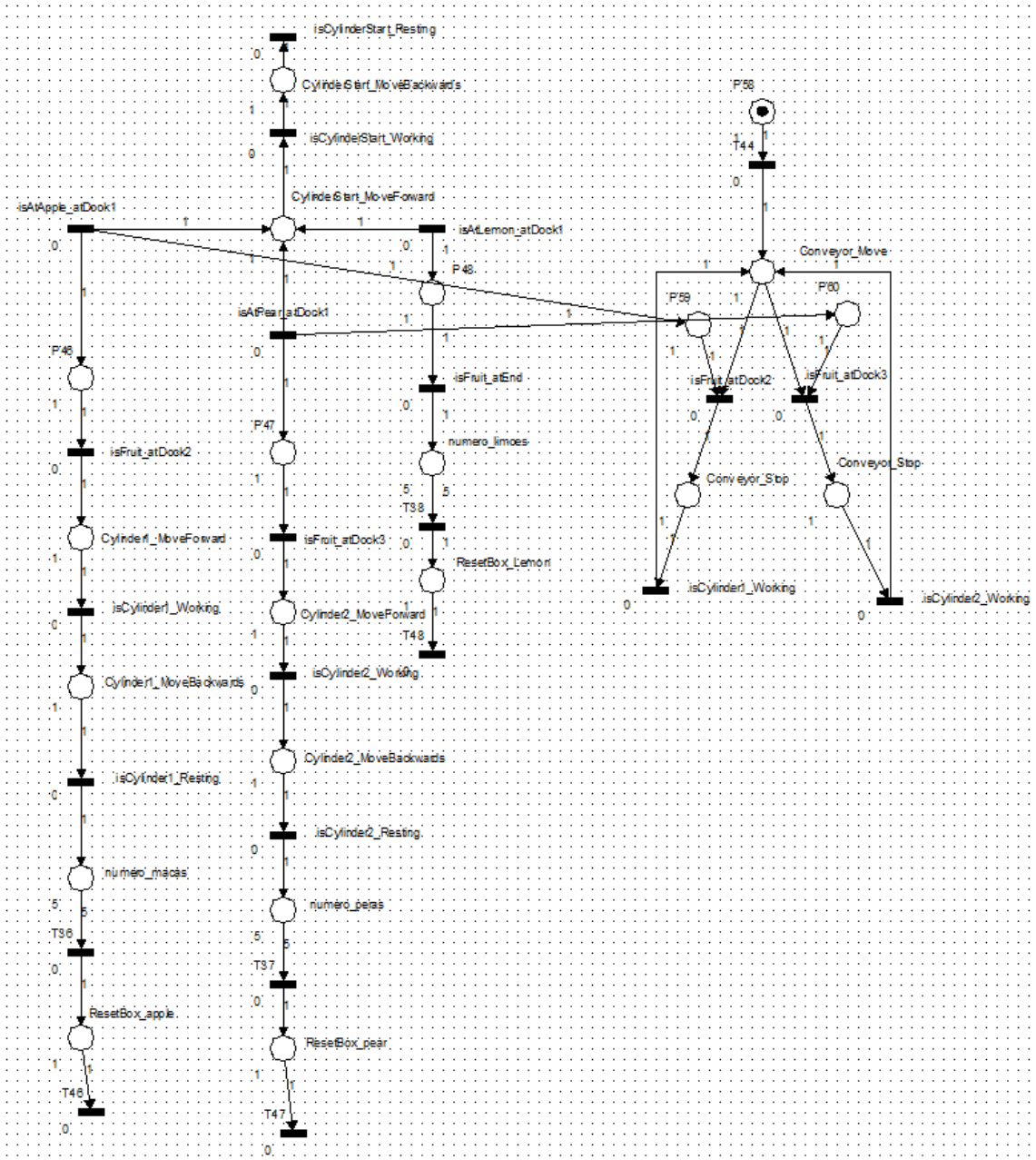
Neste cenário, verifica-se que a rede atinge o impasse após algumas iterações porque o Place P12 esgota o numero de tokens disponiveis. Como resultado, a rede fica impossibilitada de progredir, uma vez que não existem tokens disponíveis, levando à paralisação completa do sistema.

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

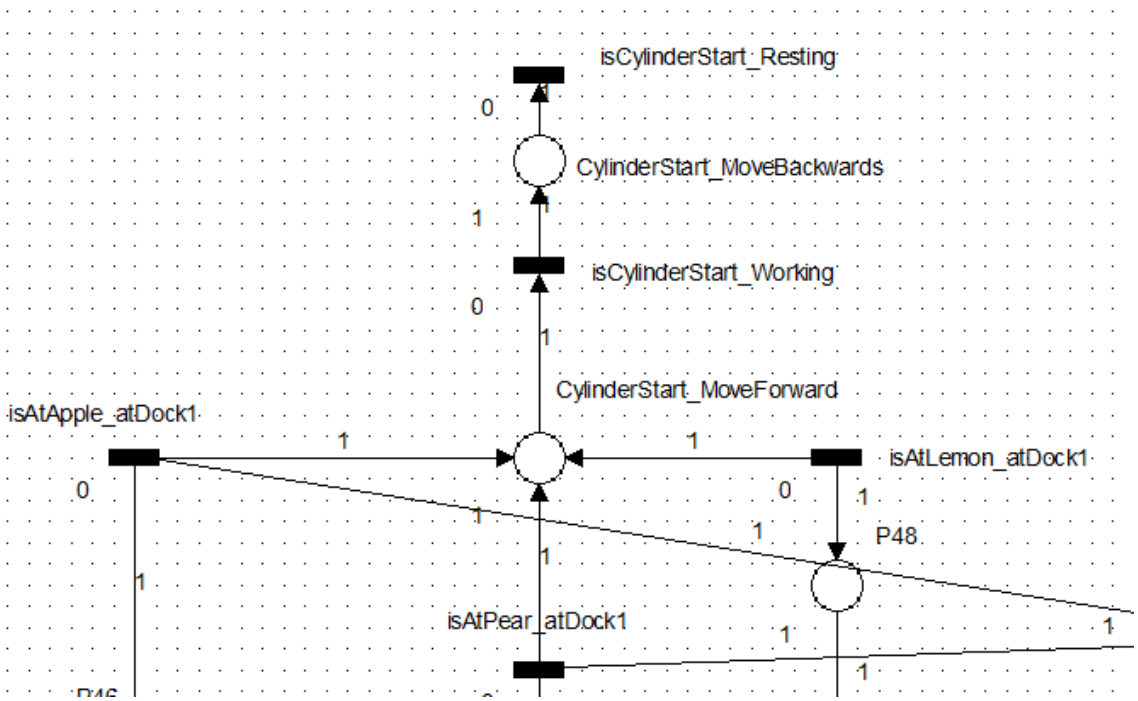
Departamento de Engenharia Electrotécnica e de Computadores

Secção de Robótica e Manufactura Integrada

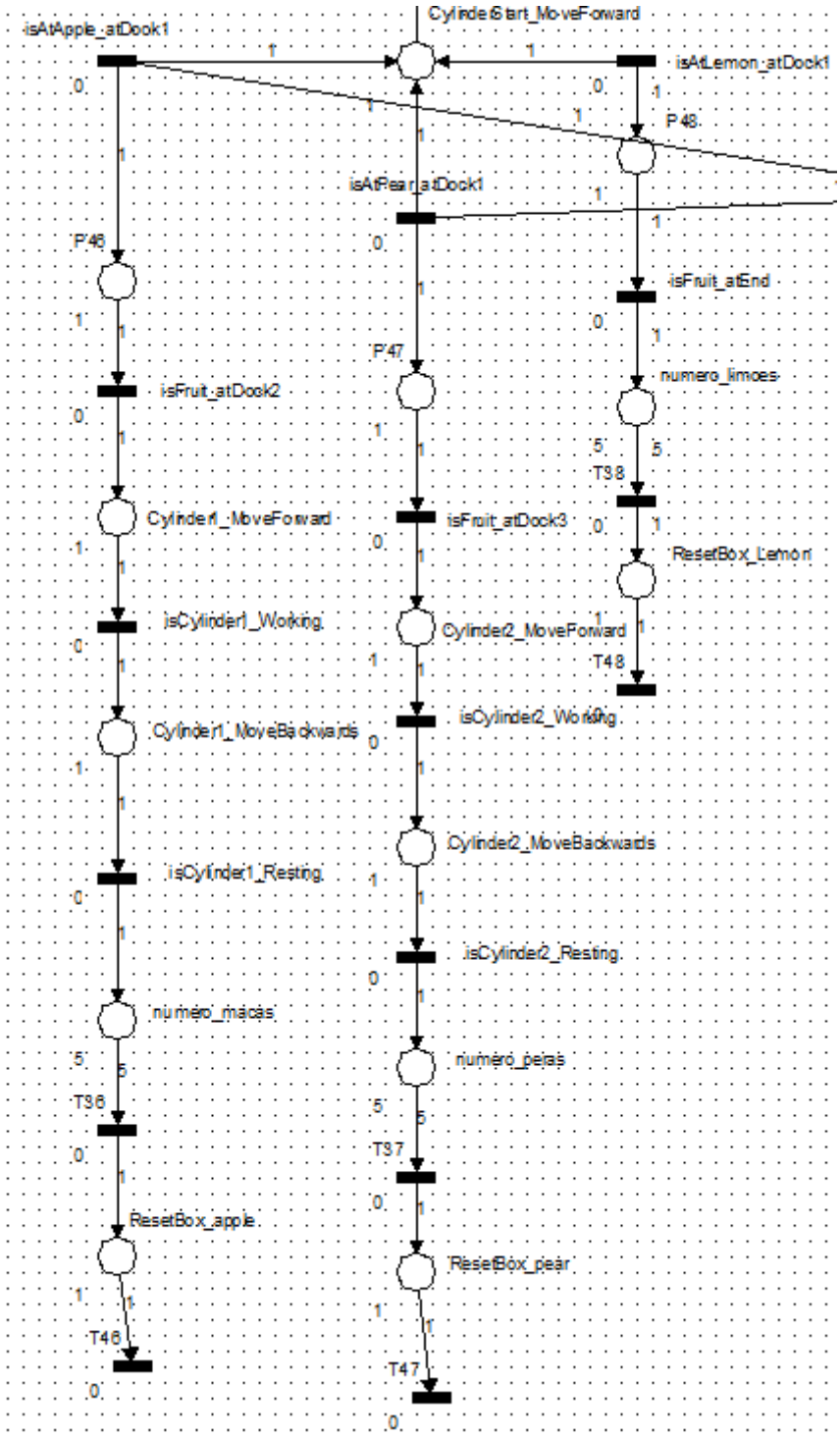
# PN to control the Fruit Splitter



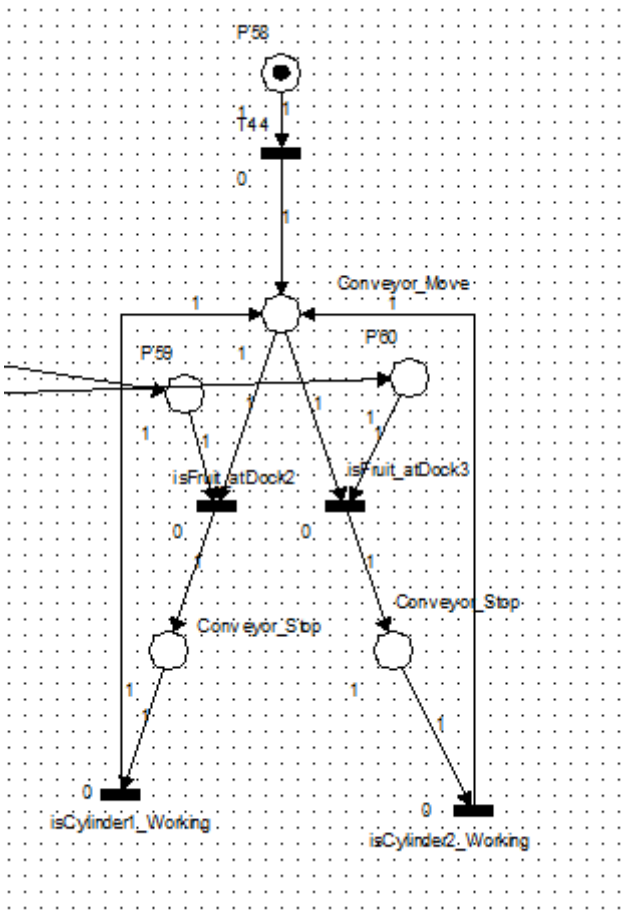
O cylinder_Start começa cada vez que uma fruta é detetada na Dock1.



O processamento de cada peça de fruta é realizado por um dos ramos onde esta é detetada na Dock 1, criando um semáforo respetivo (P46, P47 e P48). Estes semáforos servem para que a Dock 2, Dock 3 e Dock End saibam o tipo de fruta. Cada cilindro é também controlado aqui, juntamente com os contadores de fruta e de caixas cheias, sendo que, neste caso, definimos 5 como o número de peças necessário para encher uma caixa.



Por fim, temos o controlo do Conveyor, que é iniciado sempre que uma peça de fruta é detetada na Dock 1 e é parado quando a peça de fruta correspondente à Dock 2 e Dock 3 é detetada, permitindo assim o processamento correto da fruta.




O nosso programa permite também mostrar as estatísticas pressionando ‘x’ ou ‘y’, mas também realizar o Emergency Break, pressionando ‘e’, onde todas as tarefas ficam bloqueadas até que o utilizador as retome, pressionando ‘r’.