

Contexto

Em Análise de Confiabilidade, presente, por exemplo, no livro “Probabilidade e Estatística para Ciências Exatas” do William Navidi ([Minha Biblioteca: Probabilidade e Estatística para Ciências Exatas](#)), capítulo 3, seção 3.2 e nos exercícios 10 e 11 da Lista 1, temos sistemas em série, em paralelo e misto.

Esse projeto visa gerar um solucionador de sistemas mistos (que já inclui os puramente em série e puramente em paralelo).

Para criar o solucionador, será necessário representar os sistemas com uma string.

Uma possibilidade é indicar os componentes em série com o “e comercial”, usando o símbolo & (por exemplo, A&B) e os em paralelo com um pipe, usando o símbolo | (por exemplo, A|B).

Para saber onde está cada componente, em série ou em paralelo, pode-se, por exemplo, separar por números.

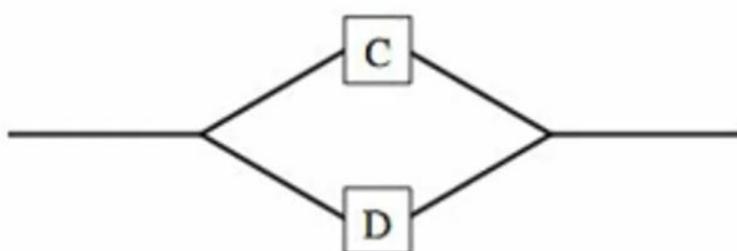
O sistema em série abaixo seria representado por:

sistema = “1A&B1”



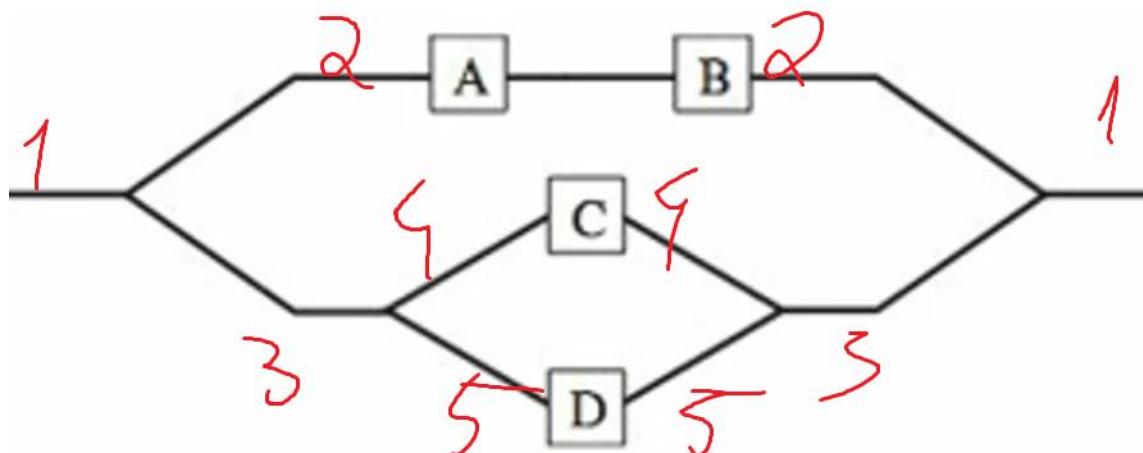
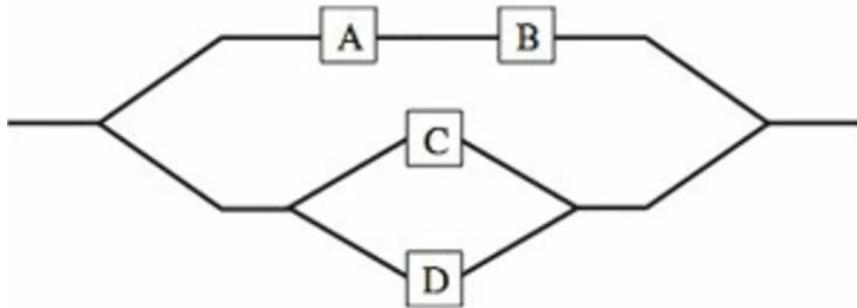
O sistema em paralelo a abaixo seria representado por:

sistema = “1 2C2 | 3D3 1”



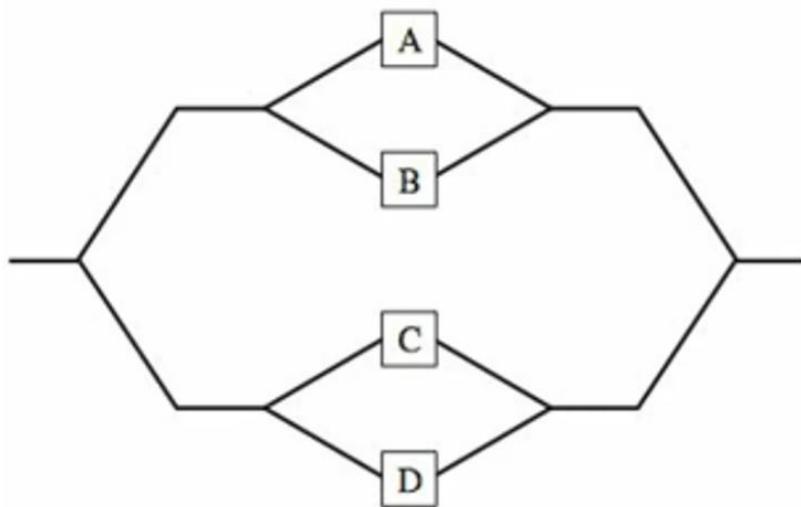
O sistema misto abaixo seria representado por:

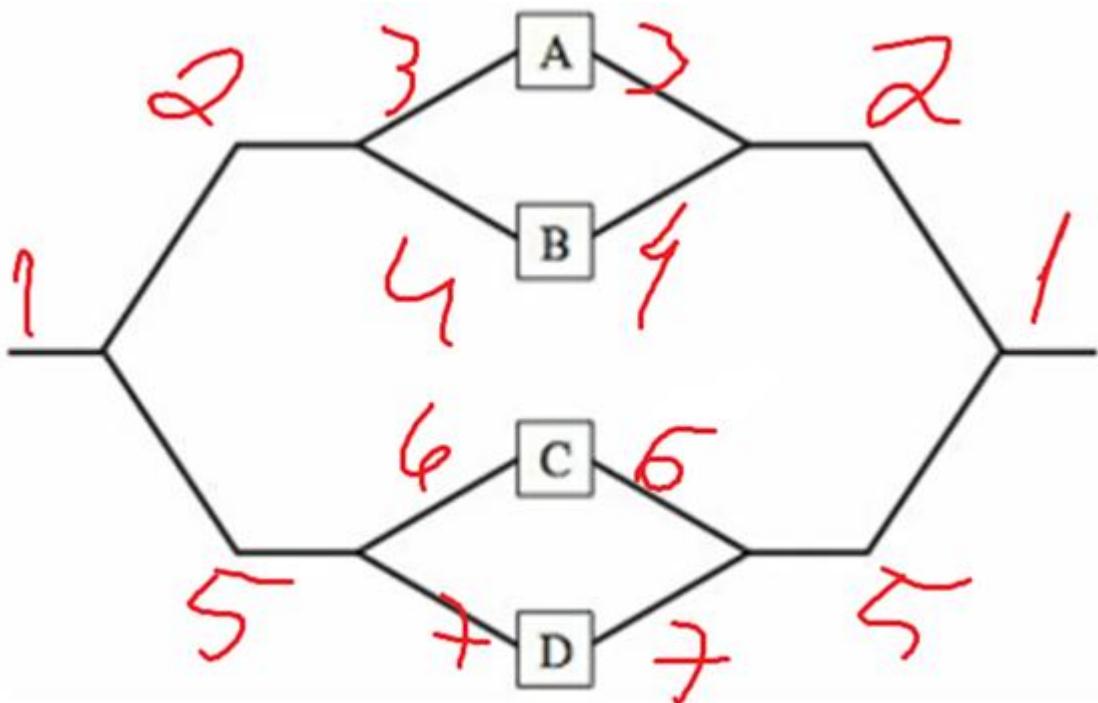
sistema = “1 2A&B2 | 34C4|5D53 1”



O sistema em paralelo, com subsistemas em paralelo, seria representado por:

sistema = “1 2 3A3|4B4 2 | 5 6C6|7D751”



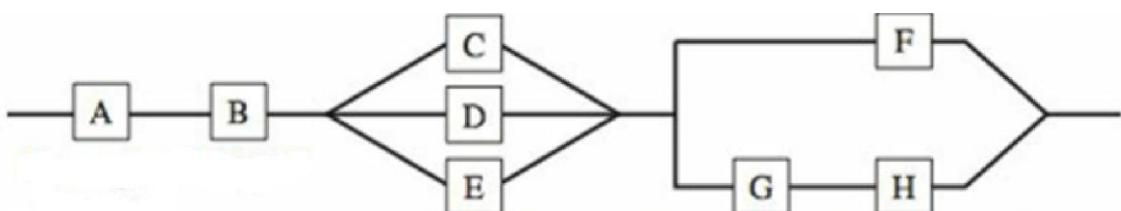


Cada componente tem uma probabilidade que será representada por um dicionário. Por exemplo:

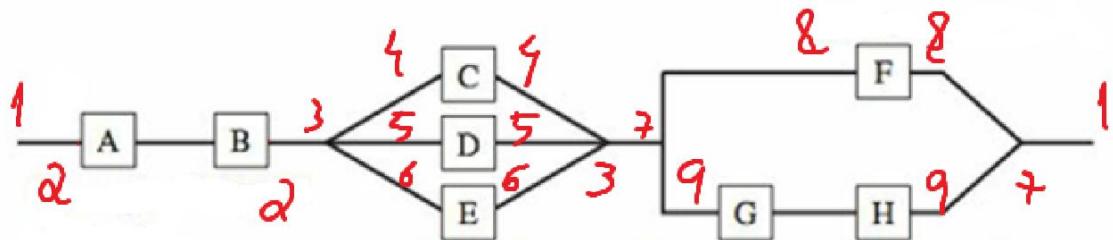
```
prob = {
    "A": 0.98,
    "B": 0.97
}
```

Por último, o sistema misto abaixo seria por:

sistema = “1 2A &B2 & 3 4C4|5C5|6C5 3 & 7 8F8|9G&H9 7 1”



sistema = “1 2A &B2 & 3 4C4|5C5|6C5 3 & 7 8F8|9G&H9 7 1”



Observe que alguns sistemas acima poderiam ser modelados de outras maneiras, com números diferentes. Sugiro que testem com os sistemas que foram passados. Observe, também, que é permitido ter espaços, mas esses deverão ser retirados pelo programa antes de começar a resolução.

Tarefa

Assim, a função a ser implementada em Python receberá a string **sistema** e o dicionário de probabilidades dos componentes **prob** e retornará a probabilidade do sistema funcionar. Faça essa tarefa em um Notebook (Jupyter, por exemplo), com extensão *.ipynb. A função deverá ser chamada **resolve_sistema** e deverá ter a assinatura **resolve_sistema(sistema, prob)**.

Vou passar um testador automático e nomes diferentes de variáveis ou funções resultará em erros, podendo a nota dessa atividade ficar comprometida.

Configurações:

O máximo de componentes será 10 (de A até J) e o máximo de componentes puras (só em série ou só em paralelo) deverá ser 9 (de 1 até 9). Mais exatamente, o seu programa precisa funcionar com números entre 0 e 9. Se houver um 12, por exemplo, deveria ser visto como 1 e 2.

Sugestão:

Substitua cada sistema puro (em série ou em paralelo) por um único resultado. Por exemplo, cada novo resultado será uma letra diferente começando por M.

Exemplo (por etapa):

- 1) sistema = “1 2A&B2 & 3 4C4 | 5D5 3 1” → esse é a entrada original
- 2) sistema = “1 2 **M** 2 & 3 4C4 | 5D5 3 1” → substituímos A&B por M (resultado de A&B)
- 3) sistema = “1 2M2 & 3**N**3 1” → substituímos 4C4 | 5D5 por N.
- 4) sistema = “1 **O** 1” → sistema final

Em cada etapa, atualize o dicionário de probabilidades (prob) adicionando a nova chave e o novo valor. Por exemplo, após a etapa 2, o dicionário – inicialmente com as 4 chaves - deverá conter (os valores de probabilidade são fictícios, apenas para esse exemplo, exceto o M que tem o valor correto vindos do cálculo com A e B):

```
prob = {"A": 0.98,  
        "B": 0.97,  
        "C": 0.96,  
        "D": 0.95,  
        "M": 0.9506  
    }
```

Testes:

Para você poder saber se seu programa está funcionando, verifique a mão o sistema. Pode utilizar os exemplos resolvidos de livros, também.

Prazo:

12/09/2025 (domingo), até às 23h50min.