Métodos de Análise de Circuitos - Análise Nodal

Jupyter Notebook desenvolvido por [Gustavo S.S. (https://github.com/GSimas)](https://github.com/GSimas)

> "Nenhuma grande obra é realizada de forma apressada. Realizar uma grande descoberta científica, pintar uma grande tela, escrever um poema imortal, tornar-se um ministro ou um general famoso – realizar qualquer coisa de grandioso requer tempo, paciência e perseverança. Essas coisas são realizadas gradualmente, "pouco a pouco" - W. J. Wilmont Buxton

Transformações Y-Delta (estrela-triângulo)

Estes circuitos apresentados correspondem à rede ípsilon (Y) ou tê (T), mostrada na Figura 2.47, e à rede delta (Δ) ou pi (π), ilustrada na Figura 2.48. Essas redes ocorrem por si só ou como parte de uma rede maior e são usadas em redes trifásicas, filtros elétricos e circuitos adaptadores

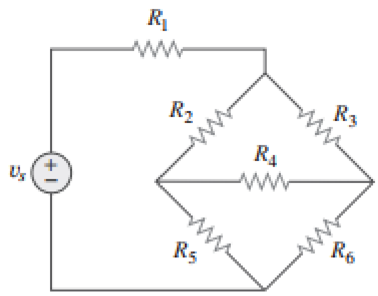

Figura 2.46 Circuito em ponte.

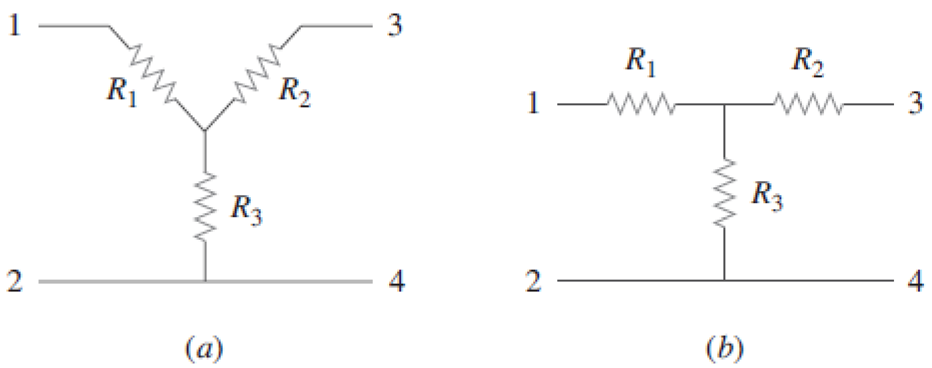

Figura 2.47 Duas formas da mesma rede: (*a*) Y; (*b*) T.

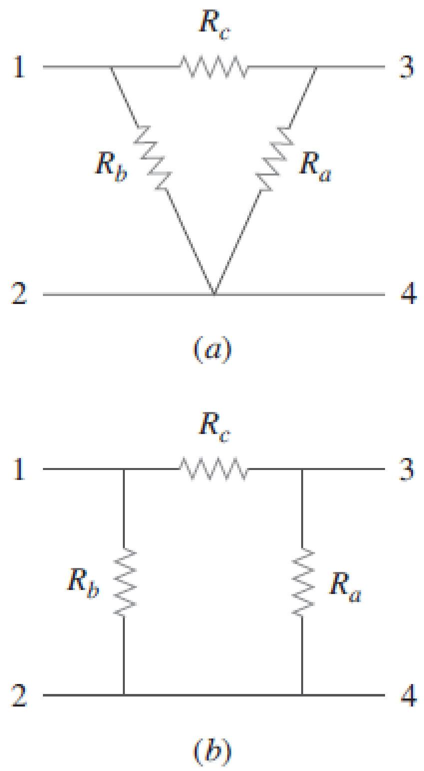

Figura 2.48 Duas formas da mesma rede: (*a*) Δ; (*b*) Π.

Conversão delta-Y (triângulo-estrela)

Em alguns casos é mais conveniente trabalhar com uma rede Y em um ponto em que o circuito contém uma configuração delta. Sobrepomos uma rede Y à rede delta existente e encontramos as resistências equivalentes na rede Y, e, para obtê-las, comparamos as duas redes e nos certificamos de que a resistência entre cada par de nós na rede Δ (ou π) é a mesma que a resistência entre o mesmo par de nós na rede Y (ou T)

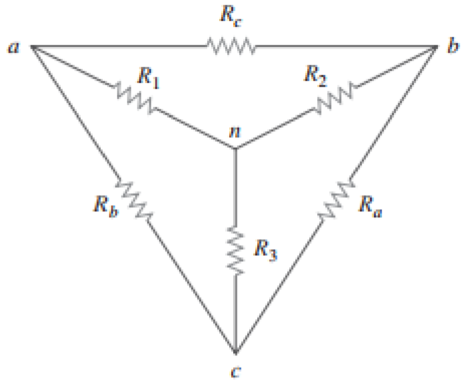

FIGURA 2.49 Superposição das redes Y e Δ como uma ferramenta na transformação de uma em outra.

Assim:

$$R_1 = \frac{R_b R_c}{R_a + R_b + R_c}$$
$$R_2 = \frac{R_a R_c}{R_a + R_b + R_c}$$
$$R_3 = \frac{R_a R_b}{R_a + R_b + R_c}$$

Resumindo:

Cada resistor na rede Y é o produto dos resistores nos dois ramos adjacentes Δ, dividido pela soma dos três resistores Δ.

Conversão Y-Delta

Para se obter o valor de resistores numa representação Delta a partir de uma Y, basta realizar o seguinte cálculo:

$$R_a = \frac{R_1 R_2 + R_2 R_3 + R_3 R_1}{R_1}$$
$$R_b = \frac{R_1 R_2 + R_2 R_3 + R_3 R_1}{R_2}$$
$$R_c = \frac{R_1 R_2 + R_2 R_3 + R_3 R_1}{R_3}$$

Resumindo:

Cada resistor na rede Δ é a soma de todos os produtos possíveis de Y resistores extraídos dois a dois, dividido pelo resistor Y oposto.

No caso, como os resistores são de mesmo valor, fica:

$$R_Y = \frac{R_\Delta}{3}$$
$$ou$$
$$R_\Delta = 3R_Y$$

Note que, ao fazer a transformação, não retiramos nem inserimos nada de novo no circuito, porque substituímos padrões de rede de três terminais diferentes, porém matematicamente equivalentes para criar um circuito no qual os resistores estão em série ou em paralelo, o que possibilita que calculemos a Req, caso necessário.

Exemplo 2.14

Converta a rede Δ da Figura 2.50a em uma rede Y equivalente.

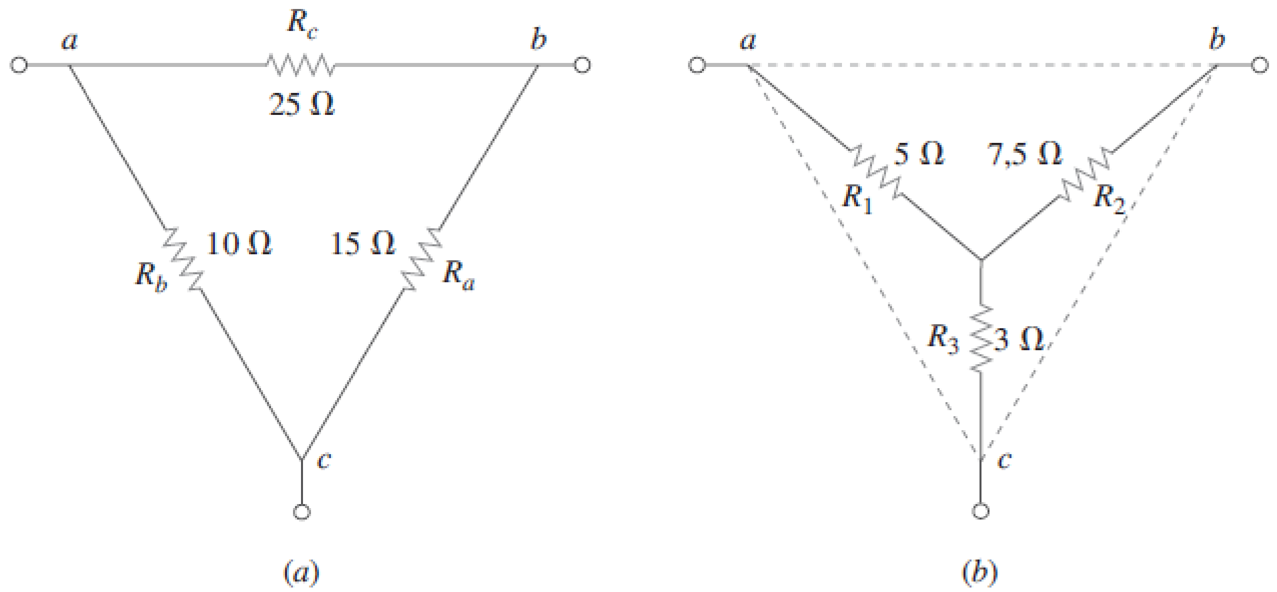

Figura 2.50 Esquema para o Exemplo 2.14: (*a*) rede Δ original; (*b*) rede Y equivalente.

In [1]:

```
print("Exemplo 2.14")
def delta_y(Ra,Rb,Rc):
    r1 = Rb*Rc/(Ra+Rb+Rc)
    r2 = Ra*Rc/(Ra+Rb+Rc)
    r3 = Ra*Rb/(Ra+Rb+Rc)
    return r1,r2,r3

r1,r2,r3 = delta_y(15,10,25)
print("R1:",r1,"ohms")
print("R2:",r2,"ohms")
print("R3:",r3,"ohms")
```

```
Exemplo 2.14
R1: 5.0 ohms
R2: 7.5 ohms
R3: 3.0 ohms
```

Problema Prático 2.14

Transforme a rede Y da Figura 2.51 em uma rede delta.

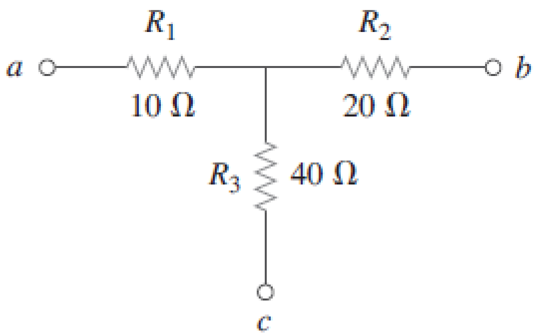

Figura 2.51 Esquema para o Problema prático 2.14.

In [3]:

```
print("Problema Prático 2.14")
def y_delta(R1,R2,R3):
    Ra = (R1*R2 + R2*R3 + R3*R1)/R1
    Rb = (R1*R2 + R2*R3 + R3*R1)/R2
    Rc = (R1*R2 + R2*R3 + R3*R1)/R3
    return Ra,Rb,Rc
ra,rb,rc = y_delta(10,20,40)
print("Ra:",ra,"ohms")
print("Rb:",rb,"ohms")
print("Rc:",rc,"ohms")
```

```
Problema Prático 2.14
Ra: 140.0 ohms
Rb: 70.0 ohms
Rc: 35.0 ohms
```

Exemplo 2.15

Obtenha a resistência equivalente Rab para o circuito da Figura 2.52 e a use para encontrar a corrente i.

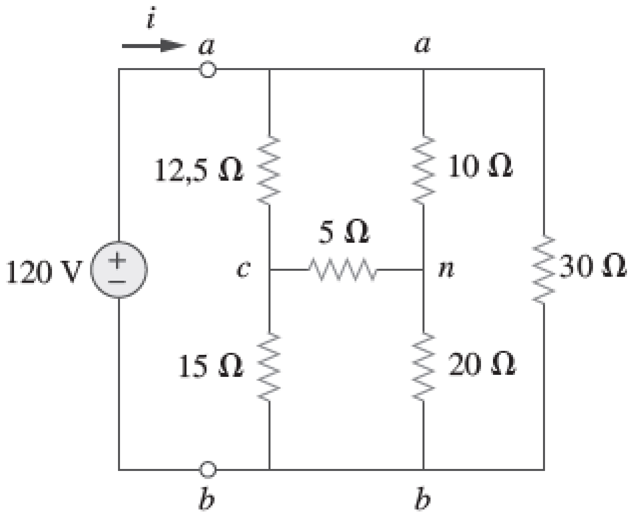

Figura 2.52 Esquema para o Exemplo 2.15.

In [7]:

```
print("Exemplo 2.15")
#analisando a triade 5ohms (rc) 15ohms (rb) e 20ohms (ra) como associacao em delta
r1,r2,r3 = delta_y(20,15,5)
Req1 = 12.5+r1 #em serie com resistor 12,5 ohm
Req2 = 10+r2 #em serie com resistor 10 ohm
Req3 = Req1*Req2/(Req1 + Req2) #paralelo entre Req1 e Req2
Req4 = Req3 + r3
Reqf = 30*Req4/(Req4 + 30)
V = 120
i = V/Reqf
print("Corrente resultante:",i,"A")
```

```
Exemplo 2.15
Corrente resultante: 12.459016393442623 A
```

Problema Prático 2.15

Para o circuito em ponte da Figura 2.54, determine Rab e i.

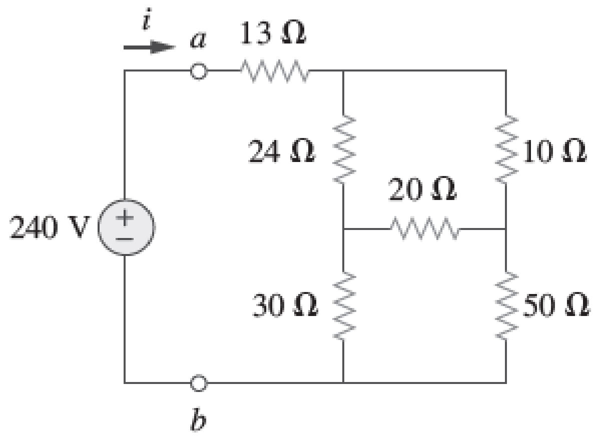

Figura 2.54 Esquema para o Problema prático 2.15.

In [9]:

```
print("Problema Prático 2.15")
r1,r2,r3 = delta_y(50,30,20)
Req1 = 24 + r1
Req2 = 10 + r2
Req3 = Req1*Req2/(Req1 + Req2)
Req4 = Req3 + r3
Reqf = Req4 + 13
V = 240
i = V/Reqf
print("Rab:",Reqf,"ohm")
print("Corrente resultante:",i,"A")
```

```
Problema Prático 2.15
Rab: 40.0 ohm
Corrente resultante: 6.0 A
```

Exemplo 2.16

Três lâmpadas são conectadas a uma fonte de 9 V, conforme mostrado na Figura 2.56a. Calcule:

(a) corrente total fornecida pela fonte;

(b) corrente que passa por cada lâmpada;

(c) resistência de cada lâmpada.

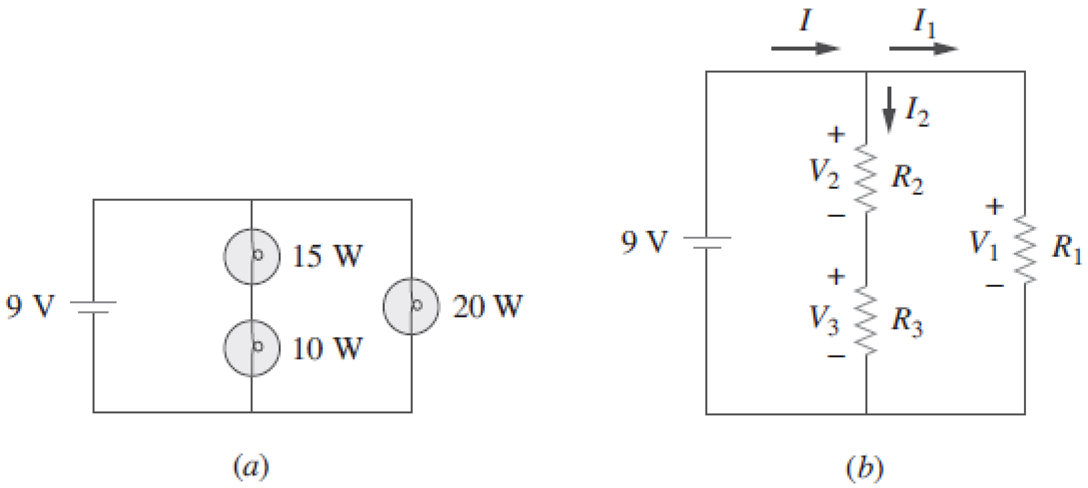

Figura 2.56 (*a*) Sistema de iluminação com três lâmpadas; (*b*) Modelo equivalente de circuito resistivo.

In [12]:

```
print("Exemplo 2.16")
V = 9
p2 = 15 #potencia da primeira Lampada
p3 = 10
p1 = 20
pt = p1 + p2 + p3
i = pt/V
i1 = p1/V
i2 = i - i1
r1 = V/i1
r2 = p2/(i2**2) # da equacao p = r*(i^2) => r = p/(i^2)
r3 = p3/(i2**2)
print("Corrente total:",i,"A")
print("Corrente na lampada 20W:",i1,"A")
print("Corrente nas lampadas 15W e 10W:",i2,"A")
print("Resistencia lampada 15W:",r2,"ohm")
print("Resistencia lampada 10W:",r3,"ohm")
print("Resistencia lampada 20W:",r1,"ohm")
```

```
Exemplo 2.16
Corrente total: 5.0 A
Corrente na lampada 20W: 2.2222222222222223 A
Corrente nas lampadas 15W e 10W: 2.7777777777777777 A
Resistencia lampada 15W: 1.9440000000000002 ohm
Resistencia lampada 10W: 1.296 ohm
Resistencia lampada 20W: 4.05 ohm
```

Problema Prático 2.16

Consulte a Figura 2.55 e considere a existência de dez lâmpadas que podem ser associadas em paralelo e dez que podem ser ligadas em série, cada uma das quais com potência nominal de 40 W. Se a tensão da rede elétrica for 110 V para as ligações em série e em paralelo, calcule a corrente através de cada lâmpada para ambos os casos.

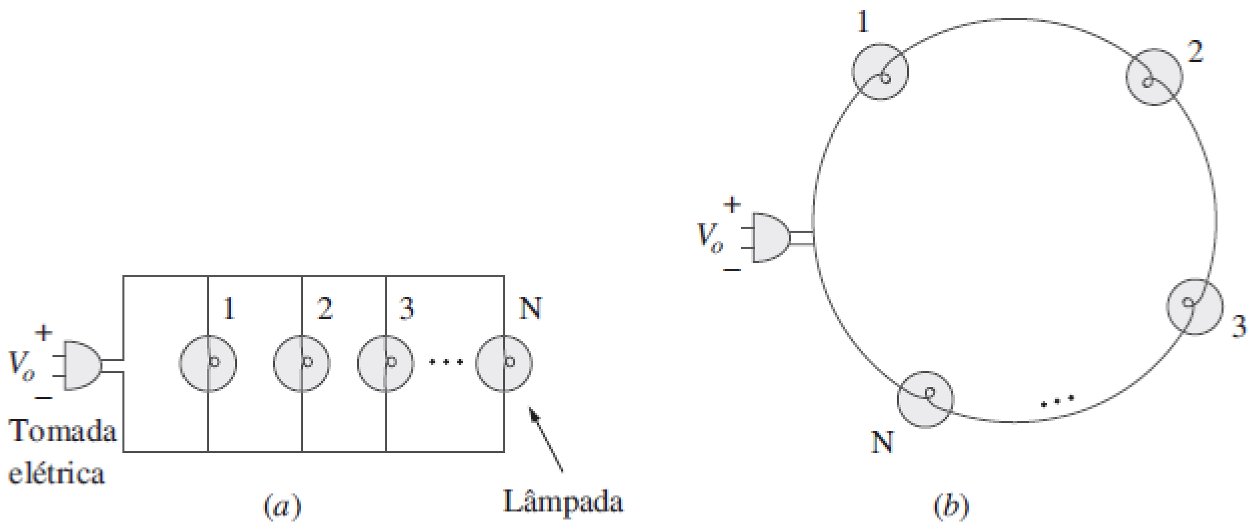

Figura 2.55 (*a*) Conexão em paralelo de lâmpadas; (*b*) Conexão em série de lâmpadas.

In [14]:

```
V = 110
p = 40
n = 10
pt = p*n  #potencia serie
i = pt/V
print("Corrente em cada lampada em serie:",i,"A")
i = p/V
print("Corrente em cada lampada em paralelo:",i,"A")
```

```
Corrente em cada lampada em serie: 3.6363636363636362 A
Corrente em cada lampada em paralelo: 0.36363636363636365 A
```

Análise Nodal

A análise nodal também é conhecida como método do nó-tensão.

Etapas para determinar tensões nodais:

1. Selecione um nó como referência.
2. Atribua tensões v1, v2, ..., vn–1 aos n – 1 nós restantes. As tensões são medidas em relação ao nó de referência.
3. Aplique a LKC a cada um dos n – 1 nós que não são de referência.
4. Use a lei de Ohm para expressar as correntes nos ramos em termos de tensões nodais.
5. Resolva as equações simultâneas resultantes para obter as tensões nodais desconhecidas.

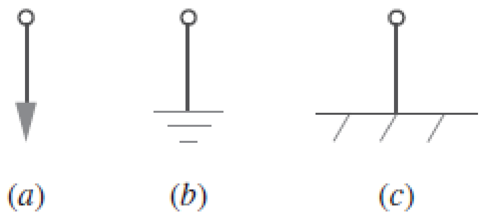

Figura 3.1 Símbolos comuns para indicar um nó de referência: (*a*) terra comum; (*b*) terra; (*c*) terra (chassi).

O número de nós que não são de referência é igual ao número de equações independentes que vamos deduzir.

Em um resistor, a corrente flui de um potencial mais elevado para um potencial mais baixo.

$$i = \frac{v_{maior} - v_{menor}}{R}$$

Exemplo 3.1

Calcule as tensões nodais no circuito mostrado na Figura 3.3a.

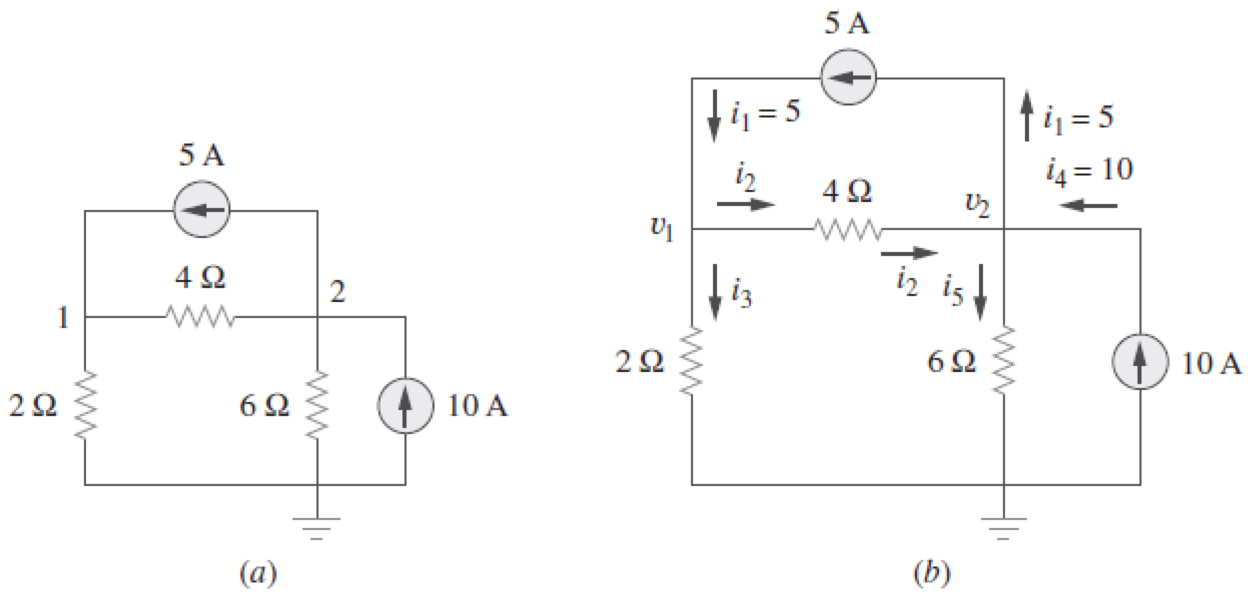

Figura 3.3 Esquema para o Exemplo 3.1: (*a*) circuito original; (*b*) circuito para análise.

In [29]:

```
print("Exemplo 3.1")
import numpy as np
#i2 + i3 = 5
#i2 + 10 = i5 + 5 => i5 = i2 + 5
#i2 = (v1-v2)/4
#i3 = v1/2
#i5 = v2/6
#(v1-v2)/4 + v1/2 = 5 => 3v1 - v2 = 20
# v2/6 = (v1-v2)/4 + 5 => 2v2 = 3v1 - 3v2 + 60 => 3v1 - 5v2 = -60
coef = np.matrix('3 -1; 3 -5')
res = np.matrix('20;-60')
V = np.linalg.inv(np.transpose(coef)*coef)*np.transpose(coef)*res #algebra linear B =
 (X'X)^(-1) * X'Y
print("V1:",float(V[0]),"V")
print("V2:",float(V[1]),"V")
```

```
Exemplo 3.1
V1: 13.333333333333334 V
V2: 20.0 V
```

Problema Prático 3.1

Obtenha as tensões nodais no circuito da Figura 3.4

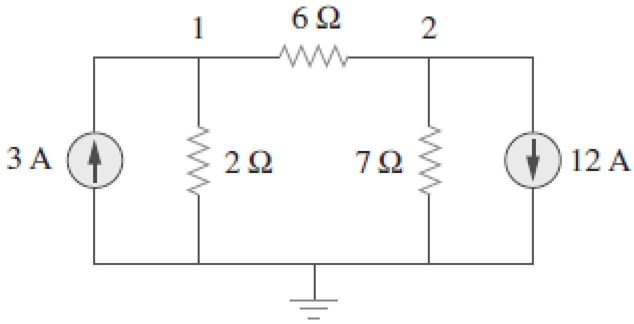

Figura 3.4 Esquema para o Problema prático 3.1.

In [35]:

```
print("Problema Prático 3.1")
#i1 + i3 = 3
#i3 = i2 + 12
#i1 = v1/2
#i2 = v2/7
#i3 = (v1-v2)/6
#v1/2 + (v1-v2)/6 = 3 => 4v1 - v2 = 18
#(v1-v2)/6 = v2/7 + 12 => 7v1 - 7v2 = 6v2 + 504 => 7v1 - 13v2 = 504
coef = np.matrix('4 -1; 7 -13')
res = np.matrix('18; 504')
V = np.linalg.inv(np.transpose(coef)*coef)*np.transpose(coef)*res
print("Tensão v1:",float(V[0]),"V")
print("Tensão v2:",float(V[1]),"V")
```

```
Problema Prático 3.1
Tensão v1: -5.999999999999962 V
Tensão v2: -41.999999999999986 V
```

Exemplo 3.2

Determine as tensões na Figura 3.5a.

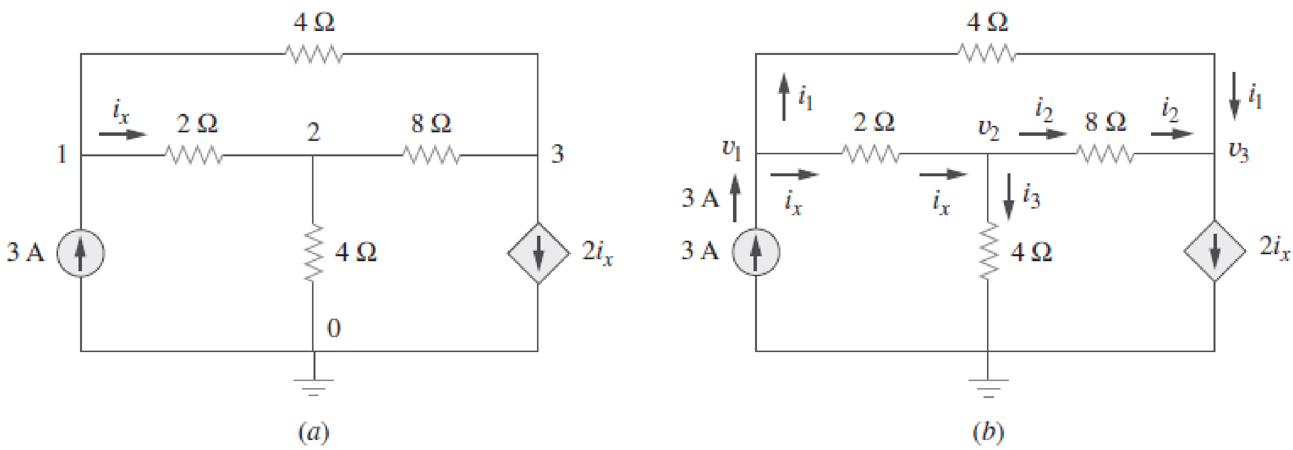

Figura 3.5 Esquema para o Exemplo 3.2: (*a*) circuito original; (*b*) circuito para análise.

In [44]:

```
print("Exemplo 3.2")
#i1 + ix = 3
#ix = i2 + i3
#i1 + i2 = 2*ix
#i1 = (v1-v3)/4
#i2 = (v2-v3)/8
#i3 = v2/4
#ix = (v1-v2)/2
#(v1-v3)/4 + (v1-v2)/2 = 3            =>     9v1 - 6v2 - 3v3 = 36
#(v1-v2)/2 = (v2-v3)/8 + v2/4         =>     4v1 - 7v2 + v3 = 0
#(v1-v3)/4 + (v2-v3)/8 = v1 - v2      =>     -6v1 + 9v2 - 3v3 = 0
coef = np.matrix('9 -6 -3;4 -7 1;-6 9 -3')
res = np.matrix('36;0;0')
V = np.linalg.inv(coef)*res
print("Tensão V1:",float(V[0]),"V")
print("Tensão V2:",float(V[1]),"V")
print("Tensão V3:",float(V[2]),"V")
```

```
Exemplo 3.2
Tensão V1: 4.799999999999999 V
Tensão V2: 2.3999999999999995 V
Tensão V3: -2.4000000000000004 V
```

Problema Prático 3.2

Determine as tensões nos três primeiros nós que não são de referência no circuito da Figura 3.6.

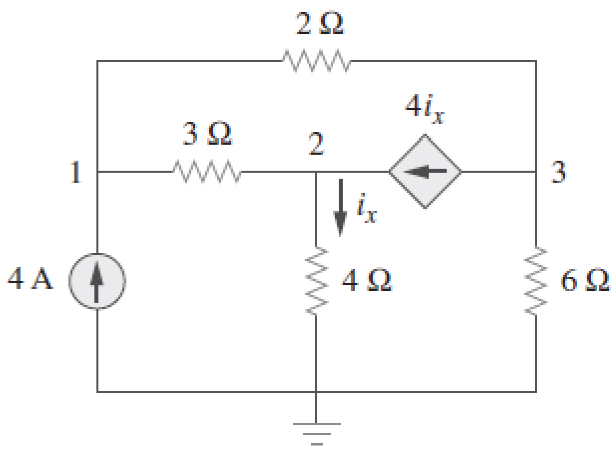

Figura 3.6 Esquema para o Problema prático 3.2.

In [47]:

```
print("Problema Prático 3.2")
#i2 = i1 + 4
#ix = i2 + 4*ix
#i1 + i3 + 4*ix = 0
#i1 = (v3-v1)/2
#i2 = (v1-v2)/3
#i3 = v3/6
#ix = v2/4
#(v1-v2)/3 = (v3-v1)/2 + 4       =>     10v1 - 4v2 -6v3 = 48
#v2/4 = (v1-v2)/3 + v2           =>     4v1 + 5v2 = 0
#(v3-v1)/2 + v3/6 + v2 = 0       =>     -3v1 + 6v2 + 4v3= 0
coef = np.matrix("10 -4 -6;4 5 0;-3 6 4")
res = np.matrix("48;0;0")
V = np.linalg.inv(coef)*res
print("Tensão V1:",float(V[0]),"V")
print("Tensão V2:",float(V[1]),"V")
print("Tensão V3:",float(V[2]),"V")
```

```
Problema Prático 3.2
Tensão V1: 32.00000000000004 V
Tensão V2: -25.600000000000044 V
Tensão V3: 62.4000000000001 V
```