Outro exemplo

```
Endereços( rua, cidade, codpostal )

DF's:

  (1) rua, cidade --> codpostal
  (2) codpostal --> cidade

Existem 2 chaves:

    * rua, cidade
    * rua, codpostal

A DF (2) viola BCNF. Vamos tentar decompor em:

   S( codpostal, cidade )
   T( rua, codpostal )

   codpostal --> cidade
```

é verificado em S. Mas

rua, cidade --> codpostal

não pode ser verificado.

EXEMPLO: as 2 relações seguintes são válidas segundo as novas DF's de S e T.

rua	codpostal
Av. 5 de Outubro	8000-039
Av. 5 de Outubro	8000-038

codpostal	cidade
8000-039	Faro
8000-038	Faro

mas ao juntarmos S com T obtemos:

rua	codpostal	cidade
Av. 5 de Outubro	8000-039	Faro
Av. 5 de Outubro	8000-038	Faro

que viola a DF original

rua, cidade --> codpostal

* A solução nestes casos é relaxar a condição BCNF.

3ª Forma Normal (3FN)

- Uma relação R está na 3FN sse para cada DF não trivial $X \rightarrow A$ se verificar que:

  1. X é superchave de R ou

  2. A é membro de pelo menos uma chave de R.

Se uma relação obedecer à condição (1) está em BCNF. Se obedecer a (1) ou a (2) está na 3FN.

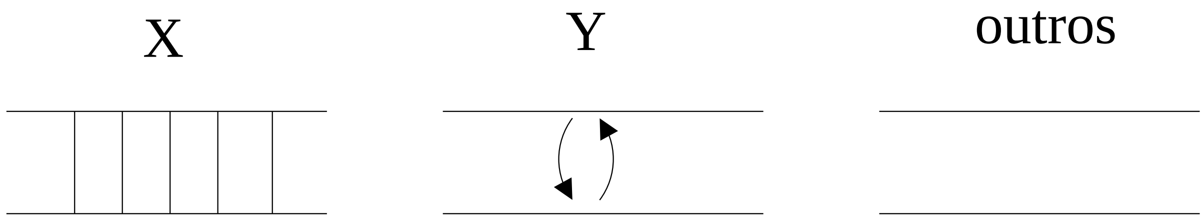
Logo, BCNF $\Rightarrow$ 3FN

Dependências Multi-Valor (DMV)

`Notação:`

```
X -->> Y
```

- A DMV $X -->> Y$ verifica-se na relação R se sempre que tivermos dois tuplos de R que tenham os mesmos valores para os atributos X, então podemos trocar os valores de Y e obter dois novos tuplos que também estão em R.



Exemplo

nome	morada	telefone	disciplina
maria	a	91-123123	BD
maria	a	91-123123	Algoritmos
maria	a	91-123123	POO
maria	a	289-444777	BD
maria	a	289-444777	Algoritmos
maria	a	289-444777	POO

```
  nome -->> telefone
  nome -->> disciplina
  nome -->> morada
```

são dependências multi-valor.

=> Dados 2 tuplos com o mesmo nome, podemos trocar o telefone e obtemos 2 tuplos que também estão na relação. A mesma coisa para disciplina.

Regra 1:

Se X --> Y é uma DF,
então X -->> Y é uma DMV.

Porquê?

Regra 2:

```
Se X -->> Y, então X -->> Z,
onde Z são todos os atributos excepto X U Y

No nosso exemplo,

  nome -->> telefone
  nome -->> disciplina
  nome -->> morada

Logo:

  nome -->> displina morada
  nome -->> telefone morada
  nome -->> telefone displina
```

4ª Forma Normal (4FN)

- Eliminar redundância devido ao efeito multiplicativo das DMV.

- Uma relação R está na 4FN se sempre que existir uma DMV: $X -- >> Y$ (não trivial), X for superchave de R.

- $X -- >> Y$ é não trivial se:

  1. nenhum dos $Y's$ está contido nos $X's$.

  2. existem atributos de R que não estão contidos nos $X's \bigcup Y's$.

No nosso exemplo...

- A relação não está na 4FN. Aliás, nem sequer está em BCNF.

```
R( nome, morada, telefone, disciplina )

DFs:
  nome --> morada

DMV:
  nome -->> telefone
  nome -->> disciplina
  nome -->> morada

A chave é {nome, telefone, disciplina}.

R não está em BCNF porque 'nome' não
contém a chave.
```

Decompomos R

Decompomos R em:

```
 * R1( nome, morada )
       ----


 * R2( nome, disciplina, telefone )
       ----  ----------  --------
```

R1 está em BCNF e também está na 4FN.

R2 está em BCNF mas não está na 4FN.

```
As DMV:
  nome -->> telefone
  nome -->> disciplina
```

violam a condição da 4FN.

Temos de decompor R2

- A solução é decompor a relação utilizando a DMV que viola a 4FN.

- Decompõe-se R usando $X -- >> Y$ em XY e $X \bigcup (R - Y)$.

4FN $\Rightarrow$ BCNF $\Rightarrow$ 3FN

No nosso exemplo, temos

```
  R2( nome, disciplina, telefone )
      ----  ----------  --------

Decompomos R2 em:

       *  R2a( nome, disciplina )
              ----  ----------

       *  R2b( nome, telefone )
              ----  --------

R2a e R2b estão ambas em BCNF e estão
ambas na 4FN.

As DMV:

  nome -->> telefone
  nome -->> disciplina

passaram a ser triviais.
```