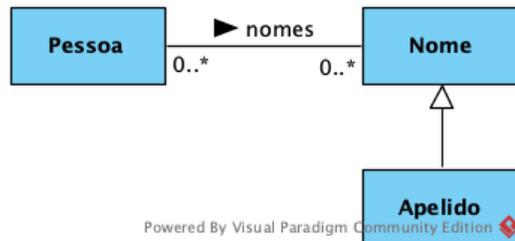


Breve introdução ao Alloy

O Alloy usa *conjuntos* para representar entidades e *relações binárias* para estabelecer associações entre elas. Um conjunto é declarado com a keyword **sig**. É possível também declarar um conjunto como um subconjunto de outro com a keyword **extends**. Uma relação binária que associa elementos de A com B é declarada dentro da **sig** A com tipo **set** B. Por exemplo, o modelo Alloy abaixo à esquerda corresponde ao modelo de domínio à direita, e declara um conjunto de pessoas, um conjunto de nomes, um subconjunto dos nomes que são apelidos, e uma relação que associa cada pessoa com os seus nomes.

```
sig Pessoa {
  nomes : set Nome
}
sig Nome {}
sig Apelido extends Nome {}
```



Para especificar requisitos em Alloy podemos usar os seguintes operadores lógicos e de conjuntos.

all x : A P	P é válido para todo o x contido em A		
P and Q	P é válido e Q é válido	no A	A é vazio
P or Q	P é válido ou Q é válido	lone A	A tem um ou menos elementos
P implies Q	Se P for válido então Q é válido	some A	A tem um ou mais elementos
not P	P não é válido	one A	A tem um elemento
A in B	A é um subconjunto de B	A + B	União de A com B
A not in B	A não é um subconjunto de B	A & B	Intersecção de A com B
A = B	A é igual a B	A - B	Diferença de A com B
A != B	A é diferente de B	A . R R . A	Conjunto dos elementos relacionados com A através de R

Por exemplo, no modelo acima Nome - Apelido é o conjunto dos nomes que não são apelidos (nomes próprios) e **some** Apelido testa se existe pelo menos um apelido. Estes operadores podem também ser aplicados a variáveis quantificadas. Por exemplo, se x é um A qualquer, a expressão x + B representa o conjunto com todos os elementos de B mais x, e a expressão x in B testa se x pertence ao conjunto B.

O operador . é muito usado em Alloy pois serve para calcular conjuntos de elementos relacionados através de uma relação binária. Por exemplo, no modelo acima, se p for uma Pessoa, então p.nomes é o conjunto de todos os nomes de p; se n for um Nome, nomes.n é o conjunto de todas as pessoas que têm pelo menos o nome n; já a expressão nomes.Apelido representa o conjunto de todas as pessoas que têm algum apelido. Podemos também encadear este operador de “navegação” para obter conjuntos mais complexos. Por exemplo, a partir da expressão p.nomes (o conjunto de todos os nomes de p) podemos voltar a “navegar” na direção oposta com a expressão nomes.(p.nomes), e obter o conjunto de todas as pessoas que têm algum nome em comum com p (note que p será obviamente uma dessas pessoas).

Usando todos estes operadores, podemos, por exemplo, especificar que todas as pessoas têm pelo menos um apelido e pelo menos um nome próprio da seguinte forma:

```
all p : Pessoa | some (p.nomes & Apelido) and some (p.nomes - Apelido)
```