

# Análise, Modelação e Teste

Teste sobre Alloy

7 de Fevereiro de 2013

Uma empresa fabrica componentes numa linha de montagem organizada em fases de fabrico sucessivas. Um componente é fabricado numa das fases de fabrico usando materiais base e/ou sub-componentes fabricados previamente. Cada fase de fabrico deve ser suportada por máquinas que fazem a montagem dos componentes lá fabricados. O seguinte modelo Alloy (incompleto) especifica formalmente uma visão estática deste processo.

```
open util/ordering[Fase]

sig Fase {}

abstract sig Produto {}

sig Componente extends Produto {
  partes : set Produto,
  fase : one Fase
}

sig Material extends Produto {}

sig Maquina {
  fase : one Fase
}
```

1. (6) Especifique invariantes que garantam as seguintes propriedades:
  - Um componente não pode ser fabricado a partir do nada.
  - Um componente não pode ser um dos seus sub-componentes.
  - Todos os sub-componentes de um componente tem que ser fabricados em fases prévias.
  - Todas as fases envolvidas no fabrico de algum componente tem que ser suportadas por máquinas.
2. (2) Altere o modelo por forma a permitir a modificação do catálogo de produtos fabricados, ou seja, permitir que as relações `Componente`, `partes`, e

*fase* (do componente) sejam mutáveis, continuando a garantir os invariantes anteriores e potenciais restrições de integridade referencial. Utilize o *local-state idiom*, utilizando a assinatura **State** para denotar o estado.

3. (4) Especifique uma operação que permita acrescentar um novo sub-componente ou material (um **Produto**) às partes necessárias para fabricar um componente, garantindo que essa operação preserva os invariantes anteriores.
4. (4) Considere agora que existe um stock de materiais especificado da seguinte forma:

```
sig Material extends Produto {
  stock : Int -> State
}

fact {
  all s : State, m : Material {
    one m.(stock.s) and gte[m.(stock.s),0]
  }
}
```

Especifique um predicado que teste se um componente pode ser fabricado num determinado estado, ou seja, existe material suficiente em stock para o fabricar (incluindo todos os sub-componentes).

5. (2) Usando o modelo apresentado no início do teste, apresente três exemplos de expressões que apresentem erros devido a *bounding types* vazios, *relevance types* vazios, e ambiguidade, respectivamente. Justifique formalmente a sua resposta.
6. (2) Considere o seguinte modelo dinâmico, com duas operações **X** e **Y** modeladas com eventos e *frame conditions* ao estilo de Reiter.

```
sig State {}

sig A {
  r : set State, s : set State, t : set State
}

abstract sig Event {
  pre: State, pos: State,
  a : A
} {
  r.pos = r.pre + a
  s.pos != s.pre implies this in X
  t.pos != t.pre implies this in Y
}
```

```
}  
  
sig X extends Event {} {  
  s.pos = s.pre + a  
}  
  
sig Y extends Event {} {  
  t.pos = t.pre + a  
}
```

Defina um modelo equivalente especificando as operações usando predicados.