

15 de dezembro de 2016

Iniciação Científica – “Abordagem BETA e o projeto de testes a partir de modelos formais em notação B”

Cobertura de Requisitos de Teste

A partir da máquina “Marriage.mch” separar as características, arrumar os requisitos de teste e comparar com os casos de teste com o BETA.

Larissa Monteiro da Fonseca Galeno

Matheus Henrique Panno Guimarães

Ver. 5.0

- Passo a passo para a atividade:
 - I. Definir a estratégia de teste;
 - II. Definir o Critério de cobertura;
 - III. Escolher a operação que será coberta;
 - IV. A partir de uma análise da máquina concluir quais são as características;
 - V. Com isso separar os possíveis valores em blocos;
 - VI. E então, dependendo do critério escolhido, combino os blocos e as características;
 - VII. Gero os requisitos de teste;
 - VIII. Assim, rodar a máquina na ferramenta BETA e gerar os relatórios em HTML;
 - IX. Checo e vejo quais requisitos a BETA cobriu.

Máquina escolhida:

MACHINE Marriage

USES Life

VARIABLES marriage

INVARIANT marriage : male >+> female

INITIALISATION

marriage := {}

OPERATIONS

wed(mm, ff) =

PRE

mm : male &

mm /: dom(marriage) &

ff : female &

ff /: ran(marriage)

THEN

marriage(mm) := ff

END;

part(mm, ff) =

PRE

mm : male &

ff : female &

mm |-> ff : marriage

THEN

```

marriage := marriage - {mm |-> ff}
END;

pp <-- partner(nn) =
  PRE
  nn : dom(marriage) ∨ ran(marriage)
  THEN
  IF nn : dom(marriage)
  THEN pp := marriage(nn)
  ELSE pp := marriage~(nn)
  END
  END
END

```

1) **Estratégia de teste:** *Input Space Partitioning;*

Todos os casos de Teste presentes nesse documento foram gerados pela ferramenta BETA.

Abaixo segue a legenda para as tabelas que se encontram ao longo deste documento, a fim de compreender quais Casos de Teste (CTy) estão sendo gerados pela ferramenta por meio de um esquema de cores.

| | |
|-----|--|
| CTy | amarelo – possíveis bugs |
| CTy | vermelho - Casos de teste INF CONSIDERADOS pelo BETA |
| CTy | verde - Casos de teste COBERTOS pelo BETA |

Critério de Cobertura: *Equivalent Classes;*

a) **Operação:** Casamento (wed(mm,ff))

Características:

c1 – mm : male
 c2 - mm /: dom(marriage)
 c3 - ff : female
 c4 - ff /: ran(marriage)

Blocos:

- Para c1 :

c1b1 - mm: male
 c1b2 - not(mm: male)

- Para c2:

c2b1 -mm/:dom(marriage)
c2b2 - not(mm/:dom(marriage))

- Para c3:

c3b1 - ff: female
c3b2- not(ff:female)

-Para c4:

c4b1 - ff/:ran(marriage)
c4b2 - not(ff/:ran(marriage))

a.1)***Each-Choice:***

Requisitos para Each Choice:

TR = {c1b1 , c1b2, c2b1, c2b2, c3b1,c3b2, c4b1, c4b2}

O BETA cobriu os requisitos c1b1, c2b1, c3b1 e c4b1, porém foram identificados diversos outros que poderiam ter sido cobertos.É possível com a geração de mais casos de teste que esse problema seja resolvido. Foi montado uma tabela para melhorar o entendimento dos resultados.

| Req X CT | CT1 | CT2 |
|-------------|-----|-----|
| c1b1 | | |
| c1b2 | | |
| c2b1 | | |
| c2b2 | | |
| c3b1 | | |
| c3b2 | | |
| c4b1 | | |
| c4b2 | | |

| |
|------------------------------|
| CT1 - c1b1, c2b1, c3b1. c4b1 |
| CT2 - c1b2, c2b2, c3b2, c4b2 |

Casos de Teste gerados pelo BETA para Each Choice:

- CT1(Positivo):
Variáveis:
female ={PERSON2};
male={PERSON1} ;
marriage={} ;
Valores de Entrada:

mm=PERSON1 , ff=PERSON2;

a.2)**Pairwise:**

Requisitos para Pairwise:

TR = {(c1b1,c2b1), (c1b1,c3b1), (c1b1,c4b1), (c1b1,c2b2), (c1b1,c3b2), (c1b1,c4b2), (c1b2,c2b1), (c1b2,c3b1), (c1b2,c4b1), (c1b2,c2b2),(c1b2,c3b2), (c1b2,c4b2), (c2b1,c3b1), (c2b1,c4b1), (c2b1, c3b2), (c2b1, c4b2) ,(c2b2,c3b1), (c2b2,c4b1), (c2b2,c3b2), (c2b2,c4b2), (c3b1,c4b1), (c3b1,c4b2), (c3b2,c4b1), (c3b2,c4b2)}

O BETA cobriu os requisitos (c1b1,c4b1), (c2b1,c4b1), (c1b1,c3b2), (c3b1,c4b1) e (c1b2,c2b1), porém foram identificados diversos outros que poderiam ter sido cobertos. Ao montarmos uma tabela para melhorar o entendimento dos resultados foi observado que a ferramenta considerou o bloco c4b2 (*not(ff/:ran(marriage))*) como insatisfazível em combinação com outros blocos. Com isso, foi feito uma análise mais minuciosa da veracidade desse fato, já que não havia razão visível para que isso acontecesse.

Foi feita uma máquina parcial “Teste.mch” onde nela continha apenas a operação wed com a adição do invariante, da Marriage.mch, e a cláusula do bloco c4b2 em suas pré-condições, substituindo a que havia no lugar .Em seguida a máquina foi “carregada” para o ProB para que fosse feito a animação da mesma e foi constatado que o bloco c4b2 não é insatisfazível por si só, uma vez que o ProB foi capaz de realizar a animação da máquina. Logo é possível que o resultado obtido no BETA seja um *bug*.

Máquina Parcial:

MACHINE Teste
SEES Life

OPERATIONS

```
wed(mm, ff,marriage) =  
  PRE  
    marriage : male >+> female &  
    mm : male &  
    mm /: dom(marriage) &  
    ff : female &  
    not(ff /: ran(marriage))  
  THEN  
    skip  
  END  
END
```

Tabela utilizada:

| Req X CT | CT1 | CT2 | CT3 | CT4 | CT5 | CT6* |
|-------------|-----|-----|-----|-----|-----|------|
| (c1b1,c2b1) | | | | | | |
| (c1b1,c3b1) | | | | | | |
| (c1b1,c4b1) | | | | | | |
| (c1b1,c2b2) | | | | | | |
| (c1b1,c3b2) | | | | | | |
| (c1b1,c4b2) | | | | | | |
| (c1b2,c2b1) | | | | | | |
| (c1b2,c2b2) | | | | | | |
| (c1b2,c3b1) | | | | | | |
| (c1b2,c3b2) | | | | | | |
| (c1b2,c4b1) | | | | | | |
| (c1b2,c4b2) | | | | | | |
| (c2b1,c3b1) | | | | | | |
| (c2b1,c3b2) | | | | | | |
| (c2b1,c4b1) | | | | | | |
| (c2b1,c4b2) | | | | | | |
| (c2b2,c3b1) | | | | | | |
| (c2b2,c3b2) | | | | | | |
| (c2b2,c4b1) | | | | | | |
| (c2b2,c4b2) | | | | | | |
| (c3b1,c4b1) | | | | | | |
| (c3b1,c4b2) | | | | | | |
| (c3b2,c4b1) | | | | | | |
| (c3b2,c4b2) | | | | | | |

** Bug que considera a combinação de c4b2 com outros blocos infeasible.*

CT1 – (c1b1,c2b1),(c3b2,c4b1)

| |
|---------------------------------|
| CT2 - (c1b1,c2b2),(c3b2,c4b1) |
| CT3 - (c1b2, c2b1),(c3b1, c4b1) |
| CT4 - (c1b2, c2b2),(c3b1, c4b1) |
| CT5 - (c1b2,c2b2),(c3b2, c4b2) |
| CT6 - (c1b1, c2b1),(c3b1,c4b2) |

Casos de Teste gerados pelo BETA para o Pairwise:

- CT1(Negativo):
Variáveis:
female={}
male={PERSON1}
marriage={}
Valores de entrada:
female=PERSON1; male=PERSON1
- CT2(Negativo):
Variáveis:
female={PERSON1}
male={}
marriage={}
Valores de entrada:
female=PERSON1; male=PERSON1
- CT3(Negativo)
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={(PERSON1|->PERSON2)}
Valores de entrada:
female=PERSON1; male=PERSON1

a.3) **All Combinations:**

Requisitos para o All Combination:

TR = { (c1b1, c2b1, c3b1, c4b1), (c1b1, c2b2, c3b1, c4b1), (c1b1, c2b1, c3b2, c4b1), (c1b1, c2b1, c3b1, c4b2), (c1b1, c2b2, c3b2, c4b1), (c1b1, c2b2, c3b1, c4b2), (c1b1, c2b1, c3b2, c4b2), (c1b1, c2b2, c3b2, c4b2), (c2b2, c2b1, c3b1, c4b1), (c2b2, c2b2, c3b1, c4b1), (c1b2, c2b1, c3b2, c4b1), (c1b2, c2b1, c3b1, c4b2), (c1b2, c2b2, c3b2, c4b1), (c1b2, c2b2, c3b1, c4b2), (c1b2, c2b1, c3b2, c4b2), (c1b2, c2b2, c3b2, c4b2) }

O BETA não cobriu todos requisitos possíveis, mesmo gerando todos os casos de teste possíveis. Foi observado o requisito c1b1,c2b2,c3b1,c4b1 (ff /: ran(marriage) & ff : female & mm : male & not(mm /: dom(marriage)) é posto como infeseable, apesar de não aparentar ser. Para confirmar criamos outra máquina parcial e ao animarmos vimos que ela rodou normalmente mostrando que, de fato, esse requisito não é infeseable.

Máquina Parcial usada:

MACHINE Teste

SEES Life

OPERATIONS

wed(mm, ff,marriage) =

PRE

marriage : male >+> female &

ff /: ran(marriage) &

ff : female &

mm : male &

not(mm /: dom(marriage))

THEN

skip

END

END

Podendo ser esse, então, outro bug que o BETA possui.Foi feita uma tabela para melhorar o entendimento dos resultados.

| Req X CT | CT1 | CT2** | CT3 | CT4* | CT5 | CT6 | CT7* | CT8 | CT9 | CT10 | CT11 | CT12 | CT13 | CT14 | CT15 | CT16 |
|-----------------------|-----|-------|-----|------|-----|-----|------|-----|-----|------|------|------|------|------|------|------|
| (c1b1,c2b1,c3b1,c4b1) | | | | | | | | | | | | | | | | |
| (c1b1,c2b2,c3b1,c4b1) | | | | | | | | | | | | | | | | |
| (c1b1,c2b1,c3b2,c4b1) | | | | | | | | | | | | | | | | |
| (c1b1,c2b1,c3b1,c4b2) | | | | | | | | | | | | | | | | |
| (c1b1,c2b2,c3b2,c4b1) | | | | | | | | | | | | | | | | |
| (c1b1,c2b2,c3b1,c4b2) | | | | | | | | | | | | | | | | |
| (c1b1,c2b1,c3b2,c4b2) | | | | | | | | | | | | | | | | |
| (c1b1,c2b2,c3b2,c4b2) | | | | | | | | | | | | | | | | |
| (c1b2,c2b1,c3b1,c4b1) | | | | | | | | | | | | | | | | |
| (c1b2,c2b2,c3b1,c4b1) | | | | | | | | | | | | | | | | |
| (c1b2,c2b1,c3b2,c4b1) | | | | | | | | | | | | | | | | |
| (c1b2,c2b1,c3b1,c4b2) | | | | | | | | | | | | | | | | |
| (c1b2,c2b2,c3b2,c4b1) | | | | | | | | | | | | | | | | |
| (c1b2,c2b2,c3b1,c4b2) | | | | | | | | | | | | | | | | |
| (c1b2,c2b1,c3b2,c4b2) | | | | | | | | | | | | | | | | |
| (c1b2,c2b2,c3b2,c4b2) | | | | | | | | | | | | | | | | |

** Bug que considera a combinação de c4b2 com outros blocos infeasible.*

*****Com isso, foi observado o requisito c1b1,c2b2,c3b1,c4b1 (ff /: ran(marriage) & ff : female & mm : male & not(mm /: dom(marriage)) é posto como infesable, apesar de não aparentar ser. Para confirmar criamos outra máquina parcial e ao animarmos vimos que ela rodou normalmente mostrando que, de fato, esse requisito não é infesable.***

| |
|---------------------------------|
| CT1 - (c1b1,c2b1,c3b1,c4b1) |
| CT2 - (c1b1,c2b2,c3b1,c4b1) |
| CT3 - (c1b1, c2b1,c3b2,c4b1) |
| CT4 - (c1b1, c2b1,c3b1,c4b2) |
| CT5 - (c1b1,c2b2,c3b2,c4b1) |
| CT6 - (c1b1,c2b2,c3b1,c4b2) |
| CT7 - (c1b1,c2b1,c3b2,c4b2) |
| CT8 - (c1b1,c2b2,c3b2,c4b2) |
| CT9 - (c1b2, c2b1,c3b1, c4b1) |
| CT10 - (c1b2, c2b2,c3b1, c4b1) |
| CT11 - (c1b2, c2b1,c3b2, c4b1) |
| CT12 - (c1b2, c2b1, c3b1, c4b2) |
| CT13 - (c1b2, c2b2, c3b2, c4b1) |
| CT14 - (c1b2, c2b2,c3b1, c4b2) |
| CT15 - (c1b2, c2b1, c3b2, c4b2) |
| CT16 - (c1b2, c2b2, c3b2, c4b2) |

Casos de Teste gerados pelo BETA para All Combinations:

- CT1(Positivo):
Variáveis:
female = {PERSON2};
male = {PERSON1} ;
marriage{} ;
Valores de Entrada:
mm=PERSON1 , ff=PERSON2;

- CT3(Negativo):
Variáveis:
female = {};
male = {PERSON1} ;
marriage{} ;
Valores de Entrada:
mm=PERSON1 , ff=PERSON1;

- CT5(Negativo):
Variáveis:
female = {PERSON1};
male = {} ;
marriage{} ;
Valores de Entrada:
mm=PERSON1 , ff=PERSON1;

- CT6(Negativo):
 Variáveis:
 female ={};
 male={}
 marriage{} ;
 Valores de Entrada:
 mm=*PERSON1* , ff=*PERSON1*;

- CT9(Negativo):
 Variáveis:
 female ={*PERSON1*};
 male={*PERSON2*}
 marriage{(*PERSON2*|->*PERSON1*)} ;
 Valores de Entrada:
 mm=*PERSON1* , ff=*PERSON1*;

- CT11(Negativo):
 Variáveis:
 female ={*PERSON2*};
 male={*PERSON1*}
 marriage{(*PERSON1*|->*PERSON2*)} ;
 Valores de Entrada:
 mm=*PERSON1* , ff=*PERSON1*;

- CT12(Negativo):
 Variáveis:
 female ={*PERSON2*};
 male={*PERSON1*}
 marriage{(*PERSON1*|->*PERSON2*)} ;
 Valores de Entrada:
 mm=*PERSON1* , ff=*PERSON2*;

b) **Operação:**Divórcio (part(mm,ff))

Características:

c1 - mm : male
 c2 - ff : female
 c3 - mm |-> ff : marriage

Blocos:

- Para c1 :

c1b1 - mm:male
 c1b2 - not(mm:male)

- Para c2:

c2b1 - ff: female
c2b2- not(ff:female)

- Para c3:

c3b1 - mm |-> ff : marriage
c3b2 - not(mm |-> ff : marriage)

b.1) ***Each-Choice:***

Requisitos para Each Choice:

TR = {c1b1 , c1b2, c2b1, c2b2, c3b1,c3b2}

O BETA cobriu todos os requisitos. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 |
|----------|-----|-----|
| c1b1 | | |
| c1b2 | | |
| c2b1 | | |
| c2b2 | | |
| c3b1 | | |
| c3b2 | | |

| |
|------------------------|
| CT1 - c1b1, c2b1, c3b1 |
| CT2 - c1b2, c2b2, c3b2 |

Casos de Teste gerados pelo BETA para Each-Choice:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1 |-> PERSON2}
Valores de entrada:
female=PERSON1; male=PERSON2
- CT2(Negativo):
Variáveis:
female={}
male={}
marriage={}
Valores de entrada:

female=PERSON1; male=PERSON1

b.2) **Pairwise:**

Requisitos para Pairwise:

TR = {(c1b1, c2b1), (c1b1, c2b2), (c1b1, c3b1), (c1b1, c3b2), (c1b2, c2b1), (c1b2, c2b2), (c1b2, c3b1), (c1b2, c3b2), (c2b1, c3b1), (c2b1, c3b2), (c2b2, c3b1), (c2b2, c3b2)}

A ferramenta BETA não foi capaz de cobrir todos os requisitos impostos pelo pairwise. Com isso, se faz necessário a geração de mais casos de teste para que seja alcançada uma cobertura de 100% da operação. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 | CT3 | CT4 |
|--------------|-----|-----|-----|-----|
| (c1b1, c2b1) | | | | |
| (c1b1, c2b2) | | | | |
| (c1b1, c3b1) | | | | |
| (c1b1, c3b2) | | | | |
| (c1b2, c2b1) | | | | |
| (c1b2, c2b2) | | | | |
| (c1b2, c3b1) | | | | |
| (c1b2, c3b2) | | | | |
| (c2b1, c3b1) | | | | |
| (c2b1, c3b2) | | | | |
| (c2b2, c3b1) | | | | |
| (c2b2, c3b2) | | | | |

| |
|--------------------------------|
| CT1 - (c1b1,c2b1),(c3b1,c1b1) |
| CT2 - (c1b1,c2b1),(c3b2,c1b1) |
| CT3 - (c1b2, c2b1),(c3b2,c1b2) |
| CT4 - (c1b2,c2b2),(c3b1,c1b2) |

Casos de Teste gerados pelo BETA para Pairwise:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1 |-> PERSON2}
Valores de entrada:
female=PERSON1; male=PERSON2

- CT2(Negativo):
Variáveis:
female={}
male={PERSON1}
marriage={}
Valores de entrada:
female=PERSON1; male=PERSON1

- CT3(Negativo):
Variáveis:
female={PERSON1}
male={}
marriage={}
Valores de entrada:
female=PERSON1; male=PERSON1

b.3) **All Combinations:**

Requisitos para o All Combination:

TR = { (c1b1, c2b1, c3b1), (c1b1, c2b2, c3b1), (c1b1, c2b1, c3b2), (c1b1, c2b2, c3b2), (c1b2, c2b1, c3b1), (c1b2, c2b2, c3b1), (c1b2, c2b1, c3b2), (c1b2, c2b2, c3b2) }

O BETA não foi capaz de cobrir todos os requisitos existentes, mesmo gerando todos os casos de teste possíveis. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 | CT3 | CT4 | CT5 | CT6 | CT7 | CT8 |
|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| (c1b1, c2b1, c3b1) | | | | | | | | |
| (c1b1,c2b2,c3b1) | | | | | | | | |

| | | | | | | | | |
|--------------------|--|--|--|--|--|--|--|--|
| (c1b1,c2b1,c3b2) | | | | | | | | |
| (c1b1,c2b2,c3b2) | | | | | | | | |
| (c1b2,c2b1,c3b1) | | | | | | | | |
| (c1b2,c2b2,c3b1) | | | | | | | | |
| (c1b2, c2b1, c3b2) | | | | | | | | |
| (c1b2, c2b2, c3b2) | | | | | | | | |

| |
|-------------------------|
| CT1 - (c1b1,c2b1,c3b1) |
| CT2 - (c1b1,c2b1,c3b2) |
| CT3 - (c1b1, c2b2,c3b2) |
| CT4 - (c1b2, c2b1,c3b2) |
| CT5 - (c1b2,c2b2,c3b2) |
| CT6 - (c1b1,c2b2,c3b1) |
| CT7 - (c1b2,c2b1,c3b1) |
| CT8 - (c1b2,c2b2,c3b1) |

Casos de Teste gerado pelo BETA para All Combination:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1|->PERSON2}
Valores de entrada:
female=PERSON2; male=PERSON1
- CT2(Negativo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={}
Valores de entrada:
female=PERSON2; male=PERSON1
- CT3(Negativo):
Variáveis:
female={}
male={PERSON1}
marriage={}
Valores de entrada:
female=PERSON1; male=PERSON1

- CT4(Negativo):
 Variáveis:
 female={PERSON1}
 male={}
 marriage={}
 Valores de entrada:
 female=PERSON1; male=PERSON1
- CT5(Negativo):
 Variáveis:
 female={}
 male={}
 marriage={}
 Valores de entrada:
 female=PERSON1; male=PERSON1

c) Operação: Parceiro (partner(nn))

Características:

c1 - nn : dom(marriage) \vee ran(marriage)
 c2 - nn : dom(marriage)

Blocos:

- Para c1 :

c1b1 - nn : dom(marriage) \vee ran(marriage)
 c1b2 - not(nn : dom(marriage) \vee ran(marriage))

- Para c2:

c2b1 - nn : dom(marriage)
 c2b2 - not(nn : dom(marriage))

c.1) *Each-Choice*:

Requisitos para Each Choice:

TR = {c1b1 , c1b2, c2b1, c2b2}

O BETA cobriu todos os requisitos. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 |
|----------|-----|-----|
| c1b1 | | |
| c1b2 | | |
| c2b1 | | |
| c2b2 | | |

| |
|------------------|
| CT1 - c1b1, c2b1 |
| CT2 - c1b2, c2b2 |

Casos de Teste gerados pelo BETA para Each-Choice:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1|->PERSON2}
Valores de entrada:
nn=PERSON1

- CT2(Negativo):
Variáveis:
female={}
male={}
marriage={}
Valores de entrada:
nn=PERSON1

c.2) **Pairwise:**

Requisitos para Pairwise:

TR = {(c1b1, c2b1), (c1b1, c2b2), (c1b2, c2b1), (c1b2, c2b2)}

A ferramenta BETA não foi capaz de cobrir todos os requisitos,mesmo gerando todos os casos de teste possíveis. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 | CT3 | CT4 |
|--------------|-----|-----|-----|-----|
| (c1b1, c2b1) | | | | |
| (c1b1, c2b2) | | | | |
| (c1b2, c2b1) | | | | |
| (c1b2, c2b2) | | | | |

| |
|--------------------|
| CT1 - (c1b1, c2b1) |
| CT2 - (c1b1, c2b2) |
| CT3 - (c1b2, c2b2) |
| CT4 - (c1b2, c2b1) |

Casos de Teste gerados pelo BETA para Pairwise:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1|->PERSON2}
Valores de entrada:
nn=PERSON1
- CT2(Negativo):
Variáveis:
female={PERSON1}
male={PERSON2}
marriage={PERSON2|->PERSON1}
Valores de entrada:
nn=PERSON1
- CT3(Negativo):
Variáveis:
female={}
male={}
marriage={}
Valores de entrada:
nn=PERSON1

c.3) All Combinations:

Requisitos para o All Combination:

TR = {(c1b1, c2b1), (c1b1, c2b2), (c1b2, c2b1), (c1b2, c2b2)}

Foi obtido o mesmo resultado que no pairwise, o BETA não cobriu todos os requisitos, mesmo gerando todos os casos de teste. Foi feita uma tabela para melhor visualização da combinação dos requisitos.

| Req X CT | CT1 | CT2 | CT3 | CT4 |
|--------------|-----|-----|-----|-----|
| (c1b1, c2b1) | | | | |
| (c1b1, c2b2) | | | | |
| (c1b2, c2b1) | | | | |
| (c1b2, c2b2) | | | | |

| |
|--------------------|
| CT1 - (c1b1, c2b1) |
| CT2 - (c1b1, c2b2) |
| CT3 - (c1b2, c2b2) |
| CT4 - (c1b2, c2b1) |

Casos de Teste gerados pelo BETA para All Combinations:

- CT1(Positivo):
Variáveis:
female={PERSON2}
male={PERSON1}
marriage={PERSON1|->PERSON2}
Valores de entrada:
nn=PERSON1
- CT2(Negativo):
Variáveis:
female={PERSON1}
male={PERSON2}
marriage={PERSON2|->PERSON1}
Valores de entrada:
nn=PERSON1
- CT3(Negativo):
Variáveis:
female={}
male={}
marriage={}
Valores de entrada:
nn=PERSON1