UNIVERSIDADE FEDERAL DE PERNAMBUCO

CIN – CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

2009.1

USO DE ESPECTROS DE EXECUÇÃO PARA ANÁLISE DE MODIFICAÇÕES EM UM PROGRAMA

TRABALHO DE GRADUAÇÃO

RAFAEL ARAÚJO SANTANA DE OLIVEIRA

Junho de 2009

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CIN – CENTRO DE INFORMÁTICA

GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

2009.1

USO DE ESPECTROS DE EXECUÇÃO PARA ANÁLISE DE MODIFICAÇÕES EM UM PROGRAMA

Este trabalho foi apresentado ao programa de Graduação em Ciência da Computação pelo Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Bacharel em Ciências da Computação.

Orientador: Marcelo Bezerra d'Amorim

ASSINATURAS

Este Trabalho de Graduação representa o esforço do aluno **Rafael Araújo Santana de Oliveira**, orientado pelo professor **Marcelo Bezerra d'Amorim**, sob o título **"Uso de espectros de execução para análise de modificações em um programa"**. Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste trabalho de graduação.

_____

Marcelo Bezerra d'Amorim

(orientador)

_____

Rafael Araújo Santana de Oliveira

(aluno)

À minha família

AGRADECIMENTOS

Aos meus pais por me terem me apoiado durante todo o período de graduação. Sem o apoio deles este trabalho não seria possível.

Ao professor Marcelo Bezerra d'Amorim por estimular a produção deste trabalho e pelas inúmeras conversas que contribuíram para o refinamento das ideias finais aqui trazidas.

Ao amigo Elton Renan Magalhães Alves por ter contribuído enormemente como co-autor da ideia original do comparador do espectro de *branches* o usado neste trabalho.

A todos os outros amigos aqui não citados por não serem diretamente ligado à produção deste trabalho. Estes não são menos importantes.

RESUMO

É pratica comum no desenvolvimento de programas executar a *test suite* após modificações para detectar erros de regressão. Este procedimento nem sempre é efetivo e depende diretamente da qualidade das sequências de testes escolhidos pelos desenvolvedores que invariavelmente pensam no custo da execução de toda a *test suite*. Somente uma *test suite* adequada é capaz de detectar grande parte dos erros de regressão. Este trabalho se propõe a fornecer uma alternativa que pode melhorar a qualidade dos testes de regressão tanto conjuntamente com o uso de *test suites* existentes, sejam adequadas ou não, quanto com a geração automática de testes. A alternativa proposta, o uso de espectro de *branch*, se mostra tão eficiente ou mais que os testes usuais.

Palavras chave: oracle automático, testes de diferenciação, espectro de branches

ABSTRACT

It is common during program development to execute a test suite after modification to provide regression error detection. This procedure is not always effective and is affected directly by the quality of the test sequences chosen by developers who invariably think about the execution costs of the entire test suite. Only a good test suite is capable of detecting a good portion of the regression errors. This work aims to provide an alternative solution that can improve regression test quality in cooperation with existing test suites or with automatic generated tests. The proposed alternative, the use of branch-spectrum, shows to be equally efficient or more than the usual test methods.

Keywords: automatic test-oracle, differential testing, branch spectrum

SUMÁRIO

Introdução....1

Espectros de execução....1

Problema....2

Solução....2

Estrutura do trabalho....3

1. Concepção....4

1.1. O espectro de branches....4

1.2. Identificação de equivalência entre branches....6

1.3. Equivalência entre espectros....7

2. Implementação....10

2.1. Instrumentação estática....10

2.2. Ajustes após a instrumentação....13

2.3. Registros no espectro....14

2.4. Comparação....15

3. Experimentação....16

3.1. Metodologia....16

3.2. Resultados....17

4. Conclusões....19

4.1. Limitações....19

4.2. Trabalhos relacionados....20

4.3. Trabalhos futuros....20

Bibliografia....22

A. Resultados detalhados da classificação....23

INTRODUÇÃO

Garantir que erros e falhas estejam ausentes em um *software* é tarefa delicada. Quase sempre é impossível ou inviável garantir esta ausência devido a grande velocidade de mudança de código seja por adição de novas funcionalidades ou melhoria das já existentes.

Grande parte do custo de desenvolvimento de *software* é diretamente relacionado ao processo de testes e depuração. Redução no custo envolvido no processo de teste é possível com uma melhoria no *framework* de testes através de melhorias na classificação de erros e falhas com novas técnicas. O uso de espectros para classificar ou até mesmo localizar a origem destes problemas é uma técnica promissora.

ESPECTROS DE EXECUÇÃO

Espectros de execução são registros provenientes de um programa durante seu funcionamento. As diferenças entre os espectros indicam modificações no comportamento decorrentes das modificações ocorridas no código entre duas versões. O uso de espectros tem mostrado potencial em classificar erros num programa[1][2]. Comparando espectros de duas versões diferentes de um mesmo programa é também possível localizar a fonte dos desvios de comportamento no código[3].

Analisando as diferenças entre os espectros e as diferenças dos código fonte das duas versões é possível classificar cada diferença do espectro como diretamente relacionada com as mudanças do código. No caso deste diferença de espectro ser devida diretamente a uma modificação legítima de código, tem-se que esta diferença é inserida pelo programador, como *refactoring* ou adição de funcionalidade, no momento da modificação do código. No caso contrário, se a diferença entre os espectros não é causada diretamente por linhas de código modificadas então esta diferença é um efeito colateral introduzido pelo programador.

Efeitos colaterais indicam diferenças na semântica do programa. São diferenças entre

o funcionamento de dois programas que podem causar resultados finais diferentes. Modificações semânticas entre duas versões de um programa são comumente detectadas em testes de regressão convencionais com o uso de asserções sobre os resultados obtidos na execução.

Problema

O foco deste trabalho é apresentar o espectro de *branches* e comparar sua qualidade de classificação com a qualidade de testes de unidade convencionais. A aplicação de espectros se destaca por complementar os métodos tradicionais de teste e depuração. É uma área que apresenta muitos pontos passíveis de contribuição, tanto na melhoria do custo de execução e espaço quanto na qualidade dos espectros usados.

O objetivo deste trabalho é usar os espectros gerados na execução do programa para detectar desvios de comportamento não esperados. Um exemplo seria a substituição do algoritmo de ordenação *bubblesort* pelo algoritmo de ordenação *quicksort*. Neste caso, as modificações no código, se bem feitas, acarretariam somente no aumento da performance, já que, se o estado inicial for o mesmo, o estado final de ordenação é idêntico em ambos algoritmos de ordenação. Num caso ruim de substituição, o resultado da ordenação seria diferente e poderia causar desvios de funcionalidade no restante da execução do programa.

Solução

Este trabalho propõe a instrumentação de *branches* e posterior análise dos espectros gerados pela execução para classificar falhas com qualidade semelhante a obtida com testes de unidade.

O código fonte de duas versões são instrumentados estaticamente e pós-instrumentados para remover os registros que ocorreriam em áreas não relevantes para análise. Após a execução uma sequência de registros em um arquivo representa os espectros que serão comparados. A comparação dos espectros constata se houve erro ou não na evolução de uma versão para outra.

Estrutura do trabalho

Este trabalho organiza-se da seguinte forma:

A concepção da ideia usada é vista no Capítulo 1 juntamente com o porquê da escolha do espectro de *branch* em favor de outros espectros e como fazer a comparação destes espectros com relativa qualidade.

Informações sobre a implementação estão no Capítulo 2. Lá encontra-se detalhes sobre a instrumentação dos programas para gerar o espectro, a forma de compará-los e uma segunda alternativa explorada durante o desenvolvimento do trabalho capaz de localizar o desvio de comportamento no código, abandonada por ser muito custosa tanto em espaço quanto tempo de execução.

Experiências e resultados são encontrados no Capítulo 3. Neste capítulo são descritas a metodologia e métricas de qualidade usadas na experiência. Resultados com detalhes sobre a qualidade do classificador com espectro também estão disponíveis.

Por fim, no Capítulo 4, encontra-se a conclusão deste trabalho juntamente com as limitações da ideia e onde estas se encontram. Uma análise dos trabalhos relacionados também é feita, embasando os resultados correntes e trabalhos futuros para a melhora da classificação.

1. CONCEPÇÃO

O espectro escolhido foi o de *branches* e determinado que a instrumentação seria estática pelas limitações das ferramentas existentes em não instrumentar *branches* nativamente. A instrumentação de *branches* é um conceito simples de entender como será visto.

1.1. O ESPECTRO DE *BRANCHES*

O espectro de *branches* é baseado no registro das expressões condicionais durante a execução do programa. O registro deve manter uma relação entre uma entrada no arquivo contendo o registro de todas as expressões e sua respectiva avaliação condicional no código fonte. Desta forma, deve-se ser capaz de identificar uma entrada no espectro como proveniente de determinada linha de código do programa e que determinada linha de código é responsável pelos eventuais registros no arquivo contendo o espectro.

```
public int max (int x, int y) {
❶     if (x > y) return x;
      else return y;
}
public int min (int a, int b) {
❷     if (x > y) return y;
      else return x;
}
/* calcula distância entre 5 e 3 */
public int distancia (void) {
      int a = 5;
      int b = 3;
      return (max(a,b) - min(a,b));
}
```

Exemplo 1: Identificação única de expressões

No Exemplo 1 temos que ser capazes de registrar o primeiro *branch* como diferente do

segundo, mesmo sendo a expressão lógica utilizada a mesma. Somente desta forma é possível fazer a comparação entre espectros de dois programas, já que uma expressão lógica pode se repetir diversas vezes no código.

Para identificar o fluxo corretamente, para cada *branch* visitado deve-se registrar seu identificador único e a sua validação. Abaixo está representado o espectro gerado pela execução do da função `distancia()` do Exemplo 1:

❶:true; ❷:true

Deve-se ser registrado no espectro os controles de fluxo *if-else* e laços de repetição *while*, *for* e *do-while*. Apesar de chamadas de funções serem consideradas instruções de *branch* não foram consideradas por não caracterizarem estritamente como *branch* condicional. *Switches* também não foram considerados por sua complexidade para instrumentar.

```
public int repeticoes (void) {
      int i;
❶     for (i=0; i<5; i++) {
            /* i é igual a 5 ao fim /*
      }
❷     while (i != 2) {
            /* i é igual a 2 ao fim */
            i--;
      }
}
```

Exemplo 2: Laços de repetição

O Exemplo 2 é uma extensão do Exemplo 1. Neste caso, devem ser registradas todas as decisões tomadas durante a execução. Cada avaliação da expressão do laço é registrada. O espectro toma então a seguinte configuração para a execução da função `repeticoes()`:

❶:true; ❶:true; ❶:true; ❶:true; ❶:true; ❶:false; ❷:true; ❷:true; ❷:true; ❷:false

Em resumo, o espectro de *branches* deve conter todas as decisões tomadas no decorrer da execução do programa onde houve a instrumentação.

1.2. IDENTIFICAÇÃO DE EQUIVALÊNCIA ENTRE *BRANCHES*

O maior problema da precisão do espectro de *branches* é identificar um *branch* de uma versão para outra do programa no caso deste ter passado por modificações de código. O *branch* não executa a mesma tarefa exatamente como na versão anterior, possivelmente se encontra em uma linha de código diferente e em algumas situações pode se encontrar em arquivos diferentes, no caso de *refactoring* mais agressivo do código.

```
public int repeticoes (void) {
❶     while (eval_1()) {
            //tarefa 1
      }
❷     while (eval_2()) {
            //tarefa 2
      }
❸     while (eval_3()) {
            //tarefa 3
      }
      return 0;
}
```

```
public int repeticoes (void) {
①     while (eval_1()) {
            //tarefa 1
      }
②     while (eval_2()) {
            //tarefa 2
            //diferente
      }
③     while (eval_3()) {
            //tarefa 3
      }
      return 0;
}
```

Exemplo 3: Identificação de branches *em duas versões diferentes*

No Exemplo 3, tem duas versões de uma função em um programa qualquer. A expressão condicional permanece a mesma somente com `eval_2()`. As funções `eval_1()` e `eval_3()` sofreram modificações internas no sentido de que o número de repetições dos laços de controle poderá mudar. Os comentários indicam tarefas sem uso de *branches*. Alterações nestas tarefas não se propagam ao espectro.

Os três laços de controle são os mesmos em ambas versões. Não foi inserido nenhum laço de controle entre uma versão e outra. A primeira e a terceira função de avaliação são diferentes. A identificação de seções dos registros como equivalentes se transforma no problema da equivalência de programas.

É de se esperar que os registros de *branch* ❷ e o *branch* ② sejam idênticos nos espectros das duas versões mesmo estes *branches* estando entre dois laços que mudaram seu comportamento e número de repetições. É importante notar que as funções `eval_1()` e `eval_3()` possivelmente devem usar internamente outros *branches* que também serão registrados. Estas modificações nestas funções possivelmente tornarão os espectros totalmente diferentes na parte inicial, pelos registros dos branches ❶ e ①, e no final, pelos registros dos branches ❸ e ③. Neste caso, só é possível a identificação dos *branches* pela sequência em que acontecem no espectro, ou seja, no contexto em que são registrados.

Pela impossibilidade da precisa identificação de equivalência de *branches* a identificação por contexto pode apresentar problemas e nem sempre é satisfatória. Apesar desta limitação, o uso é possível e apresenta bons resultados.

1.3. Equivalência entre espectros

O problema da equivalência entre espectros se traduz em identificar os registros de áreas de código modificadas e registros de áreas não modificadas e separá-los. Ao final, a sequência de registros de áreas não modificadas deve ser igual entre as duas versões para classificarmos a execução das versões como equivalentes.

O objetivo é classificar se duas versões de um mesmo programa apresentam o mesmo comportamento em áreas não modificadas. Se um comportamento anômalo é inserido nas alterações de código, um efeito colateral provavelmente deve se propagar a uma parte do programa que não foi modificada. Esta modificação de comportamento pode ser um erro ou correção de erro. Se nenhum efeito colateral se propaga para fora da área modificada do programa, a secção de código alterada é equivalente entre as duas versões do programa.

A exemplo dos testes de unidade convencionais, para fazer esta comparação o estado inicial dos dois programas deve ser o mesmo. As duas execuções partem do mesmo ponto e devem chegar ao mesmo ponto se não houver modificações semânticas entre as versões.

As áreas de código modificadas fazem o fluxo do programa diferir nos momentos que as instruções desta área são executadas. A comparação dos espectros identifica os registros provenientes de áreas de código modificado e os remove do espectro. A remoção é justificada e desejada pois as diferenças entre estes registros são obviamente herdadas das modificações de código.

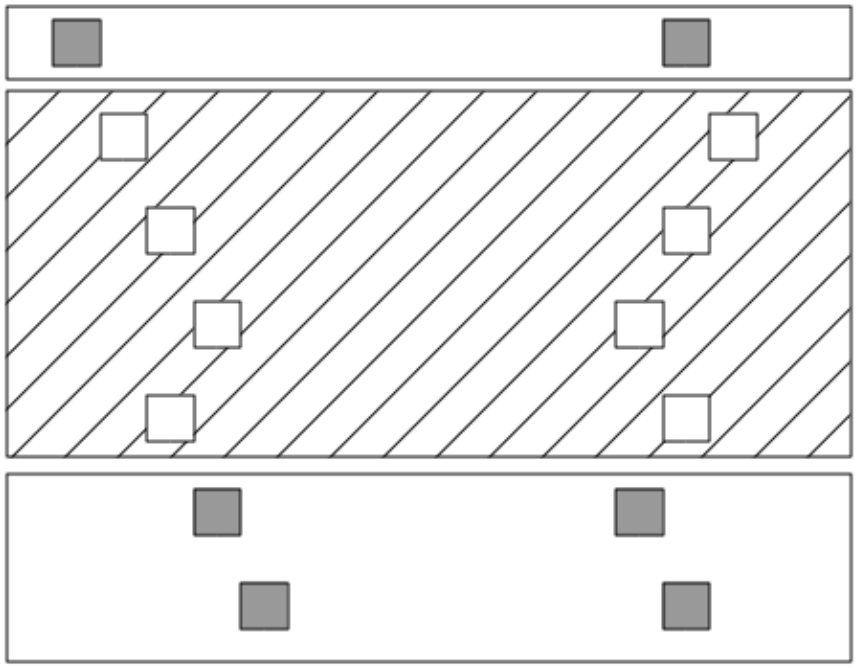

Ilustração 1: Sequências de branches *diferentes*

Na Ilustração 1, temos duas versões de um programa e as diferenças do espectro de *branches* durante a execução de ambos. A execução se dá de cima a baixo. Os blocos representam entradas no espectro. Blocos cinzas são blocos de códigos não modificados, blocos brancos são provenientes de código modificado. Deslocamentos para direita são devido a validações *true* e deslocamentos a esquerda são devidos a validações *false*. Toda a área hachurada foi detectada como diferença do espectro.

Como todas as diferenças dentro do bloco hachurado são devido a seções de código modificadas, as divergências de comportamento dentro do bloco são esperadas. Não foram inseridas modificações que se propagaram para outras áreas do programa. Neste caso temos então que de uma versão para outra não foram inseridos erros ou falhas.

Um exemplo que se adequa a Ilustração 1 é a substituição de algoritmos de ordenação. O início dos espectros registrados é o mesmo nos dois programas, seguem diferenças devido às modificações nos algoritmos de ordenação e, por fim, o restante do espectro segue idêntico se a troca de algoritmos foi feita de forma correta.

Em resumo, se após a remoção dos registros provenientes das áreas de código modificado os dois espectros forem diferentes, é porque uma área de código não modificado sofreu influências do código modificado apresentando um efeito não desejado no funcionamento do programa. No exemplo, se a troca de algoritmos de ordenação foi mal feita, a ordem dos elementos após a ordenação estará diferente entre as duas versões. O resto da execução dos programas, que usam estes elementos ordenados, se comportaria então de forma diferente devido à falha inserida.

2. IMPLEMENTAÇÃO

Este projeto foi implementado para a linguagem Java mas poderia ser aplicado a outras linguagens com poucas alterações. Somente são registrados o controlador de fluxo *if-else* e os lações *while*, *for* e *do-while*. O controlador de fluxo *switch* e o laço *foreach* não foram considerados no projeto. Em Java, o uso de exceções, herança, *overload* de métodos e alguns outros recursos tem no seu funcionamento interno o uso de *branches*. O registro destes *branches* foi abandonado em favor de simplificar a instrumentação.

O projeto se divide em dois módulos. Um instrumentador de código, responsável por registrar todos os *branches* em consideração, e um comparador, responsável por classificar se dois espectros são equivalentes dado duas versões de um programa.

No projeto, a única ligação com a linguagem Java se encontra no instrumentador. Um código instrumentado, mesmo em uma linguagem de programação arbitrária, pode se beneficiar do mesmo comparador com a condição que seu espectro gerado obviamente mantenha o formato compatível.

Na realidade, no desenvolver da ideia, o algoritmo do comparador degenerou para um simples comando `diff` entre os espectros. O comparador original e a nova abordagem de comparação serão explicados no decorrer deste capítulo com suas respectivas vantagens.

2.1. INSTRUMENTAÇÃO ESTÁTICA

A instrumentação escolhida para uso foi estática. Na instrumentação estática o código é alterado antes da compilação. O código das duas versões passa por um *parser* que adiciona um pedido de registro para cada *branch* visitado.

Cada *branch* tem dois registros possíveis: *true* e *false*. O sistema deve garantir que no caso de um *branch* não ter sua expressão condicional avaliada com *true*, o registro deste *branch* deve ser feito como *false*. O instrumentador então deve adicionar pedidos de registro

considerando duas possibilidades para cada controle de fluxo ou laço de execução. O Exemplo 4 mostra uma secção de código antes da instrumentação.

```
public int comparador (int x, int y) {
❶     if (x = y) return 0;
❷     else if (x > y) return 1;
      else return –1;
}
```

Exemplo 4: Código com if-else não instrumentado

O Exemplo 4 usa dois *branches* em uma função usada para comparação. A instrumentação desta secção resulta no código do Exemplo 5. É importante notar que cada *if* deve ser complementado com seu *else*.

```
public int comparador (int x, int y) {
      if (x = y) {
            registrar(❶,true);
            return 0;
      } else {
            registrar(❶,false);
            if (x > y) {
                  registrar(❷,true);
                  return 1;
            else {
                  registrar(❷,false);
                  return –1;
            }
      }
}
```

Exemplo 5: Código com if-else instrumentado

Os laços de repetição são tratados de forma similar. Os laços *for* e *while* têm dentro de seu bloco de execução o pedido de registro da expressão condicional validada como *true*.

Após a execução do laço, o instrumentador deve inserir um pedido de registro com a expressão validada como *false*. É fácil ver que, para sair do laço de repetição, a expressão foi validada como *false*. O Exemplo 6 mostra o Exemplo 2 instrumentado.

```
public int repeticoes (void) {
      int i;
      for (i=0; i<5; i++) {
            registrar(❶,true);
            /* i é igual a 5 ao fim /*
      }
      registrar(❶,false);
      while (i != 2) {
            registrar(❷,true);
            /* i é igual a 2 ao fim */
            i--;
      }
      registrar(❷,false);
}
```

Exemplo 6: Laços de repetição instrumentados

No laço de repetição *do-while* deve-se somente considerar que a execução do seu corpo pela primeira vez não deve ser registrado.

```
/* gasta ciclos ( x deve ser maior que 1 ) */
public int esperar (int x) {
      int i = 0;
      do {
            i++;
❶     } while (i < x);
}
```

Exemplo 7: Código com do-while não instrumentado

A instrumentação do laço *do-while* é feita similarmente aos laços *for* e *while*. O registro continua no bloco do laço, mas a primeira execução do corpo é checada através do uso de

uma variável auxiliar gerada durante a instrumentação. O Exemplo 8 mostra o uso de tal variável e o código instrumentado gerado.

```
/* gasta ciclos ( x deve ser maior que 1 ) */
public int esperar (int x) {
      int i = 0;
      boolean auxiliar_8976238 = false;
      do {
            if ( auxiliar_8976238 ) {
                  registrar(❶,true);
            }
            i++;
            auxiliar_8976238 = true;
      } while (i < x);
      registrar(❶,false);
}
```

Exemplo 8: Código com do-while instrumentado

A declaração da variável `auxiliar_8976238` é gerada durante a instrumentação. É importante notar que o *if* que avalia a variável não é registrado já que não faz parte do código original.

Um detalhe notável é que nem sempre um laço de repetição registrará sua saída. No caso de haver uma exceção ou retorno da função no corpo do laço, este é interrompido e não mais executa registros no arquivo de espectro.

2.2. AJUSTES APÓS A INSTRUMENTAÇÃO

A instrumentação leva em conta somente o código de uma versão. Como ela não se utiliza das diferenças entre as duas versões, o código é integralmente instrumentado. A execução deste código instrumentado resultará num espectro com registros de código modificado e de código não modificado.

Na apresentação da concepção do algoritmo, foi exposto que a única parte relevante do espectro é proveniente de código não modificado entre as duas versões. É preciso então um passo para remover as entradas de instrumentação de código modificado.

Isto foi obtido comparando os códigos instrumentados e gerando dois *patches*. Os *patches* são então processados para remover registros de *branchs* que não foram identificados como os mesmos de uma versão para a outra e que portanto não podem ser comparados. A seguir os *patches* são aplicados removendo os registros de áreas modificadas em ambas versões. Como será visto, a remoção dos registros de código modificado tornará muito mais rápida e eficiente a comparação dos espectros.

2.3. Registros no espectro

Há duas abordagens para o registro do espectro. A primeira envolve registrar todas as entradas não provenientes de áreas de código modificadas em um arquivo. A segunda utiliza as mesmas entradas para gerar um *hash* da execução.

A primeira abordagem possibilita encontrar onde a primeira divergência do espectro ocorreu no código do programa estudado. Esta característica dá ao espectro a possibilidade de ser usado também para localizar as falhas inseridas no código. O fato de cada ocorrência de um *branch* ser registrado torna o arquivo contendo o espectro e o tempo de execução muito grandes.

A segunda abordagem aumenta em muito a velocidade de processamento. A comparação é realizada simplesmente constatando se o *hash* de um espectro é igual ao do outro.

Para execução das experiências contidas neste trabalho, a segunda abordagem foi escolhida. Este trabalho tem como objetivo mostrar que o classificador e o espectro propostos tem qualidade similar ao uso de testes de unidade convencionais. O uso da primeira abordagem não contribui em melhoras na classificação e reduz a eficiência do tempo de execução.

2.4. COMPARAÇÃO

Se o passo de pós-instrumentação foi seguido, a comparação dos espectros é feita simplesmente observando se os dois arquivos contendo os registros dos *branches* são idênticos.

No caso da pós-instrumentação não ter sido executada, o comparador terá que checar se cada divergência dos espectros é iniciada por áreas de código modificado. No caso afirmativo, o desvio de comportamento é esperado e o comparador deve seguir para a próxima modificação. Se todas as divergências do espectro serem provenientes de código modificado o comportamento dos programas são equivalentes. Se uma única divergência seja de proveniente de uma área não modificada, uma modificação semântica aconteceu originando uma falha.

A comparação da primeira abordagem ainda pode dizer onde ocorreu cada divergência de comportamento nos espectros e ajudar a localizar o local onde foi inserida a falha, já que as entradas divergentes no registro podem ser usadas de forma a identificar no código fonte suas origens. A abundância de entradas nos registros do espectro torna esta comparação mais lenta.

A comparação na segunda abordagem somente relata se os espectros são equivalentes ou não. É feita uma simples comparação entre os dois *hashs* de execução registrados. Esta metodologia é obviamente muito mais eficiente.

3. EXPERIMENTAÇÃO

Para avaliar a qualidade do espectro de *branches*, foi usado o Jtopas disponível no repositório SIR[5]. O Jtopas é uma biblioteca usada para realizar *parsing* de texto. O pacote disponível no SIR possui quatro versões originais, sequencialmente tratadas aqui como *A00*, *B00*, *C00* e *D00*, e suas respectivas *test suites*, tratadas aqui como *A*, *B*, *C* e *D*. Além destas versões, existe a possibilidade de aplicação de falhas nas versões originais.

A versão A00 não tem falhas injetáveis, *B00*, *C00* e *D00* possuem respectivamente 10, 12 e 16 falhas possíveis. As versões com falhas são tratadas como *Vnn*, onde *V = {A, B, C, D}* e *nn* é o número da possível falha. A versão *D02* foi removida por não ser compilável, e a versão *D10*, somente quando usada junto com a *test suite D*, foi removida por apresentar um loop infinito.

As *test suites A*, *B*, *C* e *D* possuem 95, 126, 128 e 209 testes, respectivamente. Dois testes, `testParallelModification` e `testParallelParsing`, ambos na suíte *D* fazem uso de threads e impossibilitam o uso do espectro. Estes dois testes foram removidos das análises a seguir.

3.1. METODOLOGIA

A experiência baseia-se em mostrar a qualidade da detecção de erros usando o espectro em comparação com os testes de unidade fornecidos. Dois procedimentos foram utilizados: o uso da *test suite V* para comparar os espectros de *V00* com os espectros de *(V+1)nn* e o uso da *test suite V* para comparar os espectros de *V00* com os espectros de *Vnn*. As versões *A00*, *B00*, *C00* e *D00* todas executam sem erros nas suas respectivas *test suites*.

As *test suites* nem sempre detectam as falhas. Há casos de nenhum problema ser detectado em várias versões. Das 73 versões com faltas injetadas os testes convencionais fornecidos detectam somente 35 delas com problemas. São 15 versões com problemas para o primeiro procedimento e 20, para o segundo.

No experimento foram realizados no total 10076 testes de unidade. No primeira procedimento, são realizados 4382 testes que detectam no total 5 falhas e 119 erros. No segundo procedimento, 5694 testes detectam no total 50 falhas e 41 erros.

O classificador reporta somente se há um desvio de comportamento (D) ou se há conformidade dos espectros (C). Os testes de unidade classificam sucesso (S), falha (F) ou erro (E). Os resultados são cruzados gerando as seguintes classificações:

DS – desvio nos espectros e sucesso nos testes de unidade,

CS – conformidade dos espectros e sucesso nos testes de unidade,

DF – desvio nos espectros e falha nos testes de unidade,

CF – conformidade dos espectros e falha nos testes de unidade,

DE – desvio nos espectros e erro nos testes de unidade,

CE – conformidade dos espectros e erro nos testes de unidade.

3.2. RESULTADOS

Procedimento	**Classificação**	**Número de testes**
V00 com *(V+1)nn*	DS	**80**
	CS	4179
	DF	4
	CF	0
	DE	119
	CE	0
V00 com *Vnn*	DS	**189**
	CS	5414
	DF	45
	CF	**5**
	DE	41
	CE	0

Tabela 1: Resultados da classificação

A Tabela 1 contém os resultados das classificações cruzadas para os dois procedimentos citados. Deve-se notar que as falhas não detectadas são poucas (somente 5 no segundo procedimento) e as muitas falhas detectadas com desvios do espectro não detectadas pelos tes-

tes de unidade convencionais (DS). Mais detalhes são encontrados no Apêndice A.

Na Tabela 2, pode-se ver que o classificador utilizando o espectro proposto é capaz de classificar 97,66% das falhas e erros detectados pelos teste de unidade convencional. Somente 2,34% de todas as falhas e erros não foram detectados, devido às 5 classificações CF no segundo procedimento. As cinco falhas não detectadas são provenientes dos testes `testNestedExceptions` e `testExceptionList`. As discrepâncias ocorrem provavelmente devido às limitações da instrumentação, que não leva em consideração o registro das exceções, como visto nos detalhes da implementação.

Procedimento	**Falhas/erros detectados**	**Falhas/erros não detectados**
V00 com *(V+1)nn*	100,00%	0,00%
V00 com *Vnn*	94,50%	5,50%
Todos os testes	97,66%	2,34%

Tabela 2: Porcentagens de falhas detectadas

A relação de conformidade dos espectros e sucesso nos testes, calculada a como a razão do número de classificações CS com a soma de DS e CS (total de testes de unidade com sucesso) é de 97,27%.

A porcentagem de falha/erros encontrados e a relação de conformidade e sucesso mostram que o classificador e o espectro proposto tem qualidade semelhante aos testes de unidade convencionais. O sistema proposto também classifica 269 desvios que os testes de unidade não detectam como falha ou erro. Estes desvios possivelmente classificam falhas que as asserções do teste de unidade não foram capazes de detectar[2].

O custo em espaço para salvar o *hash* da execução do espectro é mínimo, em média 7,26 *bytes* por teste. O tempo de execução de uma *test suite* sobre um código instrumentado levou em média 5,17 segundos para executar com desvio padrão de 1,96 e variância 3,85. Sobre o código original, a *test suite* leva em média 2,2 segundos para executar com desvio padrão de 1,6 e variância 1,1.

4. CONCLUSÕES

O uso do espectro de *branches* é promissor. Sua capacidade para classificar a inserção de erros entre versões é similar aos testes de unidade convencionais. É possível usar como complemento aos testes de unidade convencional ou como auxiliar na seleção durante a geração automática de testes. O uso isolado desta técnica como um *oracle* automático pode ser levado em consideração em alguns casos, mas precisa de mais estudos para ser considerado realmente eficiente.

4.1. LIMITAÇÕES

A principal limitação do uso de espectros é a incapacidade de lidar com concorrência e processamento paralelo. O registro do espectro depende de uma sequência determinística de *branches*, caso contrário, os registros não seriam comparáveis.

O espectro como proposto aqui pode falhar em alguns exemplos de *refactoring*. O Exemplo 9 é uma possível fonte de problema.

```
public boolean troca (void) {        public boolean troca (void) {
❶     if ( eh_velho() ) {            ①     if ( quebrado() ) {
❷           if (quebrado()) {        ②           if (eh_velho()) {
                  return true;                        return true;
            }                                     }
      }                                     }
      return false;                         return false;
}                                     }
```

Exemplo 9: Identificação de branches *em duas versões diferentes*

Os espectro gerados têm desprezado de suas entradas os registros de ❶, ❷, ①, e ②. Já que estes fazem parte de linhas de código modificadas. Mas o interior dos métodos `quebra-`

`do()` e `eh_velho()`, pode possuir *branches* próprios que seriam registrados normalmente no espectro. A comparação destes espectros relataria diferenças semânticas já que os registros dos *branches* de `quebrado()` e `eh_velho()` não são provenientes de código modificado e são certamente diferentes.

4.2. Trabalhos relacionados

Orso et al.[1] propõem o uso de espectros para identificar diferenças de comportamento entre duas versões de um programa através de análise dinâmica. Sua abordagem consiste na geração automática de testes explorando a área de código modificada entre as duas versões e reportando as diferenças de comportamento encontradas aos desenvolvedores. Seus resultados apontam que 60% dos testes automaticamente gerados acusam desvios de comportamento.

Harrold et al.[2] comparam diversos espectros e relações de pertinência entre estes. Entre os espectros estudados são citados dois contidos no espectro aqui utilizado: *branch-hit spectrum* e *branch-count spectrum*. Nas suas análises o *branch-count spectrum* demonstrou diferenças em todos os testes que expunham um falha de regressão. A imprecisão do espectro *branch-count spectrum* ficou em torno de 7%. Na experiência relatada aqui, a imprecisão foi de 2,34%.

Xie et al.[3] utilizam um espectro diferente do aqui descrito e foca o trabalho na direção da localização da origem do problema. Apesar deste trabalho de graduação não seguir este caminho, o espectro aqui utilizado pode ser utilizado também para este fim.

Reps et al.[4] utilizam o espectro mais parecido com o aqui proposto. Sua abordagem é entretanto oposta. Em vez de comparar duas versões com as mesmas entradas e localizar diferenças de comportamento, ele aplica duas entradas na mesma versão para localizar problemas de comportamento em relação ao Problema do Ano 2000.

4.3. Trabalhos futuros

Uma instrumentação mais precisa, levando em contra exceções, *overload* de métodos e

outras situações, que internamente fazem uso de *branches* mas que não foram consideradas, seria uma forma de possivelmente melhorar a robustez do classificador.

Um estudo do classificador na localização das falhas também é uma possibilidade[3]. Isto pode tanto ser feito com o registro de todos os *branches* como explicado na primeira abordagem conceitual, quanto com o uso de uma máquina virtual para executar os dois programas simultaneamente de forma sincronizada.

Uma possível solução à limitação ilustrada no Exemplo 9, é desativar o registro durante a execução do método modificado (neste caso o método `troca()`) e reativar o registro quando o método acabar sua execução. Esta abordagem possivelmente pode melhorar o classificador em alguns casos e piorar em outros já nem sempre métodos modificados causariam o problema do Exemplo 9 e entradas preciosas poderiam ser perdidas.

BIBLIOGRAFIA

[1] Alessandro Orso e Tao Xie. BERT: BEhavioral Regression Testing. *Proceedings of the 2008 International Workshop on Dynamic Analysis*. 2008.

[2] Mary Jean Harrold, Gregg Rothermel, Kent Sayre, Rui Wu e Liu Yi. An empirical investigation of the relationship between spectra differences and regression faults. *Journal of Software Testing, Verification and Reliability.* , 10(3):171 – 194, 2000.

[3] Tao Xie e David Notkin. Checking inside the black box: Regression testing by comparing value spectra. *IEEE Trans. Softw. Eng.*, 31(10):869 – 883, 2005.

[4] Thomas Reps, Thomas Ball, Manuvir Das e James Larus. The use of program profiling for software maintenance with applications to the Year 2000 Problem. *Proceedings of ESEC/FSE 97: Lecture notes in Computer Science.* 1997.

[5] "Página do SIR", `http://sir.unl.edu`.

A. Resultados detalhados da classificação

As tabelas a seguir contêm detalhes das classificações. Todas as tabelas têm suas classificações **CS** transformadas em - para realçar as entradas mais interessantes.

As próximas três tabelas tratam do primeiro procedimento da experiência.

Test suite A* comparando *A00* com *Bnn										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	DE	DS	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	DE	DS	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	DE	DS	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	DE	DS	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	DE	DS	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	DE	DS	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	DE	DS	-	-	-	-

Test suite A comparando *A00* com *Bnn*										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
29	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	DE	DS	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	DE	DS	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	DE	DS	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	DE	DS	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	DE	DS	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-
46	-	-	-	-	DE	DS	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	-	-	-	-	-	-
49	-	-	-	-	DE	DS	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	-	-	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-
53	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	-	-	-	-
57	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-
59	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	DE	DS	-	-	-	-
63	-	-	-	-	DE	DS	-	-	-	-
64	-	-	-	-	DE	-	-	-	-	-
65	-	-	-	-	DE	DF	-	-	-	-

Test suite A comparando *A00* com *Bnn*										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
66	-	-	-	-	-	-	-	-	-	-
67	-	-	-	-	-	-	-	-	-	-
68	-	-	-	-	-	-	-	-	-	-
69	-	-	-	-	-	-	-	-	-	-
70	-	-	-	-	-	-	-	-	-	-
71	-	-	-	-	-	-	-	-	-	-
72	-	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-
78	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-
Tempo	4,81s	4,90s	5,41s	5,43s	5,60s	4,81s	8,32s	10,56s	4,87s	7,48s

Test suite B comparando *B00* com *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
1	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-

Test suite B **comparando** *B00* **com** *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-

Test suite B comparando *B00* com *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
41	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-
46	-	-	-	-	-	-	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	-	-	-	-	-	-	-	-
49	-	-	-	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	-	-	-	-	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-	-	-
53	-	-	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	-	-	-	-	-	-
57	-	-	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-	-	-
59	-	-	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	-	-	-	-	-	-	-	-
63	-	-	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	-	-	-	-	-	-	-	-
65	-	-	-	-	-	-	-	-	-	-	-	-
66	-	-	-	-	-	-	-	-	-	-	-	-
67	-	-	-	-	-	-	-	-	-	-	-	-
68	DS	DS	DF	DS	DS	DS	DS	DS	DS	DS	DS	DS
69	-	-	-	-	-	-	-	-	-	-	-	-
70	DS	DS	DE	DS	DS	DS	DS	DS	DS	DS	DS	DS
71	-	-	-	-	-	-	-	-	-	-	-	-
72	-	-	-	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-	-	-

Test suite B comparando *B00* com *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
78	-	-	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-	-	-
96	-	-	-	-	-	-	-	-	-	-	-	-
97	-	-	-	-	-	-	-	-	-	-	-	-
98	-	-	-	-	-	-	-	-	-	-	-	-
99	-	-	-	-	-	-	-	-	-	-	-	-
100	-	-	-	-	-	-	-	-	-	-	-	-
101	-	-	-	-	-	-	-	-	-	-	-	-
102	-	-	-	-	-	-	-	-	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-
104	-	-	-	-	-	-	-	-	-	-	-	-
105	-	-	-	-	-	-	-	-	-	-	-	-
106	-	-	-	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	-	-	-	-	-
108	-	-	-	-	-	-	-	-	-	-	-	-
109	-	-	-	-	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-	-	-	-	-
112	-	-	-	-	-	-	-	-	-	-	-	-
113	-	-	-	-	-	-	-	-	-	-	-	-
114	-	-	-	-	-	-	-	-	-	-	-	-

Test suite B comparando *B00* com *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
115	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
116	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
117	DS	DS	DE	DS	DS	DS	DS	DS	DS	DS	DS	DS
118	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
119	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
120	DS	DS	DE	DS	DS	DS	DS	DS	DS	DS	DS	DS
121	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
122	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
123	DF	-	DE	DS	DS	DS	DS	DS	DS	DS	DS	DS
124	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
125	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
126	DS	DS	DE	DS	DS	DS	DS	DS	DS	DS	DS	DS
Tempo	3,53s	3,58s	3,83s	7,91s	3,14s	3,18s	4,02s	5,03s	3,75s	7,80s	7,43s	5,55s

Test suite C comparando *C00* com *Dnn*															
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D10***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C **comparando** *C00* **com** *Dnn*															
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D10***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
49	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
53	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C comparando *C00* com *Dnn*															
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D10***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
59	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
63	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
65	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
66	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
67	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
69	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
70	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
71	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
72	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
78	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C **comparando** *C00* **com** *Dnn*															
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D10***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
96	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
97	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
98	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
99	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
101	-	-	DF	-	-	-	-	-	-	-	-	-	-	-	-
102	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
105	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
106	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
108	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
109	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
114	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
115	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
116	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
117	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
118	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
119	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
121	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
122	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
123	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
124	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
125	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
126	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
127	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
128	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tempo	3,25s	7,95s	5,39s	4,29s	4,84s	5,97s	5,70s	6,33s	6,59s	6,40s	6,40s	2,96s	5,72s	4,84s	4,57s

As três tabelas restantes tratam do segundo procedimento da experiência.

Test suite B* comparando *B00* com *B01										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
1	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	DE	DS	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	DE	DS	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	DE	DS	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	DE	DS	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	DE	DS	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	DE	DS	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	DE	DS	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	DE	DS	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-

Test suite B comparando *B00* com *B01*										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
35	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	DE	DS	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	DE	DS	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	DE	DS	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	DE	DS	-	-	-	-
46	-	-	-	-	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	DE	DS	-	-	-	-
49	-	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	DE	DS	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-
53	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	-	-	-	-
57	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-
59	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	-	-	-	-	-	-
63	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	DE	DS	-	-	-	-
65	-	-	-	-	DE	DS	-	-	-	-
66	-	-	-	-	DE	-	-	-	-	-
67	-	-	-	-	DE	DF	-	-	-	-
68	-	-	-	-	-	-	-	-	DE	-
69	-	-	-	-	-	-	-	-	-	CF
70	-	-	-	-	-	-	-	-	-	-
71	-	-	-	-	-	-	-	-	-	-

Test suite B **comparando** *B00* **com** *B01*										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
72	-	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-
78	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-
96	-	-	-	-	-	-	-	-	-	-
97	-	-	-	-	-	-	-	-	-	-
98	-	-	-	-	-	-	-	-	-	-
99	-	-	-	-	-	-	-	-	-	-
100	-	-	-	-	-	-	-	-	-	-
101	-	-	-	-	-	DS	-	-	-	-
102	-	-	-	-	-	DS	-	-	-	-
103	-	-	-	-	-	DS	-	-	-	-
104	-	-	-	-	-	DS	-	-	-	-
105	-	-	-	-	-	DS	-	-	-	-
106	-	-	-	-	-	DS	-	-	-	-
107	-	-	-	-	-	DS	-	-	-	-
108	-	-	-	-	-	DS	-	-	-	-

Test suite B comparando *B00* com *B01*										
Teste	***B01***	***B02***	***B03***	***B04***	***B05***	***B06***	***B07***	***B08***	***B09***	***B10***
109	-	-	-	-	-	DS	-	-	-	-
110	-	-	-	-	-	DS	-	-	-	-
111	-	-	-	-	-	DS	-	-	-	-
112	-	-	-	-	-	DS	-	-	-	-
113	-	-	-	-	-	DS	-	-	-	-
114	-	-	-	-	-	DS	-	-	-	-
115	-	-	-	-	-	-	-	-	-	-
116	-	-	-	-	-	-	-	-	-	-
117	-	-	-	-	-	-	-	-	-	-
118	-	-	-	-	-	-	-	-	DE	-
119	-	-	-	-	-	-	-	-	-	CF
120	-	-	-	-	-	-	-	-	DF	DF
121	-	DF	DE	-	-	-	-	-	-	-
122	CF	CF	-	-	-	-	-	-	-	-
123	DF	DE	DF	-	-	-	-	-	-	-
124	-	-	-	-	-	-	-	-	-	-
125	-	-	-	-	-	-	-	-	-	-
126	-	-	-	-	-	-	-	-	-	-
Tempo	3,42s	4,11s	3,52s	5,12s	6,12s	4,01s	3,71s	8,65s	3,08s	4,87s

Test suite C comparando *C00* com *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
1	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C **comparando** *C00* **com** *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
16	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-
46	-	-	-	-	-	-	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	-	-	-	-	-	-	-	-
49	-	-	-	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	-	-	-	-	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C **comparando** *C00* **com** *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
53	-	-	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	DE	-	-	DF	DE	DF
57	-	-	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-	-	-
59	-	-	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	-	-	-	-	-	-	-	-
63	-	-	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	-	-	-	-	-	-	-	-
65	-	-	-	-	-	-	-	-	-	-	-	-
66	-	-	-	-	-	-	-	-	-	-	-	-
67	-	-	-	-	-	-	-	-	-	-	-	-
68	-	-	-	-	-	-	-	-	-	-	-	-
69	-	-	-	-	-	-	-	-	-	-	-	-
70	-	-	DF	-	-	-	-	-	-	-	-	-
71	-	-	-	-	-	-	-	-	-	-	-	-
72	-	-	DE	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-	-	-
78	-	-	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-	-	-

Test suite C **comparando** *C00* **com** *Cnn*												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
90	-	-	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-	-	-
96	-	-	-	-	-	-	-	-	-	-	-	-
97	-	-	-	-	-	-	-	-	-	-	-	-
98	-	-	-	-	-	-	-	-	-	-	-	-
99	-	-	-	-	-	-	-	-	-	-	-	-
100	-	-	-	-	-	-	-	-	-	-	-	-
101	-	-	-	-	-	-	-	-	-	-	-	-
102	-	-	-	-	-	-	-	-	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-
104	-	-	-	-	-	-	-	-	-	-	-	-
105	-	-	-	-	-	-	-	-	-	-	-	-
106	-	-	-	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	-	-	-	-	-
108	-	-	-	-	-	-	-	-	-	-	-	-
109	-	-	-	-	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-	-	-	-	-
112	-	-	-	-	-	-	-	-	-	-	-	-
113	-	-	-	-	-	-	-	-	-	-	-	-
114	-	-	-	-	-	-	-	-	-	-	-	-
115	-	-	-	-	-	-	-	-	-	-	-	-
116	-	-	-	-	-	-	-	-	-	-	-	-
117	-	-	DF	-	-	-	-	-	-	-	-	-
118	-	-	-	-	-	-	-	-	-	-	-	-
119	-	-	DE	-	-	-	-	-	-	-	-	-
120	-	-	DF	-	-	-	-	-	-	-	-	-
121	-	-	-	-	-	-	-	-	-	-	-	-
122	-	-	DE	-	-	-	-	-	-	-	-	-
123	-	-	DF	-	-	-	-	-	-	-	-	-
124	CF	-	-	-	-	-	-	-	-	-	-	-
125	DF	DS	DE	-	-	-	-	-	-	-	-	-
126	-	-	DF	-	-	-	-	-	-	-	-	-

Test suite C* comparando *C00* com *Cnn												
Teste	***C01***	***C02***	***C03***	***C04***	***C05***	***C06***	***C07***	***C08***	***C09***	***C10***	***C11***	***C12***
127	-	-	-	-	-	-	-	-	-	-	-	-
128	-	-	DE	-	-	-	-	-	-	-	-	-
Tempo	6,92s	6,99s	3,03s	8,39s	3,79s	4,98s	5,12s	4,29s	3,45s	3,95s	3,47s	8,75s

Test suite D* comparando *D00* com *Dnn														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-
31	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite D **comparando** *D00* **com** *Dnn*														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
32	-	-	-	-	-	-	-	-	-	-	-	-	-	-
33	-	-	-	-	-	-	-	-	-	-	-	-	-	-
34	-	-	-	-	-	-	-	-	-	-	-	-	-	-
35	-	-	-	-	-	-	-	-	-	-	-	-	-	-
36	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-
38	-	-	-	-	-	-	-	-	-	-	-	-	-	-
39	-	-	-	-	-	-	-	-	-	-	-	-	-	-
40	-	-	-	-	-	-	-	-	-	-	-	-	-	-
41	-	-	-	-	-	-	-	-	-	-	-	-	-	-
42	-	-	-	-	-	-	-	-	-	-	-	-	-	-
43	-	-	-	-	-	-	-	-	-	-	-	-	-	-
44	-	-	-	-	-	-	-	-	-	-	-	-	-	-
45	-	-	-	-	-	-	-	-	-	-	-	-	-	-
46	-	-	-	-	-	-	-	-	-	-	-	-	-	-
47	-	-	-	-	-	-	-	-	-	-	-	-	-	-
48	-	-	-	-	-	-	-	-	-	-	-	-	-	-
49	-	-	-	-	-	-	-	-	-	-	-	-	-	-
50	-	-	-	-	-	-	-	-	-	-	-	-	-	-
51	-	-	-	-	-	-	-	-	-	-	-	-	-	-
52	-	-	-	-	-	-	-	-	-	-	-	-	-	-
53	-	-	-	-	-	-	-	-	-	-	-	-	-	-
54	-	-	-	-	-	-	-	-	-	-	-	-	-	-
55	-	-	-	-	-	-	-	-	-	-	-	-	-	-
56	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	-	-	-	-	-	-	-	-	-	-	-	-	-	-
58	-	-	-	-	-	-	-	-	-	-	-	-	-	-
59	-	-	-	-	-	-	-	-	-	-	-	-	-	-
60	-	-	-	-	-	-	-	-	-	-	-	-	-	-
61	-	-	-	-	-	-	-	-	-	-	-	-	-	-
62	-	-	-	-	-	-	-	-	-	-	-	-	-	-
63	-	-	-	-	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	-	-	-	-	-	-	-	-	-	-
65	-	-	-	-	-	-	-	-	-	-	-	-	-	-
66	-	-	-	-	-	-	-	-	-	-	-	-	-	-
67	-	-	-	-	-	-	-	-	-	-	-	-	-	-
68	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite D **comparando** *D00* **com** *Dnn*														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
69	-	-	-	-	-	-	-	-	-	-	-	-	-	-
70	-	-	-	-	-	-	-	-	-	-	-	-	-	-
71	-	-	-	-	-	-	-	-	-	-	-	-	-	-
72	-	-	-	-	-	-	-	-	-	-	-	-	-	-
73	-	-	-	-	-	-	-	-	-	-	-	-	-	-
74	-	-	-	-	-	-	-	-	-	-	-	-	-	-
75	-	-	-	-	-	-	-	-	-	-	-	-	-	-
76	-	-	-	-	-	-	-	-	-	-	-	-	-	-
77	-	-	-	-	-	-	-	-	-	-	-	-	-	-
78	-	-	-	-	-	-	-	-	-	-	-	-	-	-
79	-	-	-	-	-	-	-	-	-	-	-	-	-	-
80	-	-	-	-	-	-	-	-	-	-	-	-	-	-
81	-	-	-	-	-	-	-	-	-	-	-	-	-	-
82	-	-	-	-	-	-	-	-	-	-	-	-	-	-
83	-	-	-	-	-	-	-	-	-	-	-	-	-	-
84	-	-	-	-	-	-	-	-	-	-	-	-	-	-
85	-	-	-	-	-	-	-	-	-	-	-	-	-	-
86	-	-	-	-	-	-	-	-	-	-	-	-	-	-
87	-	-	-	-	-	-	-	-	-	-	-	-	-	-
88	-	-	-	-	-	-	-	-	-	-	-	-	-	-
89	-	-	-	-	-	-	-	-	-	-	-	-	-	-
90	-	-	-	-	-	-	-	-	-	-	-	-	-	-
91	-	-	-	-	-	-	-	-	-	-	-	-	-	-
92	-	-	-	-	-	-	-	-	-	-	-	-	-	-
93	-	-	-	-	-	-	-	-	-	-	-	-	-	-
94	-	-	-	-	-	-	-	-	-	-	-	-	-	-
95	-	-	-	-	-	-	-	-	-	-	-	-	-	-
96	-	-	-	-	-	-	-	-	-	-	-	-	-	-
97	-	-	-	-	-	-	-	-	-	-	-	-	-	-
98	-	-	-	-	-	-	-	-	-	-	-	-	-	-
99	-	-	-	-	-	-	-	-	-	-	-	-	-	-
100	-	-	-	-	-	-	-	-	-	-	-	-	-	-
101	-	-	DF	-	-	-	-	-	-	-	-	-	-	-
102	-	-	-	-	-	-	-	-	-	-	-	-	-	-
103	-	-	-	-	-	-	-	-	-	-	-	-	-	-
104	-	-	-	-	-	-	-	-	-	-	-	-	-	-
105	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Test suite D **comparando** *D00* **com** *Dnn*														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
106	-	-	-	-	-	-	-	-	-	-	-	-	-	-
107	-	-	-	-	-	-	-	-	-	-	-	-	-	-
108	-	-	-	-	-	-	-	-	-	-	-	-	-	-
109	-	-	-	-	-	-	-	-	-	-	-	-	-	-
110	-	-	-	-	-	-	-	-	-	-	-	-	-	-
111	-	-	-	-	-	-	-	-	-	-	-	-	-	-
112	-	-	-	-	-	-	-	-	-	-	-	-	-	-
113	-	-	-	-	-	-	-	-	-	-	-	-	-	-
114	-	-	-	-	-	-	-	-	-	-	-	-	-	-
115	-	-	-	-	-	-	-	-	-	-	-	-	-	-
116	-	-	-	-	-	-	-	-	-	-	-	-	-	-
117	-	-	-	-	-	-	-	-	-	-	-	-	-	-
118	-	-	-	-	-	-	-	-	-	-	DS	-	-	-
119	-	-	-	-	-	-	-	-	-	-	-	-	-	DF
120	-	-	-	-	-	-	-	-	-	-	-	-	-	DF
121	-	-	-	-	-	-	DE	-	-	-	DS	-	-	-
122	-	-	-	-	-	DS	-	DF	-	-	DS	-	-	-
123	-	-	-	-	-	DS	DE	-	-	-	DS	-	-	-
124	-	-	-	-	-	DF	-	-	-	-	DS	-	-	-
125	-	-	-	-	-	DF	-	-	-	-	DS	-	-	-
126	-	-	-	-	-	DF	DE	DF	-	-	DF	-	-	-
127	-	-	-	-	-	-	-	-	-	-	DS	-	-	-
128	-	-	-	-	DF	DF	DE	DF	-	-	DF	-	-	-
129	-	-	-	-	DF	DF	DE	DF	-	-	DF	-	-	-
130	-	-	-	-	-	-	-	-	-	-	DS	-	-	-
131	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
132	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
133	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
134	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
135	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
136	-	-	-	-	-	DF	-	DS	-	-	DS	-	-	-
137	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
138	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
139	-	-	-	-	-	DF	-	DS	-	-	DS	-	-	-
140	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
141	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
142	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-

Test suite D **comparando** *D00* **com** *Dnn*														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
143	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
144	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
145	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
146	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
147	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
148	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
149	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
150	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
151	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
152	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
153	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
154	-	-	-	-	-	DF	-	DS	-	-	DS	-	-	-
155	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
156	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
157	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
158	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
159	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
160	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
161	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
162	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
163	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
164	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
165	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
166	-	-	-	-	-	DF	-	DS	-	-	DS	-	-	-
167	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
168	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
169	-	-	-	-	-	DF	-	DS	-	-	DS	-	-	-
170	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
171	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
172	-	-	-	-	-	DS	-	DS	-	-	DS	-	-	-
173	-	-	-	-	DF	DS	DE	DE	-	-	DS	-	-	-
174	-	-	-	-	DF	DS	-	DS	-	-	DS	-	-	-
175	-	-	-	-	DF	DS	-	DS	-	-	DS	-	-	-
176	-	-	-	-	DF	DS	-	-	-	-	DS	-	-	-
177	-	-	-	-	DF	DS	DE	DS	-	-	DS	-	-	-
178	-	-	-	-	DF	DS	DE	DS	-	-	DS	-	-	-
179	-	-	-	-	DF	DS	DE	DS	-	-	DS	-	-	-

Test suite D* comparando *D00* com *Dnn														
Teste	***D01***	***D03***	***D04***	***D05***	***D06***	***D07***	***D08***	***D09***	***D11***	***D12***	***D13***	***D14***	***D15***	***D16***
180	-	-	-	-	DF	DS	DE	DS	-	-	DS	-	-	-
181	-	-	-	-	DF	DS	DE	DS	-	-	DS	-	-	-
182	-	-	-	-	-	-	-	-	-	-	DS	-	-	-
183	-	-	-	-	-	-	-	-	-	-	DS	-	-	-
184	-	-	-	-	-	-	-	-	-	-	-	-	-	-
185	-	-	-	-	-	-	-	-	-	-	-	-	-	-
186	-	-	-	-	-	-	-	-	-	-	-	-	-	-
187	-	-	-	-	-	-	-	-	-	-	-	-	-	-
188	-	-	-	-	-	-	-	-	-	-	-	-	-	-
189	-	-	-	-	-	-	-	-	-	-	-	-	-	-
190	-	-	-	-	-	-	-	-	-	-	-	-	-	-
191	-	-	-	-	-	-	-	-	-	-	-	-	-	-
192	-	-	-	-	-	-	-	-	-	-	-	-	-	-
193	-	-	-	-	-	-	-	-	-	-	-	-	-	-
194	-	-	-	-	-	-	-	-	-	-	-	-	-	-
195	-	-	-	-	-	-	-	-	-	-	-	-	-	-
196	-	-	-	-	-	-	-	-	-	-	-	-	-	-
197	-	-	-	-	-	-	-	-	-	-	-	-	-	-
198	-	-	-	-	-	-	-	-	-	-	-	-	-	-
199	-	-	-	-	-	-	-	-	-	-	-	-	-	-
200	-	-	-	-	-	-	-	-	-	-	-	-	-	-
201	-	-	-	-	-	-	-	-	-	-	-	-	-	-
202	-	-	-	-	-	-	-	-	-	-	-	-	-	-
203	-	-	-	-	-	-	-	-	-	-	-	-	-	-
204	-	-	-	-	-	-	-	-	-	-	-	-	-	-
205	-	-	-	-	-	-	-	-	-	-	-	-	-	-
206	-	-	-	-	-	-	-	-	-	-	-	-	-	-
207	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Tempo	4,90s	12,0s	6,15s	6,92s	6,92s	5,44s	5,27s	9,16s	5,17s	5,47s	5,16s	11,0s	7,88s	6,75s