

Avaliação da Etapa 4 - Turma A

Prof. Lucas Mello Schnorr

October 7, 2013

Contents

1	g1 ../../ga/compiladores-etapa4/	2
2	g2 ../../gb/etapa4-gb/	3
3	g3 ../../gc/compiladores/	3
4	g4 ../../gd/compiladores/	4
5	g5	4
6	g6 ../../gf/compilador/	4
7	g7 ../../gg/compiladores/	5
8	g8 ../../gh/compiladores/	5
9	Script utilizado na avaliação automática	6

A avaliação automática verifica o código de saída do compilador e compara este código com o código esperado considerando um erro conhecido na entrada. Esta avaliação automática utiliza um certo número de arquivos escritos em IKS como entrada, cada um com um erro semântico conhecido. Estes arquivos de entrada estão disponíveis no moodle (rubrica Projeto de Compilador: Etapa 4).

Para o relatório completo de cada grupo, veja subseções abaixo.

	Grupo	A	Testes	Encontro	Peso	Total Etapa 4	Comentários
1	A	7.27	1	8.5	1	7.885	<i>Resubmeter solução até 13 outubro 2013, 20:00</i>
2	B	8.48	1	9.5	1	8.99	<i>Resubmeter solução até 13 outubro 2013, 20:00</i>
3	C	6.66	1	8.0	1	7.33	não conseguiu abrir o arquivo output/etapa3/saida.dot <i>Resubmeter solução até 13 outubro 2013, 20:00</i>
4	D	7.87	1	8.0	1	7.935	+ segfault em todos os testes: Programa deve poder ler código para compilar da entrada padrão Exemplo: ./main < input/s00.iks <i>Resubmeter solução até 13 outubro 2013, 20:00</i>
5	E						<i>Submeter solução</i> <i>(política de atraso em vigor)</i>
6	F	2.72	1	5.0	1	3.86	Commits do Eder? <i>Resubmeter solução até 13 outubro 2013, 20:00</i>
7	G	1.51	1	3.0	0.95	2.14225	+ Git? + Código não compila: IKS_INPUT (ao invés de IKS_AST_INPUT) Corrigido manualmente pelo professor <i>Resubmeter solução até 13 outubro 2013, 20:00</i>
8	H	6.06	1	8.0	1	7.03	<i>Resubmeter solução até 13 outubro 2013, 20:00</i>

As subseções abaixo apresentam o relatório por grupo, para cada uma das entradas utilizadas na avaliação automática. A última linha de cada seção apresenta o resultado final da nota automática, seguido pela nota gerada automaticamente por uma avaliação através do valgrind (erros de programação são verificados automaticamente).

A quadrupla (w,r,c,l) significa o seguinte:

- w é a quantidade de escritas inválidas pelo programa considerando a entrada
- r é a quantidade de leituras inválidas pelo programa considerando a entrada
- c é a quantidade de jumps condicionais que se baseiam em valores jamais inicializados
- l é a quantidade de bytes perdidos contabilizados no final da execução

Os comandos para se obter cada um desses valores são os seguintes

```
INVALID_WRITES='cat $TEMP_FILE | grep "Invalid write" | wc -l'
INVALID_READS='cat $TEMP_FILE | grep "Invalid read" | wc -l'
CONDITIONAL='cat $TEMP_FILE | grep "Conditional" | wc -l'
BYTES_LOST='cat $TEMP_FILE | grep "definitely lost:" | awk '{ print $4}''
```

Onde a variável TEMP_FILE aponta para um arquivo com a saída do valgrind, executado da seguinte forma:

```
valgrind $BIN < $INPUT 2>> $TEMP_FILE >> $TEMP_FILE
```

onde BIN é o binário do grupo, INPUT é a entrada.

Nas linhas onde aparece as palavras “causes a segfault” implicam que o programa, para aquela entrada específica, gerou uma falha de segmentação. Isto impossibilita uma verificação automática, e estabelece a nota 0 para o teste em questão. Exemplo:

```
input/s12.iks causes a segfault (exit code is 139)
```

Nas linhas onde aparece as palavras “expects a” e “but we got a” implicam que o programa gerou um código de retorno diferente daquele esperado. Exemplo:

```
input/s11.iks expects a IKS_ERROR_EXCESS_ARGS(10), but we got a IKS_SUCCESS(0) (w,r,c,l)=(11,30,5,348)
```

Isso quer dizer que para a entrada input/s11.iks, o programa gerou um IKS_SUCESSO ao invés de um IKS_ERROR_EXCESS_ARGS. Nestes casos, a nota é 0 para o teste em questão.

Nas linhas onde aparece “okay” implicam que o programa gerou o erro esperado. Elas contribuem para a nota da avaliação automática.

1 g1 ../..//ga/compiladores-etapa4/

```
input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(4,6,3,121)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(7,17,6,324)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(4,6,3,121)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(3,10,2,179)
input/s04.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(3,10,1,138)
input/s05.iks IKS_ERROR_VARIABLE okay (w,r,c,l)=(6,15,3,243)
input/s06.iks IKS_ERROR_VECTOR okay (w,r,c,l)=(7,18,3,283)
input/s07.iks IKS_ERROR_FUNCTION okay (w,r,c,l)=(6,15,3,283)
input/s08.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(6,15,6,490)
input/s09.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(6,15,6,486)
input/s10.iks IKS_ERROR_MISSING_ARGS okay (w,r,c,l)=(8,22,4,340)
input/s11.iks expects a IKS_ERROR_EXCESS_ARGS(10), but we got a IKS_SUCCESS(0) (w,r,c,l)=(11,30,5,348)
input/s12.iks causes a segfault (exit code is 139)
input/s13.iks causes a segfault (exit code is 139)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(4,4,1,80)
input/s15.iks expects a IKS_ERROR_WRONG_PAR_OUTPUT(13), but we got a IKS_SUCCESS(0) (w,r,c,l)=(4,6,7,40)
input/s16.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(4,6,8,120)
input/s17.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(6,15,6,490)
input/s18.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(6,15,6,492)
input/s19.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,31,6,571)
input/s20.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,31,6,565)
input/s21.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(6,15,4,283)
input/s22.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(6,15,4,287)
input/s23.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(8,20,10,323)
input/s24.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(8,20,9,323)
input/s25.iks IKS_SUCCESS okay (w,r,c,l)=(8,26,3,242)
input/s26.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,17,10,163)
input/s27.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(8,19,11,163)
input/s28.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,37,14,486)
input/s29.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,37,13,486)
input/s30.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(9,28,10,403)
input/s31.iks IKS_SUCCESS okay (w,r,c,l)=(11,33,7,486)
input/s32.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(4,10,5,244)
g1 7.2727272727272720 0
```

2 g2 ../../gb/etapa4-gb/

```
input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,0)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,24)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,0)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,0,0)
input/s04.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,0,8)
input/s05.iks IKS_ERROR_VARIABLE okay (w,r,c,l)=(0,0,0,8)
input/s06.iks expects a IKS_ERROR_VECTOR(4), but we got a IKS_ERROR_VARIABLE(3) (w,r,c,l)=(0,0,0,0)
input/s07.iks expects a IKS_ERROR_FUNCTION(5), but we got a IKS_ERROR_VARIABLE(3) (w,r,c,l)=(0,0,0,8)
input/s08.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,16)
input/s09.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,16)
input/s10.iks IKS_ERROR_MISSING_ARGS okay (w,r,c,l)=(0,0,0,8)
input/s11.iks IKS_ERROR_EXCESS_ARGS okay (w,r,c,l)=(0,0,0,24)
input/s12.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(0,0,0,16)
input/s13.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(0,0,0,16)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a (15) (w,r,c,l)=(0,0,0,0)
input/s15.iks IKS_ERROR_WRONG_PAR_OUTPUT okay (w,r,c,l)=(0,0,0,8)
input/s16.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_ERROR_STRING_TO_X(7) (w,r,c,l)=(0,0,0,8)
input/s17.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,16)
input/s18.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,16)
input/s19.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,24)
input/s20.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,24)
input/s21.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,16)
input/s22.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,16)
input/s23.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,8)
input/s24.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,8)
input/s25.iks IKS_SUCCESS okay (w,r,c,l)=(0,0,0,40)
input/s26.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,24)
input/s27.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,32)
input/s28.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,64)
input/s29.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,64)
input/s30.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_ERROR_CHAR_TO_X(8) (w,r,c,l)=(0,0,0,40)
input/s31.iks IKS_SUCCESS okay (w,r,c,l)=(0,0,0,80)
input/s32.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,8)
g2 8.484848484848484840 7.878787878787878780
```

3 g3 ../../gc/compiladores/

```
input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,3,16)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,6,96)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,3,16)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,4,32)
input/s04.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,4,40)
input/s05.iks IKS_ERROR_VARIABLE okay (w,r,c,l)=(0,0,4,112)
input/s06.iks IKS_ERROR_VECTOR okay (w,r,c,l)=(0,0,6,160)
input/s07.iks IKS_ERROR_FUNCTION okay (w,r,c,l)=(0,0,5,136)
input/s08.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,7,128)
input/s09.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,7,128)
input/s10.iks IKS_ERROR_MISSING_ARGS okay (w,r,c,l)=(0,0,5,64)
input/s11.iks IKS_ERROR_EXCESS_ARGS okay (w,r,c,l)=(0,0,5,128)
input/s12.iks expects a IKS_ERROR_WRONG_TYPE_ARGS(11), but we got a IKS_ERROR_EXCESS_ARGS(10) (w,r,c,l)=(0,0,5,128)
input/s13.iks expects a IKS_ERROR_WRONG_TYPE_ARGS(11), but we got a IKS_ERROR_EXCESS_ARGS(10) (w,r,c,l)=(0,0,5,128)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(0,0,2,16)
input/s15.iks IKS_ERROR_WRONG_PAR_OUTPUT okay (w,r,c,l)=(0,0,2,32)
input/s16.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(0,0,1,32)
input/s17.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,7,128)
input/s18.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,8,128)
input/s19.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,7,112)
input/s20.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,7,112)
input/s21.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,4,112)
input/s22.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,5,112)
input/s23.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,5,152)
input/s24.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,5,152)
input/s25.iks IKS_SUCCESS okay (w,r,c,l)=(0,0,3,56)
input/s26.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,3,80)
input/s27.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,3,80)
input/s28.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,7,184)
input/s29.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,7,184)
input/s30.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_SUCCESS(0) (w,r,c,l)=(0,0,4,96)
input/s31.iks expects a IKS_SUCCESS(0), but we got a IKS_ERROR_WRONG_PAR_RETURN(14) (w,r,c,l)=(0,0,1,32)
```

g3 6.6666666666666666660 5.0000000000000000000

4 g4 ../../gd/compiladores/

```

input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,0)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,80)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,0)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,0,40)
input/s04.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(0,0,0,40)
input/s05.iks IKS_ERROR_VARIABLE okay (w,r,c,l)=(0,0,0,40)
input/s06.iks IKS_ERROR_VECTOR okay (w,r,c,l)=(0,0,0,0)
input/s07.iks IKS_ERROR_FUNCTION okay (w,r,c,l)=(0,0,0,0)
input/s08.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,0)
input/s09.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,0)
input/s10.iks IKS_ERROR_MISSING_ARGS okay (w,r,c,l)=(0,0,0,0)
input/s11.iks IKS_ERROR_EXCESS_ARGS okay (w,r,c,l)=(0,0,0,0)
input/s12.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(2,4,0,5)
input/s13.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(2,4,0,6)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(0,0,0,0)
input/s15.iks IKS_ERROR_WRONG_PAR_OUTPUT okay (w,r,c,l)=(2,4,0,5)
input/s16.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(2,4,0,6)
input/s17.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(0,0,0,0)
input/s18.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(0,0,0,0)
input/s19.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,40)
input/s20.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,40)
input/s21.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,40)
input/s22.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(0,0,0,40)
input/s23.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(2,4,0,6)
input/s24.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(2,4,0,5)
input/s25.iks IKS_SUCCESS okay (w,r,c,l)=(0,0,0,4)
input/s26.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(2,4,0,54)
input/s27.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(2,4,0,46)
input/s28.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(2,4,0,6)
input/s29.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(2,4,0,66)
input/s30.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(2,4,0,5)
input/s31.iks IKS_SUCCESS okay (w,r,c,l)=(0,0,0,53)
input/s32.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(0,0,0,0)
g4 7.878787878787878780 6.742424242424242420

```

5 g5

6 g6 ../../gf/compilador/

```

input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(5,3,0,170)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(10,8,1,364)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(5,3,0,170)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(10,3,0,316)
input/s04.iks expects a IKS_ERROR_DECLARED(2), but we got a IKS_SUCCESS(0) (w,r,c,l)=(5,2,0,386)
input/s05.iks expects a IKS_ERROR_VARIABLE(3), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(10,5,1,412)
input/s06.iks expects a IKS_ERROR_VECTOR(4), but we got a IKS_SUCCESS(0) (w,r,c,l)=(11,7,1,460)
input/s07.iks expects a IKS_ERROR_FUNCTION(5), but we got a IKS_SUCCESS(0) (w,r,c,l)=(6,6,1,460)
input/s08.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(12,14,0,443)
input/s09.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(12,14,0,439)
input/s10.iks expects a IKS_ERROR_MISSING_ARGS(9), but we got a IKS_SUCCESS(0) (w,r,c,l)=(16,10,0,560)
input/s11.iks expects a IKS_ERROR_EXCESS_ARGS(10), but we got a IKS_SUCCESS(0) (w,r,c,l)=(16,10,1,632)
input/s12.iks expects a IKS_ERROR_WRONG_TYPE_ARGS(11), but we got a IKS_SUCCESS(0) (w,r,c,l)=(16,10,1,584)
input/s13.iks expects a IKS_ERROR_WRONG_TYPE_ARGS(11), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(16,10,0,488)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(5,1,0,194)
input/s15.iks expects a IKS_ERROR_WRONG_PAR_OUTPUT(13), but we got a IKS_SUCCESS(0) (w,r,c,l)=(5,1,0,242)
input/s16.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(5,1,0,194)
input/s17.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(12,14,0,443)
input/s18.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(12,14,0,443)
input/s19.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,9,0,390)
input/s20.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,9,0,388)
input/s21.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,5,0,340)
input/s22.iks IKS_ERROR_WRONG_TYPE okay (w,r,c,l)=(10,5,0,342)
input/s23.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(11,7,0,436)
input/s24.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_ERROR_WRONG_TYPE(6) (w,r,c,l)=(11,7,1,508)
input/s25.iks IKS_SUCCESS okay (w,r,c,l)=(5,4,0,290)

```

[illegible]

```

input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(10,2,0,56)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(14,7,0,64)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(6,2,0,48)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(2,1,0,40)
input/s04.iks causes a segfault (exit code is 139)
input/s05.iks expects a IKS_ERROR_VARIABLE(3), but we got a IKS_SUCCESS(0) (w,r,c,l)=(16,7,0,72)
input/s06.iks expects a IKS_ERROR_VECTOR(4), but we got a IKS_SUCCESS(0) (w,r,c,l)=(12,6,0,56)
input/s07.iks causes a segfault (exit code is 139)
input/s08.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,11,0,64)
input/s09.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,11,0,64)
input/s10.iks causes a segfault (exit code is 139)
input/s11.iks causes a segfault (exit code is 139)
input/s12.iks causes a segfault (exit code is 139)
input/s13.iks causes a segfault (exit code is 139)
input/s14.iks expects a IKS_ERROR_WRONG_PAR_INPUT(12), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(0,0,0,40)
input/s15.iks expects a IKS_ERROR_WRONG_PAR_OUTPUT(13), but we got a IKS_SUCCESS(0) (w,r,c,l)=(6,4,2,48)
input/s16.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(6,3,1,40)
input/s17.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,11,0,64)
input/s18.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(13,11,0,64)
input/s19.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(14,6,0,56)
input/s20.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(14,6,0,56)
input/s21.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(12,6,0,56)
input/s22.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(12,6,0,56)
input/s23.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(20,7,2,72)
input/s24.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(20,7,2,72)
input/s25.iks expects a IKS_SUCCESS(0), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(10,2,0,56)
input/s26.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_SUCCESS(0) (w,r,c,l)=(20,7,2,72)
input/s27.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_SUCCESS(0) (w,r,c,l)=(28,7,2,104)
input/s28.iks expects a IKS_ERROR_STRING_TO_X(7), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(36,9,1,96)
input/s29.iks expects a IKS_ERROR_CHAR_TO_X(8), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(36,9,1,96)
input/s30.iks expects a IKS_ERROR_WRONG_PAR_RETURN(14), but we got a IKS_ERROR_UNDECLARED(1) (w,r,c,l)=(14,4,1,64)
input/s31.iks expects a IKS_SUCCESS(0), but we got a IKS_ERROR_WRONG_PAR_RETURN(14) (w,r,c,l)=(6,3,0,40)
input/s32.iks causes a segfault (exit code is 139)
g7 1.515151515151515150 1.43939393939393939390

```

```

input/s00.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(1,7,0,0)
input/s01.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(5,24,0,43)
input/s02.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(1,7,0,0)
input/s03.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(1,11,0,0)
input/s04.iks IKS_ERROR_DECLARED okay (w,r,c,l)=(2,10,0,574)
input/s05.iks causes a segfault (exit code is 139)
input/s06.iks expects a IKS_ERROR_VECTOR(4), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,31,0,700)
input/s07.iks expects a IKS_ERROR_FUNCTION(5), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,25,0,1,220)
input/s08.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(4,22,0,44)
input/s09.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(4,22,0,44)
input/s10.iks expects a IKS_ERROR_MISSING_ARGS(9), but we got a IKS_ERROR_EXCESS_ARGS(10) (w,r,c,l)=(3,14,0,590)
input/s11.iks IKS_ERROR_EXCESS_ARGS okay (w,r,c,l)=(5,21,0,620)
input/s12.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(5,19,0,606)
input/s13.iks IKS_ERROR_WRONG_TYPE_ARGS okay (w,r,c,l)=(5,19,0,606)
input/s14.iks IKS_ERROR_WRONG_PAR_INPUT okay (w,r,c,l)=(2,5,0,0)
input/s15.iks expects a IKS_ERROR_WRONG_PAR_OUTPUT(13), but we got a IKS_SUCCESS(0) (w,r,c,l)=(6,18,0,635)
input/s16.iks IKS_ERROR_WRONG_PAR_RETURN okay (w,r,c,l)=(4,12,0,35)
input/s17.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(4,22,0,48)
input/s18.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(4,22,0,46)
input/s19.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,26,0,669)
input/s20.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,26,0,665)
input/s21.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,29,0,658)

```

```

input/s22.iks expects a IKS_ERROR_WRONG_TYPE(6), but we got a IKS_SUCCESS(0) (w,r,c,l)=(7,29,0,664)
input/s23.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(6,23,0,50)
input/s24.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(6,23,0,50)
input/s25.iks causes a segfault (exit code is 139)
input/s26.iks IKS_ERROR_CHAR_TO_X okay (w,r,c,l)=(6,26,0,45)
input/s27.iks IKS_ERROR_STRING_TO_X okay (w,r,c,l)=(8,33,0,65)
input/s28.iks causes a segfault (exit code is 139)
input/s29.iks causes a segfault (exit code is 139)
input/s30.iks causes a segfault (exit code is 139)
input/s31.iks IKS_SUCCESS okay (w,r,c,l)=(10,35,0,1,312)
input/s32.iks IKS_ERROR_UNDECLARED okay (w,r,c,l)=(2,16,0,576)
g8 6.060606060606060600 2.424242424242424240

```

9 Script utilizado na avaliação automática

```

#!/bin/bash

for GRUPO in `ls -d g*`
do
    BIN=$GRUPO/main
    REPORT=$GRUPO/report
    declare -i nota
    declare -i quality
    ((nota = 0))
    ((quality = 0))

    if [ ! -z "$1" ]
    then
        if [ "$GRUPO" != "$1" ]
        then
            continue
        fi
    fi

    rm -f $REPORT

    ID="** $GRUPO `cat $GRUPO/GROUP`"
    echo
    echo "$ID"
    echo "$ID" >> $REPORT
    echo

    #test existence
    if [ ! -x $BIN ]
    then
        echo "# $GRUPO ignored"
        continue
    fi

    CVERBATIM="\begin{verbatim}"
    echo $CVERBATIM
    echo $CVERBATIM >> $REPORT

    for INPUT in `ls -1 input/s*.iks`
    do
        EXPECTED_ERROR=`head -n1 $INPUT | awk '{ print $2 }'`
        EXPECTED_ERROR_CODE=`grep "$EXPECTED_ERROR " input/errors.h | awk '{ print $3 }'`
        $BIN < $INPUT > /dev/null 2> /dev/null #$REPORT
        GOT_ERROR_CODE=$?

        #detect segfault
        if [ $GOT_ERROR_CODE -eq 139 ]
        then
            R="$INPUT causes a segfault (exit code is $GOT_ERROR_CODE)"
            echo $R
            echo $R >> $REPORT
            continue
        fi

        #check with valgrind
        TEMP_FILE="/tmp/e4-$RANDOM.tmp"

```

```

rm -f $TEMP_FILE
valgrind $BIN < $INPUT 2>> $TEMP_FILE >> $TEMP_FILE
INVALID_WRITES='cat $TEMP_FILE | grep "Invalid write" | wc -l'
INVALID_READS='cat $TEMP_FILE | grep "Invalid read" | wc -l'
CONDITIONAL='cat $TEMP_FILE | grep "Conditional" | wc -l'
BYTES_LOST='cat $TEMP_FILE | grep "definitely lost:" | awk '{ print $4}''
rm $TEMP_FILE
CODE_CORRECT="(w,r,c,l)=$INVALID_WRITES,$INVALID_READS,$CONDITIONAL,$BYTES_LOST)"
if [ $INVALID_WRITES -eq 0 ]
then
    ((quality = $quality + 1))
fi
if [ $INVALID_READS -eq 0 ]
then
    ((quality = $quality + 1))
fi
if [ $CONDITIONAL -eq 0 ]
then
    ((quality = $quality + 1))
fi
if [ $BYTES_LOST -eq 0 ]
then
    ((quality = $quality + 1))
fi

#analyze the result and report
if [ $GOT_ERROR_CODE -ne $EXPECTED_ERROR_CODE ]
then
    N='grep " $GOT_ERROR_CODE " input/errors.h | awk '{ print $2 }''
    R="$INPUT expects a $EXPECTED_ERROR($EXPECTED_ERROR_CODE), but we got a $N($GOT_ERROR_CODE) $CODE_CORRECT"
    echo $R
    echo $R >> $REPORT
else
    ((nota = $nota + 1))
    R="$INPUT $EXPECTED_ERROR okay $CODE_CORRECT"
    echo $R
    echo $R >> $REPORT
fi
done

NTESTES='ls -l input/s*.iks | wc -l'
NOTA_FINAL='echo "$nota / $NTESTES * 10" | bc -l'
QUALIDADE_FINAL='echo "($quality / ($NTESTES * 4)) * 10" | bc -l'
R="$GRUPO $NOTA_FINAL $QUALIDADE_FINAL"
echo $R
echo $R >> $REPORT

CVERBATIM="

```

" echo *CVERBATIM* echo *CVERBATIM* » *REPORT* done