Bill J. Ellis
April 14, 2004

Praxis Retreat: A Survey

1 Introduction

A high integrity SPARK program is surveyed. Rather than target particular points, the code is considered in large blocks, hoping to gain a more accurate picture of the genuine recurring problems.

2 Headings

The headings of the tables are discovered and refined during the survey process.

2.1 Address

- **Address -** The dotted location of the subprogram. Note that package.subprogram.subprogram is used to refer to subprograms nested within subprograms. This is not legal SPARK or Ada.

2.2 Genre

- **Straight through (ST) -** Straight through code.
- **With loop (WL) -** At least one loop in the code.
- **Regular code (RC) -** A regular looking SPARK subprogram.
- **Irregular code (IC) -** A subprogram that employs some unusual feature of SPARK, such as non decimal numbers.
- **Couple liner (CL) -** The subprogram contains a very small number of lines.
- **Maths routine (MR) -** The code is performing a maths routine of some form.

2.3 Code classification

- **Variable in type (VIT)**
- **Expressions in base type (EIBT)**
- **For loop range (FLR)**
- **Bring values through code (BVTC)**
- **Loop recurrence relations (LRR)**
- **Loop termination (LT)**
- **Exploit Boolean Assignments (EBA)**
- **Extra cut between if statements (ECBIS)**
- **Extra Cut between if statements possible but not used (ECBISPBNU)**
- **Extra cut at no special location (ECANSLU)**
- **Specification information (SI)**
- **Deep algorithm information (DAI)**
- **Propagate non variable property (PNVP)**

In these cases some program analysis is required to complete the proof.

*Notes in this series are for ϵ-baked ideas, for $1 \geq \epsilon \geq 0$. Only exceptionally should they be cited or distributed outwith the NuSPADE project (EPSRC GR/R24081).

2.4 Proof classification

- **Hard (H) -** Deeper analysis required!
- **Missing specification rules (MSR) -** Rules missing. These rules are part of the specification and are very unlikely to be manually discovered.
- **Missing Examiner rules (MER) -** Rules missing. These might sensibly be expected to be generated by the Examiner. These are typically for fringe SPARK features. Examples include, generating substitutions for tick delta.
- **Appeal to constant rules (ATCR) -** Proof will involve appealing to replace constant rules. The availability and form of these rules is a problem for both the Examiner and the Simplifier.
- **Tricky non-linear arithmetic (TNLA) -** These are not covered by the proposed changes to the Simplifier. They require some reasoning and dealing with non-linear arithmetic.
- **Stronger specification required (SSR) -** These are not provable with the current specification.

2.5 Remaining VCs

The numbers of the VCs expected not to be discharged by the SPARK toolset after the proposed changes to the Simplifier and Examiner. In these cases some user intervention is required to complete the proof.

3 Tables

3.1 Package

Address	Genre	Code	Proof	Remaining
alarm.checklimits	RC + ST	ECBIS + SI	H	27, 28, 29
alarm.checklimits.audiblealarmoff	RC + ST	-	-	-
alarm.checklimits.audiblealarmon	RC + ST + CL	-	-	-
alarm.checklimits.datalimitsfail	RC + ST + CL	-	-	-
alarm.checklimits.datalimitsok	RC + ST + CL	-	-	-
alarm.initialise	RC + ST	-	-	-
alarm.mutereset	RC + ST + CL	-	-	-
alarm.muteset	RC + ST + CL	-	-	-
alarm.overridealarms	RC + ST + CL	-	-	-
alarm.revertalarms	RC + ST + CL	-	-	-
alarm.updaterollpitchlimit	RC + ST	-	MSR	11,12,13,14

3.2 Package

Address	Genre	Code	Proof	Remaining
basictypes.addrelwinddirections	RC + ST + CL	-	H + (MSR: maybe)	2
basictypes.addtruedirections	RC + ST + CL	-	-	-
basictypes.adjustrelwinddir	RC + ST	-	-	-
basictypes.arccos	RC + WL + MR	VIT + (LT: maybe) + DAI	H + ATCR	20, 24, 25, (8: impossible? x=1, y=2)
basictypes.boundedround	RC + ST	-	-	-
basictypes.cos	RC + ST	-	-	-
basictypes.divfixedpoint	RC + ST + CL	-	MER	3
basictypes.mulfixedpoint	RC + ST + CL	-	-	-
basictypes.mulsqrt	RC + ST + CL	-	-	-
basictypes.round	RC + ST	-	TNLA	1, 2, 3, 4
basictypes.sin	RC + ST + CL	-	-	-
basictypes.sqrt	RC + WL + MR	VIT + EIBT + LT + SI + DAI	TNLA	3,4,5,6,9,11,15,16,17
basictypes.subtractrelwinddirections	RC + ST + CL	-	H + (MSR: maybe)	2
basictypes.subtracttruedirections	RC + ST	-	-	-

3.3 Package

Address	Genre	Code	Proof	Remaining
bite.inccounter	RC + ST	-	-	-
bite.incrementdbpectimeoutcounter	RC + ST + CL	-	-	-
bite.incrementmiscpectimeoutcounter	RC + ST + CL	-	-	-
bite.incrementvmewatchdogfailcounter	RC + ST + CL	-	-	-
bite.incrementwiwatchdogfailcounter	RC + ST + CL	-	-	-
bite.initialise	RC + ST	-	-	-
bite.setbitecontrolregister1bit	RC + ST + CL	-	-	-
bite.setbitecontrolregister2bit	RC + ST + CL	-	-	-
bite.setbitecontrolregister3bit	RC + ST + CL	-	-	-

3.4 Package

Address	Genre	Code	Proof	Remaining
coc1p.acceptlog	RC + ST	-	-	-
coc1p.calcselectableandfixedcones	RC + ST	-	-	-
coc1p.decrementlogtens	RC + ST	-	-	-
coc1p.decrementlogunits	RC + ST	-	-	-
coc1p.incrementlogtens	RC + ST	-	TNLA	3
coc1p.incrementlogunits	RC + ST	-	TNLA	3
coc1p.initialise	RC + ST	-	-	-
coc1p.recalccoc1p	RC + ST	-	-	-
coc1p.reinitialiseconescalculation	RC + ST + CL	-	-	-
coc1p.reinitialiseselectedlog	RC + ST	-	-	-
coc1p.showselectablelog	RC + ST	-	-	-
coc1p.updatecoc1pconescreen	RC + ST	ECBISPBNU	-	-
coc1p.updatecones	RC + ST	-	-	-
coc1p.updatedynamicscreen	RC + ST	-	-	-
coc1p.updatescreen	RC + ST	-	-	-

3.5 Package

Address	Genre	Code	Proof	Remaining
coc4p.calcadvice	RC + WL	VIT + LRR + EBA + DAI	H	7,9,10,14,19,20, (17,18,29,30: impossible)
coc4p.calcadvice.findadviceforhigherlogatotherheading	RC + WL	VIT	-	-
coc4p.calcadvice.findadviceforlowerlogatotherheading	RC + WL	VIT	-	-
coc4p.calccone	RC + ST	-	-	-
coc4p.calccone.findwidestcone	RC + WL	VIT + (DAI: hard)	-	-
coc4p.calccone.findwidestcone.tosholangle	RC + ST	-	-	-
coc4p.calccone.segmentcentre	RC + ST	-	-	-
coc4p.calcfixedlogcones	RC + WL	VIT	-	-
coc4p.clearcones	RC + WL	VIT	-	-
coc4p.conerelwind	RC + ST	-	-	-
coc4p.initialise	RC + ST	-	-	-
coc4p.recalccoc4p	RC + ST	-	-	-
coc4p.reinitialiseconescalculation	RC + ST + CL	-	-	-
coc4p.setcone	RC + WL	VIT + FLR	-	-
coc4p.updatecoc4pconescreen	RC + ST	-	-	-
coc4p.updateconedatainvalid	RC + ST	-	-	-
coc4p.updatecones	RC + ST	-	-	-
coc4p.updatedynamicscreen	RC + ST	-	-	-
coc4p.updatescreen	RC + ST + CL	-	-	-

3.6 Package

Address	Genre	Code	Proof	Remaining
cpu.fulldatabasepromtest	RC + ST	-	SSR	(3: impossible)
cpu.fullramtest	RC + ST	-	SSR	(3: impossible)
cpu.initialise	IC + ST	-	-	-
cpu.periodicpromtest	RC + ST	-	SSR	(3,5,12: impossible)
cpu.periodicramtest	RC + ST	-	SSR + H	10, (1,3,5,12: impossible)
cpu.userlinkpresent	RC + ST + CL	-	-	-
cpu.writevpu2wfailled	RC + ST + CL	-	-	-
cpu.writevpu2wsoftwarefaultled	RC + ST	-	-	-

3.7 Package

Address	Genre	Code	Proof	Remaining
cursel.cancel	RC + ST	-	-	-
cursel.initialise	RC + WL	-	-	-
cursel.pitchscale	RC + ST	-	-	-
cursel.rollscale	RC + ST	-	-	-
cursel.saveselection	RC + WL	VIT + FLR	-	-
cursel.selectecl	RC + ST	-	-	-
cursel.selectelva	RC + ST	-	-	-
cursel.unavailable	RC + ST	-	-	-
cursel.updateselection	RC + ST	-	-	-
cursel.vectorinlimits	RC + ST	-	MSR	4

3.8 Package

Address	**Genre**	**Code**	**Proof**	**Remaining**
db.convertdbsholtoshol	RC + WL	VIT + BVTC + ECANSLU + LT	H	38, (36, 39, 40, 41, 42: impossible)
db.convertdbsholtoshol.absolutearcwidth	RC + ST	-	-	-
db.convertdbsholtoshol.intersectionlimitregion	RC + WL	-	-	-
db.convertdbsholtoshol.issamepointas	RC + ST + CL	-	-	-
db.convertdbsholtoshol.setconstantdirectionwind	RC + WL	VIT + FLR	-	-
db.convertdbsholtoshol.setconstantspeed	RC + WL	VIT + FLR	-	-
db.convertdbsholtoshol.setenvelopeempty	RC + WL	VIT	-	-
db.convertdbsholtoshol.setenvelopefull	RC + WL	VIT	-	-
db.convertdbsholtoshol.setlimitregionnull	RC + WL	VIT	-	-
db.convertdbsholtoshol.setlimitregionorigin	RC + WL	-	-	-
db.convertdbsholtoshol.setline	RC + ST	VIT + BVTC + ECANSLU	SSR	(8: impossible)
db.convertdbsholtoshol.setline.averagexcomponent	RC + ST	-	MSR	(3,6,7: impossible)
db.convertdbsholtoshol.setline.averageycomponent	RC + ST	-	MSR	(3,6,7: impossible)
db.convertdbsholtoshol.setline.lineargradient	RC + ST	-	MSR + SSR	(2,3: impossible)
db.convertdbsholtoshol.setline.setconstantdirection	RC + ST	-	-	-
db.convertdbsholtoshol.setline.setconstantsidewind	RC + WL	VIT + FLR	MSR	(4: impossible)
db.convertdbsholtoshol.setline.setconstantspeedanticlockwise	RC + ST	-	-	-
db.convertdbsholtoshol.setline.setconstantspeedclockwise	RC + ST	-	-	-
db.convertdbsholtoshol.setline.setconstanttailwind	RC + WL	VIT + FLR + PNVP	MSR	(4,9: impossible)
db.convertdbsholtoshol.setline.setlinearlychanging	RC + WL	VIT	MSR	(3: impossible)
db.convertdbsholtoshol.setlinearwind	RC + WL	VIT + FLR	MSR + SSR	(2,3: impossible)
db.convertdbsholtoshol.setregion	RC + ST	ECANSLU + VIT	-	-
db.convertdbsholtoshol.setregion.setregionphase1	RC + WL	VIT	SSR	(8,9: impossible)
db.convertdbsholtoshol.setregion.setregionphase2	RC + WL	VIT	SSR	(8,9: impossible)
db.convertdbsholtoshol.setregion.setregionphase3	RC + WL	VIT	SSR	(11,12,13: impossible)
db.convertdbsholtoshol.setsholorigin	RC + WL	VIT	-	-
db.convertdbsholtoshol.setsidewind	RC + WL	VIT + FLR	MSR + SSR	(3,4: impossible)
db.convertdbsholtoshol.settailwind	RC + WL	VIT + FLR	MSR + SSR	(3,4: impossible)
db.convertdbsholtoshol.unionlimitregion	RC + WL	-	-	-
db.getsholbyindex	RC + ST	-	-	-
db.getsholkey	RC + ST + CL	-	-	-

3.9 Package

Address	Genre	Code	Proof	Remaining
dbtypes.addsholangles	RC + ST + CL	-	-	-
dbtypes.adjustsholangle	RC + ST	-	-	-
dbtypes.anglebetweensholangles	RC + ST + CL	-	-	-
dbtypes.converttorelwinddir	RC + ST + CL	-	-	-
dbtypes.converttosholangle	RC + ST	-	TNLA	8,9,13
dbtypes.nextsholangle	RC + ST	-	-	-
dbtypes.previoussholangle	RC + ST	-	-	-
dbtypes.subtractsholangles	RC + ST + CL	-	-	-

3.10 Package

Address	Genre	Code	Proof	Remaining
diag.rtassert	RC + ST	-	-	-
diag.rtassertdecimal	RC + ST	-	-	-

4 I give up

This covers about 9 percent of system S. Hopefully this is fairly reflective of the rest of the code. Now that an impression of the code has been gained it is reasonable to skip though the code with a little more haste, stopping where subprograms are encountered that do not conform to the subprograms seen here.