Project: DAZ PROJECT

Title: Implementation of First 1000 Primes Example

Ref: ISS/HAT/DAZ/WRK501 *Issue:* 1.30 *Date:* 22 July 2011

Status: Informal *Type:* Technical

Author:

Name	*Location*	*Signature*	*Date*
D.J. King	WIN01		

Authorisation for Issue:

Name	*Function*	*Signature*	*Date*
R.D. Arthan	HAT Team		

Abstract: This document gives an experimental implementation "Refinement of Z to SPARK: First 1000 Primes", DRA/CIS/CSE3/SWI/WP/5/2.

Distribution: Library

0 DOCUMENT CONTROL

0.1 Contents List

0.2 Document Cross References

[1] DRA/CIS/CSE3/SWI/WP/5/2. *Refinement of Z to SPARK: Example - First 1000 Primes.* C.M. O'Halloran, C.T. Sennett, and A. Smith, Defence Research Agency, Malvern, 11th October 1993.

[2] ISS/HAT/DAZ/HLD501. *High Level Design: Overview.* D.J. King, Lemma 1 Ltd., `http://www.lemma-one.com`.

[3] ISS/HAT/DAZ/USR501. *Compliance Tool — User Guide.* Lemma 1 Ltd., `http://www.lemma-one.com`.

0.3 Changes History

Issue 1.15 First Issue.

Issue 1.16 TL added invocation of VC browser according to shell variable `PP_USE_VC_BROWSER`

Issue 1.19 Copyright and banner updates for open source release.

Issue 1.20 DAZ-specific updates to banner for open source release

Issue 1.21 DAZ-specific updates to banner for open source release

Issue 1.23 The SPARK program is now referred to as the Ada program.

Issue 1.24 Made it output success message in same format as the module tests use.

Issue 1.25 Allowed for changes to VCs resulting from environment reforms.

Issue 1.26 Compliance Notation reserved words are now prefixed by a dollar sign.

Issue 1.27 Allowed for automated state management.

Issue 1.29 Allowed for enhancement 165.

0.4 Changes Forecast

None.

1 GENERAL

1.1 Scope

This document contains a transcription of the literate script for the "1000 Primes" example in [1] suitable for input into the Compliance Tool as specified in [2] and [3].

1.2 Introduction

1.3 Background

The literate script for the "1000 Primes" example constitutes the case study which forms part of DRA's acceptance criteria for the compliance tool. This document contains an adaptation of the script in [1] suitable for input into the compliance tool.

2 PREAMBLE

The following Standard ML script initialises the theory database and sets up the appropriate modes for processing the literate script which follows.

SML

```
new_script {name="PRIMES", unit_type="proc"};
```

3 THE LITERATE SCRIPT

This literate script aims to follow exactly that supplied in [1].

z

$$prime \mathrel{\widehat=} \{ n : \mathbb{N}_1 \mid \neg(\exists i, j : \mathbb{N}_1 \setminus \{1\} \bullet i * j = n)\}$$

z

$$not\_prime \mathrel{\widehat=} \mathbb{N}_1 \setminus prime$$

z

$$rel \quad \_\ next\_prime\ \_$$

z

$$\_\ next\_prime\ \_ : \mathbb{N}_1 \leftrightarrow \mathbb{N}_1$$

$$(\_\ next\_prime\ \_) =$$
$$\{i, j : \mathbb{N}_1 \mid i \in prime \land j \in prime \land j>i \land (i+1)..(j-1) \subseteq not\_prime\}$$

z

$$primed \mathrel{\widehat=} \{p : seq_1\ \mathbb{N}_1 \mid p(1) = 2 \land (\forall\ i : dom\ p \setminus \{1\} \bullet p(i-1)\ next\_prime\ p(i))\}$$

Compliance Notation

```
procedure primes is
subtype Indexrange is Integer range 1..1000;
type Arraytype is array (Indexrange) of Integer;
p : Arraytype;

⟨ other declarations ⟩          (1)
⟨ other declarations ⟩          (2)
⟨ other declarations ⟩          (3)

begin

Δ P [true, P ∈ primed]

end primes;
```

Compliance Notation

```
(1) ≡

j, k : integer;
```

z

$$odd \mathrel{\widehat=} \{n : \mathbb{N}_1 \mid \forall\ i : \mathbb{N}_1 \bullet n \neq 2 * i\}$$

z

Inv1

$$P : ARRAYTYPE;\quad J, K : \mathbb{Z}$$

$$((1\ ..\ K-1) \lhd P) \in primed;\quad J = P(K-1);\quad J \in odd$$

Compliance Notation

$$\sqsubseteq \Delta\ J,K,P\ [true, Inv1 \land K = 1001]$$

Compliance Notation

```
(2) ≡

ord, square : Integer;
ord_max : constant Integer := 30;
subtype mult_index_type is Integer range 2..ord_max;
type mult_type is array (mult_index_type) of Integer;
mult : mult_type;
```

z

ord_inv

$$P : ARRAYTYPE;$$
$$J,\ K,\ ORD,\ SQUARE : \mathbb{Z}$$

$$ORD > 1;$$
$$SQUARE = P(ORD) * P(ORD);$$
$$J < SQUARE;$$
$$ORD < K;$$
$$ORD = 2 \Rightarrow J \in prime$$

z

$$rel \quad \_\ factor\_of\ \_$$

z

$$\_\ factor\_of\ \_ : \mathbb{N}_1 \leftrightarrow \mathbb{N}_1$$

$$(\_\ factor\_of\ \_) =$$
$$\{i,\ n : \mathbb{N}_1 \mid \exists\ j : 2..(n-1) \bullet i * j = n\}$$

z

mult_inv

$$P : ARRAYTYPE;$$
$$MULT : MULT\_TYPE;$$
$$ORD,\ J : \mathbb{Z}$$

$$\forall\ n : 2..(ORD - 1) \bullet$$
$$MULT(n) \in odd \wedge$$
$$P(n)\ factor\_of\ MULT\ (n) \wedge$$
$$(MULT(n) < J \vee$$
$$\quad MULT(n) - 2 * P(n) < J \leq MULT(n))$$

z

$Inv \mathrel{\widehat=} ord\_inv \land mult\_inv$

Compliance Notation

$\sqsubseteq$

$p(1) := 2;$
$p(2) := 3;$
$j := 3;$
$k := 3;$
$ord := 2;$
$square := 9;$

$\Delta\ J,K,P,ORD,\ SQUARE,\ MULT\ [Inv1 \land Inv,\ Inv1 \land Inv \land K = 1001]$

Compliance Notation

$\sqsubseteq$

while $k \mathrel{/=} 1001$
loop
$\quad \Delta\ J,K,P,ORD,SQUARE,MULT\ [Inv1 \land Inv,\ Inv1 \land Inv]$
end loop;

z

Inter1

$P : ARRAYTYPE;$
$J,\ K : \mathbb{Z}$

$((1\ ..\ K - 1) \lhd P) \in primed;$
$P(K - 1)\ next\_prime\ J;$
$J \in odd$

Compliance Notation

$\sqsubseteq$

$\Delta\ J,\ ORD,\ SQUARE,\ MULT\ [Inv1 \land Inv,\ Inter1 \land Inv]$
$p(k) := j;$
$k := k+1;$

Compliance Notation

$\sqsubseteq$
$\$CON\ lastj : \mathbb{Z} \bullet \Delta\ J,\ ORD,\ SQUARE,\ MULT$
$\quad [lastj = J \land Inv1 \land Inv,\ lastj\ next\_prime\ J \land Inv]$

z

Inv2

$$P : ARRAYTYPE;$$
$$J,\ K,\ lastj : \mathbb{Z}$$

$$lastj = P(K - 1);$$
$$J \geq lastj;$$
$$(lastj{+}1)\ ..\ (J{-}1) \subseteq not\_prime;$$
$$((1..K{-}1) \lhd P) \in primed;$$
$$J \in odd$$

Compliance Notation

$\sqsubseteq$

$\$till\ [\![Inv2 \wedge J > lastj \wedge J \in prime \wedge Inv]\!]$
loop
$\quad \Delta\ J,\ ORD,\ SQUARE,\ MULT\ [Inv2 \wedge Inv \wedge (J \in prime \Rightarrow lastj = J),$
$\quad\quad Inv2 \wedge Inv \wedge J \notin prime]$

end loop;

Compliance Notation

$(3) \equiv$

$jprime : Boolean;$

Compliance Notation

$\sqsubseteq$

$\quad \Delta\ J,\ ORD,\ SQUARE,\ MULT\ [Inv2 \wedge Inv \wedge (J \in prime \Rightarrow lastj = J),$
$\quad\quad Inv2 \wedge Inv \wedge J = J_0 + 2]$
$\quad \Delta\ JPRIME,\ MULT\ [Inv2 \wedge Inv \wedge J > lastj,$
$\quad\quad Inv2 \wedge Inv \wedge J > lastj \wedge JPRIME = J\ mem\ prime]$ (4)
exit when jprime;

Compliance Notation

$\sqsubseteq$

$j := j + 2;$
if $j = square$
then
$\quad ord := ord + 1;$

$$
\begin{array}{|l}
\quad square := p(ord) * p(ord); \\
\quad mult\ (ord-1) := j; \\
end\ if;
\end{array}
$$

Compliance Notation

$$
\begin{array}{|l}
(4) \sqsubseteq \\
\\
jprime := true; \\
\Delta\ JPRIME,\ MULT\ [Inv2 \land Inv \land JPRIME = TRUE, \\
\qquad\qquad Inv2 \land Inv \land JPRIME = J\ mem\ prime]
\end{array}
$$

z

Inv3

$$
\begin{array}{|l}
\quad P : ARRAYTYPE; \\
\quad J,\ ORD,\ N : \mathbb{Z}; \\
\quad JPRIME : BOOLEAN \\
\hline
\quad N \in 2..(ORD - 1); \\
\quad \forall\ m : 2..\ (N-1) \bullet \neg(P(m)\ factor\_of\ J); \\
\quad JPRIME = TRUE; \\
\quad J \in odd
\end{array}
$$

Compliance Notation

$$
\begin{array}{|l}
\sqsubseteq \\
\\
for\ n\ in\ Integer\ range\ 2\ ..\ (ord - 1) \\
\$till\ [\![P(N)\ factor\_of\ J \land JPRIME = FALSE \land Inv]\!] \\
\\
loop \\
\quad \Delta\ JPRIME,\ MULT\ [Inv3 \land Inv, \\
\qquad\qquad Inv3 \land Inv \land \neg(P(N)\ factor\_of\ J)] \\
end\ loop;
\end{array}
$$

Compliance Notation

$$
\begin{array}{|l}
\sqsubseteq \\
\\
\Delta\ JPRIME,\ MULT\ [Inv3 \land mult\_inv,\ mult\_inv \land (P(N)\ factor\_of\ J \Leftrightarrow JPRIME = FALSE)] \\
\\
exit\ when\ not\ jprime;
\end{array}
$$

Compliance Notation

$$
\begin{array}{l}
\sqsubseteq \\
\Delta\ MULT[mult\_inv,\ mult\_inv \wedge MULT\ (N) \geq J] \\
\Delta\ JPRIME\ [mult\_inv \wedge MULT\ (N) \geq J, \\
\qquad\qquad mult\_inv \wedge MULT\ (N) \geq J \wedge JPRIME = MULT\ (N)\ noteq\ J] \quad (5)
\end{array}
$$

Compliance Notation

$$
\begin{array}{l}
\sqsubseteq \\
\\
while\ mult(n) < j \\
loop \\
\qquad \Delta\ MULT\ [mult\_inv \wedge MULT(N) < J,\ mult\_inv] \\
end\ loop;
\end{array}
$$

Compliance Notation

$$
\begin{array}{l}
\sqsubseteq \\
\\
mult(n) := mult(n) + p(n) + p(n);
\end{array}
$$

Compliance Notation

$$
\begin{array}{l}
(5) \sqsubseteq \\
\\
jprime := mult(n)\ /= j;
\end{array}
$$

SML

```
output_z_document{script="PRIMES'proc", out_file="wrk501.zdoc"};
output_ada_program{script="-", out_file="wrk501.ada"};


val wrk501_browse =
        let val shvar = get_shell_var "PP_USE_VC_BROWSER"
        in      if shvar <> "" andalso shvar <> "NO" andalso shvar <> "no"
                        andalso shvar <> "No"
                then browse_vcs() else ()
        end;
diag_line "All module tests passed";
```