From dm@Franz.COM Mon Sep 10 10:25:17 1990
From: dm@Franz.COM (David Margolies)
To: fateman@Franz.COM, fkunze@Franz.COM, jim@Franz.COM, layer@Franz.COM,
        tech@Franz.COM
Subject: Re: Know of any fp errors in ACL

            Date: Mon, 10 Sep 90 08:54:21 PDT
            From: jim (Jim Veitch)

            >From fateman Fri Sep 7 18:43:55 1990
            Return-Path: <fateman>
            Date: Fri, 7 Sep 90 18:44:27 PDT
            From: fateman (Richard Fateman)
            To: fkunze, jim
            Subject: floating point errors

            Kahan would very much like to learn of any problems with the
            floating point arithmetic elementary functions in Allegro.

The know fp errors in built in functions are described in
the file associated with [bug1328]. These were detected by
a group at MIT. He is a transcript of the bug file.

David Margolies

@@id: bug1328
@@reference:
@@summary: test suite from a user at MIT
@@product: acl
@@component: test suite
@@customer-name: Jerry Roylance
@@customer-company: MIT
@@customer-phone:
@@customer-email-address: glr@ai.mit.edu
@@email-cc:
@@hardware:
@@operating-system-and-version:
@@state: logged 8/16/90 by layer
@@state: priority 8/16/90 9 by layer
@@endstate

@@resolution:

@@discussion:

Replied: Fri, 10 Aug 90 14:36:34 -0700
Replied: "Jerry Roylance <glr@ai.mit.edu> +outbox"
Return-Path: glr@ai.mit.edu
Return-Path: <glr@ai.mit.edu>
Date: Fri, 10 Aug 90 16:51 EDT
From: Jerry Roylance <glr@ai.mit.edu>
Subject: A reminder
To: layer@Franz.COM

        Date: Tue, 07 Aug 90 13:53:07 -0700
        From: layer@franz.com (Kevin Layer)

Jerry,
    This is a reminder to forward the test file from which you generated
the listing which you showed me at AAAI. We are certainly interested
in fixing the bugs you have found. Thanks.

            Kevin Layer
            Franz Inc.
            layer@franz.com

Kevin,

Here's more information about the specific bugs.

            Jerry

Common Lisp Test Suite Version 1.0.1, 14 Feb 1990

Implementation Information:
  lisp-implementation-type: "Allegro CL"
  lisp-implementation-version: "3.1.4 [Sun4] (12/1/89)"
  machine-type: "Sun4"
  machine-version: ""
  machine-instance: "id: 1358965870"
  software-type: "SunOS"
  software-version: ""
  short-site-name: "nut-and-honey"
  long-site-name: "nut-and-honey"

GC information (room):

space	cons (free:used)	symbols (free:used)	roots (free:used)	other bytes (free:used)
New	172:4923	178:25	666:1367	798616:140896
Tenured	198:141443	111:19701	-----	2389456:5820200

;;; Checking Symbols
These warnings are obvious... LISP: package symbols that have extra
bindings...
W 1 (SYMBOLS   ): Nonstandard function binding for symbol: "DOUBLE-FLOAT"
W 2 (SYMBOLS   ): Nonstandard function binding for symbol: "SIMPLE-STRING"
W 3 (SYMBOLS   ): Nonstandard binding for symbol: "SPACE"
W 4 (SYMBOLS   ): Nonstandard function binding for symbol: "STRUCTURE"

;;; Checking Branch Cuts
These errors uncover discontinuities in the trig functions.
The error message gives two points (z0, f[z0]) and (z1, f[z1]) that
should be in the same neighborhood.
E 1 (ASIN   ): ASIN fails branch-cut test ((-2.0, #c(-1.5707964 1.316958))
                         (#c(-2.0 1.0e-5), #c(1.5707905 1.316958)))
E 2 (ACOS   ): ACOS fails branch-cut test (-2.0, #c(0.0 -1.316958))
                         (#c(-2.0 1.0e-5), #c(3.1415985 -1.316958)))
E 3 (ATAN   ): ATAN fails branch cut test (#c(0.0 2.0),
                    #c(1.5707964 0.54930615))
                         (#c(-1.0e-5 2.0), #c(-1.570793 0.54930615)))
E 4 (ASINH   ): ASINH test blew up
E 5 (ACOSH   ): ACOSH test blew up
E 6 (ATANH   ): ATANH fails branch cut test (#c(-1.5 0.0),
                    #c(-0.804719 1.5707964))
                         (#c(-1.5 -1.0e-5), #c(-0.804719 -1.5707884)))

```
(defun check-branch-cut (fcn pt eps E N W S)
   (let* ((delta 0.01)
          (u      (complex eps   0))
          (v      (complex   0 eps))
          (f      (symbol-function fcn))
          (z-E    (+ pt u))
          (z-N    (+ pt v))
          (z-W    (- pt u))
          (z-S    (- pt v))
          (y      (funcall f pt)))
     (flet ((test (z)                              ; should be continuous
```