$SPAD/src/input aseg6.input

The Axiom Team

December 3, 2016

Abstract

Contents

1 License

```
-- Copyright (c) 1991-2002, The Numerical ALgorithms Group Ltd.
-- All rights reserved.
--
-- Redistribution and use in source and binary forms, with or without
-- modification, are permitted provided that the following conditions are
-- met:
--
--     - Redistributions of source code must retain the above copyright
--       notice, this list of conditions and the following disclaimer.
--
--     - Redistributions in binary form must reproduce the above copyright
--       notice, this list of conditions and the following disclaimer in
--       the documentation and/or other materials provided with the
--       distribution.
--
--     - Neither the name of The Numerical ALgorithms Group Ltd. nor the
--       names of its contributors may be used to endorse or promote products
--       derived from this software without specific prior written permission.
--
-- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
-- IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
-- TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
-- PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
-- OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
-- EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
-- PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
-- PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
-- LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
-- NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
-- SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

—— * ——

```
#include "axiom"

C1: Category == with {
foo: % -> %;
}

C2(X: BasicType): Category == with {
foo: % -> X;
oof: X -> %;
foof: X -> X;

default {
```

```
foof(x: X): X == foo(oof x);
}

}

C3: Category == Join(Ring, C1) with {
foo2: % -> %
}

C4(X: Ring): Category == Join(Ring, C2 X) with {
foo2: % -> %;
}


D1: C1 with {
new: Integer -> %;
} == add {
Rep ==> Integer;
import from Rep;

foo(x: %):   % == x;
new(n: Integer): % == per n;
}

D2(X: BasicType): C1 with {
new: X -> %;
} == add {
Rep ==> X;
foo(x: %): % == x;
new(n: X): % == per n;
}

D3: C2(Integer) with == add {
Rep ==> Integer;
X   ==> Integer;

foo(x: %): X == rep x;
oof(x: X): % == per(x+1);
}

D4(X: BasicType): C2(X) with == add {
Rep ==> X;

foo(x: %): X == rep x;
oof(x: X): % == per x;
}

D5(X: Ring): Ring == X add {
Rep ==> X;
import from Rep;
```

```
(a: %) + (b: %): % == per(rep(a) + rep(b) + 1@Rep);
}
```

References

[1] nothing