

an r l U s r Qui

---

---



Das e i e : c øber 200

a øn røl is øpyrigh c 200 Eurøcøn røl/ aløgg a øn røl is free søf arel yøn can  
re is ribn ei an /ørf ø ify i øn ør ør søf he NU øneral Pøblic øicense as øublishe by

## bil Contents

1	Inrodu fon .....	2
---	------------------	---



12	Næ_ løsing_Na_e	28
12 1	Syn æð	
12 2	c iðn	28
13	Næ_Elabøra iðn_ alls	28
13 1	Syn æð	28
13 2	c iðn	28
13 3	■i i a iðns	28
1	Para_e er_ liasing_	28
1 1	Syn æð	28
1 2	c iðn	2
1 3	■i i a iðn	2
1	Prag_ as	

Examples of using decision rules for optimization programming	6
---	---

## 1 Introduction





## 2 Testing and Control

## 3 Program Usage

### 1 Running dmonrol

a dmonrol is a console line program, i.e. it is callable directly by a system shell, and can be integrated in shells such as PS (see [Section 2](#) [Integrating a dmonrol in a PS](#) page or Emacs (see



**Attribute** is an attribute name including the quote. No overloading information is allowed.

**FullName** designates a single entity or several overloaded entities declared in the same place (as identified by the prefix) while **simple\_name** designates all identifiers with the given name in the program irrespectively of where they appear. **Attribute** designates all occurrences of the given attribute irrespectively of where the attribute applies to.

Utility is provided with a manual to help you find the full name of an entity. See [Section 3.7.1 \[pdf\]](#) page 17.

...1 Overloading name

. . . Enumeration literal 1

























-

## 1.1 Introduction

This chapter describes each rule currently provided by the system. Note that the rules are





```
P ck.Register CB = Pr c'Access};
```

The `Register` line will report on any ask that can be made silently. It is an `unhandled` exception.

## 7. Mi

It is often useful to check that a generic is instantiated only once (at least for a given type in a project). For example, a project may have a special service in charge of releasing printers or  
 strings; it may be useful to check that `nchecked_out` is not used more than once. (leak 310( ha )TJ/F 206 Tf



```

    Entr _In ex |
L   e || |
t   t_N e |
    L   p_N e |
    B || ck_N e |
u   pr gr |
    Pr ce ure |
        Regu || r_Pr ce ure |
        Pr tecte _Pr ce ure |
        Gener || c_F r || _Pr ce ure |
    Funct || n |
        Regu || r_Funct || n |
        Pr tecte _Funct || n |
        Gener || c_F r || _Funct || n |
    Entr |
        T sk_Entr |
        Pr tecte _Entr |
P   ck ge |
    Regu || r_P ck ge |
    Gener || c_F r || _P ck ge |
T   sk |
    T sk_T pe |
    T sk_O Aect |
Pr tecte |
    Pr tecte _T pe |
    Pr tecte _O Aect |
Except || n |
Gener || c |
    Gener || c_P ck ge |
    Gener || c_p |
        Gener || c_Pr ce ure |
        Gener || c_Funct || n

```

## 11. Action



“type” “iscre\_e\_type” “in\_eger\_type” and “equal\_in\_eger\_type” ever if a file kind is preceded by “rxx” rules above it in the hierarchy are not considered (neither for itself nor is



## 1.1 . Action



## .18. Action

This can rules subprogram calls or generic ins and ins here i eren ac ual para e ers call func ions kno n o have si e e ec s. This is angerous prac ice since correc behaviu ay open on a cer ain evalua ion or er of para e ers hich is no specifie by he language

ll func ions en ionne as para e ers in he rule are assu e o in erfere ie he rule ill signal if any of hese func ions is calle ore han once in he para e ers of a call

ll is allowe o give he na e of a generic func ion or of a func ion eclare in a generic package; in his case all func ions resul ing fro ins and ins of hese generics ill be consi ere

In he case of renaings you ns give he na e of he original func ion; he rule ill ork correc ly if he call is a e hrough a renaing of his func ion

Ex:

```
check sll e_e ect_p r eters F1;
check sll e_e ect_p r eters G1, G 2;
```



generic package can be declared constant. It means that no instance of this variable from any instantiation is being written. But bear in mind that this can be true only if all units from the program are analyzed. See [link](#) page 33

It is possible to specify as parameter(s) one or several of the keywords (case irrelevant) **constant**, **write** or **initial** possibly preceded by **not**. The rule will **execute** the inference only for objects that match all the conditions given. The rule can be given once

```
select then rt_sae en c se_thers . aches a when thers pa h in a c se sae
en unn e _exit . aches an exit sae en i h a loop na e ha exi s fr a na e
loop
```

Ex:

```
se rch st te ents e|| };
check st te ents g t , rt};
```

## 2 Unnecessary\_







```
check_equality(t1, t2, ccess_t_sp);
```

Port statements shall not be used

```
check_statement(rt);
```

There shall be only one instantiation of `std::Numerics::GenerateElementaryFunctions` for each floating-point type

```
check_inst_1(t1, t2, ns)-5.5 A. Numerics::GenerateElementaryFunctions for each floating-point type TJ/F5310.9
check
```

There shall be no

search **Algorithms**;

This rule identifies all allocations and can be used to check that all allocated elements are properly deallocated.