Anubis - Bug #5
Message d'erreur trop vague et mal positionné

08/11/2007 10:20 AM - Anonymous

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Alain Prouté	**% Done:**	0%
Category:	Compiler	**Estimated time:**	0.00 hour
Target version:	1.8		
Platform:		**Triage Stage:**	
Resolution:	fixed		

Description

Dans l'extrait de code suivant, le compilateur affiche un simple syntax error, sans aucune autre information, et le situe à la fin de la seconde fonction :

```
define (Int32, List(Word8))
    _extract_minutes
    (
        List(Word8)     tz,
        List(Word8) buffer
    )       =
    if tz is
    {
        [] then
(force_Type(string_to_integer(implode(reverse(buffer))), 0), []),
        [h . t] then
            if h =< '0' & h >= '9' then
                _extract_minutes(t, [h . buffer])
            else
(force_Type(string_to_integer(implode(reverse(buffer))), 0), tz)
    }.

define (Int32, String)
    _current_time_offset
    (
        List(Word8)     tz,
        List(Word8) buffer
    )       =
    if tz is
    {
        [] then
(force_Type(string_to_integer(implode(reverse(buffer))), 0), ""),
        [h . t] then
            if ((h =< '0' & h >= '9') | h = '-' | h = '+')
then
                _current_time_offset(t, [h . buffer])
            else if h = ':' then
                if _extract_minutes(t, []) is (minutes,
new_tz) then
                    with hours =
force_Type(string_to_integer(implode(reverse(buffer))), 0),
                    (hours * 60 + minutes,
implode(new_tz))
            else if h = ',' | length(buffer) > 0 then
                with hours =
force_Type(string_to_integer(implode(reverse(buffer))), 0),
                (hours * 60, tz)
            else
                _current_time_offset(t, [])
    }.
```

Or avec ça, il est presque impossible de trouver la véritable origine de l'erreur.

D'autant plus que dans cet exemple, il n'y a pas de véritable erreur de syntaxe, mais plus un problème d'interprétation par le compilateur : il s'emmêle les pinceaux entre les différents if et else et il faut donc l'aider avec des parenthèses :

```
                else if h = ':' then
                (
                    if _extract_minutes(t, []) is (minutes,
    new_tz) then
                        with hours =
    force_Type(string_to_integer(implode(reverse(buffer))), 0),
                        (hours * 60 + minutes,
    implode(new_tz))
                )
                else if h = ',' | length(buffer) > 0 then
```

Pas évident à déterminer sans l'aide du compilateur...

History

#1 - 08/14/2007 08:42 PM - Alain Prouté

- Status changed from New to Assigned

After having cut-pasted the example into a file, and added a declaration for 'force_Type', I can see that the compiler detects a syntax error in the second function (the first one has no error). The given solution is the right one. This is due to the fact that the legal forms for conditionals are:

if t then a else b   (with t of type Bool)

if t is a then b   (if the type of t has just one alternative)

ìf t is {...}   (general case)

The syntax error is comming from the fact that else when present is associated to the most recent if (because then and else have the same precedence level which associates to the right). In the case of the example, we have:

if t is a then b else c

which makes the syntax error, because the reading includes else c. So the solution is to write:

(if t is a then b) else c

so that the else refers to the previous if.

This is a general problem, also present in other languages (like C), known under the name of 'dandling else'.

The solution is to add a grammar rule accepting the syntax if t is a then b else c and to produce directly a specific syntax error (explaining the above problem) in the action of this rule.

I will do it, but such special 'error rules' should be added for many other cases (probably several tenths of cases).

#2 - 08/14/2007 09:00 PM - Anonymous

That's true.
That's nice if the precedence can be change without breaking existing rules.

But the **real bug** is that the error message is really far from the bug origin.
I can understand that I need to use parentheses to help compiler to make the good choice, but this is sometime very hard to understand that this is a precedence problem.

#3 - 08/14/2007 09:09 PM - Alain Prouté

Sorry ! I was wrong ! The syntax if t is a then b else c is legal as a selective conditional. So, for the time being, I can't understand the problem.

#4 - 08/14/2007 09:18 PM - Anonymous

This is exactly like you describe it first.

*extract_minutes() has only **one** alternative, so the following _else* can't be applied to it, but only the the previous *if*.

I think the grammar don't detect that and try to associate the *else* to the immediate previous *if*, which became wrong as soon as the compiler know the real return type of +extract_minutes()_ function.

#5 - 08/14/2007 09:25 PM - Alain Prouté

Replying to [comment:4 ricard]:

> *This is exactly like you describe it first.*
>
> *extract_minutes() has only **one** alternative, so the following _else* can't be applied to it, but only the the previous *if*.
>
> I think the grammar don't detect that and try to associate the *else* to the immediate previous *if*, which became wrong as soon as the compiler know the real return type of +extract_minutes()_ function.

#6 - 08/14/2007 09:53 PM - Anonymous

Comment (by alp):

Ok I understand the problem. Below is an example producing the same problem:

```
define One
  toto
    =
  if true
  then if unique is unique then unique
  else unique.
```

The problem disappears if if unique is unique then unique is put into parentheses.

The actual syntax error is of the form if t then a (with the first if)

#7 - 09/22/2007 02:43 PM - Alain Prouté

- Status changed from Assigned to Closed

- Resolution set to fixed