Евгений Александрович,

Я провёл трассировку своих юнитов через парсер, постарался максимально исправить код под условия парсера. Тем не менее есть места, которые пришлось оставить "как есть", потому что другого способа пройти проверку без изменения семантики алгоритма - нет. На мой взгляд, код вполне работоспособный и отражает суть работы и принципы. Думаю, что со временем, когда парсер будет претерпевать некоторые изменения и послабления в строгости написания кода на СЛанге, код для базовой системы типов будет корректироваться в соответствии и дальше, а пока - это хороший фундамент для следующих работ выпускников, где этот код можно будет уже прогонять через компиляторы и писать юнит-тесты.

Сейчас я готовлю уже законченную версию на подпись вам в понедельник 14 мая.

Хочу выслать список заметок/уточнений к парсеру :

1. ​//Нужен побитовый левый/правый сдвиг:

if true

then

a >> 1

end

1. if (((0b1 xor 0b0) or 0b1) = 0b0)
2. //мне не удалось найти такой комбинации скобок, чтобы парсер не выдал ошибку в if'e

then

//do something

end

1. if a(BitsNumber) //очень нужная возможность вызова в условии некой рутины

// if this(BitsNumber) = 0b0 // - так тоже не получается

then

a : Integer

end​

1. Обращения к**внутренним полям**юнита из рутин считаются 'Unresolved' - приходится писать this.a  //Такие же проблемы могут​ возникать при объявлениях переменных вне цикла. // или это решается не на этапе парсинга, а на этапе компиляции?
2. Такое ощущение, что оператор простого объявления 'is' не работает.  
   Не получается объявить переменную и тут же её проинициализировать. А также не получается объявить переменную ключевым словом ​is.  
   Ни одна из нижеследующих строк не хочет быть распознана или не вполне удовлетворяет желаемому :
3. // ​a
4. is Intgeer[16].init(Integer.zero)
5. // ​b
6. : Integer[16] is Integer.zero

с : Integer[16] // но в таком случае это дефолтное количество бит

c := Integer.zero​ // и лишь потом приравнивание

// - внутри рутины можно написать две строчки,

// но если мы описываем юнит, хотелось бы описать это одной (и объявление и начальное значение),

// особенно, если это константа

1. //Код

ensure

- old this = this

//Парсер пытается распознать, как

UNARY -

OLD

​EQUAL =

​​LEFT

​​​THIS:6

​​RIGHT

​​​THIS:6

​​​Приходиться ставить скобки или менять местами this = -

(old (this))

1. / (other: as this) : as this is abstract

// No code inside

//end - вот здесь нельзя ставить end - потому что внутри тела рутины ничего нет, соответственно нечему заканчиваться - этот юнит уже абстрактный (Или можно?)

- : as this is abstract

// ​No code here, but below is

ensure

(- this) = old this

end // Вот здесь нужно ставить end - потому что внутри тела абстрактной рутины есть постусловие

1. Правильно ли, что парсер выдаёт  
   PRECONDITIONS  
   для  
   ensure  
   ?​

C уважением,

Закир Шакиров