Naming operations

Да у меня варианты. На текущий момент их два А & B

A: Здесь у нас фиксированный набор знаков операций, известных даже сканеру

RoutineName: [Identifier](#Identifier)|“**()**”|“:=”|([OperatorName](#OperatorName) [[AliasName](#AliasName)])

OperatorName: “**=**”|“**/=**”|”**<=**”|”**>=**”|“**+**”|“**-**“|“**\***”|“**/**”|“**\**”|“**\*\***”|

“**^**”|”**&**”|”|”|**and**|**or**|**not**|**xor**|**implies**|“->”|**and then**|**or else** |“+**=**”|“-**=**”|“\***=**”|“/**=**”|“++” |“--”

B: В этом случае сканер знает только набор символов с клавиатуры и понятие идентификатор. Другими словами – это подход огарниченной гибкости по введению новых операций при помощи специальных символов

------------- May be any sequence of 1 or 2 characters started with a fixed set of symbols???

RoutineName : ( [Identifier](#Identifier) [[Identifier](#Identifier)] )|“**()**”|“:=”|([OperatorName](#OperatorName) [[AliasName](#AliasName)])

OperatorName : [OperatorSign](#OperatorSign) [[OperatorSign](#OperatorSign)]

OperatorSign : “**=**”|“**/**”|”**<**”|”**>**”|“**+**”|“**-**“|“**\***”|“**\**”|“**^**”|”**&**”|”|”|

------------- May be any sequence of 1 or 2 characters started with a fixed set of symbols???

2й подход разрешает вводить операции вида ++ -+ \*\* /= <> && а также greater than или less equal и ограничивает длину имени операции либо не более двух слов, либо не более двух символов из узкого набора. Таким образом повышается с одной стороны гибкость для программиста, но не бесконечно (безконтрольно) и лексический и синтаксический анализаторы не зависят от семантки языка. Опять же интересно твое **инженерное** мнение. Спасибо.