



# Deca Compiler : Manuel Utilisateur

gl35

January 2023

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introduction au compilateur</b>         | <b>2</b> |
| 1.1      | Options du compilateur . . . . .           | 2        |
| 1.2      | Limitations du compilateur . . . . .       | 3        |
| <b>2</b> | <b>Erreurs levées</b>                      | <b>4</b> |
| 2.1      | Erreurs lexicales et syntaxiques . . . . . | 4        |
| 2.2      | Erreurs contextuelles . . . . .            | 4        |
| 2.3      | Erreurs a l'exécution . . . . .            | 8        |
| <b>3</b> | <b>Extension</b>                           | <b>9</b> |
| 3.1      | Ulp . . . . .                              | 9        |
| 3.2      | Cos . . . . .                              | 9        |
| 3.3      | Sin . . . . .                              | 10       |
| 3.4      | Asin . . . . .                             | 10       |
| 3.5      | Atan . . . . .                             | 10       |

# 1 Introduction au compilateur

La documentation utilisateur s'adresse aux utilisateurs du compilateur de langage Deca. Elle donne des informations sur la méthode d'utilisation de ce compilateur et ses différentes spécifications. En plus du compilateur classique Deca, l'extension TRIGO a été ajoutée sous forme d'une bibliothèque "Math.decah" permettant l'utilisation des fonctions trigonométriques (cos, sin, arcsin et arctan) voir section 3. Il est essentiel de lire ce document en cas d'erreur durant la compilation d'un fichier source ou un problème avec la ligne de commande.

## 1.1 Options du compilateur

Le compilateur Deca complet est un exécutable **decac** qui prend en argument un fichier Deca sous la forme `<path/file.deca>`, le résultat est généré sous forme d'un programme assembleur ayant le même nom et dans le même répertoire `<path/file.ass>`. Une exécution est faite par la suite à l'aide du machine abstraite **ima** avec la commande `ima >path/file.ass$>`. La syntaxe d'utilisation de l'exécutable **decac** est :

```
decac [[-p | -v] [-n] [-r X] [-d]* [-P] [-w] <file deca>...]] [-b]
```

La spécification des options du compilateur Deca est la suivante:

- **decac**
  - Affiche les options disponibles.
- **decac -b**
  - Affiche une bannière indiquant le nom de l'équipe.
- **decac -p <file.deca>**
  - Parse le fichier en s'arrêtant à l'étape de la construction de l'arbre et affiche sa décompilation.
- **decac -v <file.deca>**
  - Vérifie le fichier contextuellement et affiche un message d'erreur s'il y en a.
- **decac -n <file.deca>**
  - Exécute le fichier sans tester les débordements (arithmétique, mémoire, et déferencement null).
- **decac -r X <file.deca>**
  - Limite les registres utilisés à  $R_0 \dots R_{X-1}$ , avec X entre 4 et 16.
- **decac -b <file.deca>**
  - Active les traces de debug. Répéter l'option plusieurs fois permet d'augmenter les traces de debug.

- `decac -P <file_1.deca> ... <file_n.deca>`
  - Lance la compilation de plusieurs fichiers en parallèle afin d'accélérer la compilation.
  - Les options décrites ci dessus peuvent être appelées simultanément (sauf `-p` et `-v` incompatibles).
  - Si un fichier apparaît plus qu'une fois sur la ligne de commande, il n'est compilé qu'une seule fois.

## 1.2 Limitations du compilateur

Le compilateur Deca complet proposée a été implémenté conformément au cahier de charge. Parmi ses limitations on cite :

- Les flottants, les entiers et les strings ne peuvent pas dépasser des valeurs précises, si c'est le cas, un message d'erreur est affiché `the given literal is too big` (voir 2.1).
- La taille de la pile et du tas est limitée dans la machine abstrait IMA. En effet, dans le cas de déclaration de plusieurs variables globales ou la création de plusieurs objet avec *new*, une erreur de débordement de tas ou de pile sera levée.
- Utilisation d'une méthode écrite en assembleur non compatible avec l'assembleur de IMA ne génère pas une erreur a la compilation.

## 2 Erreurs levées

### 2.1 Erreurs lexicales et syntaxiques

- `Circular include for <file>`
  - Erreur levée en cas d'une inclusion du fichier lui même ou un autre fichier qui inclus ce même fichier, ainsi une inclusion circulaire.
- `token recognition at <token>'`
  - Erreur levée lorsque le compilateur ne connaît pas le `<token>` (lettre avec accent hors commentaire, caractère @ ..).
- `mismatched input <token> expecting <list>`
  - Erreur levée lorsque le compilateur ne reçoit pas ce qu'il attend (il attendait un élément de la liste suggérée `<list>`) pour former une règle de la grammaire validant la syntaxe.
- `extraneous input <token1> expecting <token2>`
  - Erreur levée lorsque le compilateur reçoit `<token1>` et il ne peut pas avoir une syntaxe valide selon Deca, il propose alors le remplacer par `<token2>` comme solution.
- `missing <token1> at <token2>`
  - Erreur levée lorsqu'on attend un caractère précis a un endroit précis (; a la fin d'une ligne, bloc non fermé, commentaire de la forme `/*Comment`)
- `left-hand side of assignment is not an lvalue`
  - Erreur levée lorsque la valeur la valeur sur cote gauche d'une affectation est invalide ( $-s = 1$ ). Elle doit être une sélection ou bien un identifier.
- `no viable alternative at input <token>`
  - Erreur levée lorsque l'entrée du programme ne correspond a aucune possibilité du grammaire. Il s'arrête au position du token.
- `the given literal is too big`
  - Erreur levée lorsque un littéral de type float, int, ou entier est très grand.
    1. Pour tout flottant f, f doit être entre  $-2^{149}$  et  $(2 - 2^{-23})2^{127}$
    2. Pour tout entier i, i doit être entre  $-2^{31}$  et  $2^{31}-1$
    3. Pour tout string s, s ne doit pas dépasser 100 caractère.

### 2.2 Erreurs contextuelles

| Liste des erreurs de compilation   |   |
|--|---|
| Message d'erreur   | Explication   |
| 1- Identifier of type is not defined.  | Une variable est déclaré avec un type inconnu par le compilateur.   |
| 2- The class already exists.   | Une déclaration de classe s'effectue avec le même nom qu'une classe déjà déclarée.  |
| 3- The given superclass does not exist, or its definition doesn't match Class type.    | Une classe est déclarée avec une classe mère qui n'est pas déjà défini, ou la classe déclarée étend d'un type qui ne correspond pas à un type Classe.   |
| 4- The given attribute has already been declared and with a different type than Field. | La déclaration du field ne peut pas s'effectuer vu que l'identifiant est déjà utilisé par une méthode dans la classe courante, ou une des classes mères.  |
| 5- The given attribute has already been declared.                                      | Une double définition d'un field dans la classe courante.   |
| 6- The field type can't be void.   | Une déclaration d'un field dans une classe sous le type void est illégal.   |
| 7- The return type of the new method isn't a subtype of the one already defined.       | La redéfinition de la méthode est erronée, vu que le nouveau type de retour n'est pas un sous-type du plus ancien.  |
| 8- The signature of the new method isn't the same as the past signature.               | La définition de méthodes avec le même identificateur, même type de retour, dans des classes liées (mère, fille) et avec des signatures différentes est illégale.                               |
| 9- The given method is already defined.  | Les noms des méthodes au sein d'une même classe doivent être distincts.   |
| 10- The given name of the method is already used for a field.                          | Définir une méthode dans une sous-classe avec un identifiant commun avec un champ est illégal.  |
| 11- The given method name is already used.   | Définir une méthode et un field dans une même classe avec un identifiant commun est illégal.  |
| 12- The method is called in the main without an identifier                             | L'appel de méthode dans le main doit s'effectuer sur un identificateur de type classe, avec la méthode appelée devra être défini dans la classe du type de l'identificateur ou une classe mère. |
| 13- The given signature doesn't match with the signature of the used method.           | L'appel de méthode s'effectue sur un identificateur adéquat, mais la signature de la méthode appelée est différente de celle défini dans la classe d'origine.                                   |
| 14- You cannot apply the given method on a such class.                                 | L'appel de méthode s'effectue sur un identificateur invalide, qui ne permet pas la reconnaissance.  |
| 15- You cannot apply the given method to a such type.                                  | L'appel d'une méthode sur un type hors les classes (Float, boolean, int) est illégal.   |

| Liste des erreurs de compilation  |  |
|---|--|
| Message d'erreur  | Explication  |
| 16- The type of a parameter can not be Void.                                  | La signature d'une fonction ne peut pas contenir un paramètre de type void.  |
| 17- The given parameter exists already.                                       | La signature d'une fonction contient une double définition.  |
| 18- A method with void type to return should not have a return statement.     | Toute méthode avec un return devra avoir un type de retour différent de void.  |
| 19- The demanded Cast is illegal due to the given types.                      | Les types donnés pour le cast ne vérifient pas les conditions de compatibilités.   |
| 20- The given type for the Cast is void.                                      | Le cast explicite avec le type void est illégal.   |
| 21- The given type for New isn't compatible.                                  | Le type donné pour l'instanciation n'est pas de type Classe, ce qui bloque la déclaration voulue.  |
| 22- You cannot use Instanceof with the given types                            | Si le type comparé n'est pas une classe ou Null, ou le type comparant n'est pas une classe, l'instanceof ne peut pas s'effectuer.                            |
| 23- A left value cannot be a method   | L'identificateur donné à gauche est de type méthode, ce qui est illégal.   |
| 23- The specification about fields doesn't allow the access to the used field | L'accès à un champ avec une visibilité protégé est illégal sans accesseurs.  |
| 24- The given type for selection isn't a class.                               | La selection demandé s'effectue sur un identificateur de type non associé à une classe.  |
| 25- This cannot be used at the main.  | L'utilisation de this est privée pour les classes, afin de permettre l'accès aux champs et méthodes pour d'une classe ou ses classes mères.                  |
| 26- The expression type doesn't allow the print.                              | Les types qui peuvent être afficher sur la sortie standrad sont: <b>float</b> , <b>int</b> et <b>string</b> .  |
| 27- The type of the given expression is not boolean.                          | Un type booléen est attendu où l'exception est levée, s'il s'agit d'une condition vérifier le type renvoyé de celle-ci.                                      |
| 28- The type of the given expression is not boolean.                          | Un type booléen est attendu où l'exception est levée, s'il s'agit d'une condition vérifier le type renvoyé de celle-ci.                                      |
| 29- The variable type is Void.  | La déclaration d'une variable ne peut pas s'effectuer avec le type void.   |
| 30- Double definition of your symbol.   | Le symbol que vous essayez d'utiliser a été déjà utiliser, essayer d'utiliser un nouveau.  |
| 31- You cannot assign a float to an int without cast.                         | L'affectation d'un float à un type sans cast explicite est illégal, il faut forcément passer par un cast explicite de la forme (int) (..).                   |
| 32- The given types are incompatible.   | Les types donnés sont incompatibles pour l'affectation, vérifier la compatibilité des types spécialement les cas suivant: int avec float, ou le sous-typage. |

| Liste des erreurs de compilation   |  |
|--|--|
| Message d'erreur   | Explication  |
| 33- The arithmetic operation (Not Modulo) cannot be done due to the given types. | Les opérations arithmétiques s'effectuent entre des entiers et des flottants, et dans le cas où l'opération s'effectue entre un int et un float, un cast implicite s'effectue sur le int pour le rendre float. |
| 34- The arithmetic operation modulo cannot be done due to the given types.       | L'opération modulo ne s'effectue qu'entre des entiers.   |
| 35 - The boolean operation cannot be done due to the given types.                | Une opération booléenne ne s'effectue qu'entre des opérandes booléens.   |
| 36 - The given comparison operator cannot be done between two boolean.           | L'opération de comparaison ne peut pas s'effectuer entre deux types booléens .   |
| 37 - The given comparison operator cannot be done between the given types.       | L'opération de comparaison ne s'effectue qu'entre les entiers ou flottants.  |
| 38 - The unary operation (Not) cannot be done due to the given type.             | L'opération Not ne s'effectue que sur un booléen, elle transforme la valeur true en false et vice versa.   |
| 39- The unary operation (Minus) cannot be done due to the given type.            | L'opération du moins unaire s'effectue que sur les entiers ou les flottants.   |

## 2.3 Erreurs a l'exécution

Les étapes de compilation et l'exécution sont très différentes au niveau des erreurs levées. Les erreurs ci-dessous ne sont pas détectées a la compilation mais peuvent être levées a l'exécution.

| Description de l'erreur                                      | Exemple  | Message                                     |
|--|--|---|
| Pile pleine  | trop d'appel récursives ou trop de variable globales déclarées                             | Erreur : Débordement de pile                |
| Débordement arithmétique                                     | opération arithmétique avec des flottants très grand ou division par zero (aussi modulo 0) | Erreur : Débordement opération arithmétique |
| Conversion de type impossible                                | caste un objet a une classe qui derive de la type dynamique de ce objet.                   | Erreur : cast impossible                    |
| Accès à un attribut ou appel d'une méthode sur un objet null | A a =null ; a.x = 1;   | Erreur : dereferencement de null            |
| Une méthode pas void n'ayant pas d'instruction return.       | int getx()[ ]  | Erreur: La méthode ne retourne rien         |
| Entrée invalide  | entrer un entier au lieu d'un flottant (readfloat)   | Erreur :Entrée/Sortie                       |
| Débordement du tas   | trop d'allocation en tas avec new()  | Erreur : Débordement de tas                 |

NB: L'option decac -n ne lève pas les erreurs d'exécution listées ci-dessus.



### 3 Extension

Pour le compilateur Deca, une extension "TRIGO" a été rélisé, elle consiste à implémenter dans une bibliothèque "Math.decah" les fonctions suivantes :

- float *ulp*(float f)
- float *sin*(float f)
- float *cos*(float f)
- float *asin*(float f)
- float *atan*(float f)

Les méthodes créées ont pu être testé en s'aidant des méthodes similaires déjà existantes dans la classe **java.lang.Math**. En effet la précision de chacune d'eux a été évaluée comme suit:

Pour la fonction ULP(Unit In the Last Place), la précision était calculé simplement en faisant la différence en valeur absolue entre la fonction du Deca et celle du Java. Alors que pour le reste des fonctions trigonométriques, les précisions ont été évaluées en fonction de l'ULP, en calculant la différence en valeur absolue entre la valeur trouvée par la fonction Deca et la fonction similaire sur Java, le tout divisé par l'ULP de la vraie valeur (celle calculée en java).

On présente ci-dessous un résumé des résultats trouvés pour les précisions des différentes fonctions implémentées.

#### 3.1 Ulp

Cette méthode donne un résultat sans erreur.

| Intervalle          | Pas                   | Nombre d'échantillons | Erreur entre notre ulp et Math.ulp |
|---------------------|-----------------------|-----------------------|------------------------------------|
| [0; 10]             | $3.8146973 * 10^{-6}$ | 2621440               | 0                                  |
| $[2^{-40}; 2^{-7}]$ | $2^{-28}$             | 2097152               | 0                                  |
| $[2^{-7}; 2^{12}]$  | $9.765625 * 10^{-4}$  | 4194296               | 0                                  |
| $[2^{12}; 2^{40}]$  | $9.765625 * 10^{-4}$  | 4194296               | 0                                  |

#### 3.2 Cos

On a utilisé un développement en série entière pour cette méthde, avec comme intervalle de réduction  $[0; \pi/4]$ , et le reste de l'intervalle de  $[-2\pi; 2\pi]$  se déduit à l'aide des relations trigonométriques de la fonction cos. On note que pour généraliser notre fonction sur tout R, on retranche directement la valeur  $2\pi$  du point où on veut évaluer notre méthode. On a trouvé les résultats suivants:

| Intervalle        | step      | erreur en ULP | max erreur | nombres d'éléments |
|-------------------|-----------|---------------|------------|--------------------|
| $[0; \pi/4]$      | $2^{-18}$ | 0.33533767    | 2.0        | 205888             |
| $[pi/4; \pi/2]$   | $2^{-23}$ | 0.43120888    | 3.0        | 6588398            |
| $[pi/2; \pi]$     | $2^{-10}$ | 0.20397955    | 0.75       | 1609               |
| $[pi; 2\pi]$      | $2^{-10}$ | 0.20091482    | 0.75       | 3217               |
| $[2\pi; 4\pi]$    | $2^{-10}$ | 0.21499515    | 0.625      | 6434               |
| $[50; 50 + 2\pi]$ | $2^{-10}$ | 0.16506171    | 0.37304688 | 6434               |

### 3.3 Sin

On a encore utilisé un développement en série entière pour cette méthode, et on a trouvé les résultats suivants:

| Intervalle        | step      | erreur en ULP | max erreur | nombres d'éléments |
|-------------------|-----------|---------------|------------|--------------------|
| $[pi/4; \pi/2]$   | $2^{-23}$ | 0.43120888    | 3.0        | 6588398            |
| $[pi/4; \pi/2]$   | $2^{-18}$ | 0.42268613    | 2.0        | 205888             |
| $[pi/2; \pi]$     | $2^{-10}$ | 0.22364366    | 0.75       | 1609               |
| $[pi; 2\pi]$      | $2^{-10}$ | 0.22142512    | 0.75       | 3217               |
| $[2\pi; 4\pi]$    | $2^{-10}$ | 0.20667103    | 0.5        | 6434               |
| $[50; 50 + 2\pi]$ | $2^{-10}$ | 0.14266211    | 0.328125   | 6434               |

**NB:** Pour le calcul des précisions en ULP dna sdes intervalles plus grand que pie, pour sin comme pour cos, on divise par  $ulp(value)$  au lieu de  $ulp(\sin(value))$ , vu que  $ulp(\sin(value))$  va donner un ordre de  $10^{-12}$  par exemple, alors que la différence entre notre sin et le sin du java est précis jusqu'à un max de  $10^{-8}$ , ce qui va entrainer des ULP grands de l'ordre de  $10^3(10^{-8}/10^{-12})$ . Alors que lorsqu'on travaille sur Deca avec du hexadécimal on rencontre jamais ce type d'erreur, vu que la representation en hexadécimal du Deca n'est pas limité à 8 chiffres après la virgule.

### 3.4 Asin

La méthode arcsin est implémentée aussi à l'aide d'un développement en série de Taylor poussé à l'ordre 31, qui donne les résultats suivants:

| Intervalle        | step      | erreur en ULP | max erreur | nombres d'éléments |
|-------------------|-----------|---------------|------------|--------------------|
| $[0; 1/\sqrt{2}]$ | $2^{-23}$ | 0.55730605    | 5.0        | 5931642            |
| $[1/\sqrt{2}; 1]$ | $2^{-23}$ | 0.5891706     | 7.0        | 3439330            |

### 3.5 Atan

La méthode atan est implémenté à l'aide des polynômes d'Hermite. Le choix fait été de prendre un polynôme d'ordre 15. L'intervalle de réduction choisi est  $[0, 0.6]$ , et on déduit le reste des valeurs de l'Atan (domaine de définition=  $\mathbb{R}$ ) à l'aide de relations trigonométriques. (Voir documentation pour plus d'informations)

| Intervalle  | step      | erreur en ULP | max erreur | nombres d'éléments |
|-------------|-----------|---------------|------------|--------------------|
| [0.0; 0.59] | $2^{-23}$ | 1.434709      | 8.0        | 4949279            |
| [0.59; 1]   | $2^{-23}$ | 0.5891706     | 7.0        | 3439330            |
| [1; 1000]   | $2^{-7}$  | 0.36701545    | 3.0        | 127872             |