**Master 1 informatique - Montpellier**

**HMIN230-Traitement automatique de la langue naturelle**

**Description de la tâche**

Concevoir et écrire dans le langage de programmation de votre choix (Java, python, C, C++, PHP,  …) un petit programme permettant à partir d'un terme donné par l'utilisateur de produire tous les termes possibles par dérivation morphologique. Les termes produits peuvent  exister dans la langue (français) ou ne pas exister. Si un terme produit existe, iil peut éventuellement avoir dans la langue une sémantique différente de celle que vous vouliez.

Votre programme devra interpréter des règles qui seront stockée dans un fichier à part (regles.txt). Vous devrez créer et remplir ce fichier, en vous inspirant des exemples de règles ci- après dans ce document.

**Schémas de transformations morphologiques**

Modèle :

X ⇒  Y //   “commentaire + exemple  ⇒ contraintes sémantiques sur X et/ou Y

(à compléter)

(attention certaines règles ci-dessous ont un format non standard)

**Lien de ce document**

https://docs.google.com/document/d/1OjL0L32R5TB5\_GFC3QDlnWkmWg\_1gxoRi\_U6ZaWn9QI/edit?usp=sharing

**Exemples (non exhaustif)**

(attention certaines règles ci-dessous ont un format non standard)

\*er ⇒   \*age // laver ⇒ lavage

\* er ⇒ \*age & r\_pos == N & r\_isa == action

\*er ⇒  \*ement // laver ⇒  lavement

\* er ⇒ \*ement & r\_pos == N & r\_isa == action

\*er ⇒  \*eur & r\_isa personne //  “verbe vers agent”

// (exemple : miner => mineur)

\*er ⇒  \*euse //  “verbe vers agent”

// (exemple : chanter => chanteuse) r\_isa Personne

\*ger ⇒  \*geoire //  “verbe vers lieu” (exemple : manger => mangeoire)

\*ger ⇒ \*goire & r\_pos == N & r\_isa lieu

\*er ⇒  \*oire’ //  “verbe vers lieu”

// (exemple : patiner => patinoire) r\_lieu

\*er ⇒  \*oir’ //  “verbe vers lieu” // (exemple : égoutter => égouttoir)

\*er ⇒  \*age’ //   “verbe vers nom d’action” // (exemple : bavarder => bavardage)

\*er & pos == V ⇒ \*e  & pos = N

\*er & pos == V  ⇒ \*ant & pos = ADJ

\*ner & pos == V ⇒ \*nier & pos = N

\*er  & pos == V ⇒ \*ette & pos = N

\*uire ⇒  \*uction’ //  “verbe vers nom” (exemple : induire => induction)

\*uire ⇒  \*ucteur’ // “verbe vers agent” (exemple : conduire => conducteur)

\*uire ⇒  \*uctrice’ //   “verbe vers agent” (exemple : conduire => conductrice)

Verbe:ister ⇒ nom:istance

verbe:er ⇒ nom:able

\*ger ⇒  \*gement’  //   “verbe vers nom” (exemple : déranger => dérangement)

verbe:ger ⇒ nom:geur

verbe:ner ⇒ nom:naison

verbe:quer ⇒ nom:cation

verbe:ner ⇒ nom:naison

verbe:ser ⇒ nom:sation

verbe:quer ⇒ nom:quage

Verbe:ter:nom:tement

Verbe:dir ⇒ nom:d

Verbe:lir ⇒ nom:lissage

verbe:er ⇒ nom:ant

verbe:ller ⇒ nom:l

Verbe:uer ⇒ nom:cation

Nom:” “ ⇒ adj:able

Verbe:”er” ⇒ adj:ible

Nom:” “ ⇒ Nom:”age”

Nom:”ier” ⇒ Nom:”eraie”

Verbe:”er” ⇒ Nom:”ail”

Verbe:”er ⇒ Nom:”aille”

adj:””:adv:”ante”

Nom:” “ ⇒ nom:”aison”

Nompropre:” “ ⇒ nom:”oise”

Verbe:” “ ⇒ nom:”ateur”

Verbe:”er” ⇒ nom:”ation”

Nom:” “ ⇒ nom:”isme”

Nom:” “ ⇒ nom:”iste”

Nom:” “ ⇒ verbe:”ir”

Verbe:” “ ⇒ nom:”iteur”

nom:eur ⇒ nom:euse

nom:er ⇒ nom:ance

nom:" " ⇒ nom:aire

nom:" " ⇒ nom:iaire

nom:" " ⇒ nom:ale

nom:" " ⇒ nom:al

nom:" " ⇒ adj:al

nom:"ale" ⇒ nom:ales

nom:"al", nom:aux

nom:" " ⇒ adj:ale

nom: " " ⇒ adj:elle

nom" " ⇒ adj:acé

nom:""  ⇒ adj:an

nom:""  ⇒ adj:ane

Adj:ant:adv:amment r\_manner

adj:ante ⇒ adv:amment

adj:tif:nom:tivité

adv:ant:nom:ance

adv:ante:nom:ance

nom:"":adj:ain

nom:"":adj:aine

nom:""nom"ard"

nom:"" ⇒ :nom"arde"

**info diverses**

pour recup des info de JDM lisez  bien la page  
<http://www.jeuxdemots.org/jdm-about.php>

et utilisez  ce formulaire  
<http://www.jeuxdemots.org/rezo-dump.php>