

# IA01 — TD05

---

**OBJECTIF :** programmer un moteur d'ordre 0+ en utilisant une stratégie de type chaînage-arrière.

**PROGRESSION :** on commencera par s'intéresser au cas d'un moteur d'ordre 0.

## 1 ORDRES

Rappeller ce qu'est un moteur d'ordre 0, 0+ et 1.

## 2 CHAÎNAGE-AVANT ET CHAÎNAGE-ARRIÈRE

Faire une simulation papier d'une recherche en chaînage-avant, puis en chaînage-arrière, avec les données suivantes :

**BASE DE RÈGLES :**

<b>R1</b>	$B \text{ et } D \text{ et } E \rightarrow F$
<b>R2</b>	$D \text{ et } G \rightarrow A$
<b>R3</b>	$C \text{ et } F \rightarrow A$
<b>R4</b>	$C \rightarrow D$
<b>R5</b>	$D \rightarrow E$
<b>R6</b>	$A \rightarrow H$
<b>R7</b>	$B \rightarrow X$
<b>R8</b>	$X \text{ et } C \rightarrow A$

**BASE DE FAITS :** B, C.

**BUT :** H.

## 3 REPRÉSENTATION

Proposer une représentation de la base de règles et de la base de faits.

## 4 ALGORITHME

Écrire un algorithme réalisant un chaînage arrière en profondeur d'abord.

## 5 RÉALISATION

Implémenter cet algorithme en LISP.

Quelles évolutions faudrait-il apporter pour passer en 0+ ?