

1. Chainage Avant :

Paramètres : in fait (le fait à démontrer)

```
if fait  $\in$  BF then
  res  $\leftarrow$  SUCCES
else
  reglesNonDéclenchées  $\leftarrow$  BR; reglesAConsidérer  $\leftarrow$  BR; res  $\leftarrow$  ECHEC
  while reglesAConsidérer  $\neq \emptyset$  et res  $\neq$  SUCCES do
    r  $\leftarrow$  choisir(reglesAConsidérer); reglesAConsidérer  $\leftarrow$  reglesAConsidérer - {r}
    if  $\forall p \in$  premisses(r), p  $\in$  BF then
      BF  $\leftarrow$  BF  $\cup$  {conclusion(r)}
      reglesNonDéclenchées  $\leftarrow$  ReglesNonDéclenchées - {r}
      reglesAConsidérer  $\leftarrow$  reglesNonDéclenchées
      if conclusion(r) = fait then
        res  $\leftarrow$  SUCCES
      end if
    end if
  end while
end if
renvoyer res
```

Attention

- ⚡ Une règle n'est déclenchée qu'une seule fois.
- ⚡ L'algorithme termine toujours
- ⚡ On ne peut démontrer que des faits atomiques ou des disjonctions de faits atomiques.
- ⚡ Les conclusions des règles peuvent être des faits négatifs.

Différentes situations possible en cas d'utilisation de faits négatifs :

$f \in \mathcal{BF}$ Le fait est établi.

$\neg f \in \mathcal{BF}$ La négation du fait est établie.

$f \notin \mathcal{BF}$ et $\neg f \notin \mathcal{BF}$ Ni le fait ni sa négation ne sont établis.

$f \in \mathcal{BF}$ et $\neg f \in \mathcal{BF}$ La base de faits est incohérente.

2. Chainage Arrière :

Fonction **chaînageArrière**

Paramètres : in BR, in BF, in listeButs.

```
if estVide(listeButs) then
  res ← SUCCES
else
  if demBut(premier(listeButs)) then
    res ← chaînageArrière(suite(listeButs))
  else
    res ← ECHEC
  end if
end if
retourner res
```

Fonction **demBut**

Paramètres : in BR, in BF, in but.

```
if but ∈ BF then
  res ← SUCCES
else
  regles ← BR; res ← ECHEC
  while regles ≠ ∅ et res ≠ SUCCES do
    r ← choix(regles); regles ← regles - {r}
    if But ∈ conclusion(r) then
      res ← chaînageArrière(BR, BF, premise(r))
    end if
  end while
  retourner res
end if
```

Attention

- ⊗ Une règle peut être utilisée plusieurs fois.
- ⊗ On peut démontrer des faits atomiques ou des conjonctions de faits atomiques

3. Chainage mixte :

Fonction **chaînageMixte**

Paramètres : in BR, in BF, in but.

```
while but n'est pas déduit, mais peut encore l'être do
  Saturer la base de faits par chaînage avant.
  if but n'est pas dans la base de faits then
    Chercher une question pertinente à poser à l'utilisateur
    Poser la question à l'utilisateur
    Ajouter la réponse à la base de faits
  end if
end while
```