

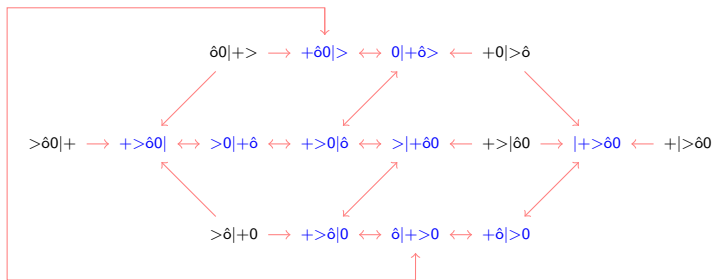
Intelligence Artificielle

Construire un chemin

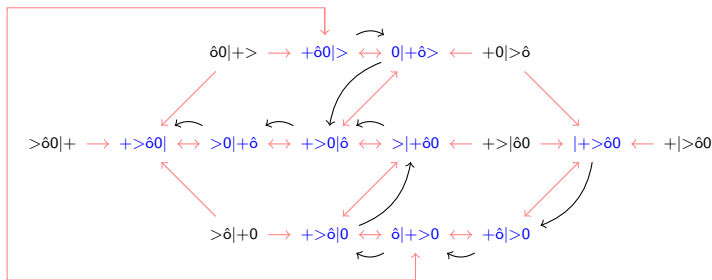
José Vander Meulen

12 octobre 2017

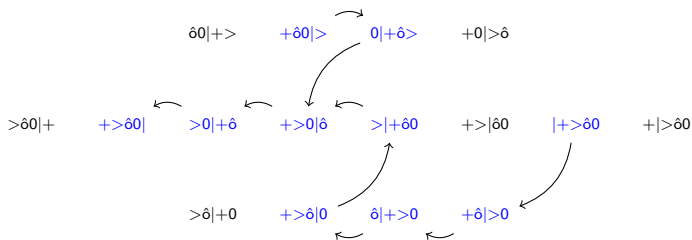
Le loup, la chèvre et le chou : trouver un chemin



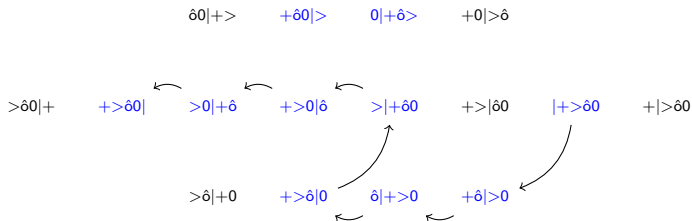
Le loup, la chèvre et le chou : construire un arbre couvrant



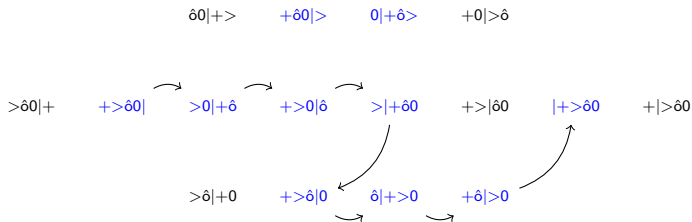
Le loup, la chèvre et le chou : construire un arbre couvrant



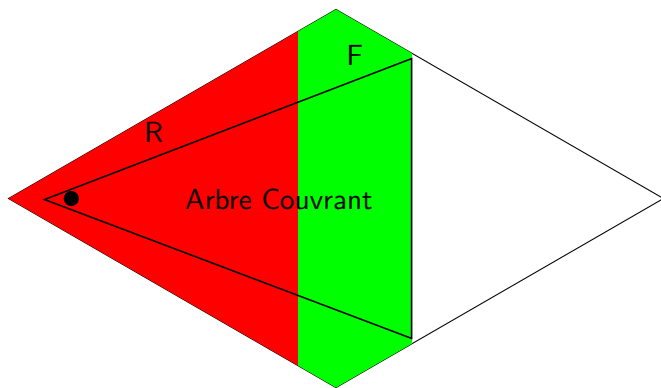
Le loup, la chèvre et le chou : isoler un but



Le loup, la chèvre et le chou : construire le chemin



Calculer les états atteignables : invariant



Algorithme et couple

(x, y)

FIND-PATH

```
procedure FIND-PATH()  
    (found, g) := CREATE-SPANNING-TREE()  
    if found then  
        g := REVERSE-PATH(g)  
    end if  
    return (found, g)  
end procedure
```

CREATE-SPANNING-TREE

procedure CREATE-SPANNING-TREE()

RF := {initial state}

F := {initial state}

found := goal{initial state}

while $F \neq \emptyset \wedge \neg \text{found}$ **do**

c := a state of F

T := succ(c) \ RF

found := goal(T)

F := F \ {c}

F := F \cup T

RF := RF \cup T

ADDPARENT(T, s)

end while

g := FIND-GOAL(RF)

return (found, g)

end procedure

ADDPARENT

```
procedure ADDPARENT(H, p)
  while H  $\neq \emptyset$  do
    s := a state of H
    H := H  $\setminus$  {s}
    création de la transition s  $\rightarrow$  p
  end while
end procedure
```

REVERSE-PATH

```
procedure REVERSE-PATH( $g$ )  
  while  $g$  a un parent do  
     $p :=$  le parent de  $g$   
    création de la transition  $p \rightarrow g$   
     $g := p$   
  end while  
  return  $g$   
end procedure
```