

## Esercizio / Riflessione sull'Unità 2

1. *Coniglio(R)*
  2.  $\exists y \text{ Levriero}(y) \wedge (\forall z \text{ Coniglio}(z) \rightarrow \text{PiuVeloce}(x,z))$
  3.  $\forall y \text{ Levriero}(y) \rightarrow \text{Cane}(y)$
  4. *Cavallo(F)*
  5.  $\forall x \forall y \text{ Cavallo}(x) \wedge \text{Cane}(y) \rightarrow \text{PiuVeloce}(x,y)$
  5.  $\forall x \forall y \forall z \text{ PiuVeloce}(x,y) \wedge \text{PiuVeloce}(y,z) \rightarrow \text{PiuVeloce}(x,z)$
- MP: modus ponens*  
*MT: modus tollens*  
*AE: and elimination*  
*AI: and introduction*  
*UI: univ. instantiation*  
*EI: exist. instantiation*

Posso definire un algoritmo per la derivazione sistematica di nuove formule, con l'apparato deduttivo di G&N?

**Separazione fra 'base di conoscenza' e 'motore inferenziale'**

- |   |   |   |
|---|---|---|
| <ul style="list-style-type: none"> <li><i>UI: univ. instantiation</i></li> <li><i>EI: exist. instantiation</i></li> <li><i>AE: and elimination</i></li> </ul> | } | regole che operano su singole formule   |
| <ul style="list-style-type: none"> <li><i>MP: modus ponens</i></li> <li><i>MT: modus tollens</i></li> <li><i>AI: and introduction</i></li> </ul>              | } | regole che operano su coppie di formule |

1. *Coniglio(R)*
  2.  $\exists y \text{ Levriero}(y) \wedge (\forall z \text{ Coniglio}(z) \rightarrow \text{PiuVeloce}(x,z))$
  3.  $\forall y \text{ Levriero}(y) \rightarrow \text{Cane}(y)$
  4. *Cavallo(F)*
  5.  $\forall x \forall y \text{ Cavallo}(x) \wedge \text{Cane}(y) \rightarrow \text{PiuVeloce}(x,y)$
  6.  $\forall x \forall y \forall z \text{ PiuVeloce}(x,y) \wedge \text{PiuVeloce}(y,z) \rightarrow \text{PiuVeloce}(x,z)$
- UI: univ. instantiation*  
*EI: exist. instantiation*  
*AE: and elimination*  
*MP: modus ponens*  
*MT: modus tollens*  
*AI: and introduction*

- 1: nessuna regola applicabile  
 2: EI, con nessun vincolo sul valore di y (7); UI, con z/R (8); AE (9,10)  
 3: UI, con y/valore nella 7 (11); MP (12)  
 ....

**Se provate ad applicare questa strategia 'grezza', troverete la soluzione in modo meno efficiente di quanto abbiamo visto nella scorsa lezione.**

Il 'motore inferenziale' deve essere progettato in modo da realizzare una strategia il più possibile efficiente, dati dei vincoli sul problema rappresentato.

1. *Coniglio(R)*
  2.  $\exists y \text{ Levriero}(y) \wedge (\forall z \text{ Coniglio}(z) \rightarrow \text{PiuVeloce}(x,z))$   
 EI:  $\text{Levriero}(L) \wedge (\forall z \text{ Coniglio}(z) \rightarrow \text{PiuVeloce}(L,z))$   
 AE:  $\text{Levriero}(L); (\forall z \text{ Coniglio}(z) \rightarrow \text{PiuVeloce}(L,z))$   
 UI:  $\text{Coniglio}(R) \rightarrow \text{PiuVeloce}(L,R)$   
 MP:  $\text{PiuVeloce}(L,R)$
  3.  $\forall y \text{ Levriero}(y) \rightarrow \text{Cane}(y)$   
 UI:  $\text{Levriero}(L) \rightarrow \text{Cane}(L)$   
 MP:  $\text{Cane}(L)$
  4. *Cavallo(F)*  
 $\forall x \forall y \text{ Cavallo}(x) \wedge \text{Cane}(y) \rightarrow \text{PiuVeloce}(x,y)$   
 UI:  $\text{Cavallo}(F) \wedge \text{Cane}(L) \rightarrow \text{PiuVeloce}(F,L)$   
 AI:  $\text{Cavallo}(F) \wedge \text{Cane}(L)$   
 MP:  $\text{PiuVeloce}(F,L)$
  5.  $\forall x \forall y \forall z \text{ PiuVeloce}(x,y) \wedge \text{PiuVeloce}(y,z) \rightarrow \text{PiuVeloce}(x,z)$   
 UI:  $\text{PiuVeloce}(F,L) \wedge \text{PiuVeloce}(L,R) \rightarrow \text{PiuVeloce}(F,R)$   
 AI:  $\text{PiuVeloce}(F,L) \wedge \text{PiuVeloce}(L,R)$   
 MP:  $\text{PiuVeloce}(F,R)$
- MP: modus ponens**  
**MT: modus tollens**  
**AE: and elimination**  
**AI: and introduction**  
**UI: univ. instantiation**  
**EI: exist. instantiation**