Logica del prim'ordine

"Rappresentiamo mondi complessi in cui gli oggetti sono in relazione gli uni con gli altri e studiamo come ragionare su queste rappresentazioni più espressive"

Pregi della logica proposizionale

- La logica proposizionale è dichiarativa, cioè:
  - Separa nettamente conoscenza da inferenza
  - Consente di derivare fatti da fatti
  - La sua semantica è data da una relazione di verità che collega formule e mondi possibili
- È composizionale: il significato (valore di verità) di una formula è ottenuto componendo il significato (valore di verità) delle sue parti
- Non è ambigua

Limiti della logica proposizionale

- La logica proposizionale non permette rappresentazioni compatte, esempio:

  ascoltaMusica(mia) ⇐ felice(mia)
  ascoltaMusica(jody) ⇐ felice(jody)
  ascoltaMusica(yolanda) ⇐ felice(yolanda)

- Non esiste un modo per codificare un'espressione generale, tipo "una persona, quando è felice, ascolta della musica"
- Non permette di esprimere relazioni fra elementi, es: padre(x, y)
- Si dice che la logica proposizionale manca di espressività

Tante logiche, qualche esempio

- **Logica temporale**:
  permette di rappresentare e ragionare sul tempo, esempio “A non è vero finché B non diventa vero”, “Quando A è vero subito dopo B sarà vero”
- **Logica epistemica** (della conoscenza):
  permette di esprimere relazioni come “l’agente i sa A”o “tutti sanno A”e di ragionare sulle implicazioni
- **Logica deontica** (normativa):
  permette di esprimere obblighi, permessi, proibizioni, commitment e di ragionare su di essi
- **Logica fuzzy** (a valori sfumati):
  introduce e ragiona su gradi di verità. I valori di verità appartengono all’intervallo [0,1]. Può esprimere efficacemente concetti come vecchio(X), per esempio.

Proposizionale → prim'ordine

- Vogliamo **mantenere le buone caratteristiche della logica proposizionale** (semantica dichiarativa e composizionale, non ambiguità, indipendenza dal contesto)
- Vogliamo aggiungere la possibilità di esprimere relazioni fra oggetti, esempi:
  - **Oggetti**: mia, jody, butch, fido, orchidea1025, matita11, ...
  - **Relazioni**: genitoreDi, affidatarioDi, proprietarioDi, maggioreDi, adiacenteA, rosso, verde, partecipante, ...

Proposizionale → prim'ordine (FOL)

- Logica proposizionale: ogni fatto o è vero o è falso
- Logica del prim'ordine: il mondo è fatto di oggetti in relazione fra di loro, una relazione può essere verificata oppure no
- La logica del prim'ordine è indicata dalla sigla **FOL** (First-Order Logic)

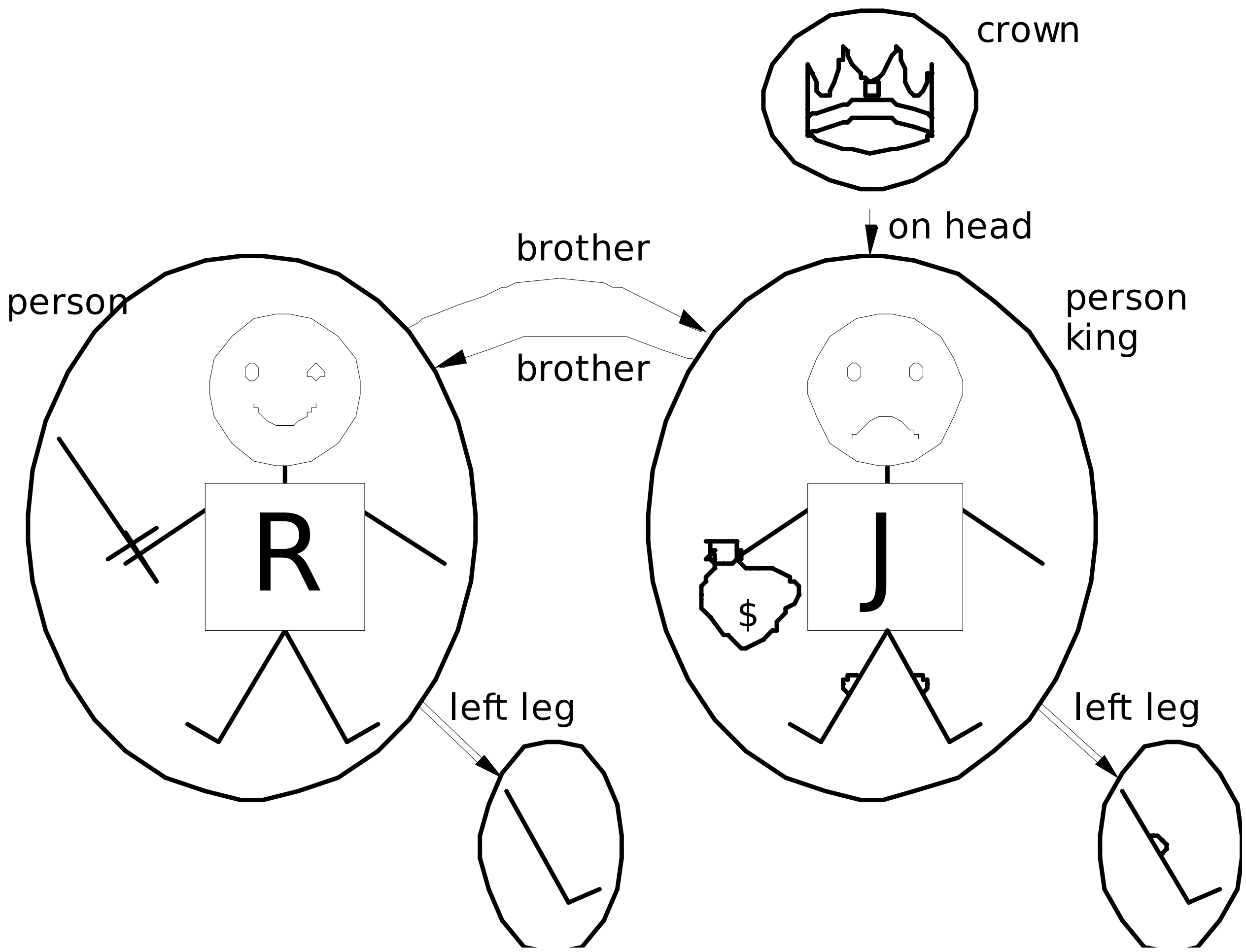
Proposiziona → prim'ordine

- **Modello proposizionale**:
  attribuzione di valori di verità ai fatti (simboli proposizionali)
- **Modello prim'ordine**:
  contiene un **dominio**, cioè l'insieme degli oggetti del mondo considerati, e delle **relazioni** fra tali oggetti (vedremo meglio più avanti)

Elementi della logica del prim'ordine

• Simboli di costante	Richard, John, 2, Gamba, Corona
• Simboli di predicato	Fratello, <, >, SullaTesta
• Simboli di funzione	+, Antenato
• Simboli di variabile	x, y, z
• Connettivi	$\Rightarrow$ $\Leftrightarrow$ $\wedge$ $\vee$ $\neg$
• Uguaglianza	=
• Quantificatori	$\forall$ $\exists$
• Punteggiatura	( ) ,

Esempio: Riccardo Cuor di Leone



Esempio

- Un **dominio di riferimento** è astratto in:
  - un insieme di oggetti (ognuno caratterizzato dalla propria **identità**)
  - Un insieme di relazioni ognuna espressa come **insieme di tuple**
- Una **relazione** è un insieme di tuple costituite da oggetti del dominio
  - per esempio:
    { <Giovanni senza terra, Riccardo cuor di leone>,
    <Riccardo cuor di leone, Giovanni senza terra> }
    che dice che Giovanni è in relazione con Riccardo e Riccardo è in relazione con Giovanni. Questo insieme ci dice chi è fratello di chi nel dominio considerato

Predicati e funzioni

- Le relazioni sono insiemi di tuple ognuna delle quali collega oggetti del dominio, si dividono in:
  - **Funzioni**: dato un insieme di oggetti restituiscono un oggetto
    esempio: più(3, 5) oppure parent(X)
  - **Predicati**: dato un insieme di oggetti ne catturano una proprietà, restituiscono vero o falso
    esempio: Uomo(Andrea) oppure Accanto(X, a)

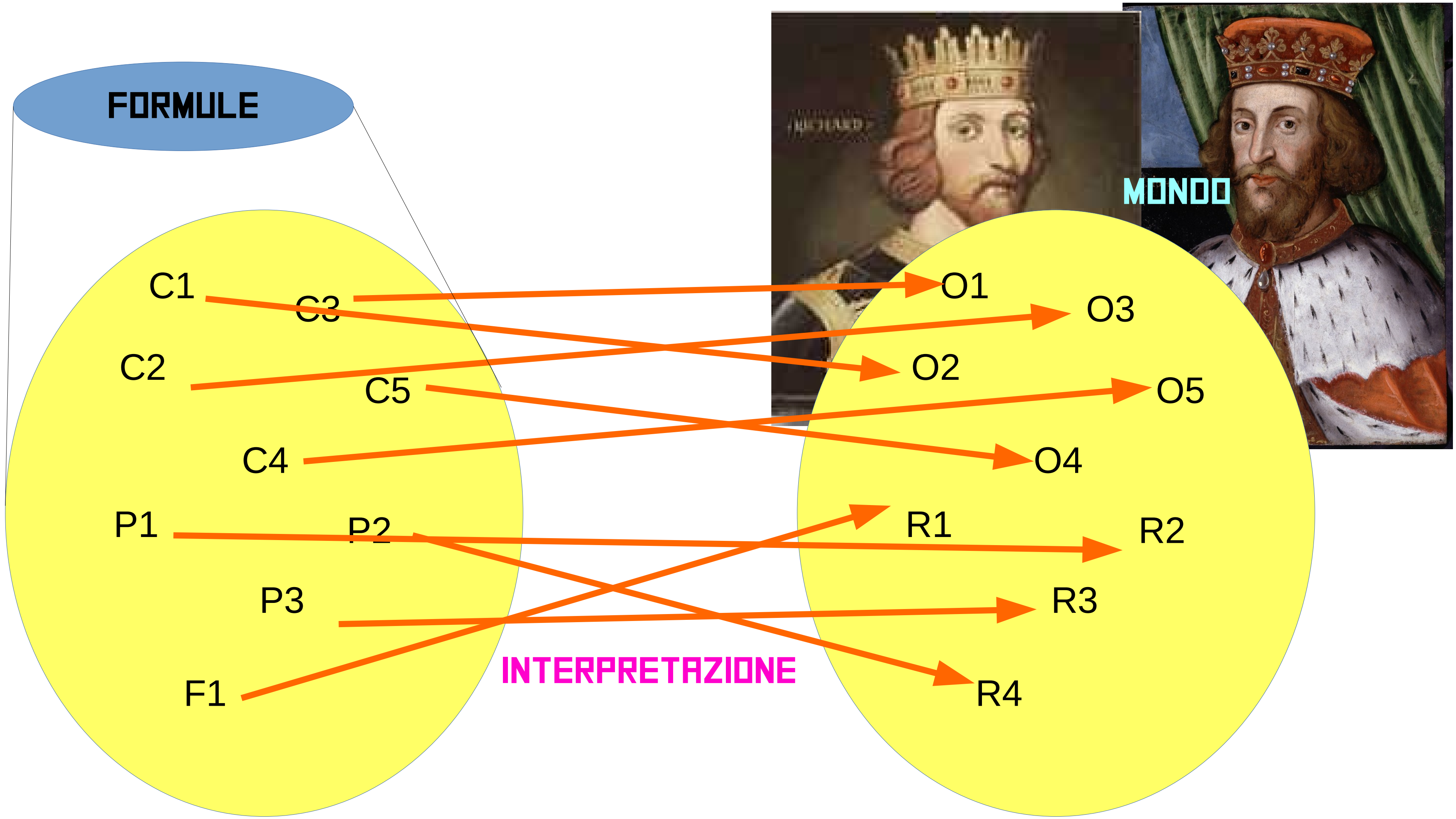
Logica prim'ordine

- Grammatica:
  - formula → formulaAtomica | (formula connettivo formula) | quantificatore variabile, … formula | ¬ formula
  - formulaAtomica → predicato(termine, …) | termine=termine
  - termine → funzione(termine, …) | costante | variabile
  - connettivo → ⇒ | ⇔ | ∧ | ∨
  - quantificatore → ∀ | ∃
  - costante → X | Y | John | Corona | …
  - variabile → x | y | …
  - predicato → PrimaDi | ColoreDi | Piove | …
  - funzione → Madre | GambaSinistra | …

Logica prim'ordine

- simboli:
  - costante → X | Y | John | Gamba | …
  - predicato → PrimaDi | ColoreDi | Piove | …
  - funzione → Madre | GambaSinistra | …
- Nel libro sono scritti iniziando con una maiuscola
- Simboli di predicato e di funzione hanno un'**arità** (numero di parametri)
- Tutti i simboli hanno **un'interpretazione**

Interpretazione

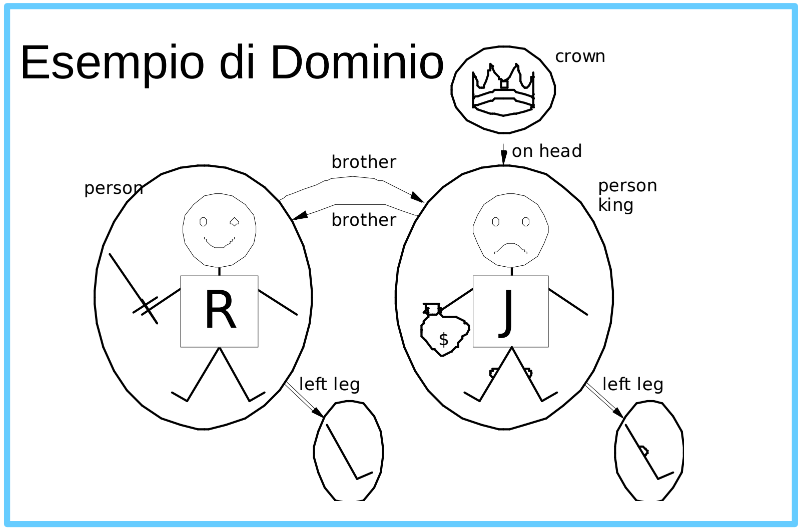


SIMBOLI
(costanti, predicati, funzioni)

DOMINIO DEL DISCORSO
oggetti e relazioni

Interpretazione

- Un modello è una coppia **M =(D, I)**, dove **D** è il dominio del discorso e **I** è un'interpretazione. L'interpretazione è il fondamento per determinare il valore di verità delle formule. È un'associazione fra i simboli e gli oggetti del dominio del discorso



- **Esempio di interpretazione**:
  - John → Giovanni senza terra
  - Richard → Riccardo cuor di leone
  - Fratello → relazione che lega i figli degli stessi genitori (e non per esempio a quella dei nodi di un albero o di fratellanza fra monaci)
  - Fratello(John, Richard) è vera in M se <Richard, John> appartiene alla relazione Fratello. Nell'esempio ciò accade, quindi la formula atomica sarà vera

Interpretazione

- **NB**: se cambio coerentemente i **simboli** le formule non cambieranno valore di verità
- **Esempi**:
  - ~~John~~ **Usurpatore →** Giovanni senza terra
  - ~~Richard~~ **Re →** Riccardo cuor di leone
  - ~~Fratello~~ **Brother →** relazione di fratellanza fra esseri umani
  - **Brother(Usurpatore, Re)** è vera

Interpretazione

- **NB**: se invece cambio **l'interpretazione** dei simboli le formule potranno cambiare valore di verità
- **Esempi**:
  - John ↣ **Corona**
  - Richard ↣ **Giovanni senza terra**
  - Fratello ↣ relazione di fratellanza fra esseri umani
  - **Fratello(John, Richard)** è falsa, Giovanni e la corona non erano fratelli

Interpretazione

- **NB**: attenzione che anche **l'interpretazione** dei simboli di funzione e di predicato può cambiare
- **Esempi**:
  - John ↣ **Giovanni senza terra**
  - Richard ↣ **Riccardo cuor di leone**
  - Fratello ↣ **relazione di fratellanza fra monaci**
  - **Fratello(John, Richard)** è falsa, Giovanni e Riccardo non erano confratelli

Interpretazione

- **NB**: se cambio il **dominio del discorso** dovrò per forza di cose cambiare l'interpretazione dei simboli e le formule potranno cambiare valore di verità
- **Esempi**:
  - John ↣ **Nobunaga**
  - Richard ↣ **Yoshimune**
  - Fratello ↣ relazione di fratellanza fra esseri umani
  - **Fratello(John, Richard)** è falsa, Nobunaga e Yoshimune non erano fratelli

Modello nella logica del prim'ordine

- Un modello M è una coppia (D, I) dove D è un dominio e I un'interpre-tazione
- D contiene un numero di oggetti maggiore o uguale a 1 (elementi del dominio) e le loro relazioni
- I specifica i riferimenti per:
  - **Simboli costanti** → elementi del dominio
  - **Simboli di predicato** → relazioni che catturano proprietà fra elementi del dominio
  - **Simboli di funzione** → relazioni funzionali fra gli oggetti del dominio
- Come nella logica proposizionale M è un modello di α se α è vera in M

Soddisfacibilità

- Una formula è soddisfacibile quando esiste almeno un modello che la rende vera
- È valida quando è vera in tutti i modelli
- È insoddisfacibile quando non è mai vera

Enumerazione dei modelli e conseguenza logica

- Consideriamo un insieme di formule e un **dominio D** di riferimento
- Ogni **interpretazione I** dei simboli crea un diverso **modello**
- Nella logica proposizionale avevamo 2^N modelli con N numero dei simboli proposizionali
- **Nel prim'ordine quanti modelli si hanno?**

Come si crea l'enumerazione dei modelli in FOL?

Per ogni possibile **numero di elementi del dominio n** da **1** a **∞**

Per ogni **predicato Pk** k-ario nel vocabolario

Per ogni possibile **relazione k-aria** su **n oggetti**

Per ogni **simbolo costante C** nel vocabolario

Per ogni **scelta di riferimento di C su n oggetti**

...

Enumerazione dei modelli e conseguenza logica

- In sunto i **modelli** M di un insieme di formule del prim'ordine **possono essere infiniti** perché:
  - se il dominio D è un insieme illimitato e se qualche formula P dell'insieme considerato contiene dei quantificatori, per determinarne il valore di verità sarebbe necessario calcolare il valore di verità delle infinite formule che si ottengono da P sostituendo alle variabili quantificate gli infiniti elementi di D
- **Conseguenza**: in generale nel prim'ordine è **impossibile** verificare la conseguenza logica (così come validità e insoddisfacibilità) tramite enumerazione dei modelli

Termini

- **termine → funzione(termine, ...) | costante | variabile**
- È un'espressione logica che si riferisce a un oggetto
- **Costanti**: danno un nome a oggetti di uso comune
- **Funzioni**:
  - permettono di riferirsi a oggetti che non hanno un nome proprio
  - **NB: non costruiscono l'oggetto restituito!**
  - **GambaSinistra(John)** è un modo per riferirsi all'oggetto del dominio gamba sinistra dell'oggetto del dominio identificato da John, non occorre sapere cosa sia una gamba sinistra o come sia fatta per ottenere questo risultato

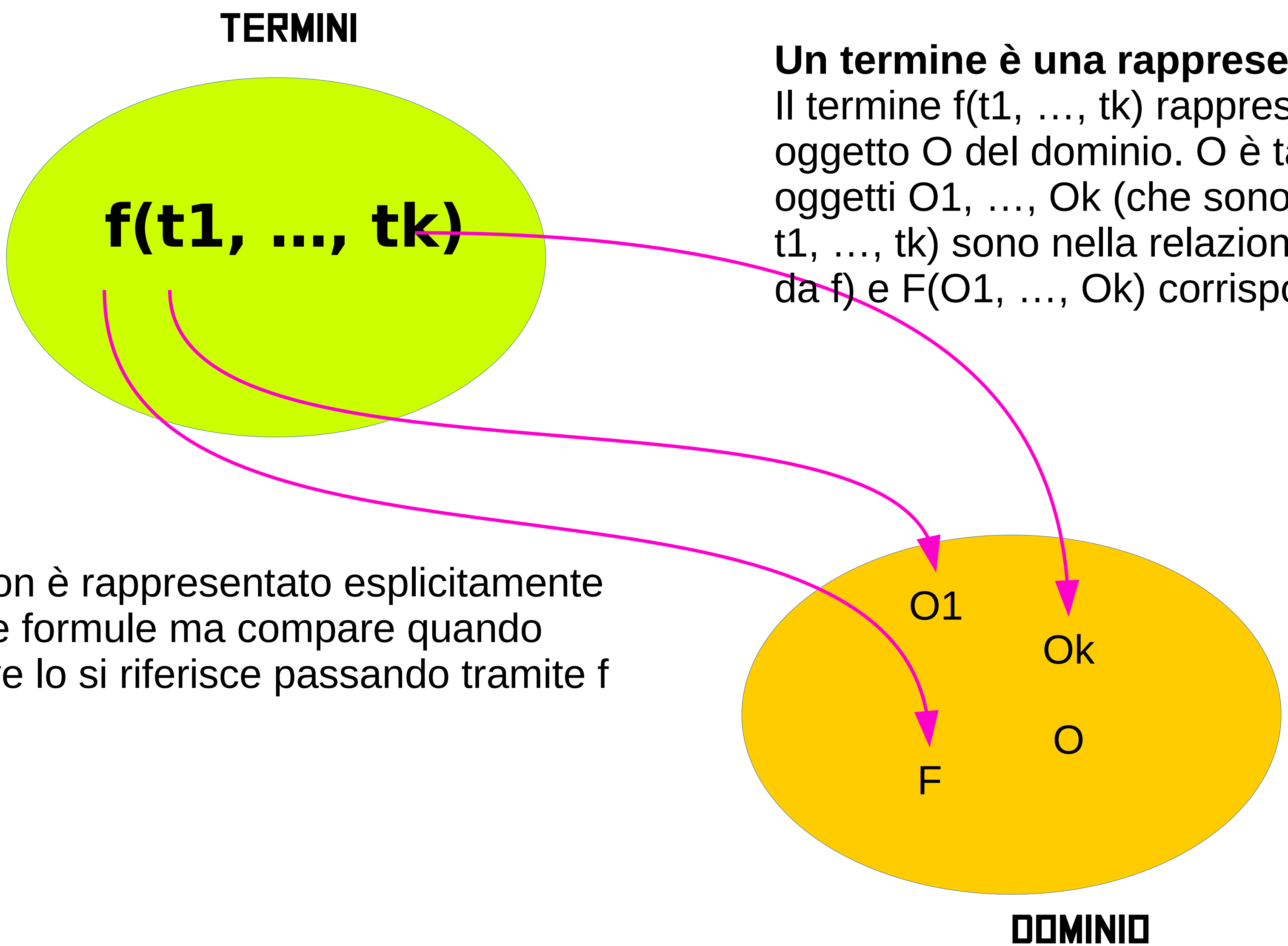
Termine ground

- Un termine è **ground** quando non contiene variabili
- Esempi
  - GambaSinistra(John)
  - Richard
  - Corona
  - Fratello(John, Richard)

Interpretazione dei termini

- È il processo tramite il quale si passa dalla scrittura di un termine all'oggetto che questo identifica
- Se il termine è una **costante** l'identificazione è immediata
- Se il termine è dato tramite una **funzione f(t1, …, tk)**:
  - f si riferisce a una funzione F del modello
  - t1, …, tk sono a loro volta dei termini
  - L'interpretazione è un processo ricorsivo:
    - prima si associa a ogni tj l'elemento del dominio a cui fa riferimento
    - poi si usa la F come specificata dal modello
- E le **variabili**? La risposta più avanti

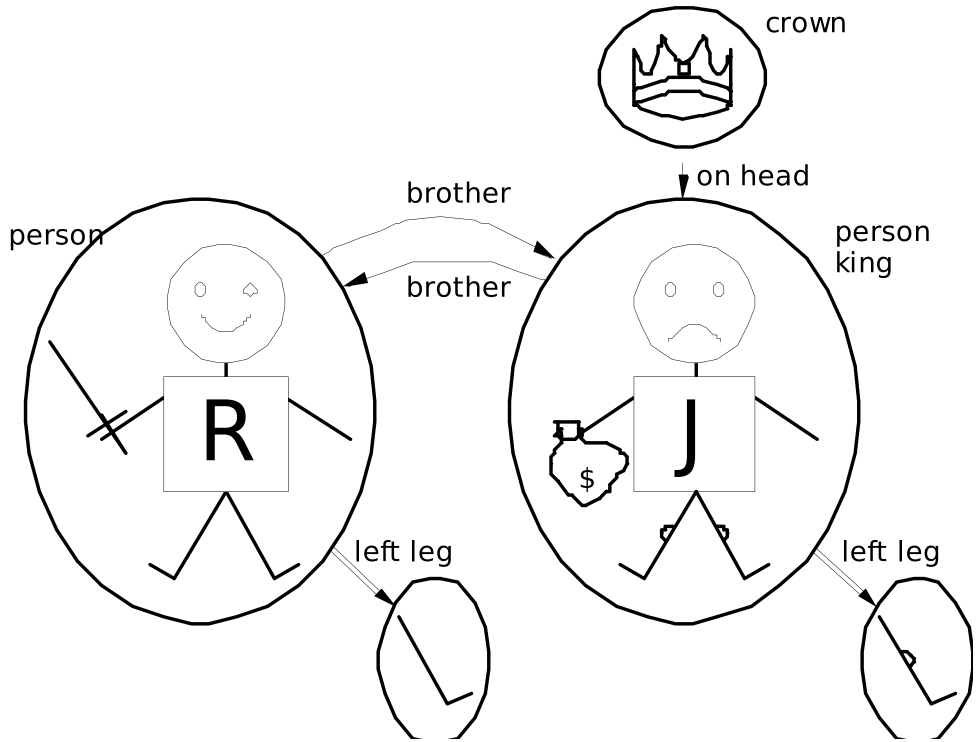
Interpretazione dei termini



Un termine è una rappresentazione
Il termine f(t1, …, tk) rappresenta un oggetto O del dominio. O è tale che gli oggetti O1, …, Ok (che sono riferiti da t1, …, tk) sono nella relazione F (riferita da f) e F(O1, …, Ok) corrisponde ad O

O non è rappresentato esplicitamente nelle formule ma compare quando serve lo si riferisce passando tramite f

Interpretazione e termini



Nell'esempio, le **gambe** di Riccardo e Giovanni non hanno una costante che le rappresenta nelle formule ma sono comunque **oggetti del dominio**

Sono riferiti da una funzione: **GambaSinistra(X)**

Quindi il termine **GambaSinistra(Giovanni)** corrisponde a un preciso oggetto esistente nel dominio

Formule atomiche

- **formulaAtomica → predicato(termine, ...) |**
  termine=termine
- Una formula atomica è vera quando:
  - dato un modello che include una specifica interpretazione,
  - la relazione a cui fa riferimento il simbolo di predicato
  - applicata agli oggetti identificati dai termini
  - è vera
- Esempi:
  sposato(madre(Richard), padre(John))
  padre(Richard) = padre(John)

Formule complesse

- Gli operatori della logica permettono di comporre predicati in fomule complesse
- **Esempi**:
  - Re(John) ⇒ ¬ Re(Richard)
  - ¬ Fratello(Corona, Fratello(John))
  - Persona(X) ∧ SullaTesta(Corona, X) ⇒ Re(X)

Quantificatori

- Quantificatore → ∀ | ∃
- I quantificatori permettono di esprimere *proprietà di collezioni di oggetti*
- Richiedono di fare riferimento a *generici oggetti* che saranno identificati da **variabili**, esempio “tutti gli uomini”diventa “tutti gli X che sono uomini”
- La logica del prim’ordine prevede:
  - ∀: per ogni (quantificatore universale)
  - ∃: esiste (quantificatore esistenziale)

Quantificatore universale

Si considerino una formula F e un modello M = (D,I). L'espressione ∀x F è vera nel modello M se e solo se F è vera per **qualsiasi** intepretazione di x in M

FORMA GENERALE: $\forall \ \langle \text{variabili} \rangle \ \ \langle \text{formula} \rangle$

ESEMPIO:

Al corso di sistemi intelligenti tutti sono intelligenti

$\forall x \ \text{Partecipa}(x, \text{SISINT}) \Rightarrow \text{Intelligente}(x)$

Quantificatore universale

Si considerino una formula F e un modello M = (D,I). L'espressione ∀x F è vera nel modello M se e solo se F è vera per qualsiasi intepretazione di x in M

FORMA GENERALE: ∀ 〈 variabili 〉 〈 formula 〉

ESEMPIO:

Al corso di sistemi intelligenti tutti sono intelligenti

∀x Partecipa(x, SISINT) ⇒ Intelligente(x)

Si può espandere in una **congiunzione** del tipo:

(Partecipa(Miriam, SISINT) ⇒ Intelligente(Miriam)) **∧**
(Partecipa(Matteo, SISINT) ⇒ Intelligente(Matteo)) **∧**
(Partecipa(Rufus, SISINT) ⇒ Intelligente(Rufus)) **∧ … ∧**
(Partecipa(SISINT, SISINT) ⇒ Intelligente(SISINT)) **∧ …**

Quantificatore universale

Si considerino una formula F e un modello M = (D,I). L'espressione ∀x F è vera nel modello M se e solo se F è vera per qualsiasi intepretazione di x in M

FORMA GENERALE: ∀ 〈 variabili 〉 〈 formula 〉

ESEMPIO:

Al corso di sistemi intelligenti tutti sono intelligenti

$\forall x$ Partecipa(x, SISINT) ⇒ Intelligente(x)

Si può espandere in una congiunzione del tipo:

(Partecipa(Miriam, SISINT) ⇒ Intelligente(Miriam)) ∧ … ∧ **(Partecipa(SISINT, SISINT) ⇒ Intelligente(SISINT))** ∧ …

Notate che "qualsiasi interpretazione" significa considerare tutti gli oggetti del dominio senza filtri. Non esistono i tipi di argomento

Quantificatore esistenziale

Si considerino una formula F e un modello M = (D,I). L'espressione ∃x F è vera nel modello M se e solo se F è vera per **qualche** intepretazione di x in M

FORMA GENERALE: ∃ ⟨ variabili ⟩ ⟨ formula ⟩

ESEMPIO:

Al corso di sistemi intelligenti qualcuno è intelligente

$\exists x$ Partecipa(x, SISINT) $\wedge$ Intelligente(x)

Quantificatore esistenziale

Si considerino una formula F e un modello M = (D,I). L'espressione ∃x F è vera nel modello M se e solo se F è vera per qualche intepretazione di x in M

FORMA GENERALE: ∃ 〈variabili〉 〈formula〉

ESEMPIO:

Al corso di sistemi intelligenti qualcuno è intelligente

∃x Partecipa(x, SISINT) ∧ Intelligente(x)

Si può espandere in una disgiunzione del tipo:

(Partecipa(Miriam, SISINT) ∧ Intelligente(Miriam)) **∨**
(Partecipa(Matteo, SISINT) ∧ Intelligente(Matteo)) **∨**
(Partecipa(Rufus, SISINT) ∧ Intelligente(Rufus)) **∨ … ∨**
(Partecipa(SISINT, SISINT) ∧ Intelligente(SISINT)) **∨ …**

Quantificatori e formule: attenzione

- Quale differenza c'è fra queste formule? Sono equivalenti?

1) $\forall x$ Partecipa(x, SISINT) $\Rightarrow$ Intelligente(x)
2) $\forall x$ Partecipa(x, SISINT) $\wedge$ Intelligente(x)

Quantificatori e formule: attenzione

- Quale differenza c'è fra queste formule? Sono equivalenti? **NO**

1) $\forall x$ Partecipa(x, SISINT) $\Rightarrow$ Intelligente(x)
   tutti quegli x che partecipano al corso di SISINT sono intelligenti

2) $\forall x$ Partecipa(x, SISINT) $\wedge$ Intelligente(x)
   tutti partecipano al corso di SISINT e sono intelligenti

Quantificatori e formule: attenzione

- Quale differenza c'è fra queste formule? Sono equivalenti?

1) $\exists x$ Partecipa(x, SISINT) $\Rightarrow$ Intelligente(x)
2) $\exists x$ Partecipa(x, SISINT) $\wedge$ Intelligente(x)

Quantificatori e formule: attenzione

- Quale differenza c'è fra queste formule? Sono equivalenti?

1) $\exists x$ Partecipa(x, SISINT) $\Rightarrow$ Intelligente(x)

   questa è equivalente a $\exists x$ ¬ Partecipa(x,SISINT) ∨ Intelligente(x) quindi significa che esiste un x tale per cui o x non partecipa a SISINT oppure x è intelligente

2) $\exists x$ Partecipa(x, SISINT) ∧ Intelligente(x)

   questa ci dice che esiste qualcuno fra i partecipanti a SISINT che è intelligente

Quantificatori annidati

1) **∃x ∃y F** è equivalente a ∃y ∃x F, si scrive anche ∃x,y F

2) **∀x ∀y F** è equivalente a ∀y ∀x F, si scrive anche ∀x,y F

3) **∀x ∃y F**: per tutti … esiste …
∀x ∃y Ama(x, y) tutti amano qualcuno (o qualcosa)

4) **∃y ∀x F**: esiste … per ogni …
∃y ∀x Ama(x, y) esiste qualcuno (o qualcosa) che tutti amano

Quantificatori e negazione

1) **$\forall x \ \neg F \equiv \neg \exists x \ F$**

$\forall x \ \neg$ Piace(x, Cavolfiore) $\equiv \neg \exists x$ Piace(x, Cavolfiore)

A nessuno piace il cavolfiore equivale a non c'è nessuno a cui piaccia il cavolfiore

2) **$\exists x \ \neg F \equiv \neg \forall x \ F$**

$\exists x \ \neg$ Piace(x, Cavolfiore) $\equiv \neg \forall x$ Piace(x, Cavolfiore)

c'è qualcuno a cui non piace il cavolfiore equivale a non a tutti piace il cavolfiore

3) **$\forall x \ F \equiv \neg \exists x \ \neg F$**

per tutti vale F equivale a non c'è nessuno per cui non vale F

4) **$\exists x \ F \equiv \neg \forall x \ \neg F$**

c'è qualcuno per cui vale F equivale a non per tutti non vale F

(Dis)uguaglianza e quantificatori

- L'uguaglianza riguarda esclusivamente termini
- Supponiamo di voler dire che John ha almeno due fratelli, come esprimere questo in formule?
- **∃ y, z Fratello(John, y) ∧ Fratello(John, z)** non è sufficiente
  Se y, z=Richard la formula sarà vera!
- Occorre aggiungere che i due termini con cui saranno unificate le variabili y e z debbono riferirsi a oggetti diversi
- **∃ y, z Fratello(John, y) ∧ Fratello(John, z) ∧ ¬(y=z)**

Verso l'unicità dei nomi

- Consideriamo l'asserzione
  Fratello(John, Richard) ∧ Fratello(John, Ramon)
- Sebbene l'intuizione sia che John ha due fratelli questa formula è soddisfatta anche quando le due costanti Richard e Ramon si riferiscono alla stessa persona!! Anche in questo caso occorre aggiungere **¬(Richard=Ramon)**
- Ancora peggio se vogliamo catturare che John ha solo due fratelli e non di più, dovremmo scrivere
  Fratello(John, Richard) ∧ Fratello(John, Ramon) ∧
  ¬(Richard=Ramon) ∧
  ∀x (Fratello(John, x) ⇒ (x=Richard ∨ x=Ramon))
- La semantica non è intuitiva, le formule sono troppo complesse

Una semantica più intuitiva: database semantics

- **Database semantics**: è la semantica usata nella programmazione logica e si basa su tre assunti
  - **Unicità dei nomi:**
    assumiamo che costanti diverse si riferiscano a oggetti del dominio diversi
  - **Closed-world assumption:**
    assumiamo che le formule atomiche delle quali non si conosce la verità siano false
  - **Domain closure:**
    un modello non contiene più elementi di quelli nominati dalle costanti
- Usando questa semantica la formula Fratello(John, Richard) ∧ Fratello(John, Ramon) rappresenta il fatto che John ha due fratelli, Richard e Ramon
- La database semantics può essere usata quando siamo sicuri dell'identità di tutti gli elementi. Da notare che riduce il numero di modelli possibili, rendendoli tipicamente finiti

Nota: nel libro si usa la semantica standard di FOL anche dopo aver presentato questa

Numeri, insiemi, liste

- Leggere da 8.3.1 a 8.3.3

Inferenza

1) Proposizionalizzazione di una KB e uso di un algoritmo per la logica proposizionale
2) Lifting delle regole di inferenza al prim'ordine e unificazione

Interrogazione di KB FOL

- Le interrogazioni a una KB FOL sono di due tipi:
  - **ask(KB, Re(John))**: viene chiesto se una formula <u>in cui compaiono solo termini ground</u> sia vera o falsa.
    La risposta sarà **true** o **false**
  - **ask(KB, Re(x))**: viene chiesto se <u>esiste un qualche valore per la variabile x</u> tale per cui la formula è vera.
    La risposta sarà **false** nel caso non esista tale valore, se invece esiste la riposta indicherà un **termine ground** che usato al posto di x rende vera la formula.

Sostituzione

- Una **sostituzione θ** è un insieme **{x1/g1, x2/g2, …, xn/gn }** dove le varie xi sono variabili e le varie gi sono termini ground
- Data una formula F e una sostituzione θ, la scrittura **F/θ** indica la formula ottenuta sostituendo le occorrenze delle variabili indicate in θ con i relativi termini ground
- **Esempio**:
  - F = Fratello(x, y)
  - θ ={ x/John}
  - F/θ= Fratello(John, y)

Trasformazione in formule proposizionali

- Primo passo dell'inferenza:
  formule del prim'ordine → formule proposizionali
- Si usano:
  - Regola di istanziazione universale
  - Regola di istanziazione esistenziale

Inferenza UI (universal instantiation)

- Regola di istanziazione universale UI:

$$\frac{\forall x \ \ \alpha}{\mathrm{SUBST}(\{x/g\}, \alpha)}$$

- Da una formula quantificata universalmente si possono inferire tutte le formule ottenute sostituendo un termine ground del vocabolario alla variabile quantificata
- SUBST identifica il processo di sostituzione, ha per argomenti la sostituzione in questione e la formula

Inferenza UI (universal instantiation)

- Data la formula **F :**
  ∀x (Partecipa(x, SISINT) ⇒ Intelligente(x))
- Date le sostituzioni alternative **θ1 = {x/Rufus}** e **θ2 = {x/Adele}** l'applicazione di UI permette di inferire rispettivamente:
  - **F θ1** è Partecipa(Rufus, SISINT) ⇒ Intelligente(Rufus)
  - **F θ2** è Partecipa(Adele, SISINT) ⇒ Intelligente(Adele)

NOTA: Fθ è un modo sintetico per scrivere SUBST(θ, F)

Regola di istanziazione esistenziale (EI)

- Le variabili compaiono anche in formule quantificate esistenzialmente. In questo caso la regola diventa

$$\frac{\exists x \;\; \alpha}{\mathrm{SUBST}(\{x/k\}, \alpha)}$$

- In questo caso **k deve essere una costante nuova**, cioè non ancora utilizzata nella KB
- L'inferenza si limita a dare un nome a questo elemento

Regola di istanziazione esistenziale

Esempio

- Data la formula:
  ∃x Corona(x) ∧ SullaTesta(x, John)
- Possiamo inferire **Corona(C1) ∧ SullaTesta(C1, John)** dove **C1** è un nome di costante nuovo inventato appositamente tramite un processo detto Skolemizzazione
- C1 non compare nella KB fino alla sua creazione
- **Attenzione**: usando la semantica standard di FOL, che non prevede unicità dei nomi, potremo poi inferire che **C1 = CoronaInglese**

Differenza UI fra e EI

- Le formule quantificate universalmente hanno tante istanze prodotte da UI:
  - La nuova KB è **logicamente equivalente** a quella originaria
- Quelle quantificate esistenzialmente sono istanziate tramite EI una volta sola e poi scartate:
  - La nuova KB **non è logicamente equivalente** a quella originaria ma è **soddisfacibile** se la prima lo era
  - Le possibili alternative di istanziazione sono inferenzialmente equivalenti

Proposizionalizzazione

- Data una KB:
  1) Applicare le regole di eliminazione dei quantificatori sostituendo le formule quantificate con le loro istanze del caso, costruite considerando il vocabolario dei possibili termini ground
  2) Applicare un algoritmo di inferenza completo per la logica proposizionale
- Ne risulta un metodo generale e completo per trattare l'implicazione … ma e se i termini contengono funzioni?

Problema delle funzioni

- Un funzione può essere applicata ricorsivamente: fz(fz(fz(…fz(x)…)))
- Di conseguenza l'insieme delle possibili sostituzioni diventa potenzialmente ∞
- nell'applicare la regola UI a una formula occorrerà considerare tutti i termini ground fz(x), fz(fz(x)), fz(fz(fz(x))), … senza fine?

Teorema di Herbrand

- Herbrand ha dimostrato che se una formula è conseguenza logica della base di conoscenza originaria (del prim'ordine) allora partendo dalla base di conoscenza proposizionalizzata esiste una dimostrazione finita della sua verità
- I termini sono costruiti "in ampiezza":
  - Prima si provano le sostituzioni delle variabili con le costanti, esempio: {x/Richard}, {x/John}
  - Poi si provano le sostituzioni con termini ground che prevedono una sola applicazione delle funzioni, esempio: {x/fz(Richard)}, {x/fz(John)}
  - Poi quelle con una chiamata ricorsiva: {x/fz(fz(Richard))}, {x/fz(fz(John))}
  - ...

Semidecidibilità di FOL

- **FOL è semidecidibile, cioè:**
  - Completezza: Come dimostrato da Herbrand se una formula consegue da una KB_{FOL} si troverà una dimostrazione finita della sua verità
  - Se però l'**non vale la consequenzialità**, la presenza di funzioni applicabili ricorsivamente porterà l'inferenza su di un percorso infinito
- In altri termini non esiste un algoritmo per dimostrare in FOL che una certa conseguenza non vale

Inefficienza della proposizionalizzazione

- Consideriamo la KB:
  ∀x Re(x) ∧ Avido(x) ⇒ Malvagio(x)
  Re(John)
  Avido(John)
  Fratello(Richard, John)
- La proposizionalizzazione creerà:
  Re(Richard) ∧ Avido(Richard) ⇒ Malvagio(Richard)
  Re(John) ∧ Avido(John) ⇒ Malvagio(john)
  Re(John)
  Avido(John)
- A un essere umano è evidente che solo John può risultare malvagio. La proposizionalizzazione è inefficiente. “Perde tempo”a creare istanze dell’implicazione (quella in grassetto) che sono ininfluenti

Inefficienza della proposizionalizzazione

- Complichiamo l'esempio:

  $\forall x$ Re(x) $\wedge$ Avido(x) $\Rightarrow$ Malvagio(x)
  Re(John)
  $\forall y$ Avido(y)
  Fratello(Richard, John)

- Anche in questo caso a un occhio umano è ovvio che solo John può essere malvagio (è l'unico re) ma la proposizionalizzazione creerà anche:

  Avido(Richard), Re(Richard) $\wedge$ Avido(Richard) $\Rightarrow$ Malvagio(Richard)

Guidare la scelta della sostituzione

- È possibile evitare la creazione di formule irrilevanti f**ocalizzando la costruzione della sostituzione** da considerare
- nell'esempio vorrei poter limitare la scelta a quelle sostituzioni che rendono vera **Re(x) ∧ Avido(x)** da un lato e **Re(John)** e **Avido(y)** dall'altro. Solo **{ x/John, y/John }** ha questi requisiti

Modus Ponens Generalizzato (MPG)

$$\frac{p'_1, \; p'_2, \dots, p'_n, \qquad p_1 \wedge p_2 \wedge \dots \wedge p_n \Rightarrow q}{q\theta}$$

Dove $p'_i\theta = p_i\theta$ per ogni $i \in [1,n]$

La regola ha come premesse n formule atomiche e una singola implicazione. La conclusione è il risultato dell'applicazione della sostituzione θ alla formula q, conseguenza dell'implicazione

NB: qθ è un modo sintetico per scrivere SUBST(θ, q)

Esempio

$$\frac{Re(John),\ \ Avido(y),\ \ Re(x) \wedge Avido(x) \Rightarrow Malvagio(x)}{Malvagio(John)}$$

Esempio

John x

y x

$$\frac{Re(John),\ \ Avido(y),\ \ Re(x) \land Avido(x) \Rightarrow Malvagio(x)}{Malvagio(John)}$$

Questa conclusione si appoggia alla sostituzione {x/John, y/John}

Le clausole focalizzano la ricerca della sostituzione e permettono di ragionare direttamente in FOL

Nella proposizionalizzazione, di contro, si costruiscono sostituzioni usando in modo esaustivo l'intero vocabolario di costanti

Sulla generalità del MPG: Lifting

- Il modus ponens non pone restrizioni sulla forma della formula antecedente dell'implicazione
- Il modus ponens generalizzato richiede invece che sia una congiunzione
- L'appellativo "generalizzato"deriva dall'avere "sollevato"la regola dalla logica proposizionale a quella del prim'ordine. Questo processo si chiama **lifting**
- Inoltre consente di avere considerare un numero qualsiasi di formule da cui trarre la conclusione (invece di due solamente)

Unificazione

- **Unificazione**: algoritmo chiave di tutte le tecniche di inferenza sul prim'ordine:
  - Date due formule F1 e F2
  - UNIFY(F1, F2) = θ tale che F1 θ = F2 θ
- Il risultato è una sostituzione che, applicata a entrambe le formule, le rende identiche
- Tale sostituzione, se esiste, è detta **UNIFICATORE**
- Se vi sono più unificatori si calcola e si usa il **Most General Unifier** (**MGU, unificatore più generale**)

Esempi

- UNIFY(knows(John, x), knows(John, Richard))
  - {x/Richard}
- UNIFY(knows(John, x), knows(y, Richard))
  - {x/Richard, y/John}
- UNIFY(knows(John, x), knows(y, MotherOf(y)))
  - { x/MotherOf(John), y/John }
- UNIFY(knows(John, x), knows(x, Richard))
  - **Fallisce** perché x non può assumere due valori contemporaneamente
  - **Standardizzazione separata (standardizing apart)**: rinoma delle variabili di una formula per evitare collisioni con un'altra formula.

Abbiamo visto

- Per rispondere alle query occorre fare **inferenza**
- L'inferenza del prim'ordine è diversa da quella proposizionale: abbiamo **quantificatori** e **variabili**
- È possibile ridurre una KB del prim'ordine in una proposizionale e poi applicare algoritmi di inferenza proposizionali:
  - Se la KB **non contiene quantificatori esistenziali**, quella trasformata sarà equivalente a quella FOL
  - Se la **KB contiene quantificatori esistenziali**, non sarà equivalente ma sarà inferenzialmente equivalente (se derivo una formula dalla seconda tale formula sarà derivabile anche da quella originaria)
- **Rispondere alle query richiede di identificare opportune sostituzioni**
- Tali sostituzioni si trovano più efficientemente se se si usano regole di inferenza di cui è stato fatto il **lifting al prim'ordine**
- È necessario tradurre la KB in **clausole di Horn del prim'ordine**

Clausole di Horn del prim'ordine

- La definizione è analoga a quelle delle clausole di Horn (e clausole definite) proposizionali, sono **disgiunzioni di letterali di cui al più uno è positivo**:
  - **Atomiche**:
    - Avido(x) la variabile è intesa come universalmente quantificata
    - Avido(John)
  - **Implicazione il cui antecedente è costituito da letterali positivi**
    Re(x) ∧ Avido(x) ⇒ Malvagio(x)

- Non tutte le KB possono essere tradotte in clausole di Horn ma molte possono. A queste è possibile applicare il forward chaining (e il modus ponens generalizzato) per fare inferenze

Una KB di esempio

Sapendo: “La legge dice che è un reato per un negozio vendere alcolici a un minorenne. Marco, minorenne, possiede della birra. Tale birra gli è stata venduta del minimarket Sotto Casa”

Obiettivo: “Dimostrare la reità di Sotto Casa”

Useremo questo esempio per imparare a scrivere una KB in clausole di Horn del prim’ordine e come funzionano le inferenze nel prim’ordine

Una KB di esempio: simboli

Costanti: Marco, SottoCasa

Predicati: Vende, Negozio, Supermarket, Birra, Alcolico, Minorenne, Possiede, Reo

Funzioni: nessuna

Interpretazione:

- Marco: il protagonista della storia
- Sotto Casa: minimarket presso il quale Marco fa acquisti abitualmente
- Vende: relazione ternaria che lega chi ha venduto qualcosa a chi
- Negozio: relazione unaria che indica la proprietà di essere un negozio
- ...

Una KB di esempio

La legge dice che è un reato per un negozio vendere alcolici a un minorenne.

$Negozio(x) \land Vende(x, y, z) \land Alcolico(y) \land Minorenne(z) \Rightarrow Reo(x)$

Una KB di esempio

La legge dice che è un reato per un negozio vendere alcolici a un minorenne.

Negozio(x) ∧ Vende(x, y, z) ∧ Alcolico(y) ∧ Minorenne(z) ⇒ Reo(x)

Marco possiede della birra.

∃x Possiede(Marco, x) ∧ Birra(x)

Applicando la regola di istanziazione dell'esistenziale (EI) questa formula si traduce nelle due formule atomiche:

Possiede(Marco, B), Birra(B)

B è una nuova costante, non utilizzata altrove. È la birra comperata da Marco. Le diamo il nome "B". Le costanti come B sono dette costanti di Skolem.

Una KB di esempio

La legge dice che è un reato per un negozio vendere alcolici a un minorenne.

Negozio(x) ∧ Vende(x, y, z) ∧ Alcolico(y) ∧ Minorenne(z) ⇒ Reo(x)

Marco possiede della birra.

Possiede(Marco, B), Birra(B)

Tale birra gli è stata venduta del minimarket Sotto Casa. Esprimiamo che se Marco possiede della birra l'ha comperata al minimarket

Possiede(Marco, x) ∧ Birra(x) ⇒ Vende(SottoCasa, x, Marco)

La birra è un alcolico

Birra(x) ⇒ Alcolico(x)

Una KB di esempio

Negozio(x) ∧ Vende(x, y, z) ∧ Alcolico(y) ∧ Minorenne(z) ⇒ Reo(x)

Possiede(Marco, B), Birra(B)

Possiede(Marco, x) ∧ Birra(x) ⇒ Vende(SottoCasa, x, Marco)

Birra(x) ⇒ Alcolico(x)

Poi ancora

Minimarket(SottoCasa)

Minimarket(x) ⇒ Negozio(x)

Minorenne(Marco)

La KB in clausole di Horn

C1) Negozio(x) ∧ Vende(x, y, z) ∧ Alcolico(y) ∧ Minorenne(z) ⇒ Reo(x)

C2) Possiede(Marco, B)

C3) Birra(B)

C4) Possiede(Marco, x) ∧ Birra(x) ⇒ Vende(SottoCasa, x, Marco)

C5) Birra(x) ⇒ Alcolico(x)

C6) Minimarket(SottoCasa)

C7) Minimarket(x) ⇒ Negozio(x)

C8) Minorenne(Marco)

NOTA: questa KB è costituita da **clausole di Horn del prim'ordine**. In particolare si tratta di clausole che non fanno uso di funzioni. Questa particolare classe di KB è detta **DATALOG**