**Sintesi Completa ed Esaustiva degli Appunti: Propositions and Inference**

**5. Propositions and Inference**

**Introduzione**

* La logica proposizionale è uno strumento efficace per rappresentare vincoli in modo conciso, evitando tabelle complesse.
* **Vantaggi**:
  + Maggiore leggibilità.
  + Facilita l'espressione di relazioni tra variabili.
  + Consente inferenze logiche per risolvere problemi complessi.

**5.1 Propositions**

**5.1.1 Syntax of Propositional Calculus**

* **Proposizione atomica (Atom)**:
  + Un simbolo (es. a, b) con valore di verità (true o false).
  + Es: ai\_is\_fun può essere true o false in base all'interpretazione.
* **Proposizioni composte**:
  + Formate da connettivi logici:
    - **¬p**: Negazione.
    - **p ∧ q**: Congiunzione.
    - **p ∨ q**: Disgiunzione.
    - **p → q**: Implicazione.
    - **p ↔ q**: Equivalenza.
* **Precedenza degli operatori**:
  + ¬
  + ∧
  + ∨
  + →
  + ↔

**5.1.2 Semantics of Propositional Calculus**

* **Semantica**:
  + Mappa ogni proposizione atomica in un valore di verità {true, false}.
  + Le proposizioni composte derivano il loro valore usando la **tabella di verità**.
* **Interpretazione**:
  + Funzione che assegna un valore {true, false} a ciascun atomo.
  + Esempio: Se π(ai\_is\_fun) = true, allora ai\_is\_fun è vero in quella interpretazione.

**5.1.3 Knowledge Base (KB) e Logical Consequence**

* **Knowledge Base (KB)**:
  + Insieme di proposizioni considerate vere.
  + Ogni proposizione nella KB è un **assioma**.
* **Modello di una KB**:
  + Interpretazione in cui tutte le proposizioni nella KB sono vere.
* **Conseguenza logica**:
  + Una proposizione g è conseguenza logica di KB (indicato con KB ⊨ g) se g è vera in tutti i modelli di KB.
  + Esempio: Se KB = {a → b, a}, allora KB ⊨ b.

**5.2 Propositional Definite Clauses**

**Definizione**

* **Clausola definita**:
  + Regola logica del tipo h ← b1 ∧ b2 ∧ ... ∧ bn, dove:
    - h: Testa (Head).
    - b1 ∧ b2 ∧ ... ∧ bn: Corpo (Body).
  + Se tutte le condizioni nel corpo sono vere, allora anche la testa è vera.
* **Knowledge Base (KB)**:
  + Insieme di clausole definite.

**5.2.2 Proofs**

* **Obiettivo della prova**:
  + Verificare se una proposizione è conseguenza logica di KB.
* **Metodi di prova**:
  + **Bottom-Up Proof Procedure**:
    - Deduce tutte le conseguenze logiche partendo dai fatti noti.
    - Usa la regola di inferenza (modus ponens): Se h ← b1 ∧ b2 e b1, b2 sono veri, allora h è vero.
  + **Top-Down Proof Procedure**:
    - Parte dalla query (g) e risale ai fatti necessari per dimostrarla.

**5.3 Knowledge Representation Issues**

**Debugging della Base di Conoscenza**

* **Errori comuni**:
  1. **False-positive**:
     + Una risposta derivata è falsa nell'interpretazione reale.
  2. **False-negative**:
     + Una risposta vera non viene derivata.
  3. **Loop infinito**:
     + Il sistema non trova una soluzione e continua a esplorare.
* **Risoluzione**:
  1. Identificare clausole mancanti o errate.

**5.4 Proving by Contradictions**

**Clausole di Horn**

* Estensione delle clausole definite, che includono regole con false come conclusione.
* Forma generale: false ← a1 ∧ a2 ∧ ... ∧ an.
* Permettono **dimostrazioni per contraddizione**:
  + Dimostrano che un insieme di assunzioni non può essere vero.

**Diagnosi Basata su Consistenza**

* Identifica conflitti tra osservazioni e regole per determinare errori o guasti.
* **Conflitti minimi**:
  + Il più piccolo insieme di assunzioni che causa inconsistenza.

**5.5 Complete Knowledge Assumption**

**Closed-World Assumption (CWA)**

* Gli atomi non derivabili come veri sono considerati falsi.
* Contrapposto alla **open-world assumption**, in cui gli atomi non derivabili sono ignoti.

**Negazione come Fallimento**

* La negazione di un atomo ~a è vera se a non può essere derivato dalla KB.

**Clark’s Completion**

* Completa la KB assumendo che ogni clausola copra tutti i possibili casi.

**5.6 Abduction**

**Definizione**

* Ragionamento per spiegare osservazioni basandosi su assunzioni.
* Usa clausole di Horn e **assumables** (ipotesi).

**Diagnosi Abduttiva**

* Identifica guasti o anomalie spiegando sintomi osservati.
* **Implementazioni**:
  1. **Bottom-Up**:
     + Calcola spiegazioni minime per ogni osservazione.
  2. **Top-Down**:
     + Genera conflitti e prova spiegazioni per risolverli.

**Conclusioni**

Questa sintesi offre una visione dettagliata dei concetti di logica proposizionale, clausole definite, debugging, diagnosi e abduzione, fornendo una base solida per comprendere e applicare queste tecniche. Se hai bisogno di ulteriori chiarimenti o approfondimenti su un argomento specifico, non esitare a chiedere! 😊