

Representação de Conhecimento: Lógica Proposicional

Representando Conhecimento para Inferência em IA

Márcio Nicolau

2025-09-11

Table of contents

Introdução	2
Objetivo de Aprendizagem	2
O que é Representação de Conhecimento (RC)?	2
Lógica na IA	2
Lógica Proposicional (LP)	4
Sintaxe da Lógica Proposicional	4
Símbolos Proposicionais (Átomos)	4
Conectivos Lógicos	4
Fórmulas Bem Formadas (FBFs)	4
Semântica da Lógica Proposicional	5
Valoração (Interpretação)	5
Tabelas Verdade	5
Classificações de FBFs	5
Inferência em Lógica Proposicional	6
Consequência Lógica	6
Modelos	6
Regras de Inferência	6
Algoritmos de Prova	6
Exemplos Práticos e Implementação em Python	7
Verificação por Tabela Verdade (Conceptual)	7
Exemplo em Python: Gerador de Tabela Verdade Simples	7
Limitações da Lógica Proposicional	10
Verificação de Aprendizagem	10
Referências Bibliográficas	11

List of Figures

1	Importância da Representação de Conhecimento em Agentes Inteligentes	3
---	--	---

Introdução

Para que os agentes inteligentes possam interagir de forma eficaz com o mundo, resolver problemas complexos e tomar decisões racionais, eles precisam de uma maneira de representar e manipular o conhecimento. A **Representação de Conhecimento (RC)** é o ramo da Inteligência Artificial dedicado a projetar formalismos e mecanismos para que os computadores possam armazenar e raciocinar sobre informações de uma maneira que seja próxima ao entendimento humano. Nesta aula, exploraremos a **Lógica Proposicional (LP)**, uma das formas mais fundamentais de representação de conhecimento, que nos permite codificar fatos e regras de forma precisa e inferir novas conclusões.

Objetivo de Aprendizagem

Ao final desta aula, você será capaz de:

- Definir o conceito de Representação de Conhecimento e sua importância na IA.
- Compreender os fundamentos da Lógica Proposicional.
- Construir Fórmulas Bem Formadas (FBFs) utilizando símbolos proposicionais e conectivos lógicos.
- Determinar o valor de verdade de FBFs usando tabelas verdade.
- Explicar e aplicar conceitos de inferência em Lógica Proposicional.
- Identificar as limitações da Lógica Proposicional para a RC.

O que é Representação de Conhecimento (RC)?

A Representação de Conhecimento (RC) é o estudo de como um agente pode armazenar o que sabe sobre o mundo de tal forma que ele possa usá-lo para tomar decisões e inferir novas informações. É um componente chave da IA porque os agentes inteligentes frequentemente precisam de mais do que apenas percepções e ações; eles precisam de um modelo explícito do mundo, de suas propriedades, e de como suas ações o afetam (Russell; Norvig, 2004, p. 207).

As principais razões para a RC em IA incluem:

- **Inferência:** Capacidade de derivar conclusões não explícitas a partir de informações existentes.
- **Compreensão:** Permitir que o agente entenda o significado de suas percepções e o impacto de suas ações.
- **Comunicação:** Facilitar a comunicação de conhecimento entre agentes (humanos e artificiais).
- **Aquisição de Conhecimento:** Ajudar na incorporação de novas informações.
- **Explicação:** Permitir que o agente justifique suas ações e conclusões.

Lógica na IA

A lógica tem sido, desde o início, um dos formalismos mais proeminentes para a representação de conhecimento na IA. Ela oferece uma linguagem precisa para expressar informações e um sistema de inferência bem definido

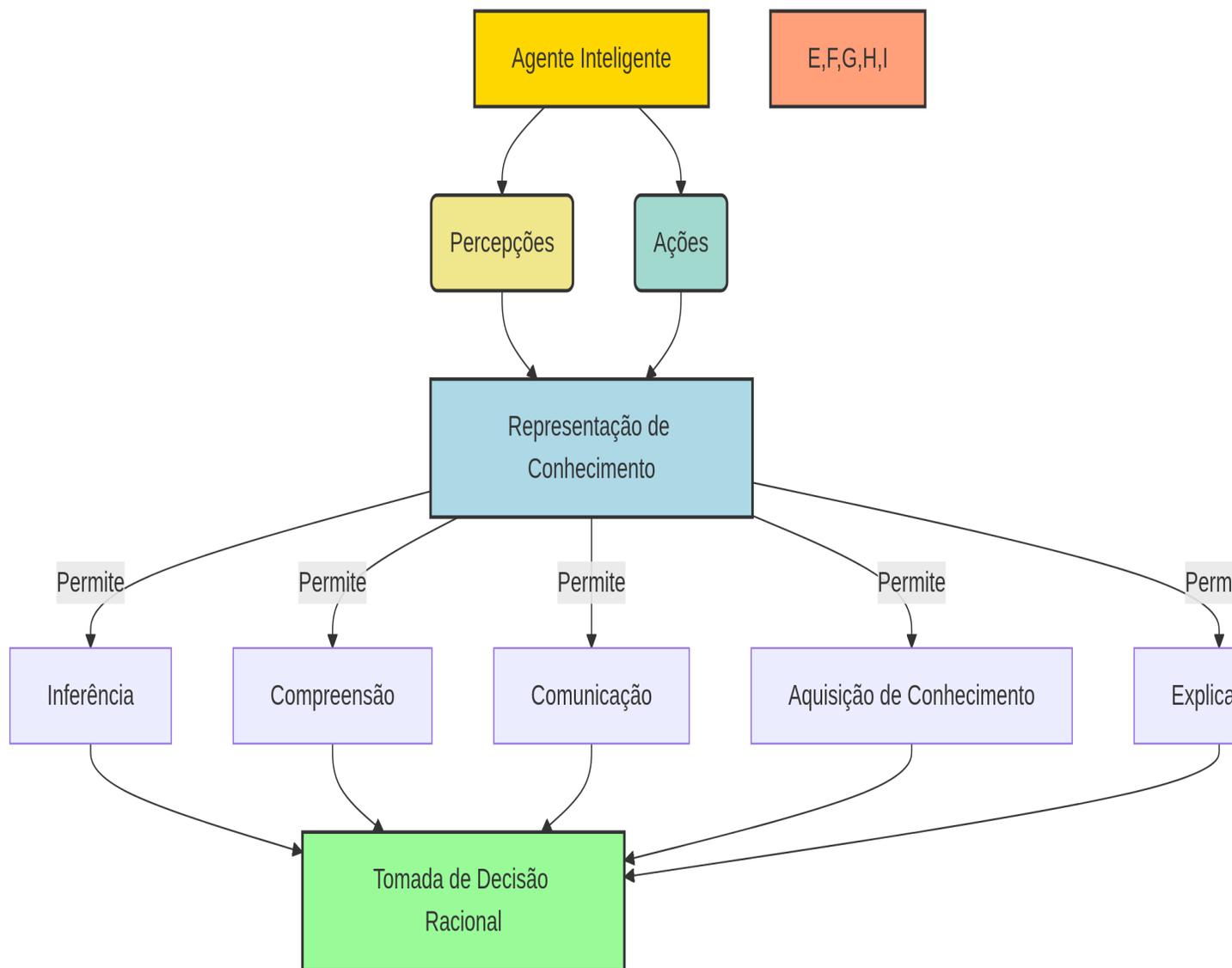


Figure 1: Importância da Representação de Conhecimento em Agentes Inteligentes

para derivar conclusões.

- **Linguagem Formal:** Define uma sintaxe (regras para construir sentenças) e uma semântica (significado das sentenças).
- **Sistema de Inferência:** Permite que conclusões logicamente válidas sejam derivadas de premissas.

Existem vários tipos de lógica, cada um com diferentes níveis de expressividade e complexidade computacional. Começaremos com a Lógica Proposicional.

Lógica Proposicional (LP)

A Lógica Proposicional (também conhecida como Cálculo Proposicional) é a forma mais simples de lógica formal. Ela lida com proposições inteiras (sentenças declarativas que podem ser verdadeiras ou falsas) e como elas se conectam.

Sintaxe da Lógica Proposicional

A sintaxe define quais são as “sentenças bem formadas” na lógica proposicional.

Símbolos Proposicionais (Átomos)

São os elementos mais básicos, representando fatos simples que podem ser verdadeiros (V) ou falsos (F). Geralmente são representados por letras maiúsculas: P, Q, R, S.

- **Exemplos:**
 - Chuva: “Está chovendo.”
 - Sol: “Há sol.”
 - Molhado: “O chão está molhado.”
 - Verdade: “A afirmação é verdadeira.”

Conectivos Lógicos

São operadores que combinam proposições mais simples para formar proposições mais complexas.

Símbolo	Nome	Significado	Exemplo
\neg	Negação	“não”	$\neg Chuva$ (Não está chovendo)
\wedge	Conjunção	“e”	$Chuva \wedge Sol$ (Está chovendo e há sol)
\vee	Disjunção	“ou”	$Chuva \vee Sol$ (Está chovendo ou há sol)
\rightarrow	Implicação (Condicional)	“se...então...”	$Chuva \rightarrow Molhado$ (Se chove, então o chão está molhado)
\leftrightarrow	Bicondicional	“se e somente se”	$Chuva \leftrightarrow Molhado$ (Chove se e somente se o chão está molhado)

Fórmulas Bem Formadas (FBFs)

As regras para construir FBFs são recursivas:

1. Um símbolo proposicional é uma FBF.
2. Se ϕ é uma FBF, então $\neg\phi$ é uma FBF.

3. Se ϕ e ψ são FBFs, então $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$ e $(\phi \leftrightarrow \psi)$ são FBFs.
4. Nada mais é uma FBF.

- **Exemplos de FBFs:**

- P
- $\neg Q$
- $(P \wedge Q)$
- $((\text{Chuva} \wedge \text{Vento}) \rightarrow \text{Molhado})$
- $((P \rightarrow Q) \wedge (\neg P \wedge Q))$

Semântica da Lógica Proposicional

A semântica define o significado das FBFs, ou seja, como determinar seu valor de verdade.

Valoração (Interpretação)

Uma valoração (ou interpretação) é uma atribuição de valor de verdade (V ou F) a cada símbolo proposicional.

- **Exemplo:**
 - Se Chuva = V e Molhado = F.

Tabelas Verdade

Uma tabela verdade lista todos os possíveis valores de verdade para os símbolos proposicionais em uma FBF e, a partir deles, calcula o valor de verdade da FBF para cada combinação.

Conectivos Básicos:

P	$\neg P$
V	F
F	V

P	Q	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$
V	V	V	V	V	V
V	F	F	V	F	F
F	V	F	V	V	F
F	F	F	F	V	V

Classificações de FBFs

- **Tautologia:** Uma FBF que é sempre verdadeira, independentemente da valoração dos seus símbolos proposicionais. (Ex: $(P \vee \neg P)$)
- **Contradição:** Uma FBF que é sempre falsa, independentemente da valoração dos seus símbolos proposicionais. (Ex: $(P \wedge \neg P)$)
- **Contingência:** Uma FBF que pode ser verdadeira ou falsa, dependendo da valoração dos seus símbolos proposicionais. (Ex: $P \wedge Q$)

Inferência em Lógica Proposicional

Inferência é o processo de derivar novas sentenças a partir de sentenças existentes. Em LP, a inferência é **dedutiva** e **monotônica** (adicionar novas informações não invalida conclusões anteriores).

Consequência Lógica

Uma sentença α é uma **consequência lógica** de um conjunto de sentenças KB (Knowledge Base) se α é verdadeira em todos os modelos nos quais KB é verdadeira. Notação: $KB \models \alpha$.

Modelos

Um **modelo** é uma atribuição de valores de verdade para os símbolos proposicionais que faz com que todas as sentenças em uma base de conhecimento sejam verdadeiras.

Regras de Inferência

São padrões de raciocínio que nos permitem derivar uma conclusão válida a partir de um conjunto de premissas.

- **Modus Ponens:**
 - Se P e $(P \rightarrow Q)$, então Q .
 - *Exemplo:* Choveu. Se choveu, então o chão está molhado. Conclusão: O chão está molhado.
- **Modus Tollens:**
 - Se $(\neg Q)$ e $(P \rightarrow Q)$, então $(\neg P)$.
 - *Exemplo:* O chão não está molhado. Se choveu, então o chão está molhado. Conclusão: Não choveu.
- **Eliminação do E (And-Elimination):**
 - Se $(P \wedge Q)$, então P . (E também Q).
- **Introdução do E (And-Introduction):**
 - Se P e Q , então $(P \wedge Q)$.
- **Resolução:** Uma regra de inferência poderosa que forma a base de muitos algoritmos de prova automática de teoremas em lógica.

Algoritmos de Prova

Um **algoritmo de prova** aplica regras de inferência para construir uma prova de que uma nova sentença é uma consequência lógica da KB.

- **Verificação de Tabela Verdade:** Enumerar todos os modelos possíveis e verificar se a conclusão é verdadeira em todos os modelos que satisfazem a KB. É completa, mas exponencialmente cara (2^N linhas para N símbolos proposicionais).
- **Dedução Natural:** Aplica um conjunto de regras de inferência para derivar a conclusão.
- **Resolução (com Forma Normal Conjuntiva - FNC):** Converte todas as sentenças para FNC e então aplica a regra de resolução até que a contradição seja encontrada (prova por refutação). (Russell; Norvig, 2004, p. 238–243)

Exemplos Práticos e Implementação em Python

Vamos considerar um cenário simples: o sistema de alarme de uma casa.

- **P:** “Há um assaltante.”
- **Q:** “O alarme toca.”
- **R:** “O cachorro late.”

Podemos formalizar algumas regras:

1. $P \rightarrow Q$ (Se há um assaltante, então o alarme toca).
2. $Q \rightarrow R$ (Se o alarme toca, então o cachorro late).
3. P (Há um assaltante).

Agora, queremos saber se R (o cachorro late) é uma consequência lógica.

Verificação por Tabela Verdade (Conceptual)

P	Q	R	$P \rightarrow Q$	$Q \rightarrow R$	KB ($P \wedge (P \rightarrow Q) \wedge (Q \rightarrow R)$)	R
V	V	V	V	V	V	V
V	V	F	V	F	F	F
V	F	V	F	V	F	V
V	F	F	F	V	F	F
F	V	V	V	V	F	V
F	V	F	V	F	F	F
F	F	V	V	V	F	V
F	F	F	V	V	F	F

Nosso único modelo onde KB é verdadeira (linha 1) mostra que R também é verdadeiro. Portanto, R é uma consequência lógica de KB.

Exemplo em Python: Gerador de Tabela Verdade Simples

Construir um motor de inferência completo para LP em Python é complexo. No entanto, podemos ilustrar como uma tabela verdade pode ser gerada para verificar a validade de uma sentença simples. Usaremos uma abordagem para avaliar expressões booleanas.


```

import itertools

def evaluate_expression(expression, assignment):
    """
    Avalia uma expressão booleana usando um dicionário de atribuições.
    A expressão deve usar operadores Python ('and', 'or', 'not').
    """
    # Substitui os nomes das variáveis na expressão pelos seus valores
    # Ex: 'P and Q' com {'P': True, 'Q': False} -> 'True and False'
    for var, value in assignment.items():
        expression = expression.replace(var, str(value))

    # Avalia a string como uma expressão Python
    # Cuidado: eval() pode ser perigoso com entradas não confiáveis
    try:
        return eval(expression)
    except Exception as e:
        print(f"Erro ao avaliar expressão: {e}")
        return None

def generate_truth_table(propositions, expression):
    """
    Gera e imprime a tabela verdade para uma expressão proposicional.
    propositions: uma lista de strings com os símbolos proposicionais (e.g., ['P', 'Q'])
    expression: a FBF como uma string Python (e.g., "(P and Q) or not R")
    """
    print(" " * 2 + " ".join(propositions) + " | " + expression)
    print("-" * (len(" ".join(propositions)) + 3 + len(expression)))

    # Gera todas as combinações de valores de verdade para as proposições
    # True/False (2 valores) para cada proposição
    for values in itertools.product([True, False], repeat=len(propositions)):
        assignment = dict(zip(propositions, values))

        # Formata a linha da tabela
        row_str = " ".join(["T" if assignment[p] else "F" for p in propositions])

        result = evaluate_expression(expression, assignment)
        result_str = "T" if result else "F" if result is not None else "?"

        print(f"{row_str} | {result_str}")

if __name__ == "__main__":
    # Exemplo 1: P AND Q
    print("--- Tabela Verdade para (P and Q) ---")
    generate_truth_table(['P', 'Q'], "(P and Q)")

    # Exemplo 2: P OR NOT P (Tautologia)
    print("\n--- Tabela Verdade para (P or not P) ---")
    generate_truth_table(['P'], "(P or not P)")

    # Exemplo 3: (P and not P) (Contradição)
    print("\n--- Tabela Verdade para (P and not P) ---")
    generate_truth_table(['P'], "(P and not P)")

```

O código acima demonstra uma forma rudimentar de verificar a satisfatibilidade de uma FBF e, consequentemente, inferir consequências lógicas através da prova por refutação (verificando se a KB AND NOT conclusão é uma contradição).

Limitações da Lógica Proposicional

Apesar de sua simplicidade e clareza, a Lógica Proposicional possui limitações significativas que a tornam inadequada para representar o conhecimento em muitos domínios da IA:

- **Falta de Expressividade para Propriedades e Relações:** A LP trata proposições como unidades atômicas. Não há como expressar que um “cachorro” tem uma “cor” ou que “João” é “irmão” de “Maria” sem criar uma proposição separada para cada instância.
 - Ex: “Todos os humanos são mortais” não pode ser representado diretamente. Precisaríamos de proposições como `Humano1_Mortal`, `Humano2_Mortal`, etc., sem generalização.
- **Dificuldade com Quantificação:** Não há como expressar “todos”, “alguns” ou “nenhum”. Isso significa que não podemos expressar sentenças como “Todos os alunos são inteligentes” ou “Existe um professor que não dá nota máxima”.
- **Explosão Combinatória:** Para representar um mundo com muitas entidades e propriedades que se repetem, a quantidade de símbolos proposicionais necessários explodiria, tornando a base de conhecimento e as tabelas verdade impraticáveis.

Essas limitações levam à necessidade de formalismos mais expressivos, como a Lógica de Predicados (ou Lógica de Primeira Ordem), que será o tema da nossa próxima aula.

Verificação de Aprendizagem

Responda às seguintes questões e implemente as tarefas propostas para solidificar seu entendimento sobre a Lógica Proposicional.

1. Conceitos Fundamentais:

- a) Explique o que é Representação de Conhecimento na IA e cite três de seus propósitos.
- b) Qual a diferença entre a sintaxe e a semântica de uma lógica formal?

2. Sintaxe e FBFs:

Considere os seguintes símbolos proposicionais:

- C: “Está claro.”
- F: “Faz calor.”
- P: “Vamos à praia.”
- G: “Usamos protetor solar.”

Traduza as seguintes sentenças do português para FBFs em Lógica Proposicional:

- a) “Se está claro e faz calor, então vamos à praia.”
- b) “Não vamos à praia, a menos que esteja claro.”
- c) “Usamos protetor solar se vamos à praia e faz calor.”
- d) “Se não usamos protetor solar, então não faz calor.”

3. Tabelas Verdade e Classificação:

- a) Construa a tabela verdade para a FBF: $(P \vee Q) \rightarrow (P \wedge Q)$.
- b) Com base na tabela verdade, classifique esta FBF como Tautologia, Contradição ou Contingência. Justifique sua resposta.
- c) Construa a tabela verdade para a FBF: $(P \vee Q) \wedge (\neg P \vee \neg Q)$. Classifique-a e justifique.

4. Inferência e Consequência Lógica:

Você tem a seguinte base de conhecimento (KB):

1. `Nevou`
2. `Nevou` \rightarrow `Frio`
3. `Frio` \rightarrow `UsarCasaco`

- a) Usando a regra de inferência **Modus Ponens**, mostre passo a passo como você pode inferir `UsarCasaco` a partir da KB.
- b) Explique, usando o conceito de modelo, por que `UsarCasaco` é uma consequência lógica desta KB.

5. Limitações da Lógica Proposicional:

Explique a principal limitação da Lógica Proposicional em relação à representação de conhecimento sobre relações e propriedades de objetos no mundo real. Forneça um exemplo de uma sentença que seria difícil ou impossível de representar eficientemente em LP, mas que a Lógica de Predicados resolveria.

Referências Bibliográficas

RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial: Um Enfoque Moderno**. 2. ed. Rio de Janeiro: Prentice Hall, 2004.