

Lógica Computacional

Aula 1 - Apresentação

António Ravara Simão Melo de Sousa

Funcionamento da Unidade Curricular

Descrição geral

- Introdução à lógica proposicional e de primeira ordem : noção de linguagem (sintaxe), de tabelas de verdade e de estrutura de interpretação (semântica) e de sistema dedutivo (sistema de prova).
- Resultados de correcção e completude do sistema de prova relativamente à semântica.
- Mecanismos computacionais : forma normal conjuntiva e forma clausal; Skolemização e unificação; resolução.
- Lógica de Floyd-Hoare para programas imperativos

Saber

- Conhecer a sintaxe e a semântica da lógica de primeira ordem.
- Conhecer os sistemas de dedução natural da lógica de primeira ordem.
- Conhecer o método de Resolução para a lógica de primeira ordem.
- Conhecer a lógica de Floyd-Hoare para programas imperativos.

Fazer

- Escrever fórmulas a partir de descrições em língua natural.
- Calcular semântica e axiomáticamente a validade lógica de fórmulas.
- Usar algoritmos de resolução para estabelecer a validade lógica de fórmulas.
- Provar a correcção de pequenos programas imperativos.

Soft-Skills

- Capacidade de raciocínio abstracto e rigoroso.
- Capacidade de manipulação de estruturas formais.
- Aprender a aprender.

Os acetatos de apoio às aulas teóricas não substituem os livros de texto.

Bibliografia

- Logic in Computer Science: modelling and reasoning about systems (2nd edition). Michael Huth and Mark Ryan. Cambridge University Press, 2004.
- Mathematical Logic: a course with exercises. Part I: propositional calculus, boolean algebras, predicate calculus. René Cori e Daniel Lascar. Oxford Press, 2007.
- Rigorous Software Development, An Introduction to Program Verification. José Carlos Bacelar Almeida, Maria João Frade, Jorge Sousa Pinto, Simão Melo de Sousa. Series: UTiCS, Springer Verlag. 2011.
- A First Course in Logic: An Introduction to Model Theory, Proof Theory, Computability, and Complexity. Shawn Hedman. Oxford Texts in Logic, 2004.
- Language Proof and Logic (4th edition). Jon Barwise and John Etchemendy. CSLI Publications, 2003.

Motivação

O que é a lógica?

A lógica está na base do raciocínio

- É um processo inerentemente humano, tão básico que vimos todos de nascença equipados com essa capacidade:
temos uma ideia intuitiva e inata do que é um processo dedutivo correcto e o que é "disparatado".
- Qualquer processo racional de tirar conclusões a partir de hipóteses é baseado em lógica.
- Exemplos de afirmações que podem ou não ser válidas:
 - ou gosto ou não gosto de lógica
 - hoje chove e não chove
 - hoje é dia 18 e dia 19
 - se apanhar o comboio das 18h não chego atrasado
 - não gostava de lógica se fosse marciano
 - $\forall \varepsilon > 0 \exists \delta > 0 \forall x (|x - a| < \delta \rightarrow |f(x) - b| < \varepsilon)$

A lógica é uma língua

- Diz-se uma linguagem artificial, por contraste com as línguas naturais (como o Português ou o Inglês, ambíguas e difíceis de representar).
- É uma linguagem (formal, ou simbólica), que usa símbolos próprios ($\vee, \rightarrow, \forall, \dots$), com sintaxe e semântica claras, distinguidas e rigorosamente definidas.
- É universal, precisa e dedicada ao objectivo.
- Equipada com regras para decidir da validade de uma afirmação (escrita na linguagem em causa), eventualmente a partir de outras afirmações válidas.

Exemplos de ambiguidades e paradoxos

- O Pedro foi ao médico e ficou doente.
Causalidade?
- O Pedro está em casa ou a Rita está em casa e a Ana está feliz.
Como associar?
- Seja n o mais pequeno natural que não pode ser definido em menos de 20 palavras.
A frase tem menos de 20 palavras?!
- Seja C o conjunto que contém todos os conjuntos.
Contém-se a si mesmo?
- Só digo mentiras!
Se é verdade, não menti, logo é falsa?!
Se é falsa, menti, logo é verdade?!

Motivação

Porquê estudar lógica? Entender o raciocínio!

Consequência lógica

- Se uma afirmação A decorre de outra B , não pode B ser verdade sem que A o seja.
- Serve também para mostrar que uma teoria é falsa: se uma teoria económica diz que a inflação é consequência de uma baixa taxa de desemprego, e se se observa simultaneamente baixa inflação e baixo desemprego, a teoria tem que ser revista.
- Permite obter provas: objecto matemático irrefutável que estabelece que dado facto é verdade.

Consequência lógica

- Se o metro se atrasar e não houver táxis na estação, o Pedro chega tarde. O Pedro não chegou tarde, mas o metro atrasou-se. Logo, havia táxis na estação.
- Se está a chover e a Ana não tiver um guarda-chuva com ela, vai-se molhar. A Ana não se está a molhar, mas está a chover. Logo, tem um guarda-chuva com ela.

Verificação

Intuitivamente, as frases reflectem raciocínios válidos, mas como verificar?

Na verdade, o raciocínio que prova uma também serve para provar a outra.

Formalização

- Hipótese 1: se p e não q então r .
- Hipótese 2: não r e p .
- Tese: q .

Prova

- Hipótese 1 é equivalente a não p ou q ou r .
- Hipótese 2 diz que p e não r .
- Logo q .

Motivação

Porquê estudar lógica? A lógica na computação

Ferramentas essenciais

- Linguagem para modelar entidades computacionais (equipamentos, sistemas ou programas).
- Regras para raciocínio formal sobre propriedades de termos (palavras) da linguagem.
- O raciocínio formal pode ser executado mecanica e automaticamente.

Aplicações

- Circuitos digitais.
- Inteligência artificial.
- Teoria da complexidade.
- Verificação de programas.

Motivação

Porquê estudar lógica? Áreas de aplicação

Objetivo Desenvolver métodos e/ou dispositivos computacionais que simulem o pensamento humano.

Alguns ramos

- Computação inspirada em fenômenos naturais (genética, neuronal, etc).
- Processamento de linguagem natural.
- Robótica.

Objectivo Classificar problema de acordo com o seu grau de dificuldade.

Alguns problemas

- Determinar se um problema pode ser decidido (resolvido).
Exemplo: problema da paragem (verificar se dado programa com certo input termina).
- Calcular o tempo de execução de um programa em função do tamanho dos dados de entrada.
Programas dizem-se polinomiais (deterministas ou não) ou exponenciais.
- $P = NP$?
É um dos grandes problemas matemáticos em aberto (prémio de 1 milhão de dolares para quem o resolver).

Saber se a classe dos problemas resolúveis deterministicamente em tempo polinomial coincide com a classe dos problemas resolúveis não deterministicamente em tempo polinomial.

Objectivo Garantir que dado programa tem o comportamento pretendido.

São de facto dois problemas: especificar comportamento e provar correcção.

Algumas aplicações

- Síntese de programas correctos.
- Sistemas de tipos.
- Verificação de modelos (“model checking”).