

Lógica Computacional

Aula Teórica 1: Apresentação

António Ravara Simão Melo de Sousa

Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade
Nova de Lisboa

Departamento de Informática, Faculdade Engenharia, LISP & Release Group
Universidade Beira Interior

Objectivos

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

Objectivos

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.

Objectivos

Soft-Skills

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

Por onde estudar

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.

Funcionamento

faltas

Não há marcação de faltas nas aulas.

O que se espera dos alunos

- ▶ Trabalho contínuo!
São 6 créditos, implica investimento autónomo semanal
- ▶ Crítica (construtiva) frontal e atempada.

Lógica: de que se trata?

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
 - ▶ $\forall \varepsilon > 0 \exists \delta > 0 \forall x (|x - a| < \delta \rightarrow |f(x) - b| < \varepsilon)$

Lógica: de que se trata?

Não trata d'A Verdade... disso ocupa-se a Filosofia (ou a Religião).

A lógica é um ramo do conhecimento

- ▶ Faz parte da matemática, mas surge em inúmeros campos.
 - ▶ Os outros ramos da matemática e a física são grandes "clientes".
 - ▶ A filosofia é tanto um "cliente" como um "produtor" importante.
- ▶ Formalização começou na antiguidade.
Os Gregos deram o "pontapé de saída", mas houve muitos outros contributos fundamentais.

Lógica: as figuras clássicas

Aristóteles: o pai da lógica dedutiva

- ▶ Viveu em Atenas de 384 a 322 A.C.
- ▶ Definiu as bases axiomáticas do raciocínio (lógico).
- ▶ Junto com Sócrates e Platão, estabeleceu as bases do conhecimento científico da Europa.

Euclides: o pai da prova matemática

- ▶ Viveu em Alexandria, cerca do ano 300 A.C.
- ▶ Formalizou a Geometria, baseada em princípios matemáticos rigorosos.
- ▶ O seu livro “Os elementos” foi a referência central até ao Séc XIX.

Lógica: a formalização da Matemática

Georg Cantor: o pai da teoria de conjuntos

- ▶ Nasceu na Rússia em 1845; morreu na Alemanha em 1918.
- ▶ Inventou a teoria de conjuntos, definiu a noção de infinito e de números transfinitos; mostrou que há mais reais que naturais.
- ▶ Estabeleceu os fundamentos da Matemática moderna.

David Hilbert (veremos à frente) disse em 1926 “Ninguém nos expulsará do paraíso que Cantor criou para nós”.

Bertrand Russell: o pai da Lógica como fundação da Matemática

- ▶ Viveu em Inglaterra de 1872 a 1970.
- ▶ Estudou os fundamentos da Matemática e descobriu o paradoxo que abalou a teoria de conjuntos.
- ▶ Escreveu “The Principles of Mathematics” mostrando que a Matemática pode ser deduzida de um pequeno número de princípios.

Lógica: a metamatemática

David Hilbert: o pai da “prova automática”

- ▶ Viveu na Alemanha de 1862 a 1943.
- ▶ O mais importante matemático da época.
- ▶ Definiu os Espaços de Hilbert, a matemática da relatividade e da quântica (“A Física é demasiado difícil para os Físicos”) , a Teoria da Prova e a Lógica Matemática.
- ▶ Em 1900 estabeleceu os 23 mais importantes problemas da Matemática, alguns deles estão ainda hoje por resolver.
- ▶ Definiu um programa de investigação para formular TODA a Matemática com bases sólidas e acentes na Lógica (a partir de um conjunto finito de axiomas, provados consistentes).
- ▶ No Congresso dedicado aos seus 70 anos, em 1923, afirmou “Temos que saber, viremos a saber!” para sublinhar que a Matemática desvendaria toda a Verdade.

Lógica: a consciência dos limites

Kurt Gödel: o pai da incompletude

- ▶ Nasceu na Austria em 1906; morreu nos EUA em 1978.
- ▶ Mostrou a completude da Lógica de 1ª ordem (no doutoramento, aos 24 anos) e a incompletude de qualquer sistema axiomático capaz de descrever a aritmética dos naturais.

Anunciou o resultado no Congresso de Hilbert (destruindo assim o programa que estava nesses dias em apoteose).

- ▶ Incompletude: há proposições verdadeiras num sistema consistente que não podem ser provadas.
- ▶ Estabeleceu as bases da Computação, com a Teoria das funções recursivas, e entre outros trabalhos importantes, mostrou soluções paradoxais para as equações de campo da relatividade generalizada, universos rotativos que permitiriam

Lógica: o berço da computação

Church: o pai das linguagens de programação

- ▶ Viveu nos EUA de 1903 a 1995.
- ▶ Contribuiu para a Lógica Matemática e para a Teoria da Computação, tendo definido o Cálculo-Lambda para provar que é indecidível determinar se dada proposição arbitrária de uma teoria matemática é ou não verdadeira.
- ▶ A simplicidade e expressividade do Cálculo-Lambda, que permite codificar funções, fizeram dele a primeira linguagem de programação.

Lógica: o berço da computação

Alan Turing: o pai da computação

- ▶ Viveu em Inglaterra de 1912 a 1952.
- ▶ Definiu o primeiro modelo de computação (uma máquina que podia executar qualquer programa): a Máquina Universal de Turing.
- ▶ Mostrou a indecidibilidade do “Halting Problem”.
- ▶ Estabeleceu a Tese de Church-Turing:
Toda a função algoritmicamente calculável é computável por uma Máquina de Turing.
- ▶ Descodificou o sistema de encriptação da marinha Alemã. Churchill disse que não é um homem que ganha uma guerra, mas que sem Turing a Grã-Bretanha teria provavelmente perdido a 2ª guerra.

Lógica: uma linguagem formal

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 (\forall , \rightarrow , ∇ , \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.

Lógica: nem tudo é simples e resolúvel...

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?!

Processo dedutivo

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.

Exemplo de dedução

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.

Verificação de consequência lógica

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 .

O que dá a lógica à 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.

Inteligência artificial

Objectivo

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

Alguns ramos

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

Teoria da complexidade

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.
- ▶ $P = NP$?
Saber se a classe dos problemas resolúveis deterministicamente em tempo polinomial coincide com a classe dos problemas resolúveis não deterministicamente em tempo polinomial.

Verificação de programas

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”).