Lógica Proposicional e de 1a Ordem

Inteligência Artificial

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

> Basedo no material do Prof. Fernando S. de Aguiar Neto

Lógica de Proposicional e Inferência

Representando conhecimento e inferindo novo conhecimento

Queremos entender se há lógica entre um conjunto de premissas e uma conclusão.

Exemplos:

- Se ganhar na loteria então aposentar
- Não aposentei
- Que conclusões podemos chegar sobre a loteria?

Usando lógica proposicional

Para representar o conhecimento do slide anterior, podemos usar lógica proposicional (não será necessária a lógica de 1a Ordem). Considerem: P = ganhou na loteria; Q = aposentou

$$
\begin{array}{lll}
1 & P & \Rightarrow & Q \\
2 & \neg Q & &
\end{array}
$$

P é True ou False? Assumimos que ambas as premissas são verdadeiras, uma vez que refletem nosso conhecimento de mundo. Logo existe um modelo onde essas relações são verdadeiras

Usando lógica proposicional - Intuição

Vamos assumir que queremos verificar que P é Falso (3). Conseguimos manipular (1) e (2) para chegar em (3)? Para formalizar os métodos que poderão provar casos como esse precisaremos ver 3 conceitos para prova de teoremas e revisar algumas propriedades lógicas

$$
\begin{array}{llll}
1 & P & \Rightarrow & Q \\
2 & \neg Q & & \\
3 & \neg P & &
\end{array}
$$

Validade e Tautologia

- Uma sentença é válida ou tautologia se é verdadeira em todos os modelos possíveis
- e.g. $P \lor \neg P$; $(P \implies Q) \lor \neg Q$

Dedução ou consequência lógica

- Sejam duas sentenças α e β , dizemos que $\alpha \models \beta$, se e somente se $\alpha \implies \beta$ é válida/ tautologia
- lê se $\alpha \models \beta$, β é consequência lógica de α , ou α deduz β
- podemos pensar em α como uma premissa e β como conclusão ou consequência dessas premissas

Dedução ou consequência lógica

Quais das relações abaixo, são consequências lógicas de fato?

- $P \models Q$?
- $P \implies Q \models Q$?
- $(P \implies Q) \land P \models Q$?

Satisfabilidade

- Uma sentença é satisfazível, se existir ao menos um modelo onde é verdadeira
- e.g. $P \land \neg P$ é insatisfazível
- e.g. $P \implies Q$ é satisfazível
- Decidir se uma sentença é satisfazível ou não é chamado de SAT Problem e é NP-Completo

Satisfabilidade

Sejam α e β sentenças

- α é válido, se e somente se $\neg\alpha$ é insatisfazível

Sejam α e β sentenças

- α é válido, se e somente se $\neg\alpha$ for insatisfazível
- α é sempre verdadeiro, se e somente se $\neg\alpha$ for sempre falso
- Podemos extender esse conceito para a dedução

Usando os conceitos para provas

Sejam α e β sentenças. Notem que podemos tratar $\alpha \models \beta$ como uma sentença:

- $\alpha \models \beta$ é válido, se e somente se $\neg(\alpha \models \beta)$ for insatisfazível
- $\alpha \implies \beta$ é válido, se e somente se $\neg(\alpha \implies \beta)$ for insatisfazível
- Podemos provar que β é consequencia lógica de α , provando que não há modelo onde $\alpha \implies \beta$ é verdadeiro

Usando os conceitos para provas

Vamos manipular a fórmula para entendermos as implicações disso:

- $\alpha \models \beta$ é válido, se e somente se $\neg(\alpha \implies \beta)$ for insatisfazível
- $\neg(\alpha \implies \beta) \equiv \neg(\neg\alpha \vee \beta) \equiv \alpha \wedge \neg\beta$
- logo: $\alpha \models \beta$ é válido, se e somente se $\alpha \wedge \neg\beta$ for insatisfazível

Usando os conceitos para provas

Vamos entender $\alpha \models \beta$ é válido, se e somente se $\alpha \wedge \neg\beta$ for insatisfazível:

- Dizemos que $\alpha \models \beta$ se e somente se, não existir modelo onde as premissas e a negação da conclusão possam ser verdade simultaneamente
- Prova por contradição, refutação, ou absurdo

Revisão de Álgebra Booleana

- $(\alpha \wedge \beta) \equiv (\beta \wedge \alpha)$ comutatividade de $\wedge$
- $(\alpha \vee \beta) \equiv (\beta \vee \alpha)$ comutatividade de $\vee$
- $((\alpha \wedge \beta) \wedge \gamma) \equiv (\alpha \wedge (\beta \wedge \gamma))$ associatividade de $\wedge$
- $((\alpha \vee \beta) \vee \gamma) \equiv (\alpha \vee (\beta \vee \gamma))$ associatividade de $\vee$
- $\neg(\neg\alpha) \equiv \alpha$ eliminação de duplo negativo
- $(\alpha \implies \beta) \equiv (\neg\beta \implies \neg\alpha)$ contraposição
- $(\alpha \implies \beta) \equiv (\neg\alpha \vee \beta)$ eliminação de implicação
- $(\alpha \iff \beta) \equiv ((\alpha \implies \beta) \wedge (\beta \implies \alpha))$ eliminação de bi-implicação
- $\neg(\alpha \wedge \beta) \equiv (\neg\alpha \vee \neg\beta)$ De Morgan
- $\neg(\alpha \vee \beta) \equiv (\neg\alpha \wedge \neg\beta)$ De Morgan
- $(\alpha \wedge (\beta \vee \gamma)) \equiv ((\alpha \wedge \beta) \vee (\alpha \wedge \gamma))$ distributividade de $\wedge$ sobre $\vee$
- $(\alpha \vee (\beta \wedge \gamma)) \equiv ((\alpha \vee \beta) \wedge (\alpha \vee \gamma))$ distributividade de $\vee$ sobre $\wedge$

Consequências Lógicas

- As fórmulas apresentadas no slide anterior nos ajudam a entender e realizar diversas manipulações nas sentenças da lógica proposicional
- O conceito de inferência será explicada no contexto dessa lógica mais simples para aí então extendermos para a lógica de primeira ordem
- Além das manipulações, vamos ver algumas consequências lógicas básicas

Consequências Lógicas Básicas - Adição

Quando P é verdadeiro, P ou qualquer coisa também será verdadeiro

- Adição
- $P \models (P \vee Q)$
- Em notação de regra de inferência:

$$\frac{P}{P \vee Q}$$

Consequências Lógicas Básicas - Simplificação e Conjunção

Quando P e Q são verdadeiros podemos separar em duas premissas, P é verdadeiro e Q é verdadeiro; e vice-versa

Simplificação

- $(P \implies Q) \wedge (Q \implies R) \models (P \implies R)$
- $\frac{P \wedge Q}{P \quad Q}$

Conjunção

- $(P) \wedge (Q) \models P \wedge Q$
- $\frac{P \quad Q}{P \wedge Q}$

Consequências Lógicas Básicas - Silogismo

Silogismo

Se P implica em Q e Q implica em R, P implica em R, usando a transitividade da implicação

- $(P \implies Q) \land (Q \implies R) \models (P \implies R)$
- $\dfrac{P \implies Q \quad Q \implies R}{P \implies R}$

Silogismo Disjunto

Se um dos literais é verdade, e sabemos que um deles não é, o outro com certeza tem que ser

- $(P \lor Q) \land (\neg P) \models Q$
- $\dfrac{P \lor Q \quad \neg P}{Q}$

Consequências Lógicas Básicas - Modus Ponnens

Se temos uma implicação e o antecedente é verdadeiro, o posterior é verdadeiro

- $(P \implies Q) \land (P) \models Q$
- $\dfrac{P \implies Q \quad P}{Q}$

Consequências Lógicas Básicas - Modus Tollens

Sempre que P acontece Q também acontece, como Q não aconteceu, P também não 1

- $(P \implies Q) \land \neg Q \models \neg P$
- $\dfrac{P \implies Q \qquad \neg Q}{\neg P}$

> Cuidado, não podemos dizer 'Como P não aconteceu, Q também não'

Inferência: Forward Checking

- Com o conjunto de premissas, ou cláusulas são construídas novas regras
- Quando alguma das novas regras é igual à conclusão, temos a prova

Exercício

Usando as consequências lógicas básicas e manipulações algébricas, mostre que:

1. $(X \implies Y) \land (X) \vDash Y$
2. $\neg(\neg P \implies \neg Q) \vDash (\neg Q)$
3. $((\neg P \lor \neg Q) \implies (R \land S)) \land (R \implies T) \land (\neg T) \vDash P$

Refutação

- De $\alpha \models \beta$ é válido, se e somente se $\alpha \land \neg\beta$ for insatisfazível, notamos que podemos negar a sentença posterior a fim de fazer provas
- Essa estratégia é usada na prova por refutação
- Na prova por refutação, negamos a conclusão e a adicionamos às premissas e mostramos que uma contradição aparece, e.g. $P \land \neg P$

Exercício Refutação

Usando a estratégia de refutação e manipulações algébricas, mostre que:

1. $(X \implies Y) \land (X) \vDash Y$
2. $\neg(\neg P \implies \neg Q) \vDash (\neg Q)$
3. $((\neg P \lor \neg Q) \implies (R \land S)) \land (R \implies T) \land (\neg T) \vDash P$

Considerações Finais

- Nos exercícios todas as deduções eram verdadeiras
- Seria possível provar uma dedução inválida usando forward-checking?
- Iremos ver como normalizar as cláusulas, e então explorar métodos de dedução

Lógica de 1a Ordem

A lógica proposicional nos permite representar muitas estruturas lógicas:

- Se está quente não está frio
- Se o computador está funcionando é porque está ligado
- Gelo indica que a água está fria e sólida e vice-versa
- Especialmente relações de causa e efeito

Motivação

Mas como usar a lógica proposicional para representar as seguintes afirmações?

- Todo professor é inteligente
- Fernando é um professor
- Existem pessoas inteligentes que não são professores
- A prova do Fernando é difícil

Motivação

- Notamos que a lógica proposicional tem dificuldades em estruturar essas afirmações
- Mesmo quando conseguimos forçar uma representação, a estrutura não permite muitas manipulações
- fica difícil relacionar regras diferentes

Solução - Lógica de 1a Ordem

- Para poder representar esse tipo de relações precisaremos de uma ferramenta lógica mais poderosa que a lógica de proposicional
- Utilizaremos a lógica de primeira ordem
- Também chamada de lógica de predicados

Ideia geral

- Utilizaremos estruturas extras para representar relações e pertinência
- e.g. Fernando é estudante
- Estudante(Fernando)
- e.g. Fernando da aulas de IA
- Aula(Fernando, IA)

Ideia geral

- Muitas vezes será importante estabelecer funções, essas funções retornarão novos objetos
- Prova(Fernando)
- Prova(José)
- Prova(ProfX)
- Notem que não preciso definir um símbolo para a prova de cada professor existente no universo, defino a função prova que recebe um professor e retorna sua prova

Estrutura

- As sentenças são similares às da lógica proposicional
- mas a lógica de 1a ordem extende os conceitos da lógica proposicional
- adicionando termos, predicados, funções e quantificadores

Termo

- Os termos são expressões lógicas referentes a um objeto
- As vezes é inconveniente ter que criar uma constante para cada objeto
- e.g. ProvaFernando, ProvaJosé, ProvaMario, ...
- Podemos definir funções para sanar esse problema
- e.g. Prova(Fernando)
- A estrutura é: um símbolo de função, com termos sendo passados como parâmetro
- Notem que função aqui tem uma notação diferente da programação

Termo

- Podemos definir funções para sanar esse problema
- e.g. Prova(Fernando)
- A estrutura é: um símbolo de função, com termos sendo passados como parâmetro
- Notem que função aqui tem uma notação diferente da programação

Sentença Atômica

- Usada para representar fatos
- Pode ser apenas um predicado
- Predicado seguido de termos: indicando uma relação entre os termos, e.g.
- Depois(B,A); Gosta(Fernando, Chocolate); True
- Por fim, pode ser a relação de igualdade entre termos
    - Terá um funcionamento similar à atribuição em computação
- e.g. Professor(IA) = Fernando

Sentença Complexa

- Podemos ligar sentenças usando conectivos lógicos
  - Que possuem a mesma semântica que na lógica proposicional
- Gerando sentenças maiores e de significado mais complexo
- Além disso é possível usar Quantificadores

Quantificadores

- Para estabelecer relações de para todos ($\forall$) e existência ($\exists$) utilizamos quantificadores e variáveis
- variável é um termo especial que irá indicar um objeto depedendo do quantificador, usaremos letras minúsculas para variáveis
- a notação é bem similar à que estamos acostumados da matemática
- $\forall x\ Estudante(x) \implies \neg Gosta(x, Prova(Fernando))$
- $\exists \mathrm{x}\ Estudante(x) \implies Gosta(x, Prova(Fernando))$

Combinando Qualificadores

- Podemos aninhar dois ou mais quantificadores
- e.g. $\forall x\ \forall y\ Irmao(x,y) = Irmao(y, x)$
- e.g. $\forall x\ \exists y\ Ama(x,y)$
- e.g. $\exists x\ \forall y\ Ama(x, y)$
- As duas últimas regras são diferentes?

Combinando Qualificadores

- Cuidado ao aninhar quantificadores, usando variáveis com o mesmo nome
- e.g. $\forall x\ (\ Professor(x) \lor (\ \exists x\ Irmao(x, Fernando)\ )\ )$
- Ainda que por contexto seja possível entender os escopos, o ideal é usar variáveis diferentes para escopos diferentes, sempre
- e.g. $\forall x\ (\ Professor(x) \lor (\ \exists y\ Irmao(y, Fernando)\ )\ )$

Relações entre quantificadores

Conseguimos representar o mesmo conceito das fórmulas abaixo, sem usar o mesmo quantificador?

- 1 $\forall x\ Gosta(x, Chocolate)$
- 2 $\neg\exists x\ Gosta(x, Prova)$
- 3 $\neg\exists x\ Gosta(x, Cafe) \land Gosta(x, Achocolatado)$

Suposições Importantes

- Nomes únicos: Cada constante representa um objeto distinto
- Mundo fechado: Toda sentença atômica desconhecida é considerada falsa
- Closure: O domínio é restrito ao que está definido nas constantes

Exercício 1

Qual o significado das seguintes expressões? Quais são lógicas (i.e. existe um modelo que satisfaça a sentença)

- $(\exists\ x\ x = x) \implies (\forall y \exists z\ y = z)$
- $\forall x\ P(x) \lor \neg P(x)$
- $\forall x\ Esperto(x) \lor (x = x)$

Exercício 2

Considere um vocabulário com os seguintes símbolos:

- 1 Occupation(p, o): Predicado. Pessoa p tem uma ocupação o
- 2 Customer(p1, p2): Predicado. Pessoa p1 é clinete da pessoa p2.
- 3 Boss(p1, p2): Predicado. Pessoa p1 é chefe da pessoa p2.
- 4 Doutor, Cirurgião, Advogado, Ator: Constantes denotando ocupações.
- 5 Emily, Joe: Constantes denotando pessoas.

Exercício 2 (cont.)

Uso os símbolos da parte 1 para escrever os fatos abaixo usando lógica de primeira ordem:

- 1 Emily é cirurgiã ou advogada
- 2 Joe é um ator, mas ele também tem outro emprego
- 3 Todos os cirugiões são doutores
- 4 Joe não tem um advogado (i.e. não é cliente de nenhum advogado)
- 5 Emily tem um chefe que é advogado
- 6 Existe um advogado cujos clientes são todos doutores
- 7 Todo cirurgião tem um advogado

Inferência na Lógica de Predicados

Cláusula

- literais são sentenças atômicas negadas ou não
- e.g. Pai(x), $\neg y$
- uma cláusula é uma disjunção de literais
- e.g. $Pai(x) \vee \neg Pai(x)$

Instanciação

Regra Instanciação Universal

- Se algo é válido para todos os valores de uma variável, podemos instânciar essa variável por qualquer termo sem variáveis 1 . i.e. podemos substituir a variável por um (ground) termo. Eliminando o quantificador ∀
- e.g. $\forall x Rei(x) \implies Rico(x)$
- Podemos realizar a substituição $\beta = \{x/John\}$
- $Rei(John) \implies Rico(John)$

Regra Instanciação Existencial

- Se existe algo, podemos representar usando uma constante nova, ainda não usada na nossa base de conhecimento. Eliminando o quantificador ∃
- $\exists Coroa(x) \wedge NaCabea(x, John)$
- Podemos inferir que (aplicando a substituição $\beta = \{x/C1\}$)
- $\exists Coroa(C1) \wedge NaCabea(C1, John)$
- C1 é chamada de constante de Skolem e o processo recebe o nome de skolemização
- Notem que a sentença nova não é identica a inicial, mas só é satisfazível se a original for satisfazível, sendo útil para inferência

Regra Instanciação Existencial - Funções de Skolem

- Para garantir só seja satisfazível se a original for satisfazível precisamos cuidar de um caso especial
- $\forall x\ (\exists y\ Pai(y, x))$
- Se substituirmos cegamente alteramos o sentido completamente:
- $\forall x\ (Pai(C1,x))$
- o esperado era 'para todos os x, x tem pai', mas obtivemos que existe um C1 que é pai de todos

Regra Instanciação Existencial - Funções de Skolem

- $\forall x\ (\exists y\ Pai(y, x))$
- Sempre que um quantificador universal ($\forall$) englobar um existencial, devemos aplicar funções de Skolem para atrelar os dois escopos
- A substituição é na forma:
- $\forall x\ (Pai(F(x), x))$
- Dizemos que existe uma dependência entre o termo e a variável do quantificador universal
- podemos interpretar como: 'para todo x, F(x) é pai de x'

Reduzindo para Lógica proposicional

- Poderíamos usar a instanciação para reduzir a lógica de predicados para a lógica proposicional
- e.g. $\forall x Rei(x) \wedge Ganancioso(x) \implies Mau(x)$
- Substituindo x por todas as constantes possíveis (e.g. John e Richard), i.e. β 1 = $x/John$,β 2 = $x/Richard$
- $Rei(John) \wedge Ganancioso(John) \implies Mau(John)$
- $Rei(Richard) \wedge Ganancioso(Richard) \implies Mau(Richard)$
- podemos tratar Rei(Richard) e Rei(John), etc. como predicados, e.g. RR e RJ

Reduzindo para Lógica proposicional

- Essa ideia é quase perfeita infelizmente, ela falha em 'proposicionar' funções, e.g. Pai(John)
- Pois sempre poderiamos aninhar uma quantidade maior de funções, gerando infinitas possibilidades, logo infinitos predicados
- e.g. Pai(Pai(Pai(....Pai(John))

FCN para Lógica de Predicados

- Vimos como transformar sentenças da lógica proposicional para Forma Conjuntiva Normal
- Veremos como transformar sentenças da lógica de 1a ordem da mesma forma
- O algoritmo é similar, mas precisamos lidar com quantificadores
- Lembrando que na FCN temos conjunções de cláusulas , cada cláusula é uma disjunção de literais, e os literais são sentenças atômicas negadas ou não
- Converteremos as sentenças para uma FCN que seja inferencialmente equivalente , i.e. caso a sentença original seja insatisfazível, a sentença na FCN também será

FCN para Lógica de Predicados

- 1 Eliminar bi-implicações
- 2 Eliminar implicações
- 3 Renomear variáveis repetidas
- 4 Mover $\neg$ para próximo dos literais
- 5 Skolemizar (i.e. remover quantificadores existenciais, usando Instanciação Existencial)
- 6 Remover quantificadores universais
- 7 Distribuir $\vee$ sobre $\wedge$

FCN para Lógica de Predicados - Exemplo

- Considerem a seguinte afirmação: 'Todo mundo que gosta de todos os animais é amado por alguém'
- $\forall x[\forall y\ Animal(y) \implies Ama(x,y)] \implies [\exists y\ Ama(y,x)]$

FCN para Lógica de Predicados - Exemplo

- Passo 1: Eliminar bi-implicações. Passo 2: Eliminar Implicações:
- $\forall x\, [\forall y\, Animal(y) \implies Ama(x,y)] \implies [\exists y\, Ama(y,x)]$
- $\forall x\, [\neg \forall y\, \neg Animal(y) \vee Ama(x,y)] \vee [\exists y\, Ama(y, x)]$

FCN para Lógica de Predicados - Exemplo

- Passo 3: Renomear variáveis repetidas (i.e. escopos diferentes, nomes diferentes)
- $\forall x\ [\neg\forall y\ \neg Animal(y) \vee Ama(x,y)] \vee [\exists y\ Ama(y, x)]$
- $\forall x\ [\neg\forall y\ (\neg Animal(y) \vee Ama(x,y))] \vee [\exists z\ Ama(z, x)]$

FCN para Lógica de Predicados - Exemplo

- Passo 4: Mover $\neg$
- $\forall x\, [\neg \forall y\, (\neg Animal(y) \vee Ama(x,y))] \vee [\exists z\, Ama(z, x)]$
- Sabemos que $\neg \forall x\ p \equiv \exists x\ \neg p$ e $\neg \exists x\ p \equiv \forall x\ \neg p$
- $\forall x\, [\exists y\, \neg(\neg Animal(y) \vee Ama(x,y)))] \vee [\exists z\, Ama(z, x)]$
- $\forall x\, [\exists y\, (Animal(y) \wedge \neg Ama(x, y)))] \vee [\exists z\, Ama(z, x)]$

FCN para Lógica de Predicados - Exemplo

- Passo 5: Skolemizar
- $\forall x\ [\exists y\ (Animal(y) \land \neg Ama(x, y)))] \lor [\exists z\ Ama(z, x)]$
- Notem que temos um quantificador universal sobre toda a sentença, precisamos usar funções ($\beta = \{y/F(x), z/G(x)\}$)
- $\forall x\ [Animal(F(x)) \land \neg Ama(x,F(x)))] \lor [Ama(G(x), x)]$

FCN para Lógica de Predicados - Exemplo

- Passo 6: Remover quantificadores universais
- $\forall x\ [Animal(F(x)) \land \neg Ama(x,F(x))] \lor [Ama(G(x), x)]$
- $[Animal(F(x)) \land \neg Ama(x,F(x))] \lor Ama(G(x), x)$

FCN para Lógica de Predicados - Exemplo

- Passo 7: Distribuir ∨ sobre ∧
- $[Animal(F(x)) \land \neg Ama(x,F(x)))] \lor Ama(G(x), x)$
- $[Animal(F(x)) \lor Ama(F(x), x)] \land [\neg Ama(x, F(x) \lor Ama(F(x), x)]$

Exercício

Escreva a representação lógica da seguinte base de conhecimento:

- 1 Cavalos, vacas e porcos são mamíferos
- 2 A prole de um cavalo é um cavalo
- 3 Barbazul é um cavalo
- 4 Barbazul é pai de Charlie
- 5 Paternidade e prole são relações inversas
- 6 Todo mamífero tem um pai

Resolução

- A resolução na lógica de primeira ordem será similar à que vimos na lógica de proposicional
- Utilizaremos o conceito de instanciação para gerar uma instância da sentença original, se essa instância for insatisfazível, a sentença completa também o é
  - Note que o oposto não é verdade, essa instância ser sat. não indica que a versão genérica também seja
- Portanto, faremos prova por contradição

Resolução na Lógica de 1a Ordem

- Sendo α, β e γ literais

$$\cdot \frac{\alpha \vee \beta}{\beta \vee \gamma}$$

- Idêntico ao que vimos anteriormente na lógica proposicional
- mas precisamos por as variáveis na nossa regra

Resolução na Lógica de 1a Ordem

- Sendo α, β e γ literais sujeitos a uma ou mais variáveis
- $$\frac{\alpha(x) \vee \beta(y)}{\beta(y) \vee \gamma(z)}$$
- Caso o termo atrelado à sentença seja idêntico em literais opostos, podemos resolvê-lo

Resolução na Lógica de 1a Ordem - Instanciando

- Sendo α, β e γ literais sujeitos a uma ou mais variáveis
  - $\alpha(k) \vee \beta(y) \quad \neg\alpha(n) \vee \gamma(z), subst = \{n|k\}$
  - $\frac{\alpha(k) \vee \beta(y)}{\beta(y) \vee \gamma(z)}$

    $\beta(y) \vee \gamma(z)$
- Caso o termo atrelado à sentença seja idêntico em literais opostos, podemos resolvê-lo

Resolução na Lógica de 1a Ordem - Instanciando

- Podemos fazer a instanciação universal à vontade, pois consideramos que todas as variáveis estão associadas à quantificadores universais
- Skolemizamos os quantificadores existenciais e omitimos os quantificadores uniersais

Resolvendo - Exemplo

- 1 $P(x,y)$ $Q(z)$
- 2 $\neg P(l, k)$ $Q(n)$
- 3 $Q(z)$ $Q(n)$ de 1,2 $\beta = \{l|x, k|y\}$
- 4 $Q(z)$ simplificando 3 $\beta = \{n|z\}$

Resolvendo - Exemplo

- 1 Mas nem sempre a instanciação é possível
- 2 $P(x,y, x)$ $Q(x)$
- 3 $\neg P(l, l, m)$ $Q(n)$
- 4 Nesse caso não podemos fazer substituições que padronizem o literal P()

Resolvendo - Exemplo

- 1 Podemos substituir funções de Skolem
- 2 $P(x,F(x))$ $Q(F(x))$
- 3 $\neg P(y,n)$ $Q(n)$
- 4 $Q(F(x))$ $Q(F(x))$ $\beta = \{x|y, n|F(x)\}$
- 5 $Q(F(x))$ simplificando 3

Provando Teoremas

- Seguiremos a mesma lógica que usamos anteriormente
- Negaremos a conclusão e mostraremos que ela gera uma contradição com os axiomas

Provando - Exemplo

Considerem as seguintes premissas e questionamento:

- 1 Todos que amam todos os animais, são amados por alguém
- 2 Ninguém ama alguém que mata um animal
- 3 Jack ama todos os animais
- 4 Jack ou Curiosity mataram o gato, chamado Tuna
- 5 Curiosity matou o gato?

Provando - Exemplo

Considerem as seguintes premissas e questionamento:

- 1 Todos que amam todos os animais, são amados por alguém
- 2 Ninguém ama alguém que mata um animal
- 3 Jack ama todos os animais
- 4 Jack ou Curiosity mataram o gato, chamado Tuna
- 5 Todo gato é um animal (precisamos adicionar esse conhecimento de mundo à nossa base)
- 6 Curiosity matou o gato?

Provando - Exemplo

Transformando em lógica de predicados:

- 1 $\forall\ x[\forall\ y Animal(y) \implies Ama(x,y)] \implies [\exists\ z Ama(z,x)]$
- 2 $\forall\ x[\exists\ y Animal(y) \implies Matou(x,y)] \implies \neg\exists\ z Ama(z,x)$
- 3 $\forall\ x Animal(x) \implies Ama(Jack,x)$
- 4 $Matou(Jack,Tuna) \vee Matou(Curiosity,Tuna)$
- 5 $Gato(Tuna)$
- 6 $\forall\ x Gato(x) \implies Animal(x)$
- 7 $Mata(Curiosity, Tuna)$ (conclusão que queremos provar)

Provando - Exemplo

- 1 $\forall\ x[\forall\ y Animal(y) \Longrightarrow Ama(x,y)] \Longrightarrow [\exists\ z Ama(z,x)]$
- 2 $\forall\ x[\exists\ y Animal(y) \wedge Matou(x, y)] \Longrightarrow \neg\exists\ z Ama(z,x)$
- 3 $\forall\ x Animal(x) \Longrightarrow Ama(Jack,x)$
- 4 $Matou(Jack,Tuna) \vee Matou(Curiosity,Tuna)$
- 5 $Gato(Tuna)$
- 6 $\forall\ x Gato(x) \Longrightarrow Animal(x)$
- 7 $\neg Mata(Curiosity,Tuna)$ (negando a conclusão)

Provando - Exemplo

Trazendo para uma instância na FCN:

- $Animal(F(x)) \vee Ama(G(x),x)$
- $\neg Ama(x,F(x)) \vee Ama(G(x),x)$
- $\neg Animal(y) \vee \neg Matou(x,y) \vee \neg Ama(z,x)$
- $\neg Animal(x) \vee Ama(Jack,x)$
- $Matou(Jack,Tuna) \vee Matou(Curiosity, Tuna)$
- $Gato(Tuna)$
- $\neg Gato(x) \vee Animal(x)$
- $\neg Mata(Curiosity,Tuna)$

Provando - Exemplo

Um truque muito prático é renomear todas as variáveis em diferentes cláusulas:

1. $Animal(F(x)) \lor Ama(G(x),x)$
2. $\neg Ama(y,F(y)) \lor Ama(G(y),y)$, $\beta = \{x|y\}$
3. $\neg Animal(k) \lor \neg Matou(l,k) \lor \neg Ama(z,l)$, □={y|k, x|l}
4. $\neg Animal(m) \lor Ama(Jack,m)$, $\beta = \{x|m\}$
5. $Matou(Jack,Tuna) \lor Matou(Curiosity, Tuna)$
6. $Gato(Tuna)$
7. $\neg Gato(n) \lor Animal(n)$, $\beta = \{x|n\}$
8. $\neg Mata(Curiosity,Tuna)$

Provando - Exemplo

9. $R(8, 5)$
10. $R(1, 3)\beta = \{k|F(x), z|G(x), l|k\}$ 6
11. $\square$ $R(9, 10)\beta\{l|Jack, F(x)|Tuna\}$
12. $R(9, 3)\beta = \{l|Jack, k|T\ una\}$
13. $\square$ $R(11, 1)\beta = \{F(x)|Tuna, z|G(x), x|Jack\}$ 7

6 Notem que eu não poderia fazer F(x) | k

7 Notem que troco algo genérico, por uma constante

Exercício

Escreva a representação lógica da seguinte base de conhecimento:

- 1 Cavalos, vacas e porcos são mamíferos
- 2 A prole de um cavalo é um cavalo
- 3 Barbazul é um cavalo
- 4 Barbazul é pai de Charlie
- 5 Paternidade e prole são relações inversas
- 6 Todo mamífero tem um pai

Exercício

Usando a base de conhecimento do slide anterior, prove (ou disprove):

- 1 Charlie é um mamífero
- 2 Barbazul é um Porco
- 3 Todo mamífero tem uma prole
- 4 Todo mamífero é um cavalo