Lógica para Computação

Linguagem da Lógica Proposicional

Thiago Alves Rocha

thiagoalvesifce@gmail.com

Tópicos

Tópicos

Objetivo

- Desenvolver uma linguagem para modelar situações e informações
- Modelar situações para poder raciocinar sobre elas
- Construir argumentos válidos e que possam ser executados em computadores

Exemplo

- Se o trem chega tarde e não tem táxis na estação então João está atrasado para a reunião.
- João não está atrasado para a reunião.
- O trem chegou tarde.
- O que podemos concluir?

Exemplo

- Se o trem chega tarde e não tem táxis na estação então João está atrasado para a reunião.
- João não está atrasado para a reunião.
- O trem chegou tarde.
- O que podemos concluir?
- Logo, tinha táxi na estação.
- Temos um conjunto de sentenças seguidas por uma outra sentenças
- Um argumento é valido quando a última sentença é consequência lógica das demais

Exemplo

- Se está chovendo e Maria não está com seu guarda-chuva então ela vai ficar molhada.
- Maria não está molhada.
- Está chovendo.
- O que podemos concluir?

Exemplo

- Se está chovendo e Maria não está com seu guarda-chuva então ela vai ficar molhada.
- Maria não está molhada.
- Está chovendo.
- O que podemos concluir?
- Maria está com seu guarda-chuva.
- A sequência de sentenças dos dois exemplos tem a mesma estrutura

Exemplo

- O trem chega tarde e está chovendo
- tem táxi na estação e Maria está com o guarda-chuva
- João está atrasado para a reunião e Maria fica molhada

Exemplo

- O trem chega tarde e está chovendo
- tem táxi na estação e Maria está com o guarda-chuva
- João está atrasado para a reunião e Maria fica molhada
- Será que podemos generalizar esse tipo de raciocínio?

Exemplo

- O trem chega tarde e está chovendo
- tem táxi na estação e Maria está com o guarda-chuva
- João está atrasado para a reunião e Maria fica molhada
- Será que podemos generalizar esse tipo de raciocínio?
- Se p e não q então r. Não r. p. Logo, q.
- Muitas vezes só precisamos da estrutura das sentenças para analisar sua veracidade
- Não precisamos nos preocupar com o tipo de informação das sentenças!

Tópicos

Linguagem

- Para representar informações precisamos de uma linguagem
- A linguagem deve deixar claro a estrutura lógica das sentenças
- Vamos usar a linguagem da Lógica Proposicional
- Essa linguagem usa proposições
- Cada proposição pode ser, ou verdadeira ou falsa.

Exemplos de Proposições

- A soma dos números 3 e 5 é igual a 8
- Maria reagiu de forma violenta com a notícia
- Todo número natural par maior que 2 é a soma de dois números primos
- Todos os marcianos gostam de pizza

Não são Proposições

- Você pode me passar o sal?
- Que horas são?
- Bom dia.

Proposições Atômicas

- Para descrever a linguagem da Lógica Proposocional vamos considerar certas sentenças como atômicas
  - O número 5 é par
  - O trem chega cedo
- Vamos usar letras minúsculas com ou sem índice para representar proposições atômicas: p, q, r, p_1, p_2, x_3

Tópicos

Proposições Compostas

- Podemos compor as proposições atômicas em proposições mais complexas
- Vamos usar os conectivos $\neg, \vee, \wedge, \rightarrow$ para compor as atômicas
- $\neg p$ representa a negação de p.
- $q \vee r$ representa q ou r e é também chamado de disjunção de q e r.
- $p \wedge r$ representa p e r, ou seja, a conjunção de p e r.
- $s \rightarrow t$ representa se s então t, ou seja, s implica em t. Chamamos p de suposição e q de conclusão.

Exemplos de Proposições Compostas

- Sejam as proposições atômicas representando as sentenças:
- p: Eu ganhei na mega-sena na semana passada.
- q: Eu comprei um bilhete da mega-sena.
- r: Eu ganhei na loteria dos sonhos na semana passada.
- As proposições compostas representam as sentenças:
- $\neg p$: Eu não ganhei na mega-sena semana passada. Também pode ser: não é verdade que eu ganhei na mega-sena semana passada.
- $p \vee r$: Eu ganhei na mega-sena semana passada ou eu ganhei na loteria dos sonhos na semana passada.
- $q \wedge p$: Eu comprei um bilhete da mega-sena e eu ganhei na mega-sena na semana passada.
- $p \wedge q$: Se eu ganhei na mega-sena na semana passada então eu comprei um bilhete da mega-sena.

Exemplos de Proposições Compostas

- Podemos continuar usando as regras de construção de proposições repetidamente:
- $p \wedge q \rightarrow \neg r \vee q$
- Como deve ser lido?

Exemplos de Proposições Compostas

- Podemos continuar usando as regras de construção de proposições repetidamente:
- $p \wedge q \rightarrow \neg r \vee q$
- Como deve ser lido?
- p e se q então não r ou q
- Se p e q então não r ou q

Exemplos de Proposições Compostas

- Podemos continuar usando as regras de construção de proposições repetidamente:
- $p \wedge q \rightarrow \neg r \vee q$
- Como deve ser lido?
- p e se q então não r ou q
- Se p e q então não r ou q
- Como queremos que um computador seja capaz de manipular esses dados precisamos retirar ambiguidades
- $(p \wedge q) \rightarrow ((\neg r) \vee q)$

Exercício

Represente as senteças abaixo na linguagem da Lógica Proposicional. Especifique as atômicas.

- Se hoje faz sol, então amanhã não faz sol.
- Roberto estava com ciúmes de Marta ou ele não estava de bom humor.
- Maria e João são namorados
- Hoje vai chover ou fazer sol, mas não ambos.

Solução

- Se o sol brilha muito hoje, então ele não vai brilhar muito amanhã.
- $(p \rightarrow (\neg q))$
- Roberto estava com ciúmes de Marta ou ele não estava de bom humor.
- $(c \vee (\neg b))$
- Maria e João são namorados
- p
- Hoje vai chover ou fazer sol, mas não ambos.
- $(r \vee s) \wedge \neg(r \wedge s)$

Tópicos

1. Introdução
2. Proposições
3. Conectivos
4. Linguagem da Lógica Proposicional
5. Estrutura Indutiva
6. Precedência

Introdução

- Queremos que as fórmulas possam ser processadas por computadores
- Temos que deixar claro qual a estrutura da fórmulas
- Precisamos definir formalmente como as fórmulas são construídas
- A definição deve eliminar ambiguidades

Alfabeto da Lógica Proposicional

Definição

- Seja $Prop = \{a, b, c, ..., a_1, b_1, ..., p_1, ...\}$ o conjunto de proposições atômicas e $Con = \{\wedge, \vee, \neg, \rightarrow\}$ o conjunto de conectivos.
- O alfabeto da linguagem da lógica proposicional é $A_{lp} = Prop \cup Con \cup \{), (\}$

Alfabeto da Lógica Proposicional

Definição

- Seja $Prop = \{a, b, c, ..., a_1, b_1, ..., p_1, ...\}$ o conjunto de proposições atômicas e $Con = \{\wedge, \vee, \neg, \rightarrow\}$ o conjunto de conectivos.
- O alfabeto da linguagem da lógica proposicional é $A_{lp} = Prop \cup Con \cup \{), (\}$

- $((p \wedge (\neg r)) \rightarrow (p \vee s))$ é uma string no alfabeto A_{lp} e é uma fórmula da lógica proposicional

Alfabeto da Lógica Proposicional

Definição

- Seja $Prop = \{a, b, c, ..., a_1, b_1, ..., p_1, ...\}$ o conjunto de proposições atômicas e $Con = \{\wedge, \vee, \neg, \rightarrow\}$ o conjunto de conectivos.
- O alfabeto da linguagem da lógica proposicional é $A_{lp} = Prop \cup Con \cup \{), (\}$

- $((p \wedge (\neg r)) \rightarrow (p \vee s))$ é uma string no alfabeto A_{lp} e é uma fórmula da lógica proposicional
- $(\neg) \vee pq \rightarrow$ é uma string no alfabeto A_{lp} mas não é uma fórmula da lógica proposicional

Alfabeto da Lógica Proposicional

Definição

- Seja $Prop = \{a, b, c, ..., a_1, b_1, ..., p_1, ...\}$ o conjunto de proposições atômicas e $Con = \{\wedge, \vee, \neg, \rightarrow\}$ o conjunto de conectivos.
- O alfabeto da linguagem da lógica proposicional é $A_{lp} = Prop \cup Con \cup \{), (\}$

- $((p \wedge (\neg r)) \rightarrow (p \vee s))$ é uma string no alfabeto A_{lp} e é uma fórmula da lógica proposicional
- $(\neg) \vee pq \rightarrow$ é uma string no alfabeto A_{lp} mas não é uma fórmula da lógica proposicional
- Como podemos definir formalmente as fórmulas da lógica proposicional?

Fórmulas

Definição

As fórmulas da lógica proposicional são obtidas usando apenas as regras a seguir uma quantidade finita de vezes

- Se $p \in Prop$ então p é fórmula
- Se ϕ é uma fórmula então $(\neg\phi)$ é uma fórmula
- Se ϕ e ψ são fórmulas então $(\phi \wedge \psi)$ é uma fórmula
- Se ϕ e ψ são fórmulas então $(\phi \vee \psi)$ é uma fórmula
- Se ϕ e ψ são fórmulas então $(\phi \rightarrow \psi)$ é uma fórmula

Exemplo

- $(\neg(p_7 \rightarrow p_1))$ é uma fórmula de acordo com a definição anterior?

Exemplo

- $(\neg(p_7 \rightarrow p_1))$ é uma fórmula de acordo com a definição anterior?
- Podemos tentar gerar a string usando as regras da definição

Exemplo

- $(\neg(p_7 \to p_1))$ é uma fórmula de acordo com a definição anterior?
- Podemos tentar gerar a string usando as regras da definição
- E $p_1 \to p_7$?

Exemplo

- $(\neg(p_7 \rightarrow p_1))$ é uma fórmula de acordo com a definição anterior?
- Podemos tentar gerar a string usando as regras da definição
- E $p_1 \rightarrow p_7$?
- E $(\rightarrow p_2$?

Exemplo

- $(\neg(p_7 \rightarrow p_1))$ é uma fórmula de acordo com a definição anterior?
- Podemos tentar gerar a string usando as regras da definição
- E $p_1 \rightarrow p_7$?
- E $(\rightarrow p_2$?
- $(((\neg p) \wedge q) \rightarrow (p \wedge (q \vee (\neg r))))$ é uma fórmula de acordo com a definição anterior?

Tópicos

Estrutura Indutiva

- Podemos nos beneficiar da definição indutiva das fórmulas para provar propriedades e definir algoritmos
- Algoritmos recursivos e funções recursivas para processar as fórmulas
- Indução para provar propriedades de fórmulas

Exemplo

- Vamos definir uma função para retornar o tamanho de uma fórmula

Definição

$$tam(\phi) = \begin{cases} 1, & \text{se } \phi \text{ é atômica.} \\ 1 + tam(\psi), & \text{se } \phi = (\neg\psi) \\ 1 + tam(\psi_1) + tam(\psi_2), & \text{se } \phi = (\psi_1 \wedge \psi_2) \\ 1 + tam(\psi_1) + tam(\psi_2), & \text{se } \phi = (\psi_1 \vee \psi_2) \\ 1 + tam(\psi_1) + tam(\psi_2), & \text{se } \phi = (\psi_1 \to \psi_2) \end{cases}$$

Exemplo

$tam((p \to q)) = 3$

$tam(((p \wedge q) \to r)) = 5$

Exemplo

- Vamos construir um algoritmo para retornar o tamanho de uma fórmula

procedure $\text{TAM}(\phi)$
    if ϕ é atômica **then**
        return 1
    if $\phi = (\neg\psi)$ **then**
        return $1 + \text{TAM}(\psi)$
    if $\phi = (\psi_1 \Box \psi_2)$ **then**
        return $1 + \text{TAM}(\psi_1) + \text{TAM}(\psi_2)$

Exercício

- Vamos definir uma função para retornar a quantidade de conectivos de uma fórmula

Exercício

- Vamos definir uma função para retornar a quantidade de conectivos de uma fórmula

Definição

$$
conec(\phi) = \begin{cases} 0, & \text{se } \phi \text{ é atômica.} \\ 1 + conec(\psi), & \text{se } \phi = (\neg\psi) \\ 1 + conec(\psi_1) + conec(\psi_2), & \text{se } \phi = (\psi_1 \wedge \psi_2) \\ 1 + conec(\psi_1) + conec(\psi_2), & \text{se } \phi = (\psi_1 \vee \psi_2) \\ 1 + conec(\psi_1) + conec(\psi_2), & \text{se } \phi = (\psi_1 \rightarrow \psi_2) \end{cases}
$$

Subfórmulas

- Uma subfórmula de uma fórmula ϕ é uma substring de ϕ que é uma fórmula
- Quais as subfórmulas da fórmula $((\neg p) \wedge q)$?

Subfórmulas

- Uma subfórmula de uma fórmula ϕ é uma substring de ϕ que é uma fórmula
- Quais as subfórmulas da fórmula $((\neg p) \wedge q)$?
- $((\neg p) \wedge q)$

Subfórmulas

- Uma subfórmula de uma fórmula ϕ é uma substring de ϕ que é uma fórmula
- Quais as subfórmulas da fórmula $((\neg p) \wedge q)$?
- $((\neg p) \wedge q)$
- $(\neg p)$

Subfórmulas

- Uma subfórmula de uma fórmula ϕ é uma substring de ϕ que é uma fórmula
- Quais as subfórmulas da fórmula $((\neg p) \wedge q)$?
- $((\neg p) \wedge q)$
- $(\neg p)$
- p
- q

Subfórmulas

- Vamos definir formalmente a noção de subfórmula
- Para isso vamos definir uma função que retorna o conjunto das subfórmulas de uma fórmula

Definição

$$
\mathit{sub}(\phi) = \begin{cases} \{\phi\}, & \text{se } \phi \text{ é atômica.} \\ \{\phi\} \cup \mathit{sub}(\psi), & \text{se } \phi = (\neg\psi) \\ \{\phi\} \cup \mathit{sub}(\psi_1) \cup \mathit{sub}(\psi_2), & \text{se } \phi = (\psi_1 \wedge \psi_2) \\ \{\phi\} \cup \mathit{sub}(\psi_1) \cup \mathit{sub}(\psi_2), & \text{se } \phi = (\psi_1 \vee \psi_2) \\ \{\phi\} \cup \mathit{sub}(\psi_1) \cup \mathit{sub}(\psi_2), & \text{se } \phi = (\psi_1 \rightarrow \psi_2) \end{cases}
$$

Subfórmulas

- Vamos definir formalmente a noção de subfórmula
- Para isso vamos definir uma função que retorna o conjunto das subfórmulas de uma fórmula

Definição

$$sub(\phi) = \begin{cases} \{\phi\}, & \text{se } \phi \text{ é atômica.} \\ \{\phi\} \cup sub(\psi), & \text{se } \phi = (\neg\psi) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \land \psi_2) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \lor \psi_2) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \to \psi_2) \end{cases}$$

- Quais as subfórmulas da fórmula $((p \lor r) \to (r \land (\neg q)))$?

Subfórmulas

- Vamos definir formalmente a noção de subfórmula
- Para isso vamos definir uma função que retorna o conjunto das subfórmulas de uma fórmula

Definição

$$sub(\phi) = \begin{cases} \{\phi\}, & \text{se } \phi \text{ é atômica.} \\ \{\phi\} \cup sub(\psi), & \text{se } \phi = (\neg\psi) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \wedge \psi_2) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \vee \psi_2) \\ \{\phi\} \cup sub(\psi_1) \cup sub(\psi_2), & \text{se } \phi = (\psi_1 \rightarrow \psi_2) \end{cases}$$

- Quais as subfórmulas da fórmula $((p \vee r) \rightarrow (r \wedge (\neg q)))$?
- $\{((p \vee r) \rightarrow (r \wedge (\neg q))), (p \vee r), (r \wedge (\neg q)), (\neg q), p, r, q\}$
- Como a definição usou conjuntos, não precisamos nos preocupar com repetições de subfórmulas

Exercício

- Prove que para qualquer fórmula da Lógica Proposicional ϕ, $|sub(\phi)| \leq 2conec(\phi) + 1$.

Exercício

- Prove que para qualquer fórmula da Lógica Proposicional ϕ, $|sub(\phi)| \leq 2conec(\phi) + 1$.

Prova

Base: Seja p uma fórmula atômica qualquer. Temos que $|sub(p)| = 1$ e $conec(p) = 0$. Logo, $|sub(p)| \leq 2 \times conec(p) + 1$.
H.I: Seja ψ_1 e ψ_2 duas fórmulas tal que $|sub(\psi_1)| \leq 2 \times conec(\psi_1) + 1$ e $|sub(\psi_2)| \leq 2 \times conec(\psi_2) + 1$.
Passo: Seja $(\psi_1 \Box \psi_2)$ para $\Box \in \{\wedge, \vee, \rightarrow\}$. Temos que $sub((\psi_1 \Box \psi_2)) = sub(\psi_1) \cup sub(\psi_2) \cup \{(\psi_1 \Box \psi_2)\}$ e $conec((\psi_1 \Box \psi_2)) = conec(\psi_1) + conec(\psi_2) + 1$, para $\Box \in \{\wedge, \vee, \rightarrow\}$.
Portanto, $|sub((\psi_1 \Box \psi_2))| \leq |sub(\psi_1)| + |sub(\psi_2)| + 1$ (note que pode ter interseção). Pela **Hipótese de Indução**, $|sub((\psi_1 \Box \psi_2))| \leq 2 \times conec(\psi_1) + 1 + 2 \times conec(\psi_2) + 1 + 1$. Logo, $|sub((\psi_1 \Box \psi_2))| \leq 2 \times conec(\psi_1) + 2 \times conec(\psi_2) + 2 + 1 = 2 \times conec((\psi_1 \Box \psi_2)) + 1$.

Exercício

- Vamos definir uma função para retornar o conjunto de atômicas de uma fórmula

Exercício

- Vamos definir uma função para retornar o conjunto de atômicas de uma fórmula

Definição

$$atom(\phi) = \begin{cases} \{\phi\}, & \text{se } \phi \text{ é atômica.} \\ atom(\psi), & \text{se } \phi = (\neg\psi) \\ atom(\psi_1) \cup atom(\psi_2), & \text{se } \phi = (\psi_1 \wedge \psi_2) \\ atom(\psi_1) \cup atom(\psi_2), & \text{se } \phi = (\psi_1 \vee \psi_2) \\ atom(\psi_1) \cup atom(\psi_2), & \text{se } \phi = (\psi_1 \rightarrow \psi_2) \end{cases}$$

Tópicos

Precedência dos Conectivos

- Para simplificar a notação podemos economizar parênteses
- Vamos usar as seguintes regras de precedência:
- $\neg$ tem a maior precedência.
- $\vee$ e $\wedge$ tem mesma precedência e maior que a do $\rightarrow$.
- $\rightarrow$ tem associatividade à direita: $p \rightarrow q \rightarrow r$ é o mesmo que $(p \rightarrow (q \rightarrow r))$.

Exercício

- Coloque os parênteses nas proposições abaixo de acordo com a convenção de precedência:
- $\neg p \wedge q \rightarrow r$
- $(p \rightarrow q) \rightarrow (r \rightarrow s \vee t)$
- $p \vee q \rightarrow \neg p \wedge r$
- $p \vee q \wedge r$

Exercício

- Coloque os parênteses nas proposições abaixo de acordo com a convenção de precedência:
- $(((\neg p) \wedge q) \rightarrow r)$
- $((p \rightarrow q) \rightarrow (r \rightarrow (s \vee t)))$
- $((p \vee q) \rightarrow ((\neg p) \wedge r))$
- $p \vee q \wedge r$ está mal formulada