Sumário

I

Lógica Proposicional

1 Sintaxe

2 Semântica

3 Problema da Satisfatibilidade

II

Lógica de Primeira Ordem

Enquanto a Lógica Proposicional tem como base proposições, onde enunciados são inteiramente representados por variáveis, Gottlob Frege buscou, em 1879, obter uma linguagem simbólica mais rica, que representa enunciados na qual os objetos mencionados nesses enunciados tenham uma representação própria. Observe as seguintes sentenças declarativas abaixo.

1. Se o unicórnio é lenda, é imortal, mas se não é lenda, é mamífero.
2. O unicórnio, se é imortal ou mamífero, é chifrudo.
3. O unicórnio, se é chifrudo, é bruxaria.

Queremos saber: o unicórnio é lenda? É bruxaria? É chifrudo? Vamos representar as sentenças na lógica proposicional na seguinte forma:

- $l =$ "O unicórnio é lenda."
- $i =$ "O unicórnio é imortal."
- $m =$ "O unicórnio é mamífero."
- $c =$ "O unicórnio é chifrudo."
- $b =$ "O unicórnio é bruxaria."

1. $(l \to i) \wedge (\neg l \to m)$
2. $(i \vee m) \to c$
3. $c \to b$

Basta saber então, se $\{1, 2, 3\}$ acarreta em l, c ou b. Podemos usar algum dos métodos algorítmicos que resolvem SAT para resolver esse problema. Agora, vejamos as sentenças abaixo.

4. O jumento é primo do unicórnio.
5. Todo primo do unicórnio é chifrudo.
6. Algum primo do unicórnio não é bruxaria.
7. A fêmea do jumento é chifruda.

Na lógica proposicional, cada uma das sentenças tem que ser representada por uma variável. Isso implica em perda de expressividade, pois não podemos representar conceitos como "primo de", "todo", "algum", "fêmea de". Sendo assim, queremos usar símbolos que nos permitam representar os objetos e as relações entre eles. Temos:

Objetos j: jumento; u: unicórnio; $f(j)$: fêmea do jumento

Predicados e relações $L(x)$: x é lenda; $I(x)$: x é imortal; $M(x)$: x é mamífero; $C(x)$: x é chifrudo; $B(x)$: x é bruxaria; $P(x, y)$: x é primo de y

Adicionalmente, usamos os símbolos $\forall x$ para representar "para todo x" e $\exists x$ para "existe x". Assim, podemos representar as sentenças 1 a 7 como:

1. $(L(u) \to I(u)) \wedge (\neg L(u) \to M(u))$
2. $(I(u) \vee M(u)) \to C(u)$
3. $C(u) \to B(u)$
4. $P(j, u)$
5. $\forall x(P(x, u) \to C(x))$
6. $\exists x(P(x, u) \wedge \neg B(x))$
7. $C(f(j))$

A lógica que lida com esses símbolos é dita **Lógica de Primeira Ordem** ou **Lógica de Predicados**.

4 Estruturas

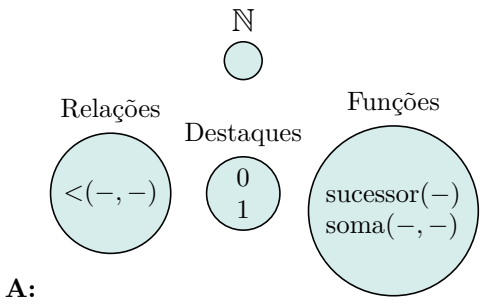
Como vimos, o vocabulário simbólico da lógica de predicados inclui símbolos para representar objetos e predicados, além de símbolos para os conectivos. Assim, a noção de valoração-verdade é incompatível com a lógica de primeira ordem, e necessitamos enriquecê-la para algo que nos permita atribuir valores aos objetos e predicados. Tomamos então, o conceito de **estrutura matemática**.

Estrutura Matemática

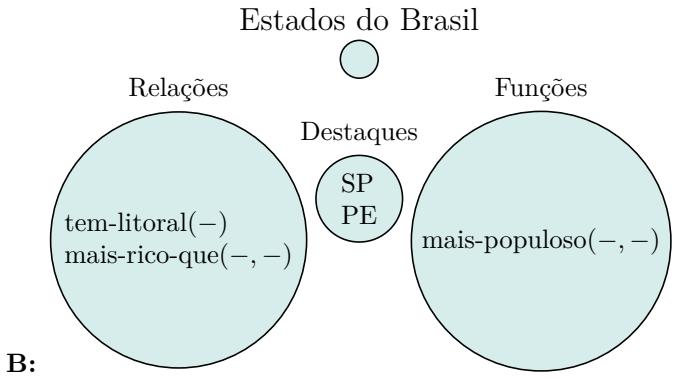
Uma estrutura A é definida por 4 componentes:

- Conjunto de objetos chamado de **domínio** ou **universo** de A – denotado por $dom(A)$.
- Subconjunto de elementos de $dom(A)$ considerados **destaques** ou **constantes**.
- Conjunto de **relações** sobre $dom(A)$, cada uma com sua aridade.
- Conjunto de **funções** sobre $dom(A)$, cada uma com sua aridade.

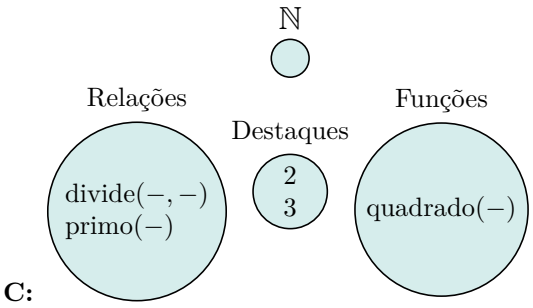
Para ilustrar o conceito, eis três exemplos:



A:



B:



Uma vez definida uma estrutura, podemos criar um vocabulário simbólico sobre a estrutura que nos permita codificar sentenças na lógica de predicados. Tal vocabulário deve dizer quão rica ou simples é a estrutura, envolvendo o número de relações, destaques e funções. Chamamos esse vocabulário de **assinatura** da estrutura.

Assinatura

Seja A uma estrutura. A assinatura L de A é definida pelos seguintes componentes:

- Quantidade de símbolos de destaques e os símbolos.
- Quantidade de símbolos de relações n-árias, onde $n \in \mathbb{N}$, e os símbolos.
- Quantidade de símbolos de funções n-árias, onde $n \in \mathbb{N}$, e os símbolos.

A é dita L-Estrutura.

A assinatura diz respeito somente à quantidade de símbolos. A definição dos mesmos é feita na **linguagem**. Mas, por simplicidade, unimos os dois conceitos. Assim, podemos definir a assinatura de A como:

- 2 símbolos de destaques: a e b;
- 1 símbolo de relação binária: R;
- 1 símbolo de função unária: f;
- 1 símbolo de função binária: g.

Uma vez definidos os símbolos, precisamos dizer o que eles representam em uma estrutura, para que possamos avaliar as sentenças que usam esses símbolos. Tal processo chama-se **interpretação**.

Interpretação

Seja L uma assinatura e A uma L-Estrutura. A interpretação de L em A é uma associação de cada símbolo de L a um elemento de cada componente de A, tal que:

- A cada símbolo c de constante, associa-se um elemento destacado do domínio de A (notação c^A).
- A cada símbolo R de relação de aridade n, associa-se uma relação de A de aridade n (notação R^A).
- A cada símbolo f de relação de aridade n, associa-se uma função de A de aridade n (notação f^A).

Assim, podemos tomar a seguinte interpretação da assinatura de A em A:

- $a^A = 0$ e $b^A = 1$;
- $R^A = <(-, -)$;

- $f^A = \text{sucessor}(-)$ e $g^A = \text{soma}(-,-)$.

Podemos então formalizar sentenças sobre a estrutura A na lógica de predicados:

2 é menor que 3. $R(f(b), f(f(b)))$

Para todo natural x, há um natural y maior que ele. $\forall x \exists y (R(x,y))$

Para todo natural x, a soma entre 1 e x é igual ao sucessor de x. $\forall x (g(b,x) = f(x))$

0 não é sucessor de nenhum natural. $\neg \exists x (f(x) = a)$

Para todo natural x, existem dois naturais cuja soma é x. $\forall x \exists y \exists z (g(y,z) = x)$

4.1 Subestruturas

Como saber se uma estrutura A é subestrutura de uma estrutura B? Se A e B forem simplesmente conjuntos, basta saber se todos os elementos de A também são elementos de B. Mas, considerando os outros componentes das estruturas A e B (relações, destaques e funções), é necessário verificar se esses componentes possuem uma relação entre si que justifique dizer que A está contida em B como estrutura.

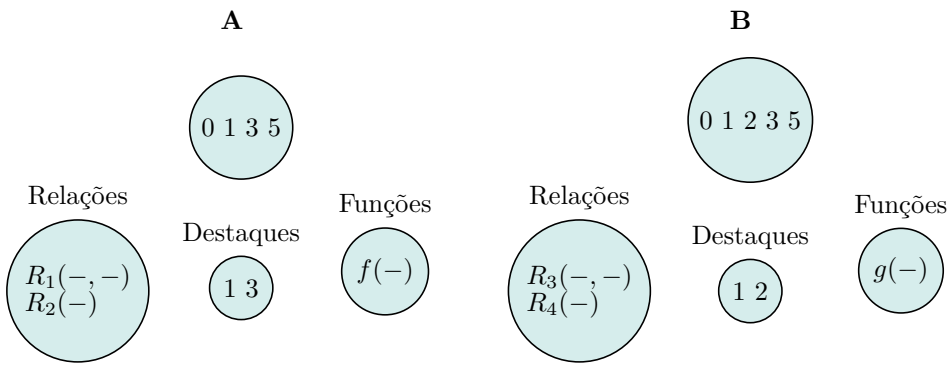
Para definir matematicamente esse possível relacionamento, tomamos emprestado da álgebra a noção de **homomorfismo**: uma função que preserva propriedades.

Homomorfismo

Sejam A e B estruturas de uma mesma assinatura L. Uma função $h : dom(A) \mapsto dom(B)$ é dita homomorfismo de A para B se as condições seguintes forem satisfeitas.

1. Para todo símbolo de constante c de L, $h(c^A) = c^B$;
2. Para todo símbolo de relação n-ária R de L e toda n-upla $(a_1, ..., a_n)$ de elementos de A, $(a_1, ...a_n) \in R^A \rightarrow (h(a_1), ..., h(a_n)) \in R^B$;
3. Para todo símbolo de função n-ária f de L e toda n-upla $(a_1, ..., a_n)$ de elementos de A, $h(f^A(a_1, ...a_n)) = f^B(h(a_1), ..., h(a_n))$.

Para ilustrar esse conceito, tomemos duas estruturas A e B:



Suponha que:

$R_1 = \{(0,3),(1,3),(3,5),(5,3)\}$ | $R_3 = \{(0,3),(1,2),(3,5),(2,3),(3,2),(3,3)\}$

$R_2 = \{0,1,5\}$ | $R_4 = \{0,1,2,3,5\}$

$f(0) = 1, f(1) = 1, f(3) = 2, f(5) = 3$ | $g(0) = 0, g(1) = 1, g(2) = 2, g(3) = 3, g(5) = 5$

Seja $h : dom(A) \mapsto dom(B)$ uma função entre as duas estruturas, definida da seguinte forma:

$$h(0) = 1$$
$$h(1) = 1$$
$$h(3) = 2$$
$$h(5) = 3$$

h é um homomorfismo de A para B? Vamos verificar cada condição:

1. A 1ª condição diz que os destaques de A são mapeados para destaques de B. Notamos que $h(1) = 1$ e $h(3) = 2$. Assim, a 1ª condição é satisfeita e dizemos que h **preserva destaques**.

2. A 2ª condição diz que se uma tupla de elementos se relaciona em A, então a tupla contendo os mapeamentos desses elementos se relaciona em B. Analisando as relações:

   R_1**:** $(0,3) \mapsto (h(0), h(3)) = (1,2) \in R_3$
   $(1,3) \mapsto (h(1), h(3)) = (1,2) \in R_3$
   $(3,5) \mapsto (h(3), h(5)) = (2,3) \in R_3$
   $(5,3) \mapsto (h(5), h(3)) = (3,2) \in R_3$

   R_2**:** $0 \mapsto h(0) = 1 \in R_4$
   $1 \mapsto h(1) = 1 \in R_4$
   $5 \mapsto h(5) = 3 \in R_4$

   Assim, a 2ª condição é satisfeita e dizemos que h **preserva relações**.

3. A 3ª condição diz que mapear a aplicação de uma função em A corresponde a mapear primeiro os argumentos e depois aplicar uma função em B. Analisando as funções:

$$h(f(0)) = g(h(0)) = g(1) = 1$$
$$h(f(1)) = g(h(1)) = g(1) = 1$$
$$h(f(3)) = g(h(3)) = g(2) = 2$$
$$h(f(5)) = g(h(5)) = g(3) = 3$$

   Assim, a 3ª condição é satisfeita e dizemos que h **preserva funções**. Por preservar destaques, relações e funções, h é um homomorfismo de A para B.

4.1.1 Imersão

Um homomorfismo $h : dom(A) \to dom(B)$ é dito **imersão** se:

- h é injetora;
- h satisfaz uma versão mais forte da 2ª condição:
  Para todo símbolo de relação n-ária R de L e toda n-upla $(a_1, ..., a_n)$ de elementos de A, $(a_1, ...a_n) \in R^A \leftrightarrow (h(a_1), ..., h(a_n)) \in R^B$.

A função h do exemplo anterior não é uma imersão, uma vez que, não só ela quebra a primeira condição (pois $h(0) = h(1) = 1$, implicando que h não é injetora) como a segunda ($(5,5) \notin R_1$, mas $(h(5), h(5)) = (3,3) \in R_2$). Além da imersão, existem outras variantes do homomorfismo:

- Uma imersão sobrejetora é dita **isomorfismo**.
- Um homomorfismo $h : dom(A) \mapsto dom(A)$ é dito **endomorfismo** de A.
- Um isomorfismo $h : dom(A) \mapsto dom(A)$ é dito **automorfismo** de A.

Agora, podemos remeter ao problema inicial e definir então as condições para que uma estrutura A seja subestrutura de uma estrutura B.

Subestrutura

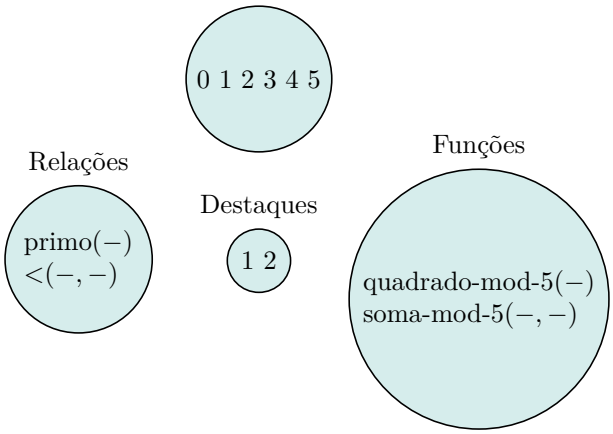
Sejam A e B estruturas de mesma assinatura. Dizemos que A é subestrutura de B se:

1. $dom(A) \subseteq dom(B)$
2. A função identidade $i : dom(A) \mapsto dom(B) \mid i(x) = x$ é uma imersão.

A notação é $A \subseteq B$.

4.1.2 O Problema da Menor Subestrutura

Seja A a estrutura a seguir e $X = \{0, 1, 3\}$ um subconjunto do domínio de A:



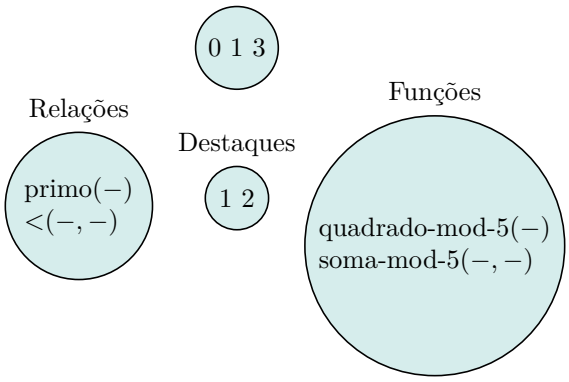
Queremos construir uma subestrutura de A que contenha o menor número de elementos em seu domínio e que contenha X. Estamos diante de um problema de otimização:

Dada: uma L-Estrutura A e um conjunto $X \subseteq dom(A)$;

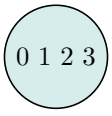
Pergunta-se: qual a menor subestrutura B de A que contém X, ou seja, $B \subseteq A$ e $X \subseteq dom(B)$?

A notação que usamos para B é $\langle X \rangle_A$. Assim, B deve conter os mesmos destaques, relações e funções que A e deve conter X em seu domínio. Além disso, precisamos adicionar elementos ao domínio de B para que a definição de estrutura se mantenha consistente.

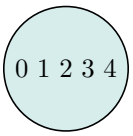
Inicialmente, temos a seguinte estrutura:



Note que ela possui o destaque 2, que não pertence ao domínio. Por definição, o conjunto de destaques é subconjunto do domínio, logo, devemos adicioná-lo a este:



Note também que a função quadrado-mod-5 aplicada a 3 retorna 4, que não é um elemento do domínio. Por definição, o domínio é fechado sob as funções, assim, devemos adicionar 4 ao domínio:



Dessa forma, $\langle X \rangle_A$ é a estrutura com domínio $\{0, 1, 2, 3, 4\}$ e com os mesmos destaques, relações e funções que A.

Podemos sintetizar o procedimento para construir $\langle X \rangle_A$ da seguinte forma:

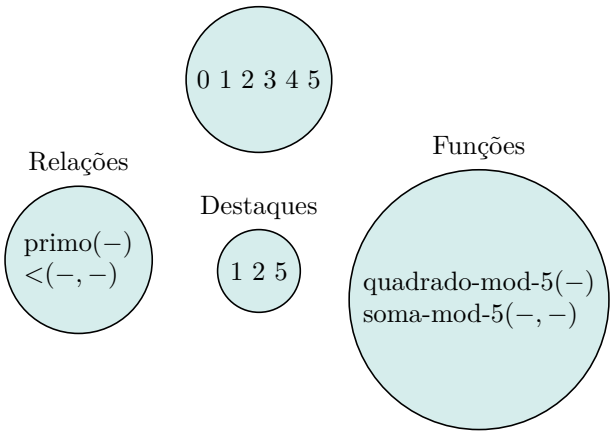
1. Inicialmente, adicione os destaques, funções e relações de A em B e faça $dom(B) = X$.
2. Adicione os destaques de B ao domínio de B.
3. Repita até que nenhum elemento novo seja adicionado:
   3.1 Adicione os conjuntos imagens das funções de B ao domínio de B.

4.1.3 Extensão de uma estrutura

Quando $\langle \emptyset \rangle_A = A$, ou seja, a menor subestrutura de A construída a partir do conjunto vazio como domínio é a própria A, todos os elementos de A são alcançáveis a partir dos destaques e funções de A.

Isso não é verdade na estrutura do exemplo anterior, uma vez que 5 é um elemento inalcançável a partir dos destaques e funções disponíveis e é considerado "sem nome" (é impossível representar 5 por meio de símbolos sobre essa estrutura). Dessa forma, podemos estender a estrutura A, adicionando ao seu conjunto de destaques os elementos inacessíveis de A. A estrutura resultante A' é chamada de **extensão** de A (e A é dita **reduto** de A').

A extensão da estrutura do exemplo anterior é, portanto:



5 Sintaxe

Vimos que, além dos conectivos e parênteses, a lógica de predicados envolve mais quatro tipos de símbolos:

Constantes e variáveis São funções de aridade zero e referenciam um objeto do domínio.

Predicados e relações Usamos esses símbolos para denotar alguma propriedade de objetos (predicados) ou uma relação entre objetos (relações). Gottlob Frege, o fundador da lógica de predicados, mostrou como esses conceitos podem ser representados por funções proposicionais (ou seja, funções que retornam verdadeiro ou falso) mesmo que, na prática, sejam conjuntos.

Funções Símbolos que representam funções de referência indireta, ou seja, servem para referenciar um objeto a partir de outros.

Quantificadores Símbolos que denotam quantidade: $\exists$ para denotar "algum objeto" e $\forall$ para denotar "todo objeto".

Assim, o alfabeto Σ da lógica de primeira ordem consiste na união desses quatro tipos de símbolos, acrescidos dos conectivos lógicos e parênteses.

5.1 Fórmulas

Assim como as expressões legítimas da lógica proposicional são chamadas proposições, as expressões legítimas da lógica de primeira ordem são chamadas **fórmulas**.

A unidade básica de uma fórmula é a **fórmula atômica**, que é uma relação entre objetos. Para representar objetos, tomamos o conceito de **termos**.

Termos

Seja L uma linguagem. O conjunto de termos de L é definido indutivamente da seguinte forma:

- Todo símbolo de constante c de L é um termo;
- Toda variável é um termo;
- Se f for um símbolo de função n-ária de L e se $t_1, ..., t_n$ forem termos de L, então $f(t_1, ..., t_n)$ é um termo.

Um termo que não contém variáveis é dito **termo fechado**.

Fórmulas Atômicas

Seja L uma linguagem. Uma fórmula atômica é uma palavra sobre o vocabulário simbólico de L com um dos dois formatos:

- $R(t_1, ...t_n)$, onde R é um símbolo de relação n-ária de L e $t_1, ..., t_n$ são termos de L;
- $t_1 = t_2$, onde t_1 e t_2 são termos de L.

Uma fórmula atômica que não contém variáveis é dita **sentença atômica**.

E assim, podemos definir indutivamente o conjunto das expressões legítimas da lógica de predicados, chamado de conjunto das **fórmulas bem formadas** ($FORM$).

Fórmula bem formada

- Toda fórmula atômica é uma fórmula bem formada;
- Se ω é uma fórmula bem formada, então $(\neg\omega)$ é uma fórmula bem formada;
- Se ω_1 e ω_2 são fórmulas bem formadas, então $(\omega_1 \wedge \omega_2)$ é uma fórmula bem formada.
- Se ω_1 e ω_2 são fórmulas bem formadas, então $(\omega_1 \vee \omega_2)$ é uma fórmula bem formada.
- Se ω_1 e ω_2 são fórmulas bem formadas, então $(\omega_1 \to \omega_2)$ é uma fórmula bem formada.
- Se ω é uma fórmula bem formada e x é uma variável livre em ω, então $(\forall x\omega)$ é uma fórmula bem formada;
- Se ω é uma fórmula bem formada e x é uma variável livre em ω, então $(\exists x\omega)$ é uma fórmula bem formada.

Uma fórmula que não contém variáveis livres é dita **sentença**.

Veremos a definição de variável livre a seguir. Temos então que $FORM$ é o fecho indutivo do conjunto base X de fórmulas atômicas sob o conjunto de funções geradoras $F = \{f_\forall, f_\exists, f_\neg, f_\vee, f_\wedge, f_\to\}$, e é possível mostrar que, da forma como definimos, $FORM$ é livremente gerado.

5.2 Variáveis

Uma variável, como vimos, é a representação de um objeto do domínio. Dada uma fórmula $(Qx\omega)$, onde Q é $\forall$ ou $\exists$, dizemos que o **escopo** do quantificador Qx é ω. Por exemplo, na fórmula a seguir:

$$S(x) \vee \exists z(P(z) \wedge \forall x(R(x,y) \to \exists y R(y,x)))$$

O escopo de $\exists z$ é $(P(z) \wedge \forall x(R(x,y) \to \exists y R(y,x))$, o de $\forall x$ é $(R(x,y) \to \exists y R(y,x))$ e o de $\exists y$ é apenas $R(y,x)$. Uma ocorrência de uma variável em uma fórmula é dita **ligada** se, e somente se, a variável está dentro do escopo de um quantificador aplicado a ela ou ela é a ocorrência do quantificador. Uma ocorrência de uma variável em uma fórmula é dita **livre** se, e somente se, essa ocorrência não é ligada.

$$S(x) \vee \exists z(P(z) \wedge \forall x(R(x,y) \to \exists y R(y,x)))$$

Assim, dizemos que uma variável é ligada em uma fórmula se há pelo menos uma ocorrência ligada dela na fórmula e, similarmente, uma variável é livre em uma fórmula se há pelo menos uma ocorrência livre dela na fórmula. Na fórmula anterior, x e y são variáveis livres, enquanto x, y e z são variáveis ligadas (é possível que uma variável seja livre e ligada ao mesmo tempo em uma fórmula, mas em ocorrências diferentes).

Podemos definir uma função recursiva que obtém o conjunto de variáveis livres em uma fórmula:

Conjunto das variáveis livres em uma fórmula

$VL : FORM \mapsto \mathcal{P}(\text{VARIÁVEIS})$
$VL(\varphi) = \{x_1, ..., x_n\}$, se φ é atômica e $x_1, ..., x_n$ ocorrem em φ
$VL((\neg\psi)) = VL(\psi)$
$VL((\rho \wedge \theta)) = VL(\rho) \cup VL(\theta)$
$VL((\rho \vee \theta)) = VL(\rho) \cup VL(\theta)$
$VL((\rho \to \theta)) = VL(\rho) \cup VL(\theta)$
$VL((\forall x\omega)) = VL(\omega) - \{x\}$
$VL((\exists x\omega)) = VL(\omega) - \{x\}$

5.2.1 Substituição de variáveis

Para atribuir um valor t à uma variável livre x em uma fórmula φ (notação $\varphi[t/x]$), devemos substituir todas as ocorrências livres dessa variável na fórmula por esse valor. Assim, queremos definir precisamente esse processo. Temos duas funções: uma aplicada a termos e uma aplicada a fórmulas.

Substituição de uma variável x por um termo t em um termo s

$s[t/x]$: TERM $\times$ TERM $\times$ VARIÁVEIS $\mapsto$ TERM
$x[t/x] = t$
$y[t/x] = y$, se $x \neq y$
$c[t/x] = c$
$f(t_1, ..., t_n)[t/x] = f(t_1[t/x], ..., t_n[t/x])$

Substituição de uma variável x por um termo t em uma fórmula φ

$\varphi[t/x]$: FORM $\times$ TERM $\times$ VARIÁVEIS $\mapsto$ FORM
$R(t_1, ..., t_n)[t/x] = R(t_1[t/x], ..., t_n[t/x])$
$(t_1 = t_2)[t/x] = (t_1[t/x] = t_2[t/x])$
$(\neg\psi)[t/x] = (\neg\psi[t/x])$
$(\rho \wedge \theta)[t/x] = (\rho[t/x] \wedge \theta[t/x])$
$(\rho \vee \theta)[t/x] = (\rho[t/x] \vee \theta[t/x])$
$(\rho \to \theta)[t/x] = (\rho[t/x] \to \theta[t/x])$
$(\forall x\omega)[t/x] = (\forall x\omega)$
$(\forall y\omega)[t/x] = (\forall y\omega[t/x])$, se $x \neq y$
$(\exists x\omega)[t/x] = (\exists x\omega)$
$(\exists y\omega)[t/x] = (\exists y\omega[t/x])$, se $x \neq y$

A função de substituição explora recursivamente a fórmula até encontrar uma ocorrência livre da variável a ser substituída, e então a substitui – isso implica que ocorrências ligadas da variável são ignoradas.

Porém, há uma condição especial para substituição que devemos considerar. Observe a seguinte fórmula φ:

$$\forall x(x = y)$$

Suponha que A seja uma estrutura com mais de um elemento em seu domínio. Isso significa que, para qualquer valor a que atribuirmos a y, a fórmula não diz a verdade nessa estrutura pois nem todo elemento de A é a^A. Isso implica (veremos com mais detalhes no próximo capítulo), que a fórmula φ não é **válida**. Porém, ao substituirmos y pela variável x, temos:

$$\forall x(x = x)$$

Nesse caso, a fórmula sempre diz a verdade, independemente da estrutura analisada, pois qualquer elemento em qualquer estrutura é igual a ele mesmo. Desse modo, a substituição causou uma fórmula não válida se tornar válida, e isso não deve acontecer. Assim, não devemos realizar substituições de um termo t em uma variável x que **causariam ocorrências livres de uma variável em t se tornarem ligadas após a substituição**. Temos então, a noção de **termo livre**.

Termo livre

Dizemos que um termo t está livre para entrar no lugar da variável x em uma fórmula φ se:

- φ é atômica;
- φ é da forma $(\neg\psi)$ e t está livre para entrar no lugar de x em ψ;
- φ é da forma $(\rho * \theta)$, t está livre para entrar no lugar de x em ρ e em θ e $* \in \{\wedge, \vee, \to\}$;
- φ é da forma $(\forall y\omega)$ ou $(\exists y\omega)$, $x = y$ ou $x \neq y$ e $x \notin VL(t)$, e t está livre para entrar no lugar de x em ω.

Assim, executamos essa função antes de proceder com a função de substituição. Na fórmula $\forall x(x = y)$, não podemos realizar a substituição $[x/y]$ pois $x \neq y$, mas $x \in VL(x)$.

6 Semântica

O matemático Alfred Tarski definiu pioneiramente uma maneira de definir quando uma sentença é verdadeira em uma estrutura, conhecida como a noção de verdade em um "modelo de Tarski" (ou por meio de uma metalinguagem).

Valor-verdade de uma sentença

Seja L uma assinatura, A uma L-Estrutura e $*^A$ uma interpretação dos símbolos de L em A. O valor-verdade de uma sentença φ de L é definida indutivamente da seguinte forma:

- $R(t_1, ..., t_n)^A$ é verdadeira se, e somente se, $(t_1^A, ..., t_n^A) \in R^A$
- $(t_1 = t_2)^A$ é verdadeira se, e somente se, t_1^A for o mesmo elemento que t_2^A
- $(\neg\psi)^A$ é verdadeira se, e somente se, ψ^A for falsa
- $(\rho \wedge \theta)^A$ é verdadeira se, e somente se, ρ^A for verdadeira e θ^A for verdadeira
- $(\rho \vee \theta)^A$ é verdadeira se, e somente se, ρ^A for verdadeira ou θ^A for verdadeira
- $(\rho \to \theta)^A$ é verdadeira se, e somente se, ρ^A for falsa ou θ^A for verdadeira
- $(\forall x\omega)^A$ é verdadeira se, e somente se, ω^A for verdadeira para todo valor de x
- $(\exists x\omega)^A$ é verdadeira se, e somente se, ω^A for verdadeira para algum valor de x

Se φ^A for verdadeira, dizemos que A satisfaz φ sob a interpretação $*^A$.

Satisfatibilidade de uma sentença

Seja φ uma sentença sobre uma assinatura L.

- φ é **satisfatível** se existe uma L-Estrutura A que satisfaz φ sob alguma interpretação de L em A.
- φ é **refutável** se existe uma L-Estrutura A que não satisfaz φ sob alguma interpretação de L em A.
- φ é **válida** se toda L-Estrutura A satisfaz φ sob toda interpretação de L em A.
- φ é **insatisfatível** se toda L-Estrutura A não satisfaz φ sob toda interpretação de L em A.

Seja Γ um conjunto de sentenças sobre L.

- Γ é **satisfatível** se existe uma L-Estrutura A que satisfaz todas as sentenças de Γ sob alguma interpretação de L em A.
- φ é **consequência lógica** de Γ se toda L-Estrutura A que satisfaz Γ também satisfaz φ sob toda interpretação de L em A.

Seja ψ uma sentença sobre uma assinatura L.

- φ e ψ são **logicamente equivalentes** se, para toda L-Estrutura A, A satisfaz ψ se, e somente se, A satisfaz ψ para toda interpretação de L em A.

E para o caso de fórmulas com ocorrências de variáveis livres? Nesse caso, devemos primeiro substituir essas variáveis por termos fechados de uma assinatura.

Satisfatibilidade de uma fórmula

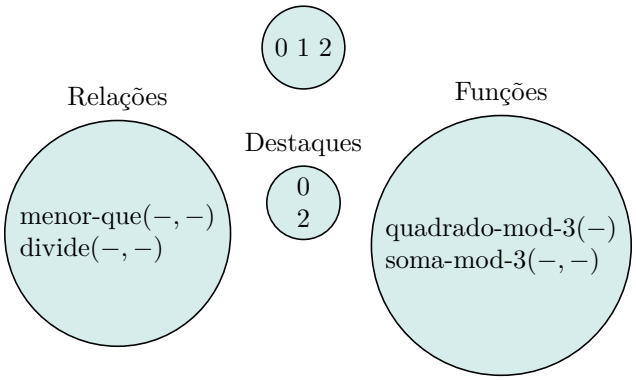
Seja φ uma fórmula sobre uma assinatura L, na qual ocorrem livremente as variáveis $x_1, ..., x_n$.

- φ é **satisfatível** se existe uma L-Estrutura A e termos $a_1, ..., a_n$ de L tal que A satisfaz $\varphi[a_1/x_1, ..., a_n/x_n]$ sob alguma interpretação de L em A.
- φ é **refutável** se existe uma L-Estrutura A e termos $a_1, ..., a_n$ de L tal que A não satisfaz $\varphi[a_1/x_1, ..., a_n/x_n]$ sob alguma interpretação de L em A.
- φ é **válida** se, para toda L-Estrutura A e toda n-upla de termos $a_1, ..., a_n$, A satisfaz $\varphi[a_1/x_1, ..., a_n/x_n]$ sob toda interpretação de L em A.
- φ é **insatisfatível** se, para toda L-Estrutura A e toda n-upla de termos $a_1, ..., a_n$, A não satisfaz $\varphi[a_1/x_1, ..., a_n/x_n]$ sob toda interpretação de L em A.

6.1 Modelos

Seja L uma assinatura, A uma L-Estrutura e φ uma sentença de L. Dizemos que A é **modelo** para φ se existe uma interpretação de L em A tal que φ^A seja verdadeira. Similarmente, dizemos que A é **contramodelo** para φ se existe uma interpretação de L em A tal que φ^A seja falsa.

Para ilustrá-los, tome a estrutura B a seguir:



A assinatura L de B pode ser definida como:

- 2 símbolos de destaques: a e b;
- 2 símbolo de relação binária: R e S;
- 1 símbolo de função unária: f;
- 1 símbolo de função binária: g.

Ao tomarmos a sentença φ abaixo:

$$\varphi = \exists x \forall y (R(x, y))$$

Notamos que a interpretação $R^B = \text{menor-que}(-, -)$ torna φ falsa em B, uma vez que não há um elemento de B que seja menor que 0, 1 e 2. Desse modo, deduzimos que B é contramodelo para φ. Porém, a interpretação $R^B = \text{divide}(-, -)$ torna φ verdadeira em B, uma vez que 1 divide 0, 1 e 2. Assim, B também é modelo para φ.

Usando as definições de modelo e contramodelo, podemos então definir os processos de criar um conjunto de sentenças atômicas a partir de uma estrutura de modo a descrevê-la; e de criar uma estrutura a partir de um conjunto de sentenças atômicas de modo a satisfazê-lo.

6.1.1 Diagrama Positivo

Dada uma estrutura A, queremos descrever minuciosamente as componentes dessa estrutura usando sentenças atômicas (uma "descrição lógica" de A). A ideia é escrever sentenças atômicas que descrevem: quais n-uplas do domínio de A pertencem a cada relação n-ária de A; e como funcionam cada função de A. Vamos começar com um exemplo simples, usando a estrutura B definida acima. Tomando a interpretação a seguir como base:

- $a^A = 0$ e $b^A = 2$;
- $R^A = \text{menor-que}(-,-)$ e $S^A = \text{divide}(-,-)$;
- $f^A = \text{quadrado-mod-3}(-)$ e $g^A = \text{soma-mod-3}(-,-)$.

Podemos usar essa linguagem simbólica para descrever B: $R(a,b)$, $R(a,f(b))$, $R(f(b),b)$, $R(g(a,a),b)$, $R(g(a,a),f(b))$, $S(f(b),a)$, $S(f(b),b)$, $S(b,a)$, $S(f(b),f(b))$, $S(b,b)$, $a = g(a,a)$, $b = g(f(b),f(b))$, $f(b) = g(b,b)$, $a = f(a)$, $b = g(b,a)$...

O que fizemos foi descrever uma base para que possamos obter o conjunto de todas as sentenças atômicas que são verdadeiras em B sob a interpretação dada. Chamamos esse conjunto de **diagrama positivo** de B. Para alcançar esse conjunto, precisamos completar essa base com o que falta para que ele inclua todas as sentenças atômicas verdadeiras em B. Assim, por exemplo, se $R(a,b)$ está nesse conjunto e $a = f(a)$ está nesse conjunto, então devemos incluir $R(f(a),b)$. Esse processo de completação é dito **fecho sob igualdade**.

Fecho sob igualdade

Seja T um conjunto de sentenças atômicas sobre uma linguagem L. O fecho sob igualdade de T é o menor conjunto T' de sentenças atômicas sobre L tal que:

- T' contém T;
- Se T' contém $\varphi(t)$ e contém $(t = t')$, então T' contém $\varphi(t')$.
- T' contém $(t = t)$ para todo termo fechado t de L.

Diagrama positivo

Seja L uma assinatura e A uma L-Estrutura. O conjunto de todas as sentenças atômicas sobre L que são verdadeiras em A sob alguma interpretação de L em A, ou seja, as sentenças atômicas para as quais A é modelo, é dito diagrama positivo de A (notação $diag^+(A)$).
Caso $\langle\emptyset\rangle_A \neq A$, usamos a extensão de A para gerar o diagrama positivo.

Construção do diagrama positivo

O processo de construção do diagrama positivo de uma estrutura A se dá da seguinte forma:

1. Inclua todos os elementos inalcançáveis do domínio de A como destaques em A.
2. Para todo símbolo de relação n-ária de L e termos $t_1, ..., t_n$, inclua $R(t_1, ..., t_n)$ se $t_1^A, ..., t_n^A \in R^A$.
3. Para todo símbolo de função n-ária de L e termos $t_1, ..., t_n, t_{n+1}$, inclua $f(t_1, ..., t_n) = t_{n+1}$ se $f^A(t_1^A, ..., t_n^A)$ for o mesmo elemento que t_{n+1}^A.

6.1.2 Modelo Canônico

Suponha que tenhamos o seguinte conjunto T de sentenças atômicas:

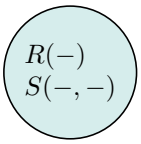
$$T = \{R(a), S(g(b),b), g(b) = g(a), g(b) = a, f(g(a),a) = b, S(a,a), R(g(b))\}$$

De modo inverso ao diagrama positivo, queremos construir uma estrutura D a partir de T, tal que D seja modelo para todas as sentenças de T. Para tal, precisamos definir as quatro componentes de D: conjunto domínio, conjunto de destaques, conjunto de relações e conjunto de funções.

Domínio O domínio, a princípio, seria o conjunto de todos os termos fechados que aparecem em T. Porém, uma vez que D deve satisfazer $g(b) = a$ e $g(b) = g(a)$, nessa estrutura, a, $g(b)$ e $g(a)$ devem ser o mesmo elemento. Desse modo, não faz sentido colocarmos os três no domínio, pois conjuntos não admitem repetição de elementos. Iremos então agrupar esses termos iguais em uma classe e colocar no domínio apenas algum termo que os represente, por exemplo, $a^{\sim}$. Assim, repetindo essa análise nas outras igualdades, o conjunto universo de D é

Destaques O conjunto de destaques é subconjunto do domínio, e deve conter todos os destaques que ocorrem em T:

Relações O conjunto de relações contém todas as relações que ocorrem em T. Porém, uma vez que D deve satisfazer as sentenças de T, é necessário incluir tais elementos nas relações.



Onde $R^D = \{a^{\sim}\}$ e $S^D = \{(a^{\sim}, b^{\sim}), (a^{\sim}, a^{\sim})\}$.

Funções O conjunto de funções contém todas as funções que ocorrem em T. Porém, as funções de D possuem como domínio o universo de D e devemos, portanto, definí-las.

Onde $f^D(a^{\sim}, a^{\sim}) = b^{\sim}$, $g^D(a^{\sim}) = a^{\sim}$ e $g^D(b^{\sim}) = a^{\sim}$.

D é uma estrutura que é modelo para todas as sentenças de T e é chamada **modelo canônico** de T. Antes de definir formalmente esse tipo de estrutura, precisamos definir o processo de selecionar um representante de uma classe que definimos na construção do domínio. Usaremos a noção de **classes de equivalência**.

A relação $\sim$

Seja L uma assinatura e T um conjunto de sentenças atômicas que é o diagrama positivo de alguma L-Estrutura. Podemos definir a relação binária $\sim$ da seguinte forma:

$$\sim = \{(t_1, t_2) \mid (t_1 = t_2) \in T\}$$

$\sim$ **é reflexiva:** Para todo $t \in L$, $(t = t) \in T$, logo $(t, t) \in \sim$.

$\sim$ **é simétrica:** Se $(t_1, t_2) \in \sim$, significa que $(t_1 = t_2) \in T$. Como a igualdade é comutativa, $(t_2 = t_1) \in T$ e, portanto, $(t_2, t_1) \in \sim$.

$\sim$ **é transitiva:** Se $(t_1, t_2) \in \sim$ e $(t_2, t_3) \in \sim$, significa que $(t_1 = t_2) \in T$ e $(t_2 = t_3) \in T$. Como a igualdade é transitiva, $(t_1 = t_3) \in T$. Desse modo, $(t_1, t_3) \in \sim$.

Sendo reflexiva, simétrica e transitiva, $\sim$ é uma **relação de equivalência** e, portanto, particiona o conjunto domínio de modo a criar classes de equivalência.

Modelo canônico

Seja T um conjunto de sentenças atômicas de uma linguagem L. A L-Estrutura B mais genérica possível que é modelo para todas as sentenças de T é dita modelo canônico de T e é definida da seguinte forma:

Domínio O domínio é o conjunto dos representantes das classes de equivalência $t^{\sim}$ dos termos fechados de L no conjunto T.

Destaques O conjunto de destaques contém todos os representantes das classes de equivalência $c^{\sim}$, onde c é um símbolo de constante de L.

Relações Seja R um símbolo de relação n-ária de L e $t_1, ..., t_n$ termos de L. Então $(t_1^{\sim}, ..., t_n^{\sim}) \in R^B$ se, e somente se, $R(t_1, ..., t_n) \in T$.

Funções Seja f um símbolo de relação n-ária de L e $t_1, ..., t_n$ termos de L. Então $f^B(t_1^{\sim}, ..., t_n^{\sim}) = f(t_1, ..., t_n)^{\sim}$.

O modelo canônico D de um conjunto de sentenças T é parecida com qualquer que tenha sido a estrutura B que foi o seu modelo "original" (ou seja, T é o diagrama positivo de B). D é chamada de **canônico** pois serve de referencial para todos os modelos de T. Isso significa que podemos construir um homomorfismo de D para qualquer outro modelo B de T:

$$h : dom(D) \mapsto dom(B)$$
$$h(t^{\sim}) = t^B$$

1. Para todo símbolo de constante c de L, $h(c^D) = c^B$. Como $c^D = c^{\sim}$, h preserva destaques.
2. Para todo símbolo de relação n-ária R de L e toda n-upla $t_1, ..., t_n$ de L, se $(t_1^{\sim}, ...t_n^{\sim}) \in R^D$, então $R(t_1, ..., t_n) \in T$. Dessa forma, $(t_1^B, ..., t_n^B) \in R^B$ e, portanto, $(h(t_1^{\sim}), ..., h(t_n^{\sim}))$. Assim, h preserva relações.
3. Para todo símbolo de função n-ária f de L e toda n-upla $t_1, ..., t_n$ de L, por definição, $f^D(t_1^{\sim}, ..., t_n^{\sim}) = f(t_1, ..., t_n)^{\sim}$. Dessa forma, $h(f^D(t_1^{\sim}, ..., t_n^{\sim})) = h(f(t_1, ..., t_n)^{\sim}) = f(t_1, ..., t_n)^D = f^D(t_1^D, ..., t_n^D) = f^D(h(t_1^{\sim}), ..., h(t_n^{\sim}))$. Assim, h preserva funções.

Por preservar destaques, relações e funções, h é um homomorfismo.

Finalmente, uma vez que podemos construir um modelo para qualquer conjunto de sentenças atômicas, **todo conjunto de sentenças atômicas é satisfatível**.

7 Problema da Satisfatibilidade

Dada: uma fórmula φ;

Pergunta-se: φ é satisfatível?

Retomamos o Problema da Satisfatibilidade (SAT) visto no capítulo 3, mas dessa vez adaptado para a lógica de primeira ordem. Diferentemente da lógica proposicional, porém, uma fórmula será satisfatível se houver uma interpretação em alguma estrutura que satisfaça a fórmula, ao invés de uma valoração-verdade sobre as variáveis de uma proposição que a satisfaça. Isso significa que o procedimento de força-bruta de avaliar se uma fórmula é válida (tautologia) sob todas as possíveis interpretações (similar ao método da tabela-verdade em lógica proposicional) em todas as estruturas não funciona, pois temos um número infinito de estruturas passíveis de avaliação. Mesmo assim, existem procedimentos que permitem verificar se uma fórmula é válida se ela de fato for válida, mas, caso não seja, esses procedimentos nunca terminam. Church e Turing provaram em 1936 que um procedimento que sempre termina não existe, classificando a lógica de primeira ordem como **semi-decidível**. De modo geral, se temos três fórmulas φ, ψ e ω, onde φ é válida e $\neg\psi$ é válida mas nem ω nem $\neg\omega$ são válidas, qualquer procedimento que resolve SAT terminará e dará uma resposta quando recebe φ ou ψ como entrada, mas nunca terminará se recebe ω. Veremos um desses procedimentos a seguir, que é a versão para a lógica de primeira ordem do método que estudamos anteriormente.

7.1 Método da Resolução

A filosofia do método persiste: ser eficiente para certas entradas e eficiente em reconhecer essas entradas. Os conceitos como a regra da resolução e a forma normal conjuntiva continuam valendo, mas dessa vez, temos diferenças cruciais.

7.1.1 O Problema da Unificação de Termos

A regra da resolução determina que, se possuírmos literais complementares L em uma cláusula C_1 e $\neg L$ em uma cláusula C_2, podemos usar a regra para criar uma cláusula nova – o resolvente de C_1 e C_2, com os outros literais das cláusulas. Porém, para isso acontecer, L em C_1 deve ser obviamente idêntico a L em C_2. Isso significa que, se tivermos dois literais $R(a)$ e $\neg R(x)$, para podermos aplicar a regra, precisamos torná-los idênticos e isso é possível ao fazermos $[a/x]$. Porém, não podemos aplicar a regra em $P(a)$ e $\neg P(b)$, pois não há substituição de variáveis em seus termos que torne $P(a) = P(b)$. De modo geral, precisamos de um método que determine se tal substituição existe. Estamos diante de um problema de decisão.

Dados: dois termos t_1 e t_2;

Pergunta-se: existe uma substituição de variáveis que torne t_1 e t_2 idênticos?

Caso tal substituição exista, dizemos que t_1 e t_2 são termos **unificáveis** e a substituição é dita **unificadora**. Em 1930, o matemático Jacques Herbrand definiu um algoritmo simples que resolve o problema usando regras de transformação de um conjunto de equações. Uma **equação** é um par de termos $(s = t)$. Um **sistema de equações** S é um multiconjunto de equações e uma substituição Θ é unificadora de S se ela unifica todas as equações de S. O conjunto de todas as substituições unificadoras de S é denotado por $U(S)$. Esse conjunto pode conter várias substituições possíveis, mas uma delas é a "ótima". Por exemplo, para os termos abaixo:

$$f(y)) \quad \text{e} \quad f(g(z))$$

A substituição $\Theta_1 = [g(b)/y, b/z]$ unifica os termos, mas não é a única possível. $\Theta_2 = [g(z)/y]$ também os unifica e é mais simples. A substituição mais simples que unifica dois termos dentre as possíveis é dita **unificadora mais geral**.

O algoritmo de Herbrand usa como base o conceito de **forma resolvida**: uma equação está na forma resolvida em um sistema S se for da forma $v = t$ (variável = termo) e v for uma variável resolvida, ou seja, v não ocorre em t e em nenhuma outra equação de S. Caso todas as equações de S estejam na forma resolvida, o sistema é unificável e o algoritmo devolve a substituição unificadora mais geral de S.

Regras de Transformação

Eliminação de Equações Triviais $S \cup \{t \stackrel{?}{=} t\} \Rightarrow S$

Decomposição de Termos $S \cup \{f(t_1, ..., t_n) \stackrel{?}{=} f(s_1, ..., s_n)\} \Rightarrow S \cup \{t_1 \stackrel{?}{=} s_1, ..., t_n \stackrel{?}{=} s_n\}$

- $S \cup \{f(t_1, ..., t_n) \stackrel{?}{=} g(t_1, ..., t_m)\}$, onde $f \neq g$ ou $n \neq m \Rightarrow$ *falha*.

Orientação $S \cup \{t \stackrel{?}{=} x\} \Rightarrow S \cup \{x \stackrel{?}{=} t\}$

Eliminação de Variáveis $S \cup \{x \stackrel{?}{=} t\} \Rightarrow S[x/t] \cup \{x \stackrel{?}{=} t\}$, se x não ocorre em t.

- Se x ocorre em $t \Rightarrow$ *falha*.

Algoritmo de Herbrand

Entrada: um sistema de equações S;

Saída: Se S for unificável, a unificadora mais geral de S; caso contrário, falha.

1. Para cada equação s de S:
   - 1.1 Se s estiver na forma resolvida, passe para a próxima.
   - 1.2 Caso contrário, aplique uma das regras de transformação, com prioridade decrescente de cima pra baixo, e vá para o início.
   - 1.3 Se não for possível aplicar uma regra, retorne **não é unificável**.
2. Caso todas as equações de s estejam na forma resolvida, retorne a **unificadora mais geral** formada pelas equações presentes em S.

Para ilustrar, vamos executar o algoritmo sobre o seguinte conjunto de equações S:

$$S = \{f(g(z), x) = f(y, x), f(y, x) = f(y, h(a)), f(g(z), x) = f(y, h(a))\}$$

$$\begin{aligned}
\textbf{DT} \quad & \{g(z) \stackrel{?}{=} y, x \stackrel{?}{=} x, f(y, x) \stackrel{?}{=} f(y, h(a)), f(g(z), x) \stackrel{?}{=} f(y, h(a))\} \\
\textbf{O} \quad & \{y \stackrel{?}{=} g(z), x \stackrel{?}{=} x, f(y, x) \stackrel{?}{=} f(y, h(a)), f(g(z), x) \stackrel{?}{=} f(y, h(a))\} \\
\textbf{EV} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, x \stackrel{?}{=} x, f(g(z), x) \stackrel{?}{=} f(g(z), h(a)), f(g(z), x) \stackrel{?}{=} f(g(z), h(a))\} \\
\textbf{EET} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, f(g(z), x) \stackrel{?}{=} f(g(z), h(a)), f(g(z), x) \stackrel{?}{=} f(g(z), h(a))\} \\
\textbf{DT} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, g(z) \stackrel{?}{=} g(z), x \stackrel{?}{=} h(a), f(g(z), x) \stackrel{?}{=} f(g(z), h(a))\} \\
\textbf{EET} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, x \stackrel{?}{=} h(a), f(g(z), x) \stackrel{?}{=} f(g(z), h(a))\} \\
\textbf{EV} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, \mathbf{x} \stackrel{?}{=} \mathbf{h(a)}, f(g(z), h(a)) \stackrel{?}{=} f(g(z), h(a))\} \\
\textbf{DT} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, \mathbf{x} \stackrel{?}{=} \mathbf{h(a)}, g(z) \stackrel{?}{=} g(z), h(a) \stackrel{?}{=} h(a)\} \\
\textbf{EET} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, \mathbf{x} \stackrel{?}{=} \mathbf{h(a)}, h(a) \stackrel{?}{=} h(a)\} \\
\textbf{EET} \quad & \{\mathbf{y} \stackrel{?}{=} \mathbf{g(z)}, \mathbf{x} \stackrel{?}{=} \mathbf{h(a)}\}
\end{aligned}$$

S é unificável e a unificadora mais geral de S é $\Theta = [g(z)/y, h(a)/x]$. Isso significa que, para os termos serem idênticos, basta substituir x por $h(a)$ e y por $g(z)$. Vejamos agora um exemplo onde o algoritmo falha:

$$S' = \{g(f(x, x)) = g(f(h(a), g(b)))\}$$

$$\begin{array}{ll} \textbf{DT} & \{f(x,x) \stackrel{?}{=} f(h(a),g(b))\} \\ \textbf{DT} & \{x \stackrel{?}{=} h(a), x \stackrel{?}{=} g(b)\} \\ \textbf{EV} & \{\mathbf{x} \stackrel{?}{=} \mathbf{h(a)}, h(a) \stackrel{?}{=} g(b)\} \\ & \{\mathbf{x} \stackrel{?}{=} \mathbf{h(a)}, h(a) \stackrel{?}{=} g(b)\} \end{array}$$

Uma vez que $h \neq g$, o algoritmo falha ao tentar aplicar uma regra em $h(a) \stackrel{?}{=} g(b)$. Desse modo, o sistema não é unificável. De fato: não existe nenhuma substituição de variáveis que torne $x = h(a)$ e $x = g(b)$ ao mesmo tempo.

No contexto de resolução, portanto, usaremos o algoritmo de Herbrand sempre que tentarmos aplicar a regra da resolução em dois literais complementares. Caso sejam unificáveis, procedemos com o resolvente. Caso contrário, não aplicaremos a regra.

7.1.2 Remoção dos quantificadores

Já concluímos um dos problemas referentes ao método da resolução, que se preocupava com a identicidade de literais. Agora, veremos o outro grande problema, sobre a sintaxe da entrada. As entradas do método da resolução devem estar na forma normal conjuntiva, que, como sabemos, é uma conjunção de cláusulas. Porém, as fórmulas da lógica de primeira ordem podem possuir quantificadores, que não estão presentes nessa definição. Dessa maneira, precisamos removê-los. A estratégia que usaremos será a seguinte: colocar todos os quantificadores no início da fórmula, e então removê-los.

Forma Normal Prenex

Uma fórmula φ está na **forma normal prenex** (*prefixed normal expression*) se, e somente se, ela tem o seguinte formato:

$$(Q_1 x_1 Q_2 x_2 ... Q_n x_n)(\psi)$$

Onde $(Q_1 x_1 Q_2 x_2 ... Q_n x_n)$ é uma parte com quantificadores e chamada de **prefixo** de φ e ψ é uma fórmula sem quantificadores chamada de **matriz** de φ.

Teorema 7.1.1

Para toda fórmula φ na forma normal prenex, existe uma fórmula ψ tal que:

- ψ está na forma normal prenex;
- ψ é logicamente equivalente a ϕ.

Prova (transformação para a FNP)

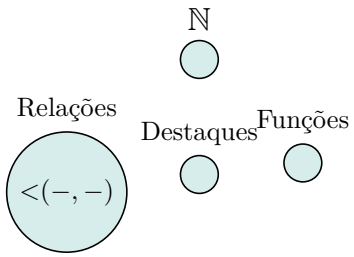
Para provar o teorema, faremos uma prova por construção mostrando um método que transforma qualquer fórmula φ para outra ψ logicamente equivalente e na forma normal prenex.

1. Elimine os conectivos $\rightarrow$ e $\leftrightarrow$.
2. Aplique repetidamente, além das equivalências conhecidas $(\neg\neg L) \equiv L$ (lei da dupla negação), as leis de De Morgan e a propriedade distributiva, as equivalências a seguir, até que a fórmula esteja na forma normal prenex.
   - $\neg\forall x(\omega) \equiv \exists x(\neg\omega)$
   - $\neg\exists x(\omega) \equiv \forall x(\neg\omega)$
   - $\forall x(\varphi(x)) \wedge \forall x(\psi(x)) \equiv \forall x(\varphi(x) \wedge \psi(x))$
   - $\exists x(\varphi(x)) \vee \exists x(\psi(x)) \equiv \exists x(\varphi(x) \vee \psi(x))$
   - $Qx(\varphi(x)) \equiv Qy(\varphi(y))$, onde $Q \in \{\forall, \exists\}$
   - $Qx(\varphi(x)) \wedge \psi \equiv Qx(\varphi(x) \wedge \psi)$, onde $Q \in \{\forall, \exists\}$
   - $Qx(\varphi(x)) \vee \psi \equiv Qx(\varphi(x) \vee \psi)$, onde $Q \in \{\forall, \exists\}$

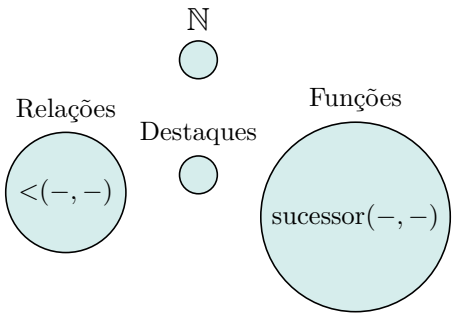
Forma Padrão de Skolem

Uma fórmula φ na forma normal prenex está na **forma padrão de Skolem** se, e somente se, o prefixo de φ não contém quantificadores existenciais. Em 1920, o lógico-matemático Thoralf Skolem definiu um método para remoção de quantificadores existenciais, que acabou ganhando seu nome: o **método da skolemização**.

Suponha a seguinte estrutura A:



A sentença $\varphi = \forall x \exists y R(x, y)$ é verdadeira em A, pois para todo natural x existe um outro maior que ele, por exemplo, seu sucessor. Dessa forma, podemos retirar o existencial, substituindo a ocorrência de sua variável pela função sucessor aplicada a x, ou seja, $\varphi' = \forall x R(x, f(x))$.



φ' é verdadeira nessa nova estrutura que é idêntica a A, adicionando-se a função f. A skolemização é, portanto, o processo de remover quantificadores existenciais e substituir as ocorrências das variáveis desses quantificadores por funções aplicadas a variáveis universalmente quantificadas anteriores a esses existenciais.

Teorema 7.1.2: Teorema de Löwenheim-Skolem

Seja φ uma fórmula na forma normal prenex sobre uma assinatura L. Seja ψ a fórmula resultante da remoção de cada quantificador existencial que ocorre em φ e cujas variáveis correspondentes são substituídas por um termo do tipo $f(x_1, ..., x_n)$, onde f é um novo símbolo de função e $x_1, ..., x_n$ são variáveis universalmente quantificadas imediatamente anteriores a esse existencial. Então, se existe uma L-Estrutura A que é modelo para φ, é possível construir uma L'-Estrutura B que é modelo para ψ, acrescentando a A uma interpretação para cada símbolo novo de função em L'.

Caso não haja quantificadores universais anteriores a um existencial, a variável correspondente deste é substituída por um novo símbolo de constante (função de aridade zero). Uma função nova adicionada na assinatura é chamada **função de Skolem**, e uma constante nova é chamada **constante de Skolem**. A skolemização não preserva a equivalência das fórmulas resultante e original, mas preserva a satisfatibilidade: a original é satisfatível se, e somente se, a resultante é satisfatível.

Para ilustrar o processo de remoção de quantificadores, vejamos as seguintes fórmulas:

- $\forall x P(x) \to \exists x S(x)$:
  $\equiv \neg \forall x P(x) \vee \exists x S(x)$
  $\equiv \exists x(\neg P(x)) \vee \exists x S(x)$
  $\equiv \underline{\exists x}(\neg P(x) \vee S(x))$ **FNP**

$\exists x \Rightarrow$ adicionando uma constante de Skolem a na assinatura;
$\equiv \neg P(a) \vee S(a)$ **FPS**

- $\exists x(R(x,y) \rightarrow \forall y(P(z,y) \wedge \exists w(S(w,u) \vee \neg R(w,y))))$:
$\equiv \exists x(\neg R(x,y) \vee \forall y(P(z,y) \wedge \exists w(S(w,u) \vee \neg R(w,y))))$
$\equiv \exists x(\neg R(x,y) \vee \forall m(P(z,m) \wedge \exists w(S(w,u) \vee \neg R(w,m))))$
$\equiv \exists x \forall m(\neg R(x,y) \vee (P(z,m) \wedge \exists w(S(w,u) \vee \neg R(w,m))))$
$\equiv \exists x \forall m(\neg R(x,y) \vee \exists w(P(z,m) \wedge (S(w,u) \vee \neg R(w,m))))$
$\equiv \underline{\exists x} \forall m \underline{\exists w}(\neg R(x,y) \vee (P(z,m) \wedge (S(w,u) \vee \neg R(w,m))))$ **FNP**
$\exists x \Rightarrow$ adicionando uma constante de Skolem a na assinatura;
$\exists w \Rightarrow$ adicionando uma função unária de Skolem $f(-)$ na assinatura;
$\equiv \forall m(\neg R(a,y) \vee (P(z,m) \wedge (S(f(m),u) \vee \neg R(f(m),m))))$ **FPS**