

2º Trabalho

Seja $\mathcal{C} = \{c_1, \dots, c_k\}$ um conjunto finito de constantes, $\mathcal{F} = \{f_1, f_2, \dots, f_s\}$ um conjunto finito de funções unárias ou binárias, e $VAR = \{x_1, x_2, x_3, \dots, x_u\}$ um conjunto finito de variáveis. Por exemplo,

$$\mathcal{C} = \{c\}$$

$$\mathcal{F} = \{f, g\} \text{ e}$$

$$VAR = \{x, y, z\}$$

Você deve pensar em uma forma de representar estruturas sem predicados. Uma sugestão é usar arrays associativos, ou seja, as chaves e os valores representam as informações importantes de uma estrutura. Por exemplo,

$$\{\text{dom}: [1, 2], \text{ f}: \{(1): 2, (2): 2\}, \text{ g}: \{(1, 1): 1, (1, 2): 2, (2, 1): 2, (2, 2): 1\}, \text{ c}: 1\}$$

representa a estrutura $\mathcal{A} = (A, f^{\mathcal{A}}, g^{\mathcal{A}}, c^{\mathcal{A}})$ em que

$$A = \{1, 2\},$$

$$f(1) = 2, f(2) = 2,$$

$$g(1, 1) = 1, g(1, 2) = 2, g(2, 1) = 2, g(2, 2) = 1 \text{ e}$$

$$c^{\mathcal{A}} = 1.$$

Perceba que as funções da estrutura também foram definidas com arrays associativos. Neste caso, as chaves são tuplas representando os argumentos da função. Você também deve pensar em uma forma de representar contextos $l : VAR \rightarrow A$ e interpretações $\mathcal{I} = (\mathcal{A}, l)$.

Seu programa também deve ser capaz de definir termos. Uma representação simples é usar listas. Por exemplo, $[g, [x, c]]$ representa o termo $t_1 = g(x, c)$.

Usando suas estruturas de dados para definir estruturas, contextos, interpretações e termos, defina uma função `interpretaTermo(interpretacao, termo)` que recebe como parâmetros uma interpretação e um termo, respectivamente. A função deve retornar o valor de `termo` em `interpretacao`. Por exemplo, na interpretação $\mathcal{I} = (\mathcal{A}, l)$ em que \mathcal{A} é definido acima e $l(x) = 2$, o valor do termo $g(x, c)$ é 2.