
Nome:

2022-06-27

Regras:

- I. Não vires esta página antes do começo da prova.
- II. Nenhuma consulta de qualquer forma.
- III. Nenhum aparelho ligado (por exemplo: celular, tablet, notebook, *etc.*).¹
- IV. Nenhuma comunicação de qualquer forma e para qualquer motivo.
- V. $(\forall x) [\text{Colar}(x) \implies \neg \text{Passar}(x, \text{FMC1})]$.²
- VI. Responda dentro das caixas indicadas.
- VII. Nenhuma prova será aceita depois do fim do tempo—mesmo se for atraso de 1 segundo.
- VIII. Escolhe até 2 dos K, L, M, N.³

Definimos os tipos de dados:

data Nat where	data List α where	data Tree $\alpha \beta$ where
O : Nat	Empty : List α	Leaf : $\alpha \rightarrow$ Tree $\alpha \beta$
S : Nat \rightarrow Nat	Cons : $\alpha \rightarrow$ List $\alpha \rightarrow$ List α	Node : $\beta \rightarrow$ List (Tree $\alpha \beta$) \rightarrow Tree $\alpha \beta$

Usamos o açúcar sintático (:) para o Cons, ao qual atribuímos associatividade (sintática) direita; também usamos [] para o Empty.

Definimos as operações

$(+) : \text{Nat} \rightarrow \text{Nat} \rightarrow \text{Nat}$	$(\cdot) : \text{Nat} \rightarrow \text{Nat} \rightarrow \text{Nat}$	$(\wedge) : \text{Nat} \rightarrow \text{Nat} \rightarrow \text{Nat}$
$n + 0 = n$	$n \cdot 0 = 0$	$n \wedge 0 = S0$
$n + Sm = S(n + m)$	$n \cdot Sm = n + n \cdot m$	$n \wedge Sm = n \cdot n \wedge m$

e atribuímos em ambas associatividade (sintática) esquerda. Atribuímos também precedência (sintática) de alta para baixa nesta ordem: (\wedge) , (\cdot) , $(+)$.

Como referir às equações. Para referir à n -ésima equação numa definição recursiva numa função f , use o rótulo « $(f.n)$ ». Por exemplo, a segunda equação da $(+)$, refere por « $(+).2$ ».

Esclarecimento. Suas demonstrações/refutações precisam ser escritas em português matemático (linguagem “mid-level” que temos elaborado nas aulas).

Boas provas!

¹Ou seja, *desligue antes* da prova.

²Se essa regra não faz sentido, melhor desistir desde já.

³Provas violando essa regra (com respostas em mais problemas) não serão corrigidas (tirarão 0 pontos).

(16) **J**

Defina recursivamente as funções:

$$\begin{aligned} & \text{[redacted]} : \text{[redacted]} \rightarrow \text{Nat} \\ & \text{[redacted]} : \text{List } \alpha \rightarrow \text{List } \alpha \rightarrow \text{List } \alpha \\ & \text{[redacted]} : \text{Nat} \rightarrow \alpha \rightarrow \text{[redacted]} \\ & \text{[redacted]} : \text{Nat} \rightarrow \alpha \rightarrow \text{[redacted]} \rightarrow \text{[redacted]} \end{aligned}$$

levando em consideração os exemplos de uso no quadro.

Podes usar as abreviações: [redacted], [redacted], [redacted].

DEFINIÇÕES.

(8) **K**

Demonstre:



Considere dadas as: associatividades e comutatividades de ambas as $(+)$, (\cdot) , e a distributividade de (\cdot) sobre $(+)$.

DEMONSTRAÇÃO.

(14) **L**

Demonstre:

$$(\forall x : \text{List } \alpha) (\forall y : \text{List } \alpha) [\text{List } \alpha = \text{List } \alpha] .$$

DEMONSTRAÇÃO.



(14) **M**

Demonstre:

$$(\forall x : \alpha) (\forall f : \alpha \rightarrow \beta) (\forall n : \text{Nat}) [\text{redacted} = \text{redacted}]$$

DEMONSTRAÇÃO.

(28) **N**

Demonstre:

$$(\forall l : \text{List } \alpha) (\forall u : \alpha) (\forall v : \alpha) (\forall n : \text{Nat}) [\text{ } = \text{ }].$$

DEMONSTRAÇÃO.

Só isso mesmo.

LEMMATA

