Um Novo Olhar sobre o Teorema Fundamental da Álgebra

Seminário de Álgebra, FCUL
30 de Abril de 2004

Luís Cruz-Filipe

Universidade de Nijmegen, Holanda

Centro de Lógica e Computação, Portugal

Aplicações

duma

Formalização

Construtiva

do TFA

Aplicações duma Formalização Construtiva do TFA

- extracção de programas
- o quê, como, porquê

Aplicações

duma

Formalização

Construtiva

do TFA

- extracção de programas
- o quê, como, porquê

- o quê, como, porquê
- interesse teórico e prático

Aplicações

duma

Formalização

Construtiva

do TFA

- extracção de programas
- o quê, como, porquê

- o quê, como, porquê
- interesse teórico e prático

- o quê
- exemplos “naturais”

Plano

Plano

1. Formalização de Matemática

Plano

1. Formalização de Matemática

2. Matemática Construtivista

Plano

1. Formalização de Matemática

2. Matemática Construtivista

3. Extracção de Programas

Plano

1. Formalização de Matemática
2. Matemática Construtivista
3. Extracção de Programas
4. O Teorema Fundamental da Álgebra

Plano

1. Formalização de Matemática
2. Matemática Construtivista
3. Extracção de Programas
4. O Teorema Fundamental da Álgebra
5. Conclusões &c.

Formalização de Matemática

Formalização de Matemática

O quê

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Porquê

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Porquê

Garantia elevada de correcção

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Porquê

Garantia elevada de correcção

Apresentação e visualização de resultados

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Porquê

Garantia elevada de correcção

Apresentação e visualização de resultados

Troca de informação

Formalização de Matemática

O quê

Representação fidedigna de *provas* num computador

Porquê

Garantia elevada de correcção

Apresentação e visualização de resultados

Troca de informação

Aplicações

Matemática Construtivista

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

$\rightsquigarrow$ não há uma interpretação intuitiva deste axioma

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

$\rightsquigarrow$ não há uma interpretação intuitiva deste axioma

$\rightsquigarrow$ realizabilidade (Kleene): uma prova de $\forall x . \exists y . P(x, y)$ define uma função [computável]

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

$\rightsquigarrow$ não há uma interpretação intuitiva deste axioma

$\rightsquigarrow$ realizabilidade (Kleene): uma prova de $\forall x.\exists y.P(x, y)$ define uma função [computável]

Exemplos "naturais":

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

$\rightsquigarrow$ não há uma interpretação intuitiva deste axioma

$\rightsquigarrow$ realizabilidade (Kleene): uma prova de $\forall x. \exists y. P(x, y)$ define uma função [computável]

Exemplos “naturais”:

- isomorfismo de Curry–Howard

Matemática Construtivista

Lógica Intuicionista (Brouwer): rejeita $A \vee \neg A$

$\rightsquigarrow$ não há uma interpretação intuitiva deste axioma

$\rightsquigarrow$ realizabilidade (Kleene): uma prova de $\forall x.\exists y.P(x,y)$ define uma função [computável]

Exemplos "naturais":

- isomorfismo de Curry–Howard
- lógica interna dum topos

Extracção de Programas

Extracção de Programas

Ideia: tornar explícito o algorimo implícito numa prova de $\forall x.\exists y.P(x, y)$

Extracção de Programas

Ideia: tornar explícito o algorimo implícito numa prova de $\forall x.\exists y.P(x,y)$

$\leadsto$ distinção entre o *algoritmo* e as suas *propriedades*

Extracção de Programas

Ideia: tornar explícito o algorimo implícito numa prova de $\forall x.\exists y.P(x, y)$

$\rightsquigarrow$ distinção entre o *algoritmo* e as suas *propriedades*

$\rightsquigarrow$ termos de prova podem influenciar o resultado dos cálculos (e.g. $\frac{1}{x}$)

Extracção de Programas

Ideia: tornar explícito o algorimo implícito numa prova de $\forall x.\exists y.P(x,y)$

$\rightsquigarrow$ distinção entre o *algoritmo* e as suas *propriedades*

$\rightsquigarrow$ termos de prova podem influenciar o resultado dos cálculos (e.g. $\frac{1}{x}$)

Útil quando correcção é mais importante que eficiência

O Teorema Fundamental da Álgebra

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

Sejam $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ com $g(0) \neq 0$

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

Sejam $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ com $g(0) \neq 0$ e $k > 0$

mínimo com $a_k \neq 0$;

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

Sejam $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ com $g(0) \neq 0$ e $k > 0$ mínimo com $a_k \neq 0$; então $g(z) = a_0 + a_k z^k + O\left(z^{k+1}\right)$.

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

Sejam $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ com $g(0) \neq 0$ e $k > 0$ mínimo com $a_k \neq 0$; então $g(z) = a_0 + a_k z^k + O\left(z^{k+1}\right)$. Tomando ε suficientemente pequeno, em $z' = \varepsilon \sqrt[k]{-\frac{a_0}{a_k}}$ o termo em $O\left(z^{k+1}\right)$ é negligenciável,

O Teorema Fundamental da Álgebra

Teorema. Seja f um polinómio não constante de coeficientes complexos. Então f tem uma raíz.

Prova. [H. Kneser, 1940] Nas condições do teorema, $|f(z)| \longrightarrow \infty$ quando $|z| \longrightarrow \infty$, logo $|f|$ tem um mínimo em $z_0 \in \mathbb{C}$.

Sejam $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ com $g(0) \neq 0$ e $k > 0$ mínimo com $a_k \neq 0$; então $g(z) = a_0 + a_k z^k + O\left(z^{k+1}\right)$. Tomando ε suficientemente pequeno, em $z' = \varepsilon \sqrt[k]{-\frac{a_0}{a_k}}$ o termo em $O\left(z^{k+1}\right)$ é negligenciável, e $|g(z')| \approx |a_0|(1-\varepsilon^k) < |g(0)|$. Absurdo.

O Teorema Fundamental da Álgebra

O Teorema Fundamental da Álgebra

$\rightsquigarrow$ não é uma prova construtiva: provámos $\neg\neg\exists z. f(z) = 0$, mais fraco que $\exists z. f(z) = 0$.

O Teorema Fundamental da Álgebra

$\rightsquigarrow$ não é uma prova construtiva: provámos $\neg\neg\exists z. f(z) = 0$, mais fraco que $\exists z. f(z) = 0$.

$\rightsquigarrow$ no entanto, dado z tal que $|f(z)| > 0$ construímos um z' com $|f(z)| > |f(z')|$

O Teorema Fundamental da Álgebra

$\leadsto$ não é uma prova construtiva: provámos $\neg\neg\exists z. f(z) = 0$, mais fraco que $\exists z. f(z) = 0$.

$\leadsto$ no entanto, dado z tal que $|f(z)| > 0$ construímos um z' com $|f(z)| > |f(z')|$

$\leadsto$ poder-se-á usar esta construção para definir uma sucessão de Cauchy que convirja para uma raíz de f?

O Teorema Fundamental da Álgebra

$\rightsquigarrow$ não é uma prova construtiva: provámos $\neg\neg\exists z. f(z) = 0$, mais fraco que $\exists z. f(z) = 0$.

$\rightsquigarrow$ no entanto, dado z tal que $|f(z)| > 0$ construímos um z' com $|f(z)| > |f(z')|$

$\rightsquigarrow$ poder-se-á usar esta construção para definir uma sucessão de Cauchy que convirja para uma raíz de f?

Problema: requisitos opostos sobre ε

Polinómios Mónicos

Polinómios Mónicos

Três problemas:

Polinómios Mónicos

Três problemas:

1. igualdade não é decidível;

Polinómios Mónicos

Três problemas:

1. igualdade não é decidível;
2. não é possível encontrar k com $\sqrt[k]{\frac{|b_0|}{|b_k|}}$ mínimo;

Polinómios Mónicos

Três problemas:

1. igualdade não é decidível;
2. não é possível encontrar k com $\sqrt[k]{\frac{|b_0|}{|b_k|}}$ mínimo;
3. não é possível encontrar k_j com $\left|b_{k_j}\right| r_j^{k_j}$ máximo.

Polinómios Mónicos (cont.)

Polinómios Mónicos (cont.)

Para resolver (1):

Polinómios Mónicos (cont.)

Para resolver (1):

reescrever o resultado como

"se $|f(z_i)| < c$ então $|f(z_{i+1})| < qc$"

com q como atrás.

Polinómios Mónicos (cont.)

Para resolver (1):

reescrever o resultado como

“se $|f(z_i)| < c$ então $|f(z_{i+1})| < qc$”

com q como atrás.

Agora podemos decidir se $|f(z_i)| < qc$ ou $|f(z_i)| > 0$, e o raciocínio anterior é válido.

Polinómios Mónicos (cont.)

Polinómios Mónicos (cont.)

Para resolver (2):

Polinómios Mónicos (cont.)

Para resolver (2): calcular um mínimo "a menos de ε"; isto é, definir ao mesmo tempo r_0 e k_0 tais que, dado $\varepsilon > 0$,

$$\begin{aligned} a_{k_0} r_0^{k_0} &= a_0 - \varepsilon \\ a_i r_0^i - \varepsilon &< a_{k_0} r_0^{k_0} \end{aligned}$$

Polinómios Mónicos (cont.)

Para resolver (2): calcular um mínimo "a menos de ε"; isto é, definir ao mesmo tempo r_0 e k_0 tais que, dado $\varepsilon > 0$,

$$\begin{aligned} a_{k_0} r_0^{k_0} &= a_0 - \varepsilon \\ a_i r_0^i - \varepsilon &< a_{k_0} r_0^{k_0} \end{aligned}$$

(Inicializar $k_0 = n$, $r_0 = \sqrt[n]{a_0 - \varepsilon}$.
Para cada i até 1:

- se $a_i r_0^i < a_0$ não fazer nada;
- se $a_i r_0^i > a_0 - \varepsilon$, redefinir $k_0 = i$ e $r_0 = \sqrt[i]{(a_0 - \varepsilon)/a_i}$.

Polinómios Mónicos (cont.)

Para resolver (2): calcular um mínimo "a menos de ε"; isto é, definir ao mesmo tempo r_0 e k_0 tais que, dado $\varepsilon > 0$,

$$\begin{aligned} a_{k_0} r_0^{k_0} &= a_0 - \varepsilon \\ a_i r_0^i - \varepsilon &< a_{k_0} r_0^{k_0} \end{aligned}$$

(Inicializar $k_0 = n$, $r_0 = \sqrt[n]{a_0 - \varepsilon}$.
Para cada i até 1:

- se $a_i r_0^i < a_0$ não fazer nada;
- se $a_i r_0^i > a_0 - \varepsilon$, redefinir $k_0 = i$ e $r_0 = \sqrt[i]{(a_0 - \varepsilon)/a_i}$.

Quando i atinge 0, k_0 e r_0 satisfazem as condições acima.)

Caso Geral

Caso Geral

Ideia: dado $f(z) = \sum_{i=0}^{n} a_n z^n$, aplicar o raciocínio anterior a f/a_n.

Caso Geral

Ideia: dado $f(z) = \sum_{i=0}^{n} a_n z^n$, aplicar o raciocínio anterior a f/a_n.

Problema: mesmo que f não seja constante não é possível garantir que $a_n \neq 0$.

Caso Geral

Ideia: dado $f(z) = \sum_{i=0}^{n} a_n z^n$, aplicar o raciocínio anterior a f/a_n.

Problema: mesmo que f não seja constante não é possível garantir que $a_n \neq 0$.

$\rightsquigarrow$ abordagem diferente, por indução em n.

O Algoritmo Implícito na Prova

O Algoritmo Implícito na Prova

$\rightsquigarrow$ a prova original de M. Kneser corresponde a aplicar o método de Newton–Raphson para encontrar uma raíz do polinómio

O Algoritmo Implícito na Prova

$\rightsquigarrow$ a prova original de M. Kneser corresponde a aplicar o método de Newton–Raphson para encontrar uma raíz do polinómio

$\rightsquigarrow$ a versão apresentada (e formalizada) é ligeiramente menos eficiente porque os k_is começam em 0 e não em -1

O Algoritmo Implícito na Prova

$\rightsquigarrow$ a prova original de M. Kneser corresponde a aplicar o método de Newton–Raphson para encontrar uma raíz do polinómio

$\rightsquigarrow$ a versão apresentada (e formalizada) é ligeiramente menos eficiente porque os k_is começam em 0 e não em -1

$\rightsquigarrow$ de momento, a aritmética básica é ineficiente; calcular raízes quadradas em $\mathbb{R}$ demora demasiado

Conclusões &c.

Conclusões &c.

- Formalizar matemática é útil

Conclusões &c.

- Formalizar matemática é útil
- “Esquecer” o princípio do terceiro excluído não é muito dramático

Conclusões &c.

- Formalizar matemática é útil
- “Esquecer” o princípio do terceiro excluído não é muito dramático
- Extracção de programas pode vir a ser “a” maneira certa de programar