Lógica Computacional (CC2003) - Folha de trabalho n. 5

1 Axiomatizações

1 Considerando a linguagem de $1^{\underline{a}}$ ordem para a aritmética dada nas aulas e os axiomas de Peano (**PA**) para a teoria dos números obtém deduções naturais (com notação de Fitch) para as fórmulas ψ seguintes (i.e $\mathbf{PA} \vdash \psi$). **Para cada alinea podes considerar que as restantes correspondem a teoremas, que poderás usar.**

Axiomas de Peano (PA)

1. $\forall x (x + 1 \neq 0)$
2. $\forall x \forall y (x + 1 = y + 1 \to x = y)$
3. $0 + 1 = 1$
4. $\forall x\ x + 0 = x$
5. $\forall x \forall y\ x + (y + 1) = (x + y) + 1$
6. $\forall x\ x \times 0 = 0$
7. $\forall x \forall y\ x \times (y + 1) = (x \times y) + x$
8. (princípio da indução) $(Q(0)\ \wedge\ \forall x (Q(x) \to Q(x + 1))) \to \forall x Q(x)$

Sugestão: Usa o Axioma 8 (principio da indução): sendo $\forall x Q(x)$ o que pretendes deduzir, deduz primeiro $Q(0)$ e depois $\forall x (Q(x) \to Q(x + 1))$ (e aplicando o axioma 8 e **modus ponens** tens o que pretendes.)

(a) $\forall x\ 0 + x = x$
(b) $\forall x\ 1 \times x = x$
(c) $\forall x\ 0 \times x = 0$
(d) $\forall x\ x + 0 = 0 + x$
(e) $\forall x\ x + 1 = 1 + x$
(f) $\forall x\ x \times 1 = x$
(g) $\forall x\ 1 \times x = x \times 1$
(h) $\forall x\ \forall y (x + y) + 0 = x + (y + 0)$
(i) $\forall z\ \forall x \forall y\ (x + y) + z = x + (y + z)$ usando indução em z.
(j) $\forall y \forall x\ (x + 1) \times y = (x \times y) + y$ usando indução em y
(k) $\forall y\ \forall x\ (x + y = y + x)$, usando indução em y
(l) $\forall x\ \forall y (x \times y) \times 0 = x \times (y \times 0)$.
(m) $\forall z\ \forall x \forall y\ (x \times y) \times z = x \times (y \times z)$, usando indução em z
(n) $\forall y\ \forall x\ (x \times y = y \times x)$, usando indução em y.

2 Seja $\mathcal{L}_N$ a linguagem de $1^{\underline{a}}$ ordem com igualdade para os números naturais tal que $\mathcal{F}_0 = \{0, 1\}$ e $\mathcal{F}_2 = \{+, \times\}$ (e onde os termos e as fórmulas atómicas são representados em notação infixa). Seja $\mathcal{N} = (\mathbb{N}, \cdot^{\mathcal{N}})$ a estrutura onde $\cdot^{\mathcal{N}}$ associa aos símbolos funcionais os correspondentes valores e operações aritméticas. Considera ainda os axiomas de Peano, **PA**. Nota: Representa o número 2 pelo termo `1 + 1`.

a) – Define uma fórmula $\texttt{Impar}(x)$ de $\mathcal{L}_N$ tal que $\mathcal{N} \models_s \texttt{Impar}(x)$ se e só se $s(x)$ é impar, i.e se o resto da divisão inteira por 2 é 1.
– Mostra, usando o sistema dedutivo de dedução natural, que $\mathbf{PA} \vdash \texttt{Impar}(\texttt{1})$.

b) – Define uma fórmula $\texttt{Par}(y)$ de $\mathcal{L}_N$, tal que $\mathcal{N} \models_s \texttt{Par}(y)$ se e só se $s(y)$ é par.
– Mostra, usando o sistema dedutivo de dedução natural, que $\mathbf{PA} \vdash \texttt{Par}(\texttt{0})$.

c) – Define uma fórmula $\texttt{Primo}(y)$ de $\mathcal{L}_N$, tal que $\mathcal{N} \models_s \texttt{Primo}(y)$ se e só se $s(y)$ é primo.
– Mostra, usando o sistema dedutivo de dedução natural, que $\mathbf{PA} \vdash \texttt{Primo}(2)$.

Lógica Computacional (CC2003) - Folha de trabalho n. 5

Resolução de exercícios selecionados

Resolução 1.1.a

1		$\forall x(x+1 \neq 0)$	
2		$\forall x \forall y(x+1=y+1 \rightarrow x=y)$	
3		$0+1=1$	
4		$\forall x\; x+0=x$	
5		$\forall x \forall y\; x+(y+1)=(x+y)+1$	
6		$\forall x\; x \times 0=0$	
7		$\forall x \forall y\; x \times (y+1)=(x \times y)+x$	
8		$(Q(0) \;\wedge\; (\forall x(Q(x) \rightarrow Q(x+1)))) \rightarrow \forall x Q(x)$	
9		$0+0=0$	$\forall$E, 4
10	u	$0+u=u$	
11		$\forall y\; 0+(y+1)=(0+y)+1$	$\forall$E, 7
12		$0+(u+1)=(0+u)+1$	$\forall$E, 7, 11
13		$0+(u+1)=u+1$	$=$**E**, 10, 12
14		$0+u=u \rightarrow 0+(u+1)=u+1$	$\rightarrow$I, 10–13
15		$\forall x(0+x=x \rightarrow 0+(x+1)=x+1)$	$\forall$I, 10–14
16		$0+0=0 \;\wedge\; \forall x(0+x=x \rightarrow 0+(x+1)=x+1)$	$\wedge$I, 10, 15
17		$\forall x(0+x=x)$	$\rightarrow$E, 8, 16

Resolução 1.1.b

1		$\forall x(x+1 \neq 0)$	
2		$\forall x \forall y(x+1=y+1 \rightarrow x=y)$	
3		$0+1=1$	
4		$\forall x\; x+0=x$	
5		$\forall x \forall y\; x+(y+1)=(x+y)+1$	
6		$\forall x\; x \times 0=0$	
7		$\forall x \forall y\; x \times (y+1)=(x \times y)+x$	
8		$(Q(0) \;\wedge\; (\forall x(Q(x) \rightarrow Q(x+1)))) \rightarrow \forall x Q(x)$	
9		$1 \times 0=0$	$\forall$E, 6
10	u	$1 \times u=u$	
11		$\forall y\; 1 \times (y+1)=(1 \times y)+1$	$\forall$E, 7
12		$1 \times (u+1)=(1 \times u)+1$	$\forall$E, 7
13		$1 \times (u+1)=u+1$	$=$**E**, 10, 12
14		$1 \times u=u \rightarrow 1 \times (u+1)=u+1$	$\rightarrow$I, 10–13
15		$\forall x(1 \times x=x \rightarrow 1 \times (x+1)=x+1)$	$\forall$I, 10–14
16		$1 \times 0=0 \;\wedge\; \forall x(1 \times x=x \rightarrow 1 \times (x+1)=x+1)$	$\wedge$I, 9, 15
17		$\forall x 1 \times x=x$	$\rightarrow$E, 8, 16

Resolução 1.1.e

1	$\forall x(x+1 \neq 0)$	
2	$\forall x \forall y(x+1 = y+1 \to x = y)$	
3	$0+1 = 1$	
4	$\forall x\ x+0 = x$	
5	$\forall x \forall y\ x+(y+1) = (x+y)+1$	
6	$\forall x\ x \times 0 = 0$	
7	$\forall x \forall y\ x \times (y+1) = (x \times y)+x$	
8	$(Q(0)\ \wedge\ (\forall x(Q(x) \to Q(x+1)))) \to \forall x Q(x)$	
9	$1+0 = 1$	$\forall$E, 4
10	$0+1 = 1+0$	=**I**, 3, 9
11	u $\quad u+1 = 1+u$	
12	$\forall y\ 1+(y+1) = (1+y)+1$	$\forall$E, 7
13	$1+(u+1) = (1+u)+1$	$\forall$E, 7, 12
14	$1+u = u+1$	= simetria, 11
15	$1+(u+1) = (u+1)+1$	=**E**, 14, 13
16	$(u+1)+1 = 1+(u+1)$	= simetria, 15
17	$u+1 = 1+u \to (u+1)+1 = 1+(u+1)$	$\to$I, 11–16
18	$\forall x(x+1 = 1+x \to (x+1)+1 = 1+(x+1))$	$\forall$I, 11–17
19	$0+1 = 1\ \wedge\ \forall x(x+1 = 1+x \to (x+1)+1 = 1+(x+1))$	$\wedge$I, 10, 18
20	$\forall x(x+1 = 1+x)$	$\to$E, 8, 19

Programação em Lógica

Forma clausal

3 Para cada uma das seguintes fórmulas encontre uma fórmula em forma clausal. Para tal:

1. Converte numa fórmula equivalente em forma normal prenexa.
2. Elimina os quantificadores existênciais introduzindo novos símbolos funcionais (*Skolemização*.
3. Converte a matriz da fórmula resultante para forma normal conjuntitiva.
4. Distribuí os quantificadores universais pelas conjunções, aplicando a regra:

$$\forall x(\phi\ \wedge\ \psi) \quad \longrightarrow \quad \forall x\phi\ \wedge\ \forall x\psi$$

5. Escreve a fórmula resultante em notação clausal.

(a) $\forall x P(x) \to \forall y Q(y)$;

(b) $(\forall x P(x) \to \exists x Q(x)) \wedge \forall z(\neg P(z) \wedge Q(z))$;

(c) $\forall y(\exists x(P(x,y) \to (Q(x) \vee Q(f(x)))))$;

(d) $\exists y \forall x P(x,y) \to \forall x \exists y P(x,y)$.

(e) $\neg \forall x R(x,u)\ \wedge\ \exists y(\neg \forall x R(y,x) \to Q(u,y))$

(f) $\forall x \neg \exists y \forall u P(x,y) \rightarrow \neg(\exists z P(x,y) \rightarrow \forall y P(u,y))$

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

(h) $\neg(\exists x Q(x,y) \rightarrow \neg \forall x(\forall z P(z,y,x) \rightarrow \neg \exists y P(y)))$

(i) $\neg(\forall x R(x,y) \rightarrow \exists x \neg(\exists z Q(z,y,x) \rightarrow \forall y P(y)))$

4 Considera o programa $\mathcal{P}$ seguinte:

$$\begin{array}{l} \mathsf{S(f(x)) \leftarrow S(x), R(y)} \\ \mathsf{R(b) \leftarrow} \\ \mathsf{R(f(x)) \leftarrow R(x)} \end{array}$$

Escreve cada fórmula do programa sem ser na *notação clausal* (i.e com os quantificadores e operações lógicas usuais).

5 Considera os seguintes programas P e objectivos G:

(a)

$$\begin{array}{ll} P: & P(a) \leftarrow P(x), Q(x) \\ & P(f(x)) \leftarrow P(x) \\ & Q(b) \leftarrow \\ & Q(f(x)) \leftarrow Q(x) \\ & \\ G: & \leftarrow Q(f(f(a))) \end{array}$$

(b)

$$\begin{array}{ll} P: & Add(x,0,x) \leftarrow \\ & Add(x,s(y),s(z)) \leftarrow Add(x,y,z) \\ & \\ G: & \leftarrow Add(s(0),s(s(0)),x) \end{array}$$

(c)

$$\begin{array}{ll} P: & Add(x,0,x) \leftarrow \\ & Add(x,s(y),s(z)) \leftarrow Add(x,y,z) \\ & \\ G: & \leftarrow Add(u,v,s(s(s(0)))) \end{array}$$

(d)

$$\begin{array}{ll} P: & P(a,b) \leftarrow \\ & P(c,b) \leftarrow \\ & P(x,z) \leftarrow P(x,y), P(y,z) \\ & P(x,y) \leftarrow P(y,x) \\ & \\ G: & \leftarrow P(a,c) \end{array}$$

(e)

$$\begin{array}{ll} P: & Lenght(nil,0) \leftarrow \\ & Lenght(cons(x,y),s(z)) \leftarrow Lenght(y,z) \\ & \\ G: & \leftarrow Lenght(x,s(s(y))) \end{array}$$

(f)

$$\begin{array}{ll} P: & Append(cons(x,x_R),x_L,cons(x,x_{RL})) \leftarrow Append(x_R,x_L,x_{RL}) \\ & Append(nil,x_L,x_L) \leftarrow \\ & \\ G: & \leftarrow Append(cons(x,nil),cons(x,nil),y) \end{array}$$

Lógica Computacional (CC2003) - Folha de trabalho n. 5

(a) Escreve cada fórmula do programa e do objectivo sem ser na *notação clausal* (i.e com os quantificadores e operações lógicas usuais).

(b) Mostra que $P \cup \{G\} \vdash \mathbf{F}$ usando o sistema de dedução natural (e a notação de Fitch) e as fórmulas na forma obtida em 5.a.

Unificação

6 Para cada uma das expressões E e cada uma das substituições σ determina $E\sigma$.

- $E = f(x, g(y, z), z)$ e $\sigma = [g(y, z)/x, a/z, z/y, f(f(b))/u]$;
- $E = P(f(z), x)$ e $\sigma = [a/y]$;
- $E = Add(x, s(y), s(z)) \leftarrow Add(x, y, z)$ e $\sigma = [s(0)/x, s(0)/y, s(s(0))/z]$.

7 Para as substituições θ e σ descritos abaixo, determina a composta $\theta\sigma$.

- $\theta = [f(x)/y, b/z]$ e $\sigma = [c/x]$;
- $\theta = [c/x, f(z)/y, u/v]$ e $\sigma = [v/u, x/z]$;
- $\theta = [f(z)/x, g(z)/y, a/u]$ e $\sigma = [f(b)/x, g(b)/y]$.

8 Para cada um dos valores abaixo determina $\theta\sigma$, $E\theta$, $(E\theta)\sigma$ e $E(\theta\sigma)$.

(a) $\theta = [f(y)/x, y/z]$, $\sigma = [a/x, b/z]$ e $E = P(x, y, g(z))$.

(b) $\theta = [h(u)/x, g(z)/y, u/v]$, $\sigma = [v/u, f(x)/z]$ e $E = g(x, f(y, u), g(z))$.

(c) $\theta = [h(y, x)/x, f(z)/y, x/u]$, $\sigma = [g(z)/x, h(u)/z, a/u]$ e $E = R(g(x), z, h(u, u))$.

(d) $\theta = [h(u, y)/x, f(z)/v, b/y]$, $\sigma = [v/u, h(x)/y, z/v]$ e $E = P(h(u, v), y, g(y, x))$.

9 Para as expressões E e F seguintes, determine se são variantes. Justifique a sua resposta. Para os pares (E, F) que são variantes determine ainda mudanças de nome σ e θ respectivamente para E e F tais que $E = F\theta$ e $F = E\sigma$.

1. $E = P(f(x, y), g(z), a)$ e $F = P(f(y, x), g(u), a)$;
2. $E = P(x, f(x))$ e $F = P(y, f(z))$;
3. $E = P(x, y)$ e $F = P(y, x)$;
4. $E = P(x, y, y)$ e $F = P(y, x, y)$.

10 Aplica o algoritmo de unificação de Robinson a cada um dos conjuntos S de expressões simples seguintes, indicando a resposta do algoritmo e ainda para cada iteração efectuada a substituição correspondente σ_i e os conjuntos D_i e $S\sigma_i$.

(a) $S = \{P(f(a), g(x)), P(y, y)\}$;

(b) $S = \{P(a, x, h(g(z))), P(z, h(y), h(y))\}$;

(c) $S = \{P(x, x), P(y, f(y))\}$;

(d) $S = \{R(f(x, g(u))), R(f(g(u), g(z))\}$;

(e) $S = \{R(f(x, g(y))), R(f(g(u), h(z))\}$;

(f) $S = \{P(x, f(z)), P(f(z), y), P(y, z)\}$

(g) $S = \{R(f(x, g(x), x)), R(f(g(u), g(g(z)), z))\}$;

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

(i) $S = \{P(f(x, f(x, y)), z), P(f(g(y), f(g(a), z)), u)\}$;

(j) $S = \{P(g(x), y), P(y, y), P(u, f(w))\}$;

(k) $S = \{P(z, g(x, f(x, w))), P(u, g(g(w), f(g(b), z)))\}$

(l) $S = \{P(g(z, h(z), z)), P(g(f(v), h(h(x)), x))\}$.

11 Sejam E e F duas expressões. Relaciona as afirmações seguintes:

a. E e F são variantes;

b. o conjunto $\{E, F\}$ é unificável.

12 O seguinte algoritmo também permite obter o unificador mais geral entre duas expressões:

Entrada: Dois termos `T1` e `T2`

Saída: `U` unificador mais geral de `T1`, `T2` ou `Falso`

Algoritmo: Inicializar `U` como a substituição vazia, uma pilha contendo a equação `T1 = T2` e $f \leftarrow 0$

Enquanto a pilha não estiver vazia e $f \neq 1$

retira `X=Y` da pilha

caso:

- `X` é uma variável que não ocorre em `Y`, substitui `X` por `Y` na pilha e adiciona `Y/X` a `U`
- `Y` é uma variável que não ocorre em `X`, substitui `Y` por `X` na pilha e adiciona `X/Y` a `U`
- `X` e `Y` são variáveis idênticas, continua
- `X=f(X1,...,Xn)` e `Y=f(Y1,...,Yn)` coloca `Xi=Yi` na pilha para i=1..n

caso contrário: $f \leftarrow 1$

Se $f = 1$ retorna *Falso* senão `U`

(a) Implementa este algoritmo.

(b) Repete o Exercício 10 utiliando este algoritmo.

Fórmulas de Horn e Programas Definidos

13 Considera o seguinte conjunto de cláusulas de Horn de lógica proposicional:

$$C = \{\{a, \neg b\}, \{b, \neg c\}, \{b, \neg d\}, \{c, \neg e\}\{d, \neg e\}, \{d\}, \{e\}, \{\neg a\}\}$$

Usando resolução mostra que $C \vdash \mathbf{F}$ (isto é constrói uma refutação).

14 Considera o algoritmo de satisfabilidade para fórmulas de Horn dado no curso.

(a) Descreve o algoritmo em pseudo-código e implementa-o.

(b) Justifica a correção do algoritmo, isto é, que atribui o valor verdade se e só se a fórmula de Horn é satisfazível.

(c) Aplica o algoritmo às seguintes fórmulas:

- $(\neg p \lor \neg q) \land (\neg q \lor r) \land q$
- $p \land (\neg p \lor q) \land (\neg q \lor p)$
- $\neg p \land (\neg p \lor q) \land \neg q$
- $p \land (\neg p \lor q) \land \neg r$

15 Justifica a validade ou a falsidade das seguintes afirmações, para uma linguagem de 1ª ordem:

Lógica Computacional (CC2003) - Folha de trabalho n. 5

1. para toda a proposição φ existe um conjunto (finito) Φ de fórmulas universais fechadas de Horn tal que $\Phi \models \varphi$ e $\varphi \models \psi$ para qualquer $\psi \in \Phi$;
2. se φ é uma fórmula de Horn, então é uma fórmula de Horn positiva ou uma fórmula de Horn negativa;
3. todo o conjunto de fórmulas de Horn positivas é satisfazível
4. todo o conjunto de fórmulas de Horn negativas é satisfazível
5. todo o conjunto de fórmulas de Horn é satisfazível

Resolução-SLD

16 Seja $\mathcal{P}$ o programa seguinte

$$\begin{array}{l} Max(x,y,x) \leftarrow Leq(y,x) \\ Max(y,x,x) \leftarrow Leq(y,x) \\ Leq(0,x) \leftarrow \\ Leq(s(x),s(y)) \leftarrow Leq(x,y) \end{array}$$

Verifica se $\mathcal{P} \models \exists x Max(s(x),x,s(x))$.

17 Considera G o objectivo $\leftarrow \mathsf{P}(\mathsf{x},\mathsf{x})$ e o programa $\mathcal{P}$ seguinte:

$$\begin{array}{l} \mathsf{P}(\mathsf{a},\mathsf{b}) \leftarrow \\ \mathsf{P}(\mathsf{x},\mathsf{z}) \leftarrow \mathsf{P}(\mathsf{x},\mathsf{y}),\mathsf{P}(\mathsf{y},\mathsf{z}) \\ \mathsf{P}(\mathsf{x},\mathsf{y}) \leftarrow \mathsf{P}(\mathsf{y},\mathsf{x}) \end{array}$$

Calcula uma refutação-**SLD** para $\mathcal{P} \cup \{G\}$ e a resposta calculada.

18 Para o programa definido P e objectivo G (da forma $\leftarrow \beta$) descritos abaixo:

- Escreve cada fórmula do programa e do objectivo sem ser em notação clausal (i.e com quantificadores e operações lógicas)
- Calcula uma refutação **SLD** para $P \cup \{G\}$, indicando as substituições em cada passo e a resposta calculada, no fim. Representa a refutação-SLD por uma árvore de derivação.
- Verifica a execução do programa com o objectivo em Prolog, usando as conversões definidas no Exercício **??**.

(a)
$$\begin{array}{ll} P: & Add(x,0,x) \leftarrow \\ & Add(x,s(y),s(z)) \leftarrow Add(x,y,z) \end{array}$$

$$G: \ \leftarrow Add(s(0),x,s(s(y))$$

(b)
$$\begin{array}{ll} P: & Lenght(nil,0) \leftarrow \\ & Lenght(cons(x,y),s(z)) \leftarrow Lenght(y,z) \end{array}$$

$$G: \ \leftarrow Lenght(x,s(s(y)))$$

(c)
$$\begin{array}{ll} P: & Append(cons(x,x_R),x_L,cons(x,x_{RL})) \leftarrow Append(x_R,x_L,x_{RL}) \\ & Append(nil,x_L,x_L) \leftarrow \end{array}$$

$$G: \ \leftarrow Append(cons(x,nil),cons(x,nil),y)$$

(d) $P:\quad Member(x, cons(x, y)) \leftarrow$
$\quad Member(x, cons(y, z)) \leftarrow Member(x, z)$

$G: \quad \leftarrow Member(x, cons(y, cons(x, z)))$

(e) $P:\quad Split(cons(x, y), cons(x, nil), y) \leftarrow$
$\quad Split(cons(x, y), cons(x, x_1), y_1) \leftarrow Split(y, x_1, y_1)$

$G: \quad \leftarrow Split(cons(y, cons(x, nil)), z, w)$

(f) $P:\quad P(s(x), y) \leftarrow I(y, z), P(x, z)$
$\quad P(\mathsf{nil}, y) \leftarrow Fi(y)$
$\quad I(a, b) \leftarrow$
$\quad I(b, c) \leftarrow$
$\quad I(c, b) \leftarrow$
$\quad Fi(b) \leftarrow$

$G: \quad \leftarrow P(s(x), a)$

(g) $P:\quad O(r(x)) \leftarrow P(x)$
$\quad P(0) \leftarrow$
$\quad P(r(r(x))) \leftarrow P(x)$

$G: \quad \leftarrow O(r(r(r(x))))$

Resolução de exercícios selecionados

Resolução (4)

$$\forall x \forall y (S(f(x)) \;\vee\; \neg S(x) \;\vee\; \neg R(y))$$
$$R(b)$$
$$\forall x (R(f(x)) \;\vee\; \neg R(x))$$

Resolução (10.f)

- $\sigma_0 = \iota$
- $\mathsf{D}_0 = \{\mathsf{x}, \mathsf{f(z)}, \mathsf{y}\}$, $\sigma_1 = [\mathsf{f(z)/x}]$ e $\mathsf{S}\sigma_1 = \{\mathsf{P(f(z), f(z)), P(f(z), y), P(y, z)}\}$
- $\mathsf{D}_1 = \{\mathsf{f(z)}, \mathsf{y}\}$, $\sigma_2 = \sigma_1[\mathsf{f(z)/y}] = [\mathsf{f(z)/x, f(z)/y}]$ e $\mathsf{S}\sigma_2 = \{\mathsf{P(f(z), f(z)), P(f(z), f(z)), P(f(z), z)}\}$
- $\mathsf{D}_2 = \{\mathsf{f(z)}, \mathsf{z}\}$, como z ocorre em $\mathsf{f(z)}$ o algoritmo termina indicando que S não é unificável.

Resolução (17)

$\leftarrow \mathsf{P(x, x)}$	$\mathsf{P(x_1, z_1) \leftarrow P(x_1, y_1), P(y_1, z_1)}$
	$\sigma_1 = [\mathsf{x/x_1, x/z_1}]$
$\leftarrow \mathsf{P(x, y_1), P(y_1, x)}$	$\mathsf{P(a, b)} \leftarrow$
	$\sigma_2 = [\mathsf{a/x/, b/y_1}]$
$\leftarrow \mathsf{P(b, a)}$	$\mathsf{P(x_2, y_2) \leftarrow P(y_2, x_2)}$
	$\sigma_3 = [\mathsf{b/x_2/, a/y_2}]$
$\leftarrow \mathsf{P(a, b)}$	$\mathsf{P(a, b)} \leftarrow$
	$\sigma_4 = \iota$
ϵ	

$\sigma_1\sigma_2\sigma_3\sigma_4 = [\mathsf{a/x_1, a/z_1, a/x, b/y_1, b/x_2, a/y_2}]$
Resposta calculada:$\sigma = [\mathsf{a/x}]$