

Počítačové algebraické systémy a nekomutativní algebry

Maxim Vrána

Prohlášení:

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a uvedl jsem veškerou použitou literaturu.

V Praze dne 11. ledna 2005

Maxim Vrána

Poděkování:

Tímto bych chtěl poděkovat Prof. Ing. Pavlu Šťovíčkovi, DrSc. za jeho rady a věcné připomínky k mé práci.

1 OBSAH

1	OBSAH.....	3
1.1	Seznam vzorců.....	4
1.2	Seznam obrázků.....	4
2	ÚVOD.....	6
2.1	Obsah práce.....	6
3	HOPFOVA ALGEBRA.....	8
3.1	Základní pojmy.....	8
3.2	Koalgebra, bialgebra a Hopfova algebra.....	10
3.2.1	Algebra.....	10
3.2.2	Koalgebra.....	12
3.2.3	Bialgebra.....	13
3.2.4	Hopfova algebra.....	14
4	q -CALCULUS.....	17
5	KVANTOVÁ GRUPA $U_q(sl_2)$	19
5.1	Algebra $U_q(sl_2)$	19
5.2	Hopfova algebra $U_q(sl_2)$	19
6	KVANTOVÁ GRUPA $SL_q(2)$	21
6.1	Algebra $SL_q(2)$	21
6.2	Hopfova algebra $SL_q(2)$	21
7	REPREZENTACE, KOREPREZENTACE, DUÁLNÍ PÁROVÁNÍ.....	23
7.1	Reprezentace.....	23
7.2	Koreprezentace.....	24
7.2.1	Matice koreprezentace T_n Hopfovy algebry $SL_q(2)$	25
7.3	Duální párování.....	27
8	VYPRACOVANÝ PROGRAM.....	31
8.1	Technické parametry vypracovaného programu.....	31
8.2	Instalace programu.....	33
8.3	Popis funkcí z hlavičkových souborů a vypracované identity.....	34
8.3.1	q -Calculus.....	34
8.3.1.1	Popis funkcí z q -calculus.mws.....	35
8.3.1.2	Vypracované identity využívající q -calculus.mws.....	37
8.3.2	Hopfova algebra $U_q(sl_2)$	40
8.3.2.1	Popis funkcí z HopfAlgebra.mws.....	40
8.3.2.2	Vypracovaná identita využívající HopfAlgebra.mws.....	47
8.3.3	Hopfova algebra $SL_q(2)$	50
8.3.3.1	Popis funkcí z DualAlgebra.mws.....	51
8.3.3.2	Vypracovaná identita využívající DualAlgebra.mws.....	54
8.3.4	Duální párování Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$	56
9	ZÁVĚR.....	60
9.1	Získané výsledky.....	60
9.2	Zjištěné problémy.....	60
9.3	Další možnosti využití práce.....	61
10	LITERATURA.....	62

1.1 Seznam vzorců

Vzorec 1: q – číslo n	17
Vzorec 2: q – faktoriál n	17
Vzorec 3: Výraz $(a; q)_n$	17
Vzorec 4: q – binomický koeficient	17
Vzorec 5: q – binomická věta	18
Vzorec 6: Hypergeometrická funkce	18
Vzorec 7: Malý q – Jacobi polynom	18
Vzorec 8: Relace v algebře $U_q(sl_2)$ – varianta 1	19
Vzorec 9: Konásobení, kojednotka, antipod v $U_q(sl_2)$ – varianta 1	20
Vzorec 10: Relace v algebře $U_q(sl_2)$ – varianta 2	20
Vzorec 11: Konásobení, kojednotka, antipod v $U_q(sl_2)$ – varianta 2	20
Vzorec 12: Relace v algebře $SL_q(2)$	21
Vzorec 13: Konásobení, kojednotka, antipod v $SL_q(2)$	21
Vzorec 14: Koeficienty $t^{(l)}_{ij}$ matice koreprezentace	26
Vzorec 15: Duální párování na $U_q(sl_2)$ a $SL_q(2)$	29
Vzorec 16: Zobecněná q - exponenciální funkce	38

1.2 Seznam obrázků

obrázek 1: Diagram 1 – algebra	11
obrázek 2: Diagram 2 – algebra	11
obrázek 3: Diagram 3, 4 – koalgebra	12
obrázek 4: Diagram 5 – Hopfova algebra	14
obrázek 5: Vytvoření hlavičkového souboru v Maple	31
obrázek 6: Vložení hlavičkového souboru do programu	32
obrázek 7: Funkce z <i>q-calculus.mws</i>	35
obrázek 8: Použití funkce „combNumber“	35
obrázek 9: Použití funkce „commute“	36
obrázek 10: Použití funkce „fact“	36
obrázek 11: Použití funkce „multiplyPol“	36
obrázek 12: První identita v <i>qIdentity.mws</i>	38
obrázek 13: Funkce z <i>HopfAlgebra.mws</i>	40
obrázek 14: Použití funkce „HopfCoCommute“	41
obrázek 15: Použití funkce „HopfComultiply“	41
obrázek 16: Použití funkce „HopfComultiplyPol“	41
obrázek 17: Použití funkce „HopfCommute“	42
obrázek 18: Použití funkce „HopfExistQNum“	42
obrázek 19: Použití funkce „HopfFillLIST“	43
obrázek 20: Použití funkce „HopfMultiplyPol“	43
obrázek 21: Použití funkce „HopfTransform“	43
obrázek 22: Příklad konásobení členů polynomu pomocí funkce „HopfComultiply“ a „HopfCoCommute“	45
obrázek 23: Použití funkcí „HopfExistQNum“ a „HopfFillLIST“	47
obrázek 24: Funkce z <i>DualAlgebra.mws</i>	51
obrázek 25: Použití funkce „DualPairing“	52

obrázek 26: Použití funkce „HyperGeom“	52
obrázek 27: Použití funkce „NNumber“	53
obrázek 28: Použití funkce „tNumber“	53
obrázek 29: Použití funkce „TMatrix“	54
obrázek 30: Postup ověřování vlastnosti koreprezentační matice T - použití funkcí.....	55
obrázek 31: Postup ověřování vlastnosti koreprezentační matice T - výsledek.....	56
obrázek 32: Rozdíl použití a výpočtu funkcí „DualPairing“ a „DualPairing2“	57
obrázek 33: Použití funkce „ProvePairing“	58
obrázek 34: Použití funkce „ProveMatrixPairing“	59

2 ÚVOD

Hlavním zdrojem pro kvantové grupy byla kvantová verze inverzní úlohy rozptylu rozvíjená L.D. Faddeevem, E.K. Sklyaninem a L.A. Takhtajanem v 80-tých letech. Kvantové grupy se postupně objevovaly v Liouvilleově problému a v kvantovém problému pro Sine - Gordonovu rovnici atd.

V.G. Drinfeld byl zřejmě první, kdo si povšiml, že hlavní algebraický objekt v těchto konstrukcích je velice speciální případ bialgebry a Hopfovy algebry. Tento objekt pak použil a zobecnil v definici kvantových grup. Nezávisle na něm se pojem kvantové grupy objevil v pracích M. Jimba. Práce V.G. Drinfelda byla přednesena na kongresu matematiků v Berkeley francouzským matematikem P. Cartierem.

Přibližně ve stejné době se kvantová algebra $SL_q(2)$ objevuje v pracích S.L. Woronowicze, ve kterých studuje „kompaktní maticové pseudogrupy“. V těchto pracích je použit přístup z C^* -algeber.

Následuje ohromný rozvoj kvantových grup. Ve fyzikálních aplikacích se kvantové grupy objevují nejen v algebraické formulaci inverzní úlohy rozptylu, ale také v exaktně řešitelných modelech, ve 2-dimensionální teorii pole, v teorii uzlu, při konstrukci invariantu pro 3-variety, v 2-topologických teoriích pole a v nekomutativní geometrii. Skupina okolo M. Jimba vytváří algebraickou formulaci XXZ-modelu na základě reprezentací a vertexových operátorů kvantové algebry $U_q(sl_2)$.

Z matematického hlediska byl pojem Hopfovy algebry zaveden Hopfem v [3]. Základní přehled lze získat v [2].

V současné době neexistuje žádná uspokojivá hlavní definice kvantových grup. Obvykle se akceptuje pravidlo, že kvantové grupy jsou určité „hezké“ Hopfovy algebry. V celé této práci budou používány dvě kvantové grupy $U_q(sl_2)$ a $SL_q(2)$.

2.1 Obsah práce

Úkolem této diplomové práce je ukázat využití moderních počítačových algebraických systémů typu Maple, nebo Mathematica v teorii kvantových grup. Přehled základní teorie potřebné k definici Hopfovy algebry je vyložen v kapitole 3. Kapitola 4 obsahuje teorii takzvaného q - calculu, definice zobecněných q - čísel, q - faktoriálu, zobecněné q - binomické věty a dalších pojmů úzce souvisejících s kvantovými grupami. Kapitoly 5 a 6 popisují konstrukci konkrétních kvantových grup $U_q(sl_2)$ resp. $SL_q(2)$. V kapitole 7 jsou definice reprezentace a korepresentace, dále pak duálního párování mezi Hopfovými algebrami. Na příkladě je ukázána dualita ve smyslu duálního párování mezi konkrétními Hopfovými algebrami $U_q(sl_2)$ a $SL_q(2)$. Samotné využití počítačových algebraických systémů (konkrétně systém Maple 8) je vyloženo v kapitole 8, kde jsou popsány veškeré vypracované funkce pro práci s q - calculem a Hopfovými algebrami $U_q(sl_2)$ a $SL_q(2)$. Vypracování sady funkcí pro ověřování identit v kvantových algebrách je stěžejním cílem celé této práce, proto i kapitola 8 obsahuje jádro celé práce. Kromě popisu a využití sady vypracovaných funkcí, obsahuje dále postup ověření vybraných identit pomocí těchto funkcí a dále i postup instalace celého programu na počítač a jeho

následné spuštění. V kapitole 9 je závěr celé práce, získané výsledky, popis problémů týkajících se vypracovaného programu a možný další rozvoj celé práce. Poslední kapitolou je pak seznam použité literatury.

3 HOPFOVA ALGEBRA

Pro definování Hopfovy algebry je třeba znát několik základních obecných algebraických pojmů.

3.1 Základní pojmy

Definice 3.1.1: Pologrupa (S, \square) je množina S spolu s asociativní binární operací $\square: S \times S \rightarrow S$.

Definice 3.1.2: Monoid (M, \square, u) je pologrupa (M, \square) s jednotkou u vzhledem k \square . Tzn.

$$u \in M \text{ a } u \square x = x \square u = x \text{ pro všechny } x \in M.$$

Definice 3.1.3: Grupa G je množina G spolu s binární operací $G \times G \rightarrow G ((a, b) \rightarrow ab)$, pro kterou platí:

1. Tato operace je asociativní.
2. Existuje prvek $u \in G$ takový, že $ua = a = au$ pro všechny $a \in G$. Prvek u se nazývá *jednotka*.
3. Ke každému prvku $a \in G$ existuje prvek $a' \in G$ takový, že $aa' = u = a'a$, kde u je jednotka a a' nazýváme inverzní prvek k prvku a .

Definice 3.1.4: Okruh $R = (R, +, \cdot, 1)$ je množina R se dvěma binárními operacemi (sčítání a násobení) a s unární operací „vybrání 1“ taková, že:

1. $(R, +)$ je komutativní grupa vzhledem ke sčítání.
2. (R, \cdot) je monoid vzhledem k násobení.
3. Násobení je distributivní (oboustranně) vzhledem ke sčítání tzn. platí:
pro každou trojici $a, b, c \in R$ platí $a(b+c) = ab + ac$, $(a+b)c = ac + bc$.

Komutativní okruh je okruh, v kterém je násobení komutativní.

Definice 3.1.5: Nechť R je okruh a nazvěme ho okruhem skalárů. Jeho prvky označme malými řeckými písmeny $\lambda, \nu, \mu \dots$

Potom R modul je aditivní komutativní grupa spolu s funkcí

$$R \times A \rightarrow A \text{ ozn. } (\lambda, a) \rightarrow \lambda a,$$

která splňuje následující axiomy:

Pro všechny $\lambda, \nu \in R$ a $a, b \in A$:

- (1) $\lambda(a + b) = \lambda a + \lambda b$,
- (2) $(\lambda + \nu)a = \lambda a + \nu a$,
- (3) $(\lambda\nu)a = \lambda(\nu a)$,
- (4) $1a = a$.

Modul je tedy aditivní komutativní grupa, jejíž prvky mohou být vynásobeny prvky z okruhu R .

Definice 3.1.6: (Asociativní) algebra je vektorový prostor A nad okruhem K spolu se zobrazením:

$(a, b) \rightarrow ab, A \times A \rightarrow A$, pro které platí:

$$a(bc) = (ab)c, (a + b)c = ac + bc, a(b + c) = ab + ac, \\ \alpha(ab) = (\alpha a)b = a(\alpha b)$$

pro všechna $a, b \in A$ a $\alpha \in K$. V celé této práci bude algebra vždy znamenat asociativní algebru s jednotkou, tzn. algebra obsahuje prvek 1 takový, že $1a = a1 = a$ pro všechna $a \in A$.

Dále je třeba znát pojem *bilineární funkce* pro definování *tenzorového součinu*, který se používá v definici Hopfovy algebry.

Definice 3.1.7: Nechť C a D jsou dva K -moduly a $C \times D$ je jejich kartézský součin jako množin.

K -bilineární funkce h na $C \times D$ do K -modulu E je taková funkce $h: C \times D \rightarrow E$, že vždy platí:

- (1) $h(\lambda_1 c_1 + \lambda_2 c_2, d) = \lambda_1 h(c_1, d) + \lambda_2 h(c_2, d)$,
- (2) $h(c, \lambda_1 d_1 + \lambda_2 d_2) = \lambda_1 h(c, d_1) + \lambda_2 h(c, d_2)$.

Rovnost (1) tvrdí, že $\forall d \in D$, parciální funkce $c \rightarrow h(c, d)$ je K -lineární zobrazení $C \rightarrow E$. Podobně rovnost (2).

Příklad: Bilineární funkce je skalární součin dvou vektorů (x_1, x_2) a (y_1, y_2) v rovině \mathbf{R}^2 . Je definován jako číslo:

$$(x_1, x_2)(y_1, y_2) = x_1 y_1 + x_2 y_2.$$

Pro pevné y je tento výraz \mathbf{R} -lineární pro vektor (x_1, x_2) , dále pro pevné x je \mathbf{R} -lineární pro vektor (y_1, y_2) . Z toho plyne, že skalární součin je \mathbf{R} -bilineární funkce $\mathbf{R}^2 \times \mathbf{R}^2 \rightarrow \mathbf{R}$.

◆

Tenzorový součin dvou modulů A a B nad komutativním okruhem K je okruh $A \otimes B$ nad K spolu s bilineární funkcí $\otimes: A \times B \rightarrow A \otimes B$, který má následující vlastnost univerzality.

Věta 3.1.8: Ke každé K - bilineární funkci h existuje právě jedna K - lineární transformace t s vlastností $t(a \otimes b) = h(a, b)$, kde $h: A \times B \rightarrow C$ a $t: A \otimes B \rightarrow C$.

Konstrukce tenzorového součinu a důkaz této věty viz [1].

Příklad: Necht' M je konečná množina $M = \{x_1, x_2, \dots, x_n\}$. Bud' $A = F(M)$ množina všech komplexních funkcí na M . Ztotožníme:

$$f, g \in A, f \equiv (f(x_1), f(x_2), \dots, f(x_n)), g \equiv (g(x_1), g(x_2), \dots, g(x_n)).$$

A je komplexní vektorový prostor, $\dim A = n$. Dále A je algebra s operací násobení

$$(fg)(x) = f(x)g(x).$$

Můžeme ztotožnit:

$$A \otimes A \equiv F(M \times M).$$

Potom pro $(f, g) \in A \times A$, $(f, g) \rightarrow f \otimes g = h \in A \otimes A$, kde $h(x, y) = f(x)g(y)$ pro $x, y \in M$.

♦

3.2 Koalgebra, bialgebra a Hopfova algebra

3.2.1 Algebra

Algebra byla již zavedena v definici 3.1.6. Nyní je možné tuto definici přeformulovat pomocí tenzorového součinu. V sekci 3.2.2. využijeme této nové formulace při zavedení koalgebry.

Definice 3.2.1: Algebra (asociativní algebra s jednotkou) je vektorový prostor A nad komutativním okruhem K s jednotkou, spolu se dvěma lineárními zobrazeními:

$m: A \otimes A \rightarrow A$ nazveme *násobení* a $\eta: K \rightarrow A$ nazveme *jednotka*, takovými že:

$$\begin{aligned} m \circ (m \otimes id) &= m \circ (id \otimes m), \\ m \circ (\eta \otimes id) &= id = m \circ (id \otimes \eta). \end{aligned}$$

Tzn. pro každé $a, b, c \in A$ platí:

$$m \circ (m \otimes id) (a \otimes b \otimes c) = m(m(a \otimes b) \otimes id(c)) = m((ab) \otimes c) = (ab)c. \text{ Podobně:}$$

$$m \circ (id \otimes m) (a \otimes b \otimes c) = m(id(a) \otimes m(b \otimes c)) = m(a \otimes (bc)) = a(bc).$$

Dále:

Nechť $e \in A$ je jednotkový prvek, tzn. $\eta(1) = e$ a $\eta(\lambda) = \lambda e$, kde $\lambda \in K$, potom:

$$\begin{aligned} m \circ (\eta \otimes id)(\lambda \otimes a) &= m(\eta(\lambda) \otimes id(a)) = m(\lambda e \otimes a) = (\lambda e)a = \lambda a, \\ m \circ (id \otimes \eta)(a \otimes \lambda) &= m(id(a) \otimes \eta(\lambda)) = m(a \otimes \lambda e) = a(\lambda e) = \lambda a. \end{aligned}$$

Přitom ztotožňujeme $K \otimes A \equiv A$, $\lambda \otimes a \equiv \lambda a$. Podobně pro $A \otimes K$.

Z uvedeného je zřejmé, že obě definice algebry jsou ekvivalentní. Asociativita násobení m znamená, že následující diagram komutuje.

$$\begin{array}{ccc} A \otimes A \otimes A & \xrightarrow{m \otimes id} & A \otimes A \\ \downarrow id \otimes m & & \downarrow m \\ A \otimes A & \xrightarrow{m} & A \end{array}$$

obrázek 1: Diagram 1 – algebra

Podobně jednotka může být vyjádřena komutováním následujícího diagramu.

$$\begin{array}{ccccc} K \otimes A & \xrightarrow{\eta \otimes id} & A \otimes A & \xleftarrow{id \otimes \eta} & A \otimes K \\ \downarrow id & & \downarrow m & & \downarrow id \\ K \otimes A & \Leftrightarrow & A & \Leftrightarrow & A \otimes K \end{array}$$

obrázek 2: Diagram 2 – algebra

Pozn. Symbol \Leftrightarrow znamená ztotožnění $K \otimes A$ a $A \otimes K$ s A .

Příklad: Nechť A je algebra. Potom $(a, b) \rightarrow ab \equiv m_A(a \otimes b)$.

Dále $A \otimes A$ je také algebra s operací násobení

$$(a_1 \otimes b_1, a_2 \otimes b_2) \rightarrow (a_1 \otimes b_1)(a_2 \otimes b_2) = a_1 a_2 \otimes b_1 b_2 \equiv m_{A \otimes A}((a_1 \otimes b_1) \otimes (a_2 \otimes b_2)).$$

Důkaz:

Ukažme, že $m_{A \otimes A} = (m_A \otimes m_A) \circ (id \otimes \tau \otimes id)$, kde $\tau: A \otimes A \rightarrow A \otimes A: a \otimes b \rightarrow b \otimes a$.

$$\begin{aligned} (a_1 \otimes b_1) \otimes (a_2 \otimes b_2) &\equiv a_1 \otimes b_1 \otimes a_2 \otimes b_2 \xrightarrow{id \otimes \tau \otimes id} a_1 \otimes a_2 \otimes b_1 \otimes b_2 \\ &\xrightarrow{m_A \otimes m_A} a_1 a_2 \otimes b_1 b_2. \quad \blacklozenge \end{aligned}$$

Definice: Necht' A a B jsou algebry. K -lineární zobrazení $\varphi: A \rightarrow B$ nazveme homomorfismus algeber, jestliže $\varphi(ab) = \varphi(a)\varphi(b)$ pro všechna $a, b \in A$ a $\varphi(1_A) = \varphi(1_B)$. Jinak napsáno:

$$\varphi \circ m_A = m_B \circ (\varphi \otimes \varphi) \text{ a } \varphi \circ \eta_A = \eta_B.$$

3.2.2 Koalgebra

K definici algebry můžeme zavést duální pojem tím, že obrátíme všechny šipky v diagramech a nahradíme všechna zobrazení příslušnými duálními zobrazeními. To znamená, že násobení m nahradíme *konásobením* Δ a jednotku η nahradíme *kojedinotkou* ε . Diagramy 1 a 2 potom přejdou na následující diagramy 3 a 4.

$$\begin{array}{ccc} A \otimes A \otimes A & \xleftarrow{\Delta \otimes id} & A \otimes A \\ \uparrow id \otimes \Delta & & \uparrow \Delta \\ A \otimes A & \xleftarrow{\Delta} & A \end{array}$$

$$\begin{array}{ccccc} K \otimes A & \xleftarrow{\varepsilon \otimes id} & A \otimes A & \xrightarrow{id \otimes \varepsilon} & A \otimes K \\ \uparrow id & & \uparrow \Delta & & \uparrow id \\ K \otimes A & \Leftrightarrow & A & \Leftrightarrow & A \otimes K \end{array}$$

obrázek 3: Diagram 3, 4 – koalgebra

Nyní můžeme napsat definici *koalgebry*.

Definice 3.2.2: Koalgebra je vektorový prostor A nad komutativním okruhem K s jednotkou a spolu se dvěma lineárními zobrazeními:

$\Delta: A \rightarrow A \otimes A$ nazvaném *konásobením* a $\varepsilon: A \rightarrow K$ nazvaném *kojedinotka*, takovými že:

$$\begin{aligned} (\Delta \otimes id) \circ \Delta &= (id \otimes \Delta) \circ \Delta, \\ (\varepsilon \otimes id) \circ \Delta &= id = (id \otimes \varepsilon) \circ \Delta. \end{aligned}$$

První rovnost lze přepsat takto: je-li $a \in A$ a $\Delta a = \sum_j a_j^1 \otimes a_j^2 \in A \otimes A$, potom

$$\sum_j \Delta(a_j^1) \otimes a_j^2 = \sum_j a_j^1 \otimes \Delta(a_j^2) \in A \otimes A \otimes A.$$

Druhá rovnost se dá zapsat obdobně jako u definice algebry.

Definice: Necht' A a B jsou koalgebry. K - lineární zobrazení $\varphi: A \rightarrow B$ nazveme homomorfismus koalgeber jestliže:

$$\Delta_B \circ \varphi = (\varphi \otimes \varphi) \circ \Delta_A \quad \text{a} \quad \varepsilon_A = \varepsilon_B \circ \varphi.$$

Příklad: Necht' G je konečná grupa. Vektorový prostor funkcí $A = F(G)$ je koalgebra. Ztotožníme-li

$$A \otimes A \equiv F(G \times G),$$

potom $\Delta: F(G) \rightarrow F(G \times G)$, $f \in F(G)$, $\Delta(f)(g_1, g_2) = f(g_1 g_2)$. Dále necht' $e \in G$ je jednotkový prvek, potom $\varepsilon: A \rightarrow K$, $\varepsilon(f) = f(e)$. ♦

Necht' A je koalgebra. Obdobně jako u algebry, značíme $\Delta_A: A \rightarrow A \otimes A$.

Navíc $A \otimes A$ je také koalgebra.

Bud'te $a, b \in A$, $a \otimes b \in A \otimes A$,

$$\begin{aligned} \Delta_A a &= \sum_j a_j^1 \otimes a_j^2 \in A \otimes A \\ \Delta_A b &= \sum_k b_k^1 \otimes b_k^2 \in A \otimes A. \end{aligned}$$

Zavádíme $\Delta_{A \otimes A}(a \otimes b) \in (A \otimes A) \otimes (A \otimes A)$ vztahem:

$$\Delta_{A \otimes A}(a \otimes b) = \sum_j \sum_k (a_j^1 \otimes b_k^1) \otimes (a_j^2 \otimes b_k^2).$$

Důkaz:

Ukažme, že $\Delta_{A \otimes A} = (id \otimes \tau \otimes id) \circ (\Delta_A \otimes \Delta_A)$.

$\Delta_A a \otimes \Delta_A b = \sum_j (a_j^1 \otimes a_j^2) \otimes \sum_k (b_k^1 \otimes b_k^2) = \sum_j \sum_k a_j^1 \otimes a_j^2 \otimes b_k^1 \otimes b_k^2$ a po použití operátoru τ na přehození pořadí (a_j^2, b_k^1) dostáváme ověřovanou rovnost. ♦

3.2.3 Bialgebra

Bialgebra je algebra a koalgebra, kde obě tyto struktury jsou rovnocenné v následujícím slova smyslu:

Lemma 3.2.3: Jestliže A je vektorový prostor, který je algebra i koalgebra, potom následující dvě podmínky jsou ekvivalentní:

- (1) $\Delta: A \rightarrow A \otimes A$ a $\varepsilon: A \rightarrow K$ je homomorfismus algebry
- (2) $m: A \otimes A \rightarrow A$ a $\eta: K \rightarrow A$ je homomorfismus koalgebry.

Podmínka (1) znamená:

$$\Delta \circ m = m_{A \otimes A} \circ (\Delta \otimes \Delta), \Delta \circ \eta = \eta_{A \otimes A}, \varepsilon \circ m = m_K \circ (\varepsilon \otimes \varepsilon), \varepsilon \circ \eta = \eta_K,$$

dále podmínka (2) znamená:

$$\Delta \circ m = (m \otimes m) \circ \Delta_{A \otimes A}, \varepsilon_{A \otimes A} = \varepsilon \circ m, \Delta \circ \eta = (\eta \otimes \eta) \circ \Delta_K, \varepsilon_K = \varepsilon \circ \eta.$$

$$\text{Zde: } \varepsilon_K = \eta_K = id_K, \quad \eta_{A \otimes A} = (\eta \otimes \eta) \circ \Delta_K, \quad \varepsilon_{A \otimes A} = m_K \circ (\varepsilon \otimes \varepsilon).$$

Definice 3.2.3: *Bialgebra* je vektorový prostor, který je současně algebra i koalgebra a takový, že lemma 2.2.3 je splněno. Tzn. jestliže A je algebra i koalgebra, potom A je bialgebra tehdy a pouze tehdy, jestliže pro každé $a, b \in A$ platí:

$$\Delta(ab) = \Delta(a) \Delta(b), \varepsilon(ab) = \varepsilon(a) \varepsilon(b), \Delta(1) = 1 \otimes 1, \varepsilon(1) = 1.$$

Příklad: Viz příklad v následující kapitole.

Homomorfismem bialgebry nazýváme zobrazení, které je homomorfismus algebry i koalgebry.

Nyní se dostáváme k základnímu pojmu v teorii kvantových grup a tím je pojem Hopfovy algebry.

3.2.4 Hopfova algebra

Definice 3.2.4: Bialgebru A nazýváme *Hopfovou algebrou*, jestliže existuje lineární zobrazení $S: A \rightarrow A$, které nazýváme *antipod*, takové že:

$$m \circ (S \otimes id) \circ \Delta = \eta \circ \varepsilon = m \circ (id \otimes S) \circ \Delta.$$

Tato rovnice se dá popsat tím, že následující diagram komutuje.

$$\begin{array}{ccccc} A \otimes A & \xleftarrow{\Delta} & A & \xrightarrow{\Delta} & A \otimes A \\ \downarrow id \otimes S & & \downarrow \eta \circ \varepsilon & & \downarrow S \otimes id \\ A \otimes A & \xrightarrow{m} & A & \xleftarrow{m} & A \otimes A \end{array}$$

obrázek 4: Diagram 5 – Hopfova algebra

Lemma 3.2.4: Antipod S v Hopfově algebře A je anti-homomorfismus v algebře a v koalgebře. Tzn. že platí:

$$S(ab) = S(b)S(a), \quad \forall a, b \in A \quad \text{a} \quad S(1) = 1,$$

$$\Delta \circ S = \tau \circ (S \otimes S) \circ \Delta \quad \text{a} \quad \varepsilon \circ S = \varepsilon,$$

kde τ značí operátor přehození tzn. $\tau(v \otimes w) = w \otimes v$.

Důkaz tohoto lemma je uveden v [2].

Tvrzení: Jestliže pro nějakou bialgebru existuje antipod S , pak je určen jednoznačně.

Důkaz tohoto tvrzení je uveden v [2].

Příklad:

Nechť G je konečná grupa, dále necht' $F(G)$ je algebra všech komplexních funkcí na G s příslušnými algebraickými operacemi. Vzhledem k vlastnostem grupy G můžeme definovat následující zobrazení:

- konásobení $\Delta: F(G) \rightarrow F(G \times G)$, $f \in F(G)$, $\Delta(f)(g_1, g_2) = f(g_1 g_2)$,
- kojednotka $\varepsilon: F(G) \rightarrow \mathbf{C}$, $\varepsilon(f) = f(e)$,
- antipod $S: F(G) \rightarrow F(G)$, $(S(f))(g) := f(g^{-1})$.

Zde e značí jednotkový prvek grupy G . Snadno lze ověřit, že Δ a ε jsou homomorfismy algebry (tzn. algebra $F(G)$ spolu s konásobením a kojednotkou tvoří koalgebru a dále i bialgebru). Pro splnění axiomů Hopfovy algebry musí být splněny následující identity:

1. Asociativita konásobení: $(\Delta \otimes id) \circ \Delta = (id \otimes \Delta) \circ \Delta$, kde id značí identické zobrazení na $F(G)$. Z definice konásobení plyne:

$$(((\Delta \otimes id) \circ \Delta) f)(g_1, g_2, g_3) = f((g_1 g_2) g_3) \in F(G \times G \times G),$$

$$(((id \otimes \Delta) \circ \Delta) f)(g_1, g_2, g_3) = f(g_1 (g_2 g_3)) \in F(G \times G \times G).$$

2. Dále relace $f(eg) = f(ge) = f(g)$ může být zapsána jako:

$$(\varepsilon \otimes id) \circ \Delta = id = (id \otimes \varepsilon) \circ \Delta.$$

3. Mějme zobrazení:

$$m: F(G \times G) \rightarrow F(G), (m(h))(g) := h(g, g), \text{ kde } h \in F(G \times G),$$

$$\eta: \mathbf{C} \rightarrow F(G), \text{ kde } \eta(1) \text{ je jednotkový prvek v algebře } F(G).$$

Potom axiom grupy: $g^{-1}g = gg^{-1} = e$, který zavádí inverzní prvek g^{-1} k prvku g , můžeme zapsat pomocí následující identity:

$$m \circ (S \otimes id) \circ \Delta = \eta \circ \varepsilon = m \circ (id \otimes S) \circ \Delta.$$

Skutečně, vzhledem k definicím zobrazení m , Δ a S dostáváme:

$$((m \circ (id \otimes S) \circ \Delta) f)(g) = (((id \otimes S) \circ \Delta) f)(g, g) = f(gg^{-1}) = f(e) = (\eta \circ \varepsilon)(f).$$

Z uvedeného dostáváme, že algebra $F(G)$ tvoří po přidání násobení a kojednotky koalgebru, dále pak bialgebru, po přidání antipodu i Hopfovou algebru. ♦

4 q – CALCULUS

Teorie kvantových grup a jejich reprezentací je úzce spojena s takzvaným q-calculem. V této kapitole jsou uvedeny základní pojmy kolem tohoto tématu. V příloženém programu je potom vypracováno několik algebraických identit na toto téma. Algoritmy jsou popsány v kapitole 8.3.1.

Definice 4.1: Necht' $q \in \mathbb{C}$, $q \neq 0$. q - číslo n je definováno následujícím způsobem:

$$[n]_q = \frac{1-q^n}{1-q} = 1 + q + q^2 + \dots + q^{n-1}, \text{ kde } n \in \mathbb{N}.$$

Vzorec 1: q – číslo n

q - faktoriál n je definován takto:

$$[n]_q! = [n]_q [n-1]_q [n-2]_q \dots [1]_q, \text{ kde } n \in \mathbb{N}.$$

Vzorec 2: q – faktoriál n

Dále výraz $(a; q)_n$ je definován:

$$(a; q)_n = (1-a)(1-aq)(1-aq^2) \dots (1-aq^{n-1}), \quad (a; q)_0 = 1.$$

Vzorec 3: Výraz $(a; q)_n$

Příklad:

$$[0]_q = 1$$

$$[1]_q = 1$$

$$[2]_q! = [2]_q [1]_q = (1+q)$$

$$[3]_q! = [3]_q [2]_q [1]_q = (1+q+q^2)(1+q) \quad \blacklozenge$$

Definice 4.2: Necht' $q \in \mathbb{C}$, $q \neq 0$. q - binomický koeficient je definován:

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{[n]_q!}{[k]_q! [n-k]_q!} = \frac{(q; q)_n}{(q; q)_k (q; q)_{n-k}}, \text{ kde } n, k \in \mathbb{N}_0 \text{ a } n \geq k.$$

Vzorec 4: q – binomický koeficient

Podobně jako u normálních kombinačních čísel, platí i u těchto q - binomických koeficientů několik identit.

Pro kombinační čísla platí následující rovnost:

$$\binom{n+1}{m} = \binom{n}{m-1} + \binom{n}{m}.$$

Pro q - binomické koeficienty potom platí:

$$\begin{bmatrix} n+1 \\ m \end{bmatrix}_q = \begin{bmatrix} n \\ m-1 \end{bmatrix}_q + q^m \begin{bmatrix} n \\ m \end{bmatrix}_q = q^{n-m+1} \begin{bmatrix} n \\ m-1 \end{bmatrix}_q + \begin{bmatrix} n \\ m \end{bmatrix}_q.$$

Hlavní význam těchto koeficientů je podobně jako u kombinačních čísel v zobecněné binomické větě.

Věta 4.1: Necht' x, y jsou nekomutativní proměnné splňující relaci $yx = qxy$. Potom platí:

$$(x+y)^n = \sum_{k=0}^n \begin{bmatrix} n \\ k \end{bmatrix}_q x^k y^{n-k} \quad \forall n \in \mathbb{N}.$$

Vzorec 5: q – binomická věta

Důkaz této věty indukcí podle n použitím předchozí identity.

Definice 4.3: *Hypergeometrická funkce* je definována následujícím vztahem:

$${}_2\phi_1(a, b; c; q, z) = \sum_{n=0}^{\infty} \frac{(a; q)_n (b; q)_n}{(c; q)_n} \frac{z^n}{(q; q)_n}.$$

Vzorec 6: Hypergeometrická funkce

Definice 4.4: *Malý q – Jacobi polynom* je definován pomocí hypergeometrické funkce vztahem:

$$p_n(x; a, b|q) = {}_2\phi_1(q^{-n}, abq^{n+1}; aq; q, xq).$$

Vzorec 7: Malý q – Jacobi polynom

5 KVANTOVÁ GRUPA $U_q(sl_2)$

Tato kapitola je věnována základnímu popisu Hopfovy algebry $U_q(sl_2)$. V příkladu uvedeném v části 3.2.4 jsme ukázali, že každé konečné grupě lze přiřadit Hopfovou algebru. Podobně některým Lieovým grupám, jako je například grupa $SL(2)$, lze přiřadit Hopfovou algebru. Přitom algebru $U_q(sl_2)$ lze chápat jako deformaci závisící na parametru q Hopfovy algebry přiřazené grupě $SL(2)$. Proto se pro tuto algebru zejména ve fyzikální literatuře často používá název kvantová grupa.

5.1 Algebra $U_q(sl_2)$

Definice 5.1: Necht' q je pevné komplexní číslo takové, že $q \neq 0$ a $q^2 \neq 1$. Označme $U_q(sl_2)$ algebru (tzn. asociativní algebru s jednotkou) nad \mathbf{C} s generátory E, F, K, K^{-1} splňujícími následující relace:

$$KK^{-1} = K^{-1}K = 1, \quad KEK^{-1} = q^2E, \quad KFK^{-1} = q^{-2}F,$$

$$[E, F] = \frac{K - K^{-1}}{q - q^{-1}}, \quad \text{kde } [E, F] = EF - FE.$$

Vzorec 8: Relace v algebře $U_q(sl_2)$ – varianta 1

Nezávislý parametr q komutuje se vším, tzn. např. $Eq = qE$, $qEqF = q^2EF$.

Věta 5.1: Soubor $\{K^m F^l E^n \mid m \in \mathbf{Z}, l, n \in \mathbf{N}_0\}$ stejně jako soubor $\{E^n F^l K^m \mid m \in \mathbf{Z}, l, n \in \mathbf{N}_0\}$, tvoří bázi $U_q(sl_2)$.

Důkaz této věty viz [2].

Věta 5.1 říká, že součin jakékoliv konečné posloupnosti generátorů $U_q(sl_2)$, lze převést na kanonický tvar, kterým je konečná lineární kombinace členů $E^n F^l K^m$. Jednou z částí vypracovaného programu, přiloženému k této práci, je právě vypracování tohoto převodu. Podrobněji o tomto v 8.3.2.

5.2 Hopfova algebra $U_q(sl_2)$

Hlavní vlastností algebry $U_q(sl_2)$ je, že zapadá do struktury Hopfovy algebry.

Věta 5.2: Existuje struktura Hopfovy algebry na $U_q(sl_2)$ s konásobením Δ , kojednotkou ε a antipodem S takovými, že:

$$\Delta(E) = E \otimes K + I \otimes E, \quad \Delta(F) = F \otimes I + K^{-1} \otimes F, \quad \Delta(K) = K \otimes K,$$

$$S(K) = K^{-1}, \quad S(E) = -EK^{-1}, \quad S(F) = -KF, \quad \varepsilon(K) = 1, \quad \varepsilon(E) = \varepsilon(F) = 0.$$

Vzorec 9: Konásobení, kojednotka, antipod v $U_q(sl_2)$ – varianta 1

Úplný důkaz této věty je opět uveden v [2]. Pro ukázkou rozebereme například poslední relaci v definici 5.1 algebry $U_q(sl_2)$. Tato relace stejně jako ostatní je splněna, zaměníme-li všechny generátory, které v ní vystupují, jejich obrazy vzhledem ke konásobení.

Důkaz:

$$\begin{aligned} [\Delta(E), \Delta(F)] &= K^{-1} \otimes EF + F \otimes E + EK^{-1} \otimes KF + EF \otimes K \\ &\quad - K^{-1} \otimes FE - K^{-1} E \otimes FK - F \otimes E - FE \otimes K = \\ &= \frac{K^{-1} \otimes (K - K^{-1}) + (K - K^{-1}) \otimes K}{q - q^{-1}} = \{\Delta(K) - \Delta(K^{-1})\} / (q - q^{-1}). \blacklozenge \end{aligned}$$

Definice 5.2: Algebru $U_q(sl_2)$ spolu se strukturou Hopfovy algebry z věty 5.2 nazýváme kvantovou grupou $U_q(sl_2)$.

Další varianta jak definovat kvantovou grupu je následující.

Nechť $q^2 \neq 1$ a označme $U_q(sl_2)$ algebru nad \mathbf{C} s generátory E, F, K, K^{-1} splňujícími tyto relace:

$$KK^{-1} = K^{-1}K = 1, \quad KEK^{-1} = qE, \quad KFK^{-1} = q^{-1}F,$$

$$[E, F] = \frac{K^2 - K^{-2}}{q - q^{-1}}, \quad \text{kde } [E, F] = EF - FE.$$

Vzorec 10: Relace v algebře $U_q(sl_2)$ – varianta 2

Podobně jako ve větě 5.2 potom z algebry $U_q(sl_2)$, vytvoříme Hopfovou algebru přidáním konásobení, kojednotky a antipodu.

$$\Delta(E) = E \otimes K + K^{-1} \otimes E, \quad \Delta(F) = F \otimes K + K^{-1} \otimes F, \quad \Delta(K) = K \otimes K,$$

$$S(K) = K^{-1}, \quad S(E) = -qE, \quad S(F) = -q^{-1}F, \quad \varepsilon(K) = 1, \quad \varepsilon(E) = \varepsilon(F) = 0.$$

Vzorec 11: Konásobení, kojednotka, antipod v $U_q(sl_2)$ – varianta 2

Výhodou této Hopfovy algebry je to, že konásobení E a F je definováno stejně.

6 KVANTOVÁ GRUPA $SL_q(2)$

V této kapitole je stručný popis Hopfovy algebry $SL_q(2)$. Později bude ukázáno, že tato algebra je vůči algebře $U_q(sl_2)$ duální ve smyslu duálního párování.

6.1 Algebra $SL_q(2)$

Definice 6.1: Necht' q je pevné komplexní číslo takové, že $q \neq 0$. Označme $SL_q(2)$ algebru (tzn. asociativní algebru s jednotkou) nad \mathbb{C} s generátory a, b, c, d splňujícími následující relace:

$$ab = qba, \quad ac = qca, \quad cd = qdc, \quad bc = cb,$$

$$ad - qbc = da - q^{-1}bc = 1.$$

Vzorec 12: Relace v algebře $SL_q(2)$

Nezávislý parametr q komutuje se vším, tzn. např. $aq = qa$, $qaqb = q^2ab$.

6.2 Hopfova algebra $SL_q(2)$

Algebra $SL_q(2)$ stejně jako algebra $U_q(sl_2)$, zapadá do struktury Hopfovy algebry.

Věta 6.2.1: Existuje struktura Hopfovy algebry na $SL_q(2)$ s konásobením Δ , kojednotkou ε a antipodem S takovými, že:

$$\Delta(a) = a \otimes a + b \otimes c, \quad \Delta(b) = a \otimes b + b \otimes d,$$

$$\Delta(c) = c \otimes a + d \otimes c, \quad \Delta(d) = c \otimes b + d \otimes d,$$

$$\varepsilon(a) = \varepsilon(d) = 1, \quad \varepsilon(b) = \varepsilon(c) = 0,$$

$$S(a) = d, \quad S(b) = -q^{-1}b, \quad S(c) = -qc, \quad S(d) = a.$$

Vzorec 13: Konásobení, kojednotka, antipod v $SL_q(2)$

Důkaz této věty je opět uveden v [2].

Vzhledem k definici relací v Hopfově algebře $SL_q(2)$ platí následující vztah:

Mějme dvě matice: $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ a $\mathbf{B} = \begin{pmatrix} d & -q^{-1}b \\ -qc & a \end{pmatrix}$.

Přímým výpočtem je možné ověřit, že jsou vzájemně inverzní.

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} d & -q^{-1}b \\ -qc & a \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Věta 6.2.2: Soubor $\{c^r b^m a^n, d^s c^r b^m \mid n, m, r, s \in \mathbf{N}_0\}$ stejně jako soubor $\{a^n b^m c^r, b^m c^r d^s \mid n, m, r, s \in \mathbf{N}_0\}$, tvoří bázi $SL_q(2)$.

Důkaz této věty viz [2].

Věta 6.2.2 říká, že součin jakékoliv konečné posloupnosti generátorů $SL_q(2)$, lze převést na kanonický tvar, kterým je konečná lineární kombinace členů $a^n b^m c^r$ a $b^m c^r d^s$. Tento převod je opět vypracován v příloženém programu, více viz 8.3.3.

7 REPREZENTACE, KOREPREZENTACE, DUÁLNÍ PÁROVÁNÍ

V této kapitole jsou uvedeny definice reprezentace a koreprezentace algeber a dále pak duálního párování. Na konci kapitoly je ukázáno, že algebry $U_q(sl_2)$ a $SL_q(2)$ jsou vůči sobě duální ve smyslu duálního párování.

7.1 Reprezentace

Definice 7.1: *Reprezentací* (maticovou) algebry A nazveme zobrazení

$$\tau: A \rightarrow \text{Mat}(n, \mathbf{C})$$

takové, že je splněna následující vlastnost:

$$\forall X, Y \in A, \quad \tau(XY) = \tau(X) \tau(Y), \quad \tau(e) = I, \text{ kde } e \text{ je jednotkový prvek.}$$

Z definice 7.1 je zřejmé, že reprezentace algebry je homomorfismus algebry.

Příklad:

Mějme algebru $U_q(sl_2)$ s generátory E, F, K, K^{-1} . Dále mějme zobrazení $\tau: U_q(sl_2) \rightarrow \text{Mat}(2, \mathbf{C})$, pro které platí:

$$(1) \quad \tau(E) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$$

$$(2) \quad \tau(F) = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}$$

$$(3) \quad \tau(K) = \begin{pmatrix} q^{\frac{1}{2}} & 0 \\ 0 & q^{\frac{1}{2}} \end{pmatrix}$$

$$(4) \quad \tau(K^{-1}) = \begin{pmatrix} q^{-\frac{1}{2}} & 0 \\ 0 & q^{-\frac{1}{2}} \end{pmatrix}$$

Ověříme, že tyto matice splňují relace, které byly uvedeny za definicí 5.2 a tím také, že takto definované zobrazení τ je reprezentací algebry $U_q(sl_2)$.

Důkaz: Algebra $U_q(sl_2)$ je dána relacemi:

$$KE = qEK, \quad KF = q^{-1}FK,$$

$$[E, F] = \frac{K^2 - K^{-2}}{q - q^{-1}}, \quad \text{kde } [E, F] = EF - FE.$$

Aplikováním zobrazení τ na jednotlivé relace dostáváme:

$$\tau(K)\tau(E) = q^{-\frac{1}{2}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \tau(E)\tau(K) = q^{\frac{1}{2}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = q\tau(K)\tau(E).$$

$$\tau(K)\tau(F) = q^{\frac{1}{2}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \quad \tau(F)\tau(K) = q^{-\frac{1}{2}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} = q^{-1}\tau(K)\tau(F).$$

$$\tau(E)\tau(F) - \tau(F)\tau(E) = - \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$\frac{1}{q - q^{-1}} (\tau(K)^2 - \tau(K)^{-2}) = \frac{1}{q - q^{-1}} \begin{pmatrix} q^{-1} - q & 0 \\ 0 & q - q^{-1} \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}. \blacklozenge$$

7.2 Koreprezentace

Definice 7.2: *Koreprezentací* koalgebry A na vektorovém prostoru V nazveme lineární zobrazení $\varphi: V \rightarrow V \otimes A$ splňující relace:

$$(\varphi \otimes id) \circ \varphi = (id \otimes \Delta) \circ \varphi, \quad (id \otimes \varepsilon) \circ \varphi = id.$$

Dále necht' \mathbf{T} je matice, jejíž prvky jsou prvky koalgebry A . Pak řekneme, že matice \mathbf{T} je koreprezentací koalgebry A , pokud platí následující relace:

$$\Delta \mathbf{T} = \mathbf{T} \otimes \mathbf{T}.$$

Tato relace se dá po jednotlivých prvcích matice zapsat následujícím způsobem:

Necht' $t_{ij} \in \mathbf{T}$, potom matice \mathbf{T} $n \times n$ je koreprezentací koalgebry A tehdy a jen tehdy, když

$$\Delta t_{ij} = \sum_{k=1}^n t_{ik} \otimes t_{kj} \quad \text{a} \quad \varepsilon(t_{ij}) = \delta_{ij}, \quad i, j = 1, 2, \dots, d.$$

Vztah mezi zobrazením φ a maticí \mathbf{T} je tento. Buď $V = \mathbb{C}^n$. Zobrazení φ stačí předeepsat na vektorech standardní báze (V_1, V_2, \dots, V_n) . Pokládáme $\varphi(V_i) = \sum_{j=1}^n V_j \otimes t_{ji}$. Snadno lze ověřit, že je-li \mathbf{T} koreprezentace, potom φ je rovněž koreprezentace ve smyslu definice 7.2.

Příklad:

Mějme Hopfovou algebru $SL_q(2)$. Dále mějme matici \mathbf{T} jejímiž prvky jsou generátory algebry $SL_q(2)$.

$$\mathbf{T} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Potom matice \mathbf{T} je koreprezentací Hopfovy algebry $SL_q(2)$.

Důkaz: Konásobení na $SL_q(2)$ je definováno:

$$(1) \quad \Delta(a) = a \otimes a + b \otimes c$$

$$(2) \quad \Delta(b) = a \otimes b + b \otimes d$$

$$(3) \quad \Delta(c) = c \otimes a + d \otimes c$$

$$(4) \quad \Delta(d) = c \otimes b + d \otimes d$$

Dle definice koreprezentace:

$$\Delta t_{ij} = \sum_{k=1}^n t_{ik} \otimes t_{kj}$$

je zřejmé, že matice \mathbf{T} splňuje tuto podmínku a je tedy skutečně koreprezentací Hopfovy algebry $SL_q(2)$. ♦

7.2.1 Matice koreprezentace \mathbf{T}_n Hopfovy algebry $SL_q(2)$

Matici koreprezentace \mathbf{T} z předešlého příkladu je možné sestavit pro libovolnou dimenzi.

Nechť $l \in \frac{1}{2}\mathbb{N}_0$, tzn. $l = 0, \frac{1}{2}, 1, \frac{3}{2}, \dots$. Indexy prvků $t_{ij}^{(l)}$ matice \mathbf{T}_l volíme v rozsahu:

$$i, j = -l, -l+1, \dots, l-1, l.$$

Dimenze matice \mathbf{T}_l je tedy $2l + 1$.

Prvky matice \mathbf{T}_l se dají vyjádřit pomocí malého q – Jacobi polynomu definovaného v kapitole 4.

Necht' $\zeta = -qbc$, dále necht' $N_{ij}^l = q^{-(l+j)(j-i)} \begin{bmatrix} l+i \\ i-j \end{bmatrix}_{q^{-2}}^{1/2} \begin{bmatrix} l-j \\ i-j \end{bmatrix}_{q^{-2}}^{1/2}$.

Potom:

$$t_{ij}^{(l)} = N_{ij}^l a^{-i-j} c^{i-j} p_{l+j}(\zeta; q^{-2(i-j)}, q^{2(i+j)} | q^{-2}), \quad \text{kde } i+j \leq 0 \text{ a } i \geq j,$$

$$t_{ij}^{(l)} = N_{ji}^l a^{-i-j} b^{j-i} p_{l+i}(\zeta; q^{-2(j-i)}, q^{2(i+j)} | q^{-2}), \quad \text{kde } i+j \leq 0 \text{ a } j \geq i,$$

$$t_{ij}^{(l)} = N_{-i,-j}^l p_{l-j}(\zeta; q^{-2(j-i)}, q^{-2(i+j)} | q^{-2}) b^{j-i} d^{i+j}, \quad \text{kde } i+j \geq 0 \text{ a } j \geq i,$$

$$t_{ij}^{(l)} = N_{-j,-i}^l p_{l-i}(\zeta; q^{-2(i-j)}, q^{-2(i+j)} | q^{-2}) c^{i-j} d^{i+j}, \quad \text{kde } i+j \geq 0 \text{ a } i \geq j.$$

Vzorec 14: Koeficienty $t_{ij}^{(l)}$ matice koreprezentace

Příklad: Matice koreprezentace Hopfovvy algebry $SL_q(2)$.

1. $l = 0 \quad \mathbf{T}_0 = (I)$

2. $l = \frac{1}{2} \quad \mathbf{T}_{\frac{1}{2}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

3. $l = 1 \quad \mathbf{T}_1 = \begin{pmatrix} a^2 & (1+q^{-2})^{1/2} ab & b^2 \\ (1+q^{-2})^{1/2} ac & 1+(q+q^{-1})bc & (1+q^{-2})^{1/2} bd \\ c^2 & (1+q^{-2})^{1/2} cd & d^2 \end{pmatrix}$

Důkaz: Pro úplný důkaz, že daná matice je koreprezentací algebry $SL_q(2)$ by bylo třeba ověřit všechny prvky matice. Pro ukázkou důkazu se omezíme na první prvek a^2 .

Z definice konásobení v Hopfově algebře $SL_q(2)$:

$$\Delta(a) = a \otimes a + b \otimes c$$

$$\Delta(a^2) = \Delta(a)\Delta(a) = (a \otimes a + b \otimes c)(a \otimes a + b \otimes c) = a^2 \otimes a^2 + ba \otimes ca + ab \otimes ac + b^2 \otimes c^2$$

Po použití komutačních pravidel dostáváme:

$$\Delta(a^2) = a^2 \otimes a^2 + q^{-2}(ab \otimes ac) + ab \otimes ac + b^2 \otimes c^2$$

Na pravé straně definičního vztahu koreprezentace pro matici \mathbf{T}_1 dostáváme výraz:

$$a^2 \otimes a^2 + (1+q^{-2})^{1/2} ab \otimes (1+q^{-2})^{1/2} bc + b^2 \otimes c^2 = a^2 \otimes a^2 + (1+q^{-2})ab \otimes bc + b^2 \otimes c^2$$

Další prvky matice se dají ověřit obdobným způsobem. Jednou z částí vypracovaného programu je ověřování koreprezentačních matic \mathbf{T}_n . Více viz kapitola 8.3.3.2. ♦

7.3 Duální párování

Definice 7.3: Necht' U, A jsou dvě bialgebry. Duálním párováním nazveme bilineární zobrazení $\langle \bullet, \bullet \rangle : U \times A \rightarrow C$ takové, že platí:

$$(1) \quad \langle \Delta_U x, f \otimes g \rangle = \langle x, m_A(f, g) \rangle, \quad \forall x \in U, \forall f, g \in A,$$

$$(2) \quad \langle x \otimes y, \Delta_A f \rangle = \langle m_U(x, y), f \rangle, \quad \forall x, y \in U, \forall f \in A,$$

$$(3) \quad \langle 1, f \rangle = \varepsilon_A(f) \text{ a } \langle x, 1 \rangle = \varepsilon_U(x) \quad \forall x \in U \text{ a } \forall f \in A.$$

Duální párování nazveme nedegenerované pokud navíc platí:

$$(4) \quad x \in U, \forall f \in A, \quad \langle x, f \rangle = 0 \Rightarrow x = 0$$

$$f \in A, \forall x \in U \quad \langle x, f \rangle = 0 \Rightarrow f = 0.$$

Příklad: V části 3.2.4 byl uveden příklad Hopfovy algebry $F(G)$. V tomto příkladu sestrojíme Hopfovou algebru $U(G)$ a dále pak duální párování mezi algebrami $F(G)$ a $U(G)$.

Necht' G je konečná grupa. Potom $U(G) = \left\{ \sum_{a \in G} \alpha_a a; \alpha_a \in C \right\}$, kde $G = \{a_0, a_1, \dots, a_{n-1}\}$ je báze $U(G)$ a $|G| = n$. Definujme příslušné operace:

- násobení - $\left(\sum_{a \in G} \alpha_a a \right) \left(\sum_{b \in G} \beta_b b \right) = \sum_{a, b \in G} \alpha_a \beta_b ab$
- jednotka - necht' e je jednotkový prvek v grupě G , potom e je jednotkou v $U(G)$.

Potom $U(G)$ s takto definovanými operacemi tvoří algebru. Dále vytvoříme z algebry $U(G)$ Hopfovou algebru přidáním konásobení, kojednotky a antipodu.

- kojednotka - $\varepsilon: U(G) \rightarrow C$ $\varepsilon \left(\sum_{a \in G} \alpha_a a \right) = \sum_{a \in G} \alpha_a$
- konásobení - $\Delta: U(G) \rightarrow U(G) \otimes U(G)$ $\Delta(a) = a \otimes a$
- antipod - $S: U(G) \rightarrow U(G)$ $S(a) = a^{-1}$.

S takto definovanými operacemi tvoří $U(G)$ Hopfovou algebru. Duální párování mezi Hopfovými algebrami $U(G)$ a $F(G)$ můžeme definovat následovně:

$$\langle \bullet, \bullet \rangle : U(G) \times F(G) \rightarrow \mathbf{C} \quad \langle a, f \rangle = f(a) \quad \varepsilon(a) = \langle a, \mathbf{1} \rangle = 1,$$

kde $a \in G$, $f \in F(G)$ a $\mathbf{1}$ je funkce identicky rovna 1. První axiom v definici duálního párování vypadá takto:

$$\langle \Delta_U x, f \otimes g \rangle = \langle x, m_A(f, g) \rangle.$$

Levá strana: $\langle \Delta(a), f \otimes g \rangle = \langle a \otimes a, f \otimes g \rangle = \langle a, f \rangle \langle a, g \rangle = f(a)g(a).$

Pravá strana: $\langle a, fg \rangle = f(a)g(a).$

Další axiomy se ověří podobně. ♦

Příklad: Na dalším příkladu duálního párování bude ukázána dualita mezi Hopfovými algebrami $U_q(sl_2)$ a $SL_q(2)$.

Věta 7.3.1: Existuje duální párování $\langle \bullet, \bullet \rangle$ mezi Hopfovými algebrami $U_q(sl_2)$ a $SL_q(2)$ takové, že:

$$\langle K, a \rangle = q^{-1/2}, \quad \langle K, d \rangle = q^{1/2}, \quad \langle E, c \rangle = \langle F, b \rangle = 1$$

$$\langle K, b \rangle = \langle K, c \rangle = \langle E, a \rangle = \langle E, b \rangle = \langle E, d \rangle = \langle F, a \rangle = \langle F, c \rangle = \langle F, d \rangle = 0.$$

Důkaz této věty viz [2]. ♦

Duální párování je jednoznačně určeno hodnotami na generátorech Hopfových algeber. Pro výpočet je však nutné znát hodnoty duálního párování na celých algebrách. Tyto hodnoty dává následující věta.

Věta 7.3.2: Duální párování na Hopfových algebrách $U_q(sl_2)$ a $SL_q(2)$ je dáno následujícím vzorcem:

$$\langle K^m E^n F^l, d^s c^r b^t \rangle = q^{(n-r)^2} \begin{bmatrix} s \\ n-r \end{bmatrix}_{q^2} \gamma_{mnl}^{srt}, \quad \text{pokud } 0 \leq n-r = l-t \leq s, \quad \text{jinak}$$

$$\langle K^m E^n F^l, d^s c^r b^t \rangle = 0 \text{ a } \langle K^m E^n F^l, a^s c^r b^t \rangle = \delta_{r,n} \delta_{t,l} \gamma_{mnl}^{-s,r,t}, \text{ kde}$$

$$\gamma_{mnl}^{srt} = \frac{q^{\frac{m(s+r-t)}{2}} q^{-\frac{s(n+l)}{2}} (q^2; q^2)_l (q^2; q^2)_n}{q^{\frac{n(n-1)}{2}} q^{\frac{l(l-1)}{2}} (1-q^2)^{l+n}}.$$

Vzorec 15: Duální párování na $U_q(sl_2)$ a $SL_q(2)$

Důkaz této věty viz [2]. ♦

Dualita Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$ dává do vztahu korepresentace $SL_q(2)$ a reprezentace $U_q(sl_2)$.

Nechť \mathbf{T}_n je koreprezentační matice Hopfovy algebry $SL_q(2)$ definovaná v kapitole 7.2.1.

Dále necht' operátor τ je operátor, který provede duální párování prvků z $U_q(sl_2)$ se všemi prvky koreprezentační matice \mathbf{T}_n . Výsledkem je komplexní matice.

Příklad: $\mathbf{T}_{\frac{1}{2}} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, potom

$$\tau(K) = \begin{pmatrix} q^{-1/2} & 0 \\ 0 & q^{1/2} \end{pmatrix}, \quad \tau(K^{-1}) = \begin{pmatrix} q^{1/2} & 0 \\ 0 & q^{-1/2} \end{pmatrix},$$

$$\tau(E) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad \tau(F) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}. \diamond$$

Dále algebra $U_q(sl_2)$ je dána relacemi:

$$KE = qEK, \quad KF = q^{-1}FK,$$

$$[E, F] = \frac{K^2 - K^{-2}}{q - q^{-1}}, \quad \text{kde } [E, F] = EF - FE.$$

Potom τ je reprezentace algebry, to znamená, že:

1. $\tau(K) \tau(E) = q \tau(E) \tau(K)$
2. $\tau(K) \tau(F) = q^{-1} \tau(F) \tau(K)$
3. $\tau(F) \tau(E) = \tau(E) \tau(F) - \frac{\tau(K) \tau(K)}{q - q^{-1}} + \frac{\tau(K^{-1}) \tau(K^{-1})}{q - q^{-1}}.$

Důkaz: Pro ukázkou se omezíme na nejsložitější třetí relaci.

Levá strana relace: $\tau(F)\tau(E) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$.

Pravá strana relace:

$$\begin{aligned} \tau(E)\tau(F) - \frac{\tau(K)\tau(K)}{q-q^{-1}} + \frac{\tau(K^{-1})\tau(K^{-1})}{q-q^{-1}} &= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} - \frac{1}{q-q^{-1}} \begin{pmatrix} q^{-1} & 0 \\ 0 & q \end{pmatrix} + \frac{1}{q-q^{-1}} \begin{pmatrix} q & 0 \\ 0 & q^{-1} \end{pmatrix} = \\ &= \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \frac{1}{q-q^{-1}} \begin{pmatrix} -q^{-1}+q & 0 \\ 0 & -q+q^{-1} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

Tím je třetí relace dokázána. První dvě relace je možné ověřit obdobně. ♦

Popsaná identita ukazuje vzájemný vztah mezi Hopfovými algebry $U_q(sl_2)$ a $SL_q(2)$. Korepresentace v algebře $SL_q(2)$ přechází pomocí duálního párování na reprezentaci v algebře $U_q(sl_2)$, dále pak konásobení v algebře $SL_q(2)$ přechází na násobení v algebře $U_q(sl_2)$.

Tato identita platí pro libovolnou koreprezentační matici algebry $SL_q(2)$ definovanou v kapitole 7.2.1. Dále identita platí pro libovolné prvky algebry $U_q(sl_2)$, pro důkaz je nutné ověřit identitu na generátorech algebry $U_q(sl_2)$, což je učiněno na předchozím příkladu. Ve vypracovaném programu je možné ověřovat libovolné prvky algebry $U_q(sl_2)$ s libovolnou koreprezentační maticí algebry $SL_q(2)$. Více o tomto v kapitole 8.3.4.

8 VYPRACOVANÝ PROGRAM

Jak již bylo mnohokrát zmíněno, nedílnou součástí této práce je vypracování programu, který ověřuje několik vybraných identit. V první části této kapitoly jsou shrnuty technické parametry programu a jeho instalace na počítač. V další jsou potom podrobněji popsány samotné algoritmy.

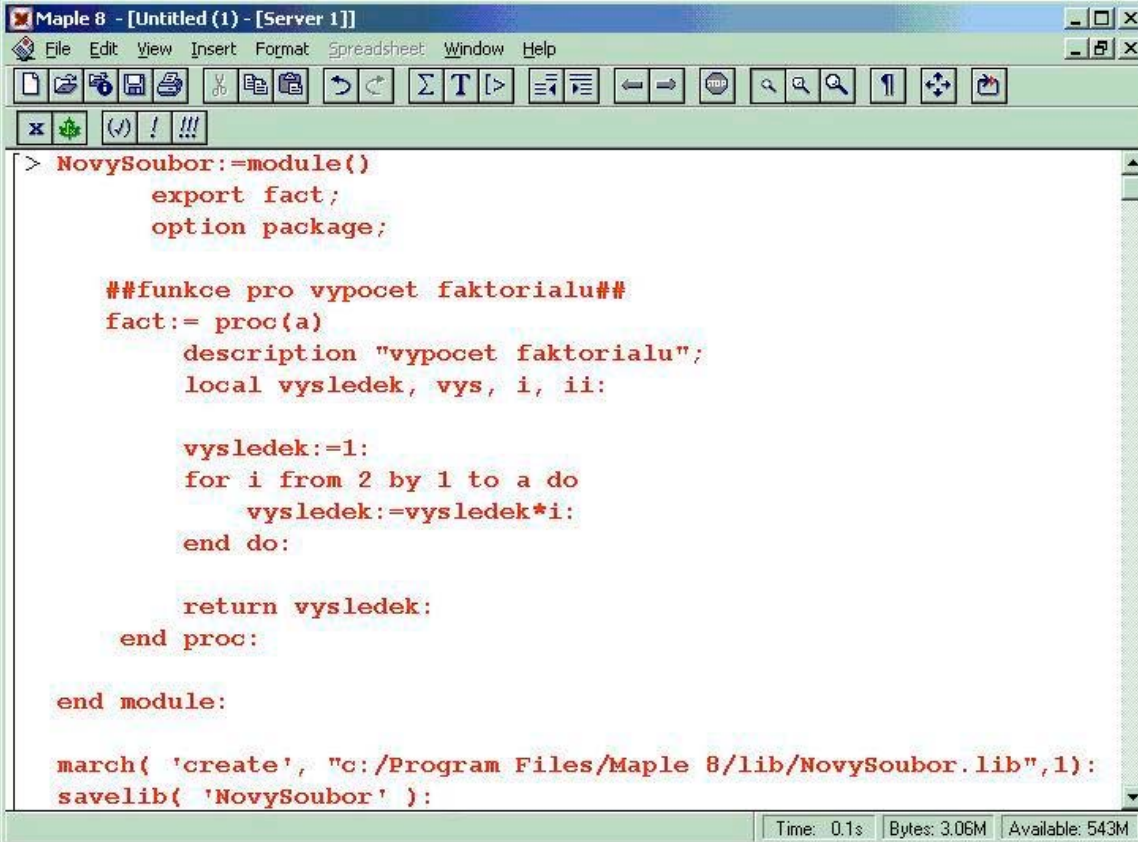
8.1 Technické parametry vypracovaného programu

V současné době existuje mnoho různých vývojových prostředí, které jsou určeny pro vývoj matematických aplikací a řešení matematických problémů, například MathLab, Mathematica a Maple.

Program k této práci byl vyvíjen v prostředí Maplu verze 8. Tato verze je podporována prostředím Microsoft Windows, i prostředím Linux.

Maple umožňuje vytvářet takzvané moduly (neboli hlavičkové soubory), to znamená, že lze vytvářet sadu funkcí, které lze potom využívat na libovolném místě v programu.

Příklad: Vytvoření hlavičkového souboru obsahujícího definice a implementace funkcí.



```
> NovySoubor := module()
  export fact;
  option package;

  ##funkce pro vypocet faktorialu##
  fact := proc(a)
    description "vypocet faktorialu";
    local vysledek, vys, i, ii;

    vysledek := 1;
    for i from 2 by 1 to a do
      vysledek := vysledek * i;
    end do;

    return vysledek;
  end proc;

end module;

march( 'create', "c:/Program Files/Maple 8/lib/NovySoubor.lib", 1):
savelib( 'NovySoubor' );
```

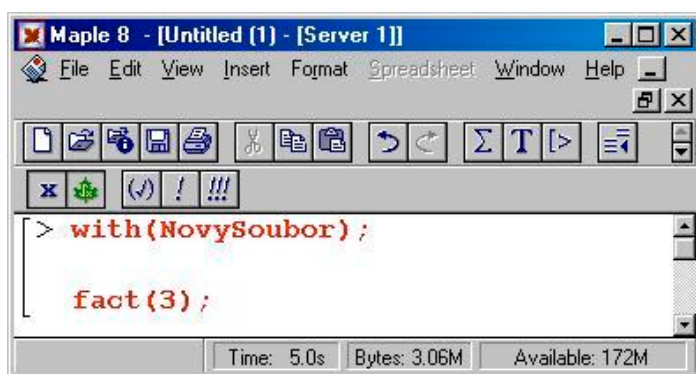
obrázek 5: Vytvoření hlavičkového souboru v Maple

Na obrázku je vytvořen jednoduchý modul s názvem „NovySoubor“, který obsahuje pouze jednu funkci, která počítá faktoriál zadaného čísla. ♦

Význam jednotlivých příkazů pro vytvoření modulu je následující:

Příkaz	Popis
„module“	Klíčový příkaz pro tvorbu modulu. Znamená, že „NovySoubor“ je modul.
„export“	Za tímto příkazem následují jména funkcí, které potom bude možné v jiných programech používat (v našem případě se jedná pouze o funkci „fact“).
„option“	Parametr „package“ za příkazem „option“ znamená, že „Novysoubor“ bude modul, který bude možné používat v libovolném programu.
„march“	Tímto příkazem (s parametrem „create“) se vytváří knihovny pro práci s modulem „NovySoubor“. Poslední parametr příkazu „march“ (v našem případě je to číslo „1“) je počet funkcí, které jsou za příkazem „export“.
„savelib“	Tímto příkazem se modul uloží pod jménem „NovySoubor“. S tímto jménem bude možné v dalších programech modul volat.

Využití vytvořeného modulu je znázorněno na následujícím obrázku.



obrázek 6: Vložení hlavičkového souboru do programu

Příkazem „with(NovySoubor)“ se zpřístupní veškeré funkce, které modul „NovySoubor“ obsahuje. Pak je možné jednotlivé funkce volat, jak je zobrazeno na obrázku. V našem případě funkce „fact(3)“ vypočítá faktoriál čísla 3. Více o modulech v systému Maple, viz nápověda k tomuto systému.

8.2 Instalace programu

Na přiloženém CD se nacházejí následující adresáře:

- *Documentation* - obsahuje dokumentaci k diplomové práci v elektronické podobě
- *Program* - obsahuje vypracovaný program, je dále rozdělený na další podadresáře
 - *Algebra UqSL2* - obsahuje soubory týkající se algebry $U_q(sl_2)$
 - *Algebra SLq2* - obsahuje soubory týkající se algebry $SL_q(2)$
 - *q - calculus* - obsahuje soubory týkající se q - calculu
 - *Pairing* - obsahuje soubory týkající se duálního párování.

Adresáře nacházející se v adresáři *Program* je nutné zkopírovat na disk. Dále je nutné zavést hlavičkové soubory do Maplu, aby mohli být použity. Celý program obsahuje tři hlavičkové soubory, které se nacházejí na následujících cestách:

- *Program \ Algebra UqSL2 \ HopfAlgebra.mws* - hlavičkový soubor pro práci s algebrou $U_q(sl_2)$
- *Program \ Algebra SLq2 \ DualAlgebra.mws* - hlavičkový soubor pro práci s algebrou $SL_q(2)$
- *Program \ q - calculus \ q-calculus.mws* - hlavičkový soubor pro práci s q - calculem.

Všechny tyto soubory je nejprve nutné otevřít v Maplu a změnit parametr příkazu „march“ viz kapitola 8.1 obrázek č. 5. Jako parametr tohoto příkazu je nutné zadat cestu, kde se nacházejí knihovny Maplu. Příkaz „march“ se nachází na konci všech třech uvedených souborů. Standardně je v nich zadaná následující cesta:

c:\Program Files \ Maple 8 \ lib \ DualAlgebra.lib

Po úpravě cesty a spuštění všech třech souborů jsou hlavičkové soubory do Maplu zavedeny a je možné veškeré funkce z nich použít příkazem „with“ v libovolném programu viz kapitola 8.1 obrázek č. 6. Dále je možné spustit zbývající soubory:

- *Program \ Algebra UqSL2 \ HopfIdentity.mws* - identita týkající se algebry $U_q(sl_2)$, využívá funkce z *HopfAlgebra.mws*.
- *Program \ Algebra SLq2 \ DualIdentity.mws* - identita týkající se algebry $SL_q(2)$, využívá funkce z *DualAlgebra.mws*.

- *Program \ q - calculus \ qIdentity.mws* - identita týkající se q - calculu, využívá funkce z *q-calculus.mws*.
- *Program \ Pairing \ pairing.mws* - identita týkající se duálního párování, využívá funkce z *HopfAlgebra.mws* a *DualAlgebra.mws*.

8.3 Popis funkcí z hlavičkových souborů a vypracované identity

Program je rozdělen na čtyři části. První se týká platných identit, které jsou popsány v kapitole 4 q - Calculus, druhá se pak týká Hopfovy kvantové algebry $U_q(sl_2)$ popsané v kapitole 5, další část je o Hopfově algebře $SL_q(2)$ popsané v kapitole 6. Poslední částí je propojení Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$ pomocí duálního párování popsáno v kapitole 7.

8.3.1 q - Calculus

V této kapitole jsou popsány identity a algoritmy související s q-calculem. Na příloženém CD se tato část programu nachází v adresáři *Program \ q - calculus*. Soubor *q-calculus.mws* obsahuje definice a deklarace funkcí, je tedy třeba pustit nejdříve tento soubor, což bylo učiněno v rámci instalace programu popsané v kapitole 8.2. Další soubor *qIdentity.mws* potom obsahuje vypracovanou identitu, ve které se využívají funkce z *q-calculus.mws*.

V celé této kapitole budeme předpokládat, že x, y jsou nekomutativní proměnné splňující následující relaci:

$$yx = qxy.$$

Věta 8.3.1: Součin libovolné konečné posloupnosti generátorů x a y lze upravit na tvar:

$$q^n x^k y^l, \quad \text{kde } k, l \in \mathbb{N} \text{ a } n \in \mathbb{Z}.$$

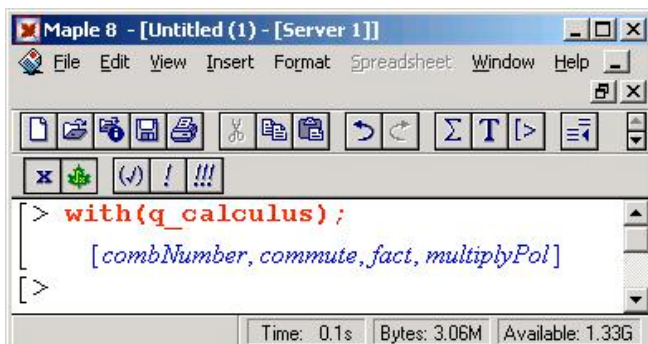
Důkaz této věty indukci pomocí předchozí relace.

Příklad: $xyyx = q^3 x^2 y^2$. ♦

Důsledkem věty 8.3.1 je, že libovolný prvek nekomutativní algebry generované x a y je roven konečné lineární kombinaci prvků $x^k y^l$ s koeficienty závisujícími na q . Navíc platí, že tato lineární kombinace je určena jednoznačně až na pořadí sčítanců (i tuto nejednoznačnost lze vyloučit, pokud uspořádáme prvky $x^k y^l$). Takovouto lineární kombinaci budeme nazývat kanonickým tvarem. Můžeme říci, že každý prvek algebry lze převést na kanonický tvar.

8.3.1.1 Popis funkcí z *q-calculus.mws*

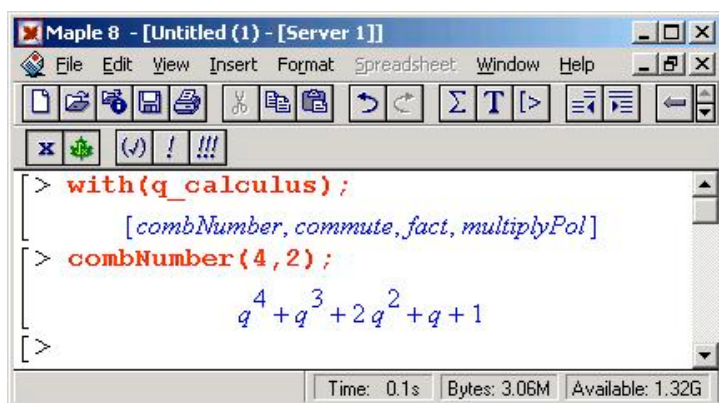
Po vložení hlavičkového souboru *q-calculus.mws* pomocí příkazu „with(q_calculus)“ je možné použít následující funkce:



obrázek 7: Funkce z *q-calculus.mws*

1. „combNumber“

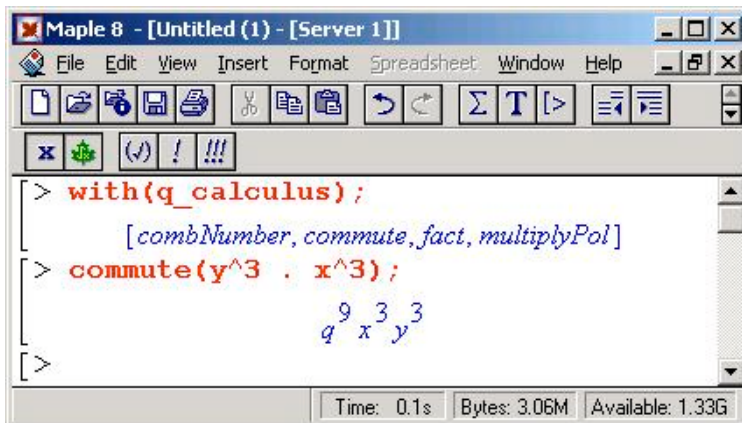
Funkce *combNumber* (n, k) počítá q -binomický koeficient $\begin{bmatrix} n \\ k \end{bmatrix}_q$ z definice 4.2.



obrázek 8: Použití funkce „combNumber“

2. „commute“

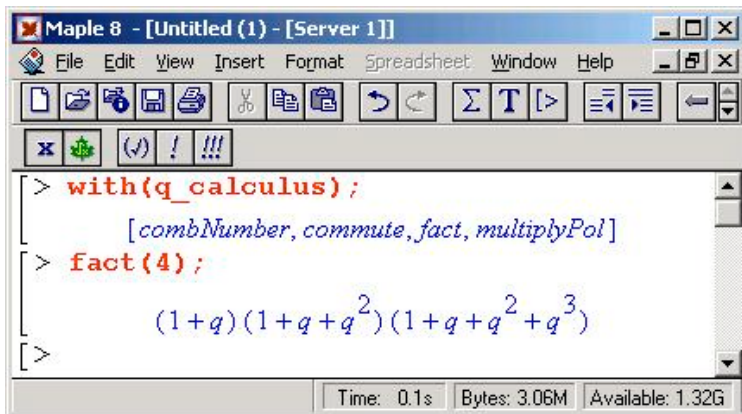
Funkce *commute*(*polynom*) převádí libovolnou posloupnost $y^j x^k$ na kanonický tvar $q^n x^k y^j$ viz věta 8.3.1.



obrázek 9: Použití funkce „commute“

3. „fact“

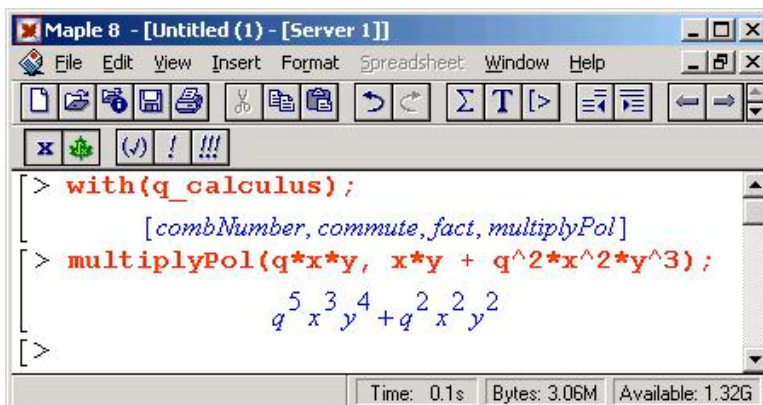
Funkce $fact(n)$ počítá $[n]_q!$ z definice 4.1.



obrázek 10: Použití funkce „fact“

4. „multiplyPol“

Funkce $multiplyPol(polynom1, polynom2)$ násobí nekomutativně dva polynomy.



obrázek 11: Použití funkce „multiplyPol“

Pro pochopení funkcí „commute“ a „multiplyPol“ je nutné znát alespoň základní informace o nekomutativním násobení v Maplu.

Mějme tedy relaci $yx = qxy$. V Maplu je nekomutativní násobení značeno tečkou „. \cdot “. Tedy chceme-li vyjádřit nekomutativnost, píšeme místo normální hvězdičky „*“ užívané pro násobení tečku „. \cdot “.

Příklad: Normální násobení v Maplu vypadá takto: $x * y$
 Zatímco nekomutativní násobení napíšeme takto: $x . y$ ♦

Pozn.

U nekomutativního násobení je třeba psát mezi písmena a operátor „. \cdot “ mezery, aby se tečka nebrala jako desetinné místo.

Příklad: (použití funkce „commute“)

Funkce *commute*($y . x . y . x$) vrátí $q^3 x^2 y^2$, kde u výrazu $q^3 x^2 y^2$ je už nekomutativní násobení nahrazeno násobením normálním, aby se dalo s takto upraveným polynomem dále dobře pracovat. Funkce „commute“ tedy převádí libovolnou posloupnost y a x na kanonický tvar. Dále tato funkce dokáže převést na kanonický tvar libovolný polynom v nekomutativních proměnných, tzn. například:

$$\text{commute}(y . x . y . x + y . x) \text{ vrátí } q^3 x^2 y^2 + q x y. \blacklozenge$$

Jak název naznačuje, funkce „multiplyPol“ násobí dva polynomy.

Příklad: (použití funkce „multiplyPol“)

Chceme nekomutativně vynásobit například polynomy $y . x + y^2 . x$ a $y + x . y . x$. Vzhledem k relaci $y . x = qxy$ by tedy výsledek měl být:

$$\begin{aligned} (y . x + y^2 . x) (y + x . y . x) &= y . x . y + y . x^2 . y . x + y^2 . x . y + y^2 . x^2 . y . x = \\ &= qxy^2 + q^4 x^3 y^2 + q^2 x y^3 + q^7 x^3 y^3. \end{aligned}$$

Funkci „multiplyPol“ můžeme použít takto:

$$\text{multiplyPol}(y . x + y^2 . x, y + x . y . x) \text{ vrátí } qxy^2 + q^4 x^3 y^2 + q^2 x y^3 + q^7 x^3 y^3. \blacklozenge$$

Tímto jsou všechny funkce z hlavičkového souboru *q-calculus.mws* popsány. Můžeme se tedy podívat do souboru *qIdentity.mws*, kde jsou tyto funkce použity k ověření několika ukázkových identit.

8.3.1.2 Vypracované identity využívající *q-calculus.mws*

Soubor *qIdentity.mws* na začátku obsahuje příkaz *with(q_calculus)*, kterým zpřístupníme použití funkcí z *q-calculus.mws*.

Dále obsahuje tři další funkce, kterými jsou „binom“, „leftExponen“ a „rightExponen“. O těchto funkcích později. Zmáčknutím klávesy Enter na začátku programu všechny tyto funkce zpřístupníme pro jejich použití, zároveň se po stisku Enter dostaneme na další část programu, což je první vypracovaná identita. Samotný kód vypadá takto:

```

[combNumber, commute, fact, multiplyPol]
> vyraz:=0:
n:=3:
for i from 0 by 1 to n do
  vyraz:=vyraz + combNumber(n,i)*(x^i)*(y^(n-i)):
end do:
print(vyraz);

vyraz2:=x+y:
for i from 1 by 1 to n-1 do
  vyraz2:=multiplyPol(vyraz2,x+y):
end do:
print(vyraz2);

vyraz - vyraz2;

```

$$y^3 + (1+q+q^2)xy^2 + (1+q+q^2)x^2y + x^3$$

$$x^3 + (1+q+q^2)x^2y + (1+q+q^2)xy^2 + y^3$$

0

obrázek 12: První identita v *qIdentity.mws*

Tento kód ověřuje správnost zobecněné binomické věty 4.1. První část kódu (první *for* cyklus) je vlastně pouze přepsání věty 4.1. Do proměnné *vyraz* se ve *for* cyklu postupně sčítají hodnoty $combNumber(n, i) * (x^i) * (y^{n-i})$ pro *i* od 0 do zadaného *n*.

Hodnoty $combNumber(n, i) * (x^i) * (y^{n-i})$ znamenají $\begin{bmatrix} n \\ i \end{bmatrix}_q x^i y^{n-i}$.

Potom se pomocí příkazu *print(vyraz)* tento polynom vypíše.

Druhý *for* cyklus počítá to samé, ale pomocí funkce *multiplyPol(vyraz2, x+y)*. Ve výrazu *vyraz2* jsou postupně mocniny *x+y*. Příkaz *print(vyraz2)* opět vypíše polynom *vyraz2* na obrazovku.

Poslední řádek *vyraz - vyraz2* nám vypíše na obrazovku 0, což znamená, že oba výrazy jsou stejné a tím je zobecněná binomická věta ověřena.

Poslední řádek celého souboru *qIdentity.mws* vypadá takto:

$$leftExponen(7) - rightExponen(7).$$

Výstupem tohoto řádku je opět 0, což značí rovnost obou funkcí. Obě tyto funkce počítají levou, resp. pravou stranu zobecněné q-exponenciály.

Zobecněná q-exponenciální funkce $e_q(z)$ je definována následujícím způsobem:

$$e_q(z) = \sum_{n=0}^{\infty} q^{\alpha_n} \frac{z^n}{[n]_q!}.$$

Vzorec 16: Zobecněná q - exponenciální funkce

Stejně jako u normální exponenciální funkce i u této zobecněné q-exponenciály platí vztah:

$$e_q(x) e_q(y) = e_q(x + y),$$

kde násobení $e_q(x) e_q(y)$ je opět nekomutativní, tedy platí $yx = qxy$.

Podívejme se podrobněji co předchozí rovnost znamená. Tečky na místech, kde je násobení opět znamenají nekomutativní násobení $y \cdot x = qxy$.

$$e_q(x) e_q(y) = \left(\sum_{m=0}^{\infty} \frac{q^{\alpha_m}}{[m]_q!} x^m \right) \left(\sum_{n=0}^{\infty} \frac{q^{\alpha_n}}{[n]_q!} y^n \right) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \frac{q^{\alpha_m + \alpha_n}}{[m]_q! [n]_q!} x^m \cdot y^n = \sum_{k=0}^{\infty} \sum_{m=0}^k \frac{q^{\alpha_m + \alpha_{k-m}}}{[m]_q! [k-m]_q!} x^m \cdot y^{k-m}$$

$$e_q(x + y) = \sum_{k=0}^{\infty} \frac{q^{\alpha_k}}{[k]_q!} (x + y)^k = \sum_{k=0}^{\infty} \left(\frac{q^{\alpha_k}}{[k]_q!} \sum_{m=0}^k \frac{[k]_q!}{[m]_q! [m-k]_q!} x^m \cdot y^{k-m} \right) = \sum_{k=0}^{\infty} \sum_{m=0}^k \frac{q^{\alpha_k}}{[m]_q! [m-k]_q!} x^m \cdot y^{k-m}$$

Z uvedeného plyne, že rovnost $e_q(x) e_q(y) = e_q(x + y)$ nastává pro $\alpha_m + \alpha_{k-m} = \alpha_k$. Tzn. aby platila rovnost, musí platit:

$$\forall m, n \quad \alpha_m + \alpha_n = \alpha_{m+n}.$$

Tato rovnost je splněna pro $\alpha_m = c m$.

Poslední identita ve vypracovaném programu ověřuje právě tuto rovnost. Podívejme se znovu, jak vypadá:

$$\text{leftExponen}(7) - \text{rightExponen}(7).$$

Funkce $\text{leftExponen}(M)$ počítá $e_q(x) e_q(y)$, zatímco $\text{rightExponen}(M)$ počítá $e_q(x + y)$. Dále je nutné uvést, co znamená parametr M v obou funkcích. Vzhledem k tomu, že sumy na obou stranách rovnosti jsou nekonečné, je třeba pro naprogramování takovýchto sum určit nějaká pravidla. Parametr M tedy určuje do jakého řádu se budou sumy počítat. Ve funkci $\text{rightExponen}(M)$ je tento parametr velice jednoduchý, prostě se provede výpočet pravé strany rovnosti, tzn.:

$$\sum_{k=0}^M \frac{q^{ck}}{[k]_q!} (x + y)^k, \text{ kde výraz } (x + y)^k \text{ se počítá pomocí zobecněné binomické věty.}$$

U funkce $\text{leftExponen}(M)$ je výpočet o něco složitější. Nejprve je třeba obě sumy navzájem nekomutativně vynásobit a pak teprve se vynechávají členy řádu vyššího než M .

$$\left(\sum_{m=0}^M \frac{q^{cm}}{[m]_q!} x^m \right) \left(\sum_{n=0}^M \frac{q^{cn}}{[n]_q!} y^n \right).$$

Tyto sumy se vynásobí a pak se kontroluje zda jednotlivé členy nejsou řádu vyššího než M .

To jsou všechny vypracované identity týkající se kapitoly q-calculus.

8.3.2 Hopfova algebra $U_q(sl_2)$

Na přiloženém CD v adresáři *Program \ Algebra UqSL2* jsou dva soubory. První z nich je hlavičkový soubor obsahující deklarace a implementace použitých funkcí. Tento soubor se jmenuje *HopfAlgebra.mws*. V dalším souboru s názvem *HopfIdentity.mws* je potom vypracována identita používající funkce z hlavičkového souboru.

Mějme Hopfovu kvantovou algebru $U_q(sl_2)$ z kapitoly 5. Tzn. máme čtyři generátory E, F, K, K^{-1} splňující tyto relace:

$$KK^{-1} = K^{-1}K = 1, \quad KEK^{-1} = qE, \quad KFK^{-1} = q^{-1}F,$$

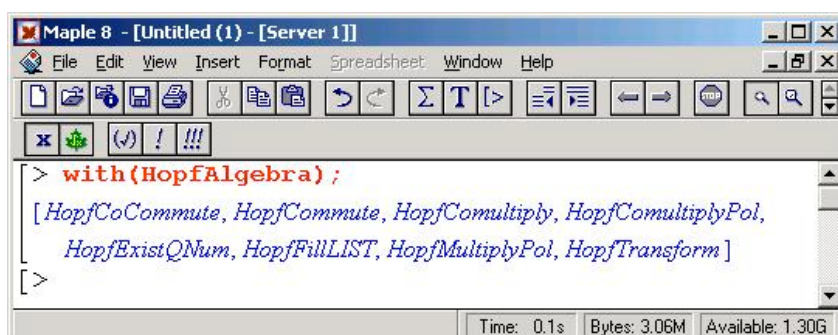
$$[E, F] = \frac{K^2 - K^{-2}}{q - q^{-1}}, \quad \text{kde } [E, F] = EF - FE,$$

$$\Delta(E) = E \otimes K + K^{-1} \otimes E, \quad \Delta(F) = F \otimes K + K^{-1} \otimes F, \quad \Delta(K) = K \otimes K,$$

$$S(K) = K^{-1}, \quad S(E) = -qE, \quad S(F) = -q^{-1}F, \quad \varepsilon(K) = 1, \quad \varepsilon(E) = \varepsilon(F) = 0.$$

8.3.2.1 Popis funkcí z *HopfAlgebra.mws*

Po vložení hlavičkového souboru *HopfAlgebra.mws* pomocí příkazu „with(HopfAlgebra)“ je možné použít následující funkce:



obrázek 13: Funkce z *HopfAlgebra.mws*

1. „HopfCoCommute“

Funkce *HopfCoCommute(Polynomial)* převádí polynom ve kterém je konásobení na kanonický tvar daný větou 5.1. Více viz níže.

```

> with(HopfAlgebra);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol,
  HopfExistQNum, HopfFillLIST, HopfMultiplyPol, HopfTransform ]
> HopfCoCommute(e . X . f . e + k . e . X . f);
(e . X . e . f) - \left( \frac{1}{q - \frac{1}{q}} \right) . e . X . (k^2) + \left( \frac{1}{q - \frac{1}{q}} \right) . e . X . \left( \frac{1}{k^2} \right) + (q . e . k . X . f)

```

obrázek 14: Použití funkce „HopfCoCommute“

2. „HopfComultiply“

Funkce *HopfComultiply(Polynomial)* provádí konásobení jednotlivých členů polynomu. Více viz níže.

```

> with(HopfAlgebra);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol, HopfExistQNum, HopfFillLIST,
  HopfMultiplyPol, HopfTransform ]
> HopfComultiply(e*f + e*k);
(e . f . X . (k^2)) + \left( e . \left( \frac{1}{k} \right) . X . k . f \right) + \left( \left( \frac{1}{k} \right) . f . X . e . k \right) + \left( \left( \frac{1}{k^2} \right) . X . e . f \right) + (e . k . X . (k^2)) + (e . k)

```

obrázek 15: Použití funkce „HopfComultiply“

3. „HopfComultiplyPol“

Funkce *HopfComultiplyPol(Polynomial1, Polynomial2)* provádí tenzorové násobení dvou polynomů. Více viz níže.

```

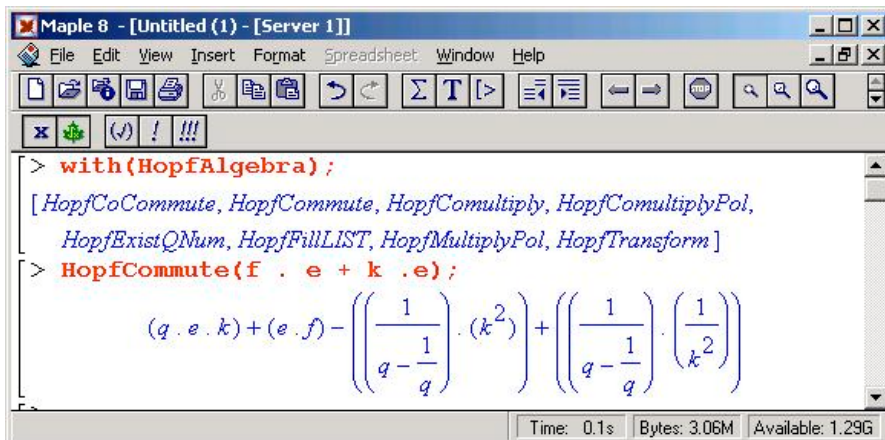
> with(HopfAlgebra);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol,
  HopfExistQNum, HopfFillLIST, HopfMultiplyPol, HopfTransform ]
> HopfComultiplyPol(e*f + e*k, e*f);
(e . f . X . e . f) + (e . k . X . e . f)

```

obrázek 16: Použití funkce „HopfComultiplyPol“

4. „HopfCommute“

Funkce $HopfCommute(Polynomial)$ převádí libovolný polynom na kanonický tvar daný větou 5.1. Více viz níže.



```

> with(HopfAlgebra);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol,
  HopfExistQNum, HopfFillLIST, HopfMultiplyPol, HopfTransform ]
> HopfCommute(f . e + k . e);

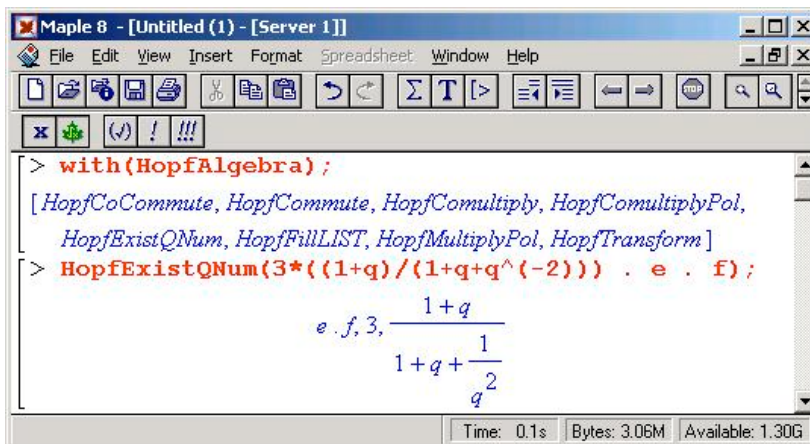
$$(q \cdot e \cdot k) + (e \cdot f) - \left( \left( \frac{1}{q - \frac{1}{q}} \right) \cdot (k^2) \right) + \left( \left( \frac{1}{q - \frac{1}{q}} \right) \cdot \left( \frac{1}{k^2} \right) \right)$$


```

obrázek 17: Použití funkce „HopfCommute“

5. „HopfExistQNum“

Funkce $HopfExistQNum(Element)$ rozděljuje výraz na čísla, parametr q a generátory algebry $U_q(sl_2)$. Více viz níže.



```

> with(HopfAlgebra);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol,
  HopfExistQNum, HopfFillLIST, HopfMultiplyPol, HopfTransform ]
> HopfExistQNum(3*((1+q)/(1+q+q^(-2))) . e . f);

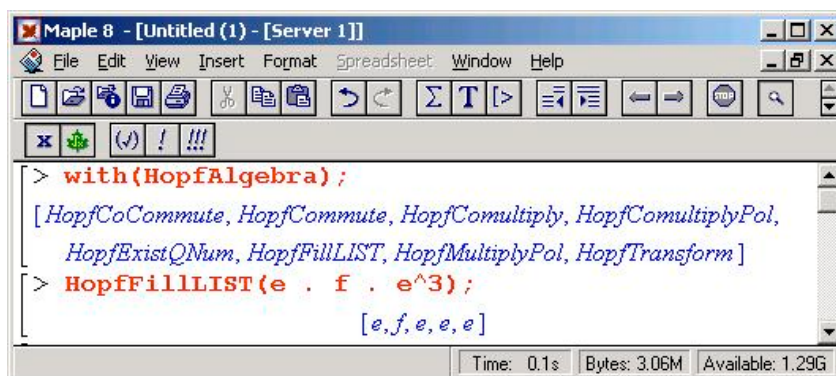
$$e \cdot f, 3, \frac{1+q}{1+q+\frac{1}{q^2}}$$


```

obrázek 18: Použití funkce „HopfExistQNum“

6. „HopfFillLIST“

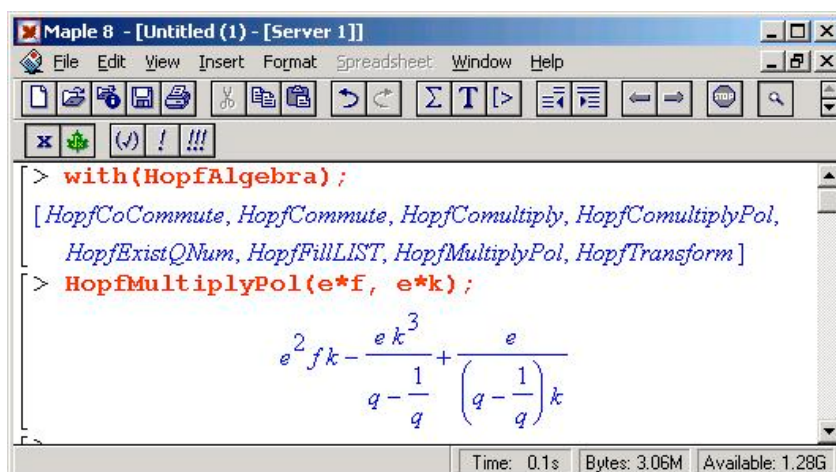
Funkce $HopfFillLIST(Element)$ naplní seznam generátory algebry $U_q(sl_2)$. Více viz níže.



obrázek 19: Použití funkce „HopfFillLIST“

7. „HopfMultiplyPol“

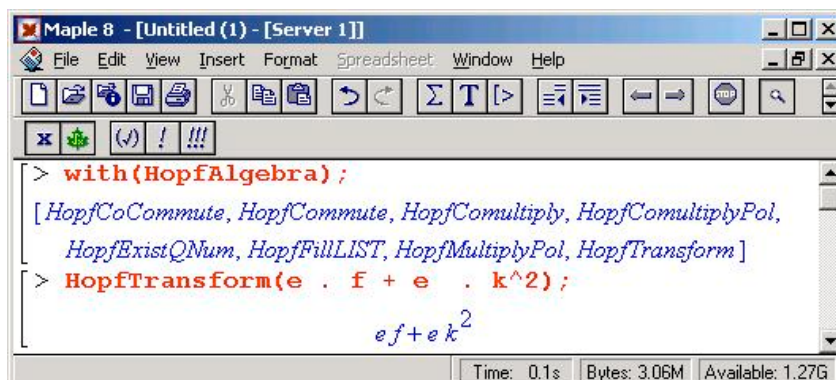
Funkce $HopfMultiplyPol(Polynomial, Polynomial)$ provádí nekomutativní násobení dvou polynomů. Více viz níže.



obrázek 20: Použití funkce „HopfMultiplyPol“

8. „HopfTransform“

Funkce $HopfTransform(Polynomial)$ převádí nekomutativní násobení na normální násobení. Více viz níže.



obrázek 21: Použití funkce „HopfTransform“

8.3.2.1.1 Podrobný popis funkcí

ad 1) „HopfCoCommute“

Funkce převádí polynom, ve kterém je konásobení, na kanonický tvar.

Příklad:

Mějme výraz $E \otimes FE + KE \otimes F$. Po použití relací v algebře $U_q(sl_2)$ dostáváme:

$$E \otimes FE + KE \otimes F = E \otimes EF - \left(\frac{1}{q - q^{-1}} E \otimes K^2 \right) + \left(\frac{1}{q - q^{-1}} E \otimes K^{-2} \right) + qEK \otimes F .$$

Tento výraz je v kanonickém tvaru daném větou 5.1.

Symbol „ \otimes “ je ve vypracovaném programu značen jako symbol „X“. Pokud tedy aplikujeme funkci „HopfCoCommute“ na výraz $E \otimes FE + KE \otimes F$ dostaneme výraz v kanonickém tvaru. Postup použití funkce „HopfCoCommute“ je zobrazen na obrázku 14. Symbol „.“ značí v Maplu nekomutativní násobení. ♦

ad 2) „HopfComultiply“

Funkce *HopfComultiply(Polynomial)* provádí konásobení jednotlivých členů polynomu.

Příklad:

Máme následující relace:

$$\Delta(E) = E \otimes K + K^{-1} \otimes E, \quad \Delta(F) = F \otimes K + K^{-1} \otimes F \quad \text{a} \quad \Delta(K) = K \otimes K.$$

Potom $\Delta(EF) = EF \otimes K^2 + EK^{-1} \otimes KF + K^{-1}F \otimes EK + K^{-2} \otimes EF$ a $\Delta(EK) = EK \otimes K^2 + EK$. Funkce „HopfComultiply“ aplikuje konásobení na každý člen polynomu. Příklad je uveden na obrázku 15. Tento polynom však není v kanonickém tvaru. Například druhý člen polynomu $EK^{-1} \otimes KF$ by měl být napsán jako $q^{-1}EK^{-1} \otimes FK$. Pokud chceme dostat tento výsledek, tak je nutné na výraz získaný funkcí „HopfComultiply“ aplikovat funkci „HopfCoCommute“. Potom máme výsledek konásobení v kanonickém tvaru. Celý postup je zobrazen na následujícím obrázku.

```

> with(HopfAlgebra);

vyraz1:=e*f + e*k;

vyraz2:=HopfComultiply(vyraz1);

vyraz3:=HopfCoCommute(vyraz2);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol, HopfExistQNum, HopfFillLIST,
HopfMultiplyPol, HopfTransform ]

vyraz1 = e f + e k
vyraz2 = (e . f . X . (k^2)) + (e . (1/k) . X . k . f) + ((1/k) . f . X . e . k) + ((1/k^2) . X . e . f) + (e . k . X . (k^2)) + (e . k)
vyraz3 =
(e . f . X . (k^2)) + ((1/q) . e . (1/k) . X . f . k) + (q . f . (1/k) . X . e . k) + ((1/k^2) . X . e . f) + (e . k . X . (k^2)) + (e . k)

```

obrázek 22: Příklad konásobení členů polynomu pomocí funkce „HopfComultiply“ a „HopfCoCommute“

Funkce „HopfComultiply“ předpokládá, že vstupní polynom je již v kanonickém tvaru a je převeden na normální násobení, tzn. není v něm již obsaženo nekomutativní násobení „ . “, ale násobení normální „*“ . Pokud polynom není převeden na kanonický tvar, je nutné na něj nejprve aplikovat funkci „HopfCommute“ a poté funkci „HopfTransform“ . ♦

ad 3) „HopfComultiplyPol“

Funkce $HopfComultiplyPol(Polynomial1, Polynomial2)$ provádí tenzorové násobení dvou polynomů.

Příklad:

Mějme dva polynomy: $Polynom1 = EF + EK$, $Polynom2 = EF$. Potom jejich tenzorový součin vypadá takto:

$$Polynom1 \otimes Polynom2 = EF \otimes EF + EK \otimes EF.$$

Použití funkce „HopfComultiplyPol“ je zobrazeno na obrázku 16. Symbol „ \otimes “ je opět nahrazen symbolem „X“. Funkce „HopfComultiplyPol“ předpokládá, že vstupní polynomy jsou již v kanonickém tvaru a jsou převedeny na normální násobení, tzn. není v nich již obsaženo nekomutativní násobení „ . “, ale násobení normální „*“ . Pokud polynomy nejsou převedeny na kanonický tvar, je nutné na ně nejprve aplikovat funkci „HopfCommute“ a poté funkci „HopfTransform“ . ♦

ad 4) „HopfCommute“

Funkce je obdobná jako funkce „commute“ v kapitole q-calculus, její výstup však není převeden na normální násobení, ve výstupním polynomu zůstává i nadále násobení nekomutativní, tzn. symbol „.“ viz obrázek 17.

ad 5) „HopfExistQNum“

Funkce *HopfExistQNum(Element)* rozděluje výraz na čísla, parametr q a generátory algebry $U_q(sl_2)$.

Příklad:

Mějme výraz:
$$3 \frac{1+q}{1+q+\frac{1}{q^2}} . e . f .$$

Pro práci s tímto výrazem je dobré umět takovéto výrazy rozdělit na čísla, parametry q a generátory algebry $U_q(sl_2)$. Toto dělá funkce „HopfExistQNum“. Její použití je znázorněno na obrázku 18. Výsledkem funkce „HopfExistQNum“, aplikované na výraz uvedený výše je vektor, který má tři složky.

- První složka jsou generátory algebry $U_q(sl_2)$, v našem případě: $e . f$
- Druhá složka je číslo, v našem případě: 3
- Poslední složka je parametr q , v našem případě: $\frac{1+q}{1+q+\frac{1}{q^2}}$ ♦

ad 6) „HopfFillLIST“

Funkce *HopfFillLIST(Element)* naplní seznam generátory algebry $U_q(sl_2)$. Pro práci s nekomutativním násobením je nutné vědět jaké generátory daný výraz obsahuje a pro zachování nekomutativnosti je dále nutné vědět i pořadí jednotlivých generátorů v daném výrazu. Z tohoto důvodu je v celém vypracovaném programu pro práci s nekomutativním násobením používán seznam (LIST). Funkce „HopfFillLIST“ je stejně jako funkce „HopfExistQNum“ pomocná funkce usnadňující práci při programování dalších funkcí týkajících se algebry $U_q(sl_2)$. Použití této funkce je zobrazeno na obrázku 19. Funkce „HopfFillLIST“ předpokládá, že vstupní parametr je výraz obsahující pouze generátory algebry $U_q(sl_2)$, tzn. pouze E, F, K . Pokud tedy chceme zpracovat výraz uvedený v příkladu funkce „HopfExistQNum“, musíme do funkce „HopfFillLIST“ přenést pouze první složku výstupu funkce „HopfExistQNum“. Celý postup je uveden na následujícím obrázku.

The screenshot shows the Maple 8 interface with the following content:

```

> with(HopfAlgebra);

vyraz:=HopfExistQNum(3*((1+q)/(1+q+q^(-2))) . e . f);

HopfFillLIST(vyraz[1]);
[ HopfCoCommute, HopfCommute, HopfComultiply, HopfComultiplyPol, HopfExistQNum,
  HopfFillLIST, HopfMultiplyPol, HopfTransform ]

vyraz := e . f, 3,  $\frac{1+q}{1+q+\frac{1}{q^2}}$ 
[e, f]

```

At the bottom of the window, the status bar shows: Time: 0.1s, Bytes: 3.06M, Available: 1.25G.

obrázek 23: Použití funkcí „HopfExistQNum“ a „HopfFillLIST“

ad 7) „HopfMultiplyPol“

Funkce $HopfMultiplyPol(Polynomial, Polynomial)$ provádí nekomutativní násobení dvou polynomů. Příklad použití je znázorněn na obrázku 20. Funkce „HopfMultiplyPol“ předpokládá, že vstupní polynomy jsou již v kanonickém tvaru a jsou převedeny na normální násobení, tzn. není v nich již obsaženo nekomutativní násobení „.“, ale násobení normální „*“. Pokud polynomy nejsou převedeny na kanonický tvar, je nutné na ně nejprve aplikovat funkci „HopfCommute“ a poté funkci „HopfTransform“.

ad 8) „HopfTransform“

Funkce „HopfTransform“ převádí nekomutativní násobení na normální násobení. Tzn. pokud máme polynom v kanonickém tvaru, tzn. byla na něj aplikována funkce „HopfCommute“, je možné upustit od nekomutativního násobení „.“ a nahradit ho násobením normálním „*“ pro jednodušší práci s polynomem. Použití funkce je znázorněno na obrázku 21.

8.3.2.2 Vypracovaná identita využívající *HopfAlgebra.mws*

Soubor *HopfIdentity.mws* obsahuje vypracovanou identitu týkající se Hopfovy algebry $U_q(sl_2)$. Na začátku tohoto souboru jsou vloženy dva hlavičkové soubory:

- $with(HopfAlgebra)$
- $with(LinearAlgebra)$

Vložení souboru *HopfAlgebra* zpřístupní použití funkcí definovaných v 8.3.2.1. Hlavičkový soubor *LinearAlgebra* je standardně implementován v Maplu a slouží například k práci s maticemi, které bude identita využívat.

Dále soubor *HopfIdentity.mws* obsahuje funkci *MultiplyMatrix(A, B, n)*. Tato funkce provádí nekomutativní násobení matic. První dva parametry A, B jsou matice, parametr n je jejich řád, jedná se tedy pouze o čtvercové matice řádu n . Tato funkce využívá funkci *HopfMultiplyPol* z kapitoly 8.3.2.1. Další funkce *MultiplyTensor(A, B)* je pouze pomocná funkce, která násobí matice tenzorovým způsobem.

Tenzorový součin matic \mathbf{A}, \mathbf{B} je matice $\mathbf{A} \otimes \mathbf{B}$ zapsaná v blokovém tvaru, přičemž (i, j) -tý blok je roven matici $a_{ij} \mathbf{B}$. Tenzorový součin matic je dobře definován nejen pro čtvercové matice, ale i pro matice libovolných rozměrů. Je zřejmé, že má-li matice \mathbf{A} rozměr $k \times l$ a matice \mathbf{B} rozměr $m \times n$, pak matice $\mathbf{A} \otimes \mathbf{B}$ má rozměr $km \times ln$.

Podívejme se nyní jakou identitu soubor *HopfIdentity.mws* ověřuje. Mějme čtyři generátory Hopfovy algebry $U_q(sl_2)$ E, F, K, K^{-1} splňující relace z kapitoly 5 a nezávislý parametr q , který komutuje se vším.

Definujme následující matice:

$$R = \begin{pmatrix} q & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & q - q^{-1} & 1 & 0 \\ 0 & 0 & 0 & q \end{pmatrix}, \text{ potom tedy } R^{-1} = \begin{pmatrix} q^{-1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -q + q^{-1} & 1 & 0 \\ 0 & 0 & 0 & q^{-1} \end{pmatrix}.$$

$$L = \begin{pmatrix} k & -\sqrt{q}(q - q^{-1})e \\ 0 & k^{-1} \end{pmatrix}, \quad L^* = \begin{pmatrix} k & 0 \\ -\sqrt{q}(q - q^{-1})f & k^{-1} \end{pmatrix}.$$

Dále můžeme spočítat další matice: „ \otimes “ značí tenzorový součin matic a $I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

$$L_1 = I_2 \otimes L = \begin{pmatrix} k & -\sqrt{q}(q - q^{-1})e & 0 & 0 \\ 0 & k^{-1} & 0 & 0 \\ 0 & 0 & k & -\sqrt{q}(q - q^{-1})e \\ 0 & 0 & 0 & k^{-1} \end{pmatrix}, \quad L_1^* = I_2 \otimes L^*,$$

$$L_2 = L \otimes I_2 = \begin{pmatrix} k & 0 & -\sqrt{q}(q - q^{-1})e & 0 \\ 0 & k & 0 & -\sqrt{q}(q - q^{-1})e \\ 0 & 0 & k^{-1} & 0 \\ 0 & 0 & 0 & k^{-1} \end{pmatrix}, \quad L_2^* = L^* \otimes I_2.$$

Potom platí následující vztahy:

- 1.) $R \cdot L_2 \cdot L_1 = L_1 \cdot L_2 \cdot R$
- 2.) $R \cdot L_1^* \cdot L_2^* = L_2^* \cdot L_1^* \cdot R$
- 3.) $L_2^* \cdot R^{-1} \cdot L_1 = L_1 \cdot R^{-1} \cdot L_2^*$

Symbol „ \cdot “ značí opět nekomutativní násobení matic, tzn. násobení matic je definováno následujícím vztahem:

$$(A \cdot B)_{ij} = \sum_{k=1}^n A_{ik} \cdot B_{kj}.$$

Uvedené vztahy jsou definiční vztahy mezi generátory Hopfovy algebry $U_q(sl_2)$ zapsány pomocí maticových identit.

Jako ověření platnosti těchto vztahů slouží identita ze souboru *HopfIdentity.mws*. Vraťme se zpět k tomuto souboru. Jak bylo již uvedeno, na začátku jsou vloženy hlavičkové soubory *HopfAlgebra* a *LinearAlgebra*. Potom následují pomocné funkce *MultiplyMatrix(A, B, n)* a *MultiplyTensor(A, B)*. Dále jsou deklarovány příslušné matice:

$$R := \langle\langle q, 0, 0, 0 \rangle \mid \langle 0, 1, q \cdot q^{-1}, 0 \rangle \mid \langle 0, 0, 1, 0 \rangle \mid \langle 0, 0, 0, 0, q \rangle\rangle:$$

$$L := \langle\langle k, 0 \rangle \mid \langle -q^{1/2} \cdot (q \cdot q^{-1}) \cdot e, k^{-1} \rangle\rangle:$$

$$LL := \langle\langle k, -q^{1/2} \cdot (q \cdot q^{-1}) \cdot f \rangle \mid \langle 0, k^{-1} \rangle\rangle:$$

$$\text{inv}R := \text{MatrixInverse}(R):$$

$$II := \langle\langle 1, 0 \rangle \mid \langle 0, 1 \rangle\rangle:$$

Kde: LL značí matici L^* , $\text{inv}R$ matice inverzní k R , tedy R^{-1} a II značí I_2 .

Pak následují matice vypočítané:

$$L1 := \text{MultiplyTensor}(II, L):$$

$$LL1 := \text{MultiplyTensor}(II, LL):$$

$$L2 := \text{MultiplyTensor}(L, II):$$

$$LL2 := \text{MultiplyTensor}(LL, II):$$

Kde: $LL1$... značí L_1^* , $LL2$... značí L_2^* a funkce *MultiplyTensor* počítá tenzorový součin matic.

Ted' máme všechny potřebné matice pro ověření vztahů 1) 2) a 3). Podívejme se jak vypadá toto ověření naprogramované. Všechny tři vztahy jsou podobné a jejich naprogramování je téměř totožné, stačí tedy ukázat jak je zapsán vztah 1).

```
xxx1:=MultiplyMatrix(R, L2, 4):
vysledek1:=MultiplyMatrix(xxx1, L1, 4):
```

```
xxx2:=MultiplyMatrix(L1, L2, 4):
vysledek2:=MultiplyMatrix(xxx2, R, 4):
```

```
vysledek1 - vysledek2;
```

Takto napsaný kód vypíše na obrazovku 0, což značí, že *vysledek1* a *vysledek2* jsou totožné. Tím je vztah 1) ověřený. U vztahu 3) je jediný rozdíl v zapsání rovnosti a to je, že poslední řádek vypadá takto:

```
simplify(vysledek5 - vysledek6);
```

Toto opět vypíše na obrazovku 0. Funkce *simplify* je standardní funkce Maplu. Jde o to, že rovnost 3) je o něco složitější než předchozí rovnosti. Maple výsledné výrazy automaticky nezjednodušuje do té míry, aby bylo možné rozpoznat, že jsou rovny nule. Proto je použita funkce *simplify*, která dané výrazy zjednoduší aplikováním pravidel pro krácení atd.

Tímto jsou všechny tři vztahy ověřeny.

8.3.3 Hopfova algebra $SL_q(2)$

Obdobně jako u Hopfovy algebry $U_q(sl_2)$ popsané v kapitole 8.3.2 jsou na CD v adresáři *Program \ Algebra SLq2* dva soubory. První z nich je hlavičkový soubor obsahující deklarace a implementace použitých funkcí. Tento soubor se jmenuje *DualAlgebra.mws*. V dalším souboru s názvem *DualIdentity.mws* je potom opět vypracována identita používající funkce z hlavičkového souboru.

Mějme Hopfovou kvantovou algebru $SL_q(2)$ z kapitoly 6. Tzn. máme čtyři generátory a, b, c, d splňující tyto relace:

$$ab = qba, \quad ac = qca, \quad cd = qdc, \quad bc = cb,$$

$$ad - qbc = da - q^{-1}bc = 1,$$

$$\Delta(a) = a \otimes a + b \otimes c, \quad \Delta(b) = a \otimes b + b \otimes d,$$

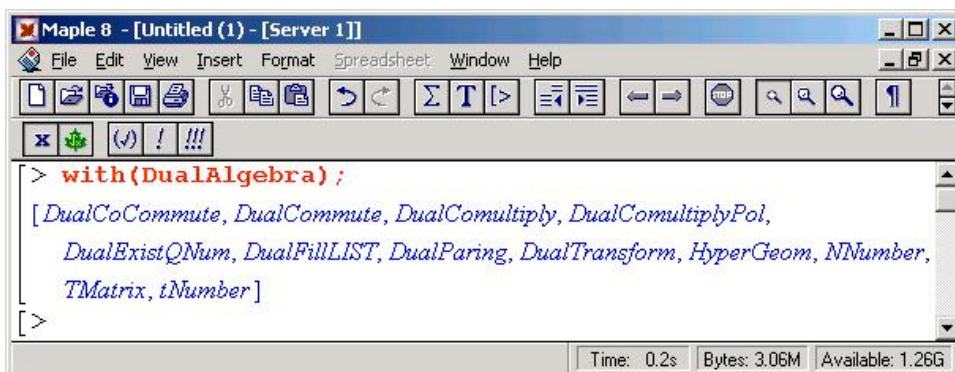
$$\Delta(c) = c \otimes a + d \otimes c, \quad \Delta(d) = c \otimes b + d \otimes d,$$

$$\varepsilon(a) = \varepsilon(d) = 1, \quad \varepsilon(b) = \varepsilon(c) = 0,$$

$$S(a) = d, \quad S(b) = -q^{-1}b, \quad S(c) = -qc, \quad S(d) = a.$$

8.3.3.1 Popis funkcí z *DualAlgebra.mws*

Po vložení hlavičkového souboru *DualAlgebra.mws* pomocí příkazu „with(DualAlgebra)“ je možné použít následující funkce:



```
> with(DualAlgebra);  
[DualCoCommute, DualCommute, DualComultiply, DualComultiplyPol,  
DualExistQNum, DualFillLIST, DualParing, DualTransform, HyperGeom, NNumber,  
TMatrix, tNumber]  
>
```

Time: 0.2s Bytes: 3.06M Available: 1.26G

obrázek 24: Funkce z *DualAlgebra.mws*

1. „DualCoCommute“

Popis viz obdobná funkce v kapitole 8.3.2.1.

2. „DualComultiply“

Popis viz obdobná funkce v kapitole 8.3.2.1.

3. „DualComultiplyPol“

Popis viz obdobná funkce v kapitole 8.3.2.1.

4. „DualCommute“

Popis viz obdobná funkce v kapitole 8.3.2.1.

5. „DualExistQNum“

Popis viz obdobná funkce v kapitole 8.3.2.1.

6. „DualFillLIST“

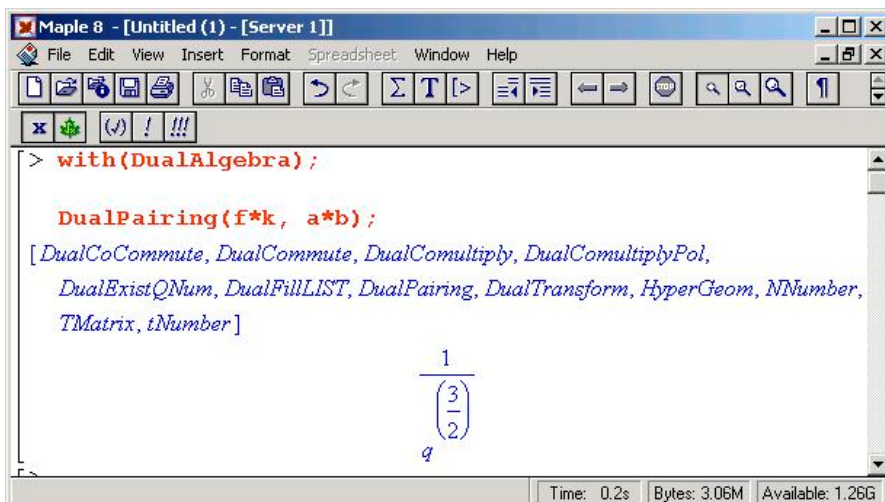
Popis viz obdobná funkce v kapitole 8.3.2.1.

7. „DualTransform“

Popis viz obdobná funkce v kapitole 8.3.2.1.

8. „DualPairing“

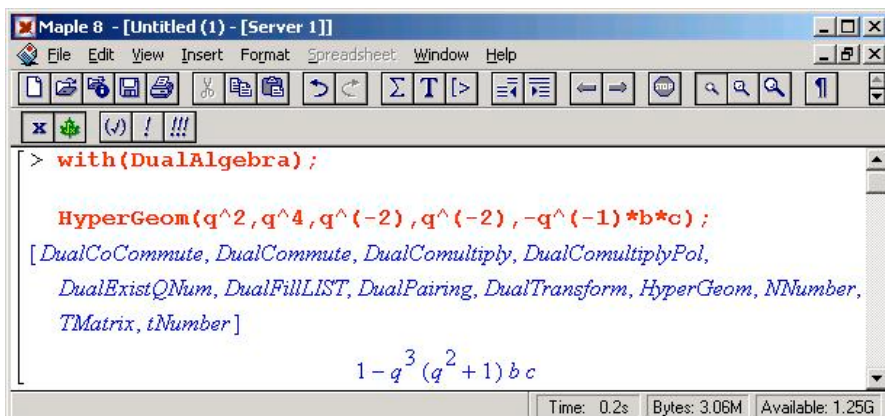
Funkce $DualPairing(Parametr1, Parametr2)$ provádí výpočet duálního párování dle věty 7.3.2 v kapitole 7.3.



obrázek 25: Použití funkce „DualPairing“

9. „HyperGeom“

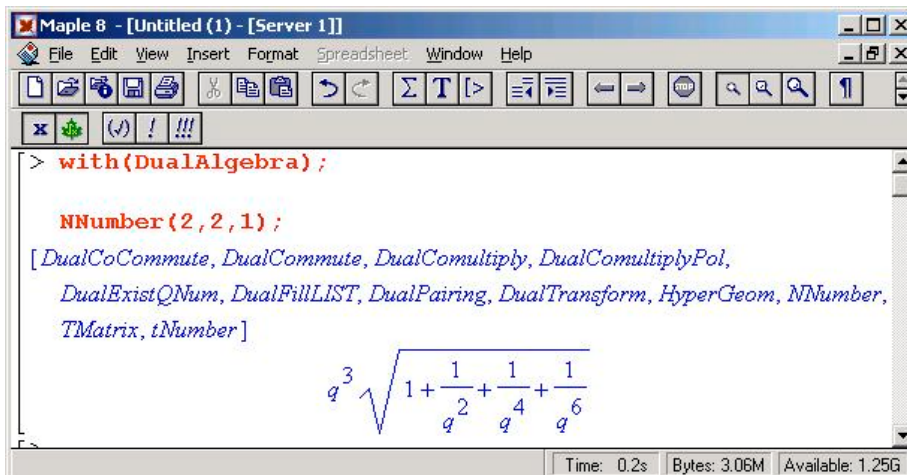
Funkce $HyperGeom(Parametr1, Parametr2, Parametr3, Parametr4, Parametr5)$ provádí výpočet hypergeometrické funkce definované v kapitole 4.



obrázek 26: Použití funkce „HyperGeom“

10. „NNumber“

Funkce $NNumber(l, i, j)$ provádí výpočet koeficientů N_{ij}^l , potřebných pro výpočet koeficientů koreprezentační matice definované v kapitole 7.2.1. U této funkce je nutné dát pozor na to, že $l \in \frac{1}{2} \mathbb{N}_0$.

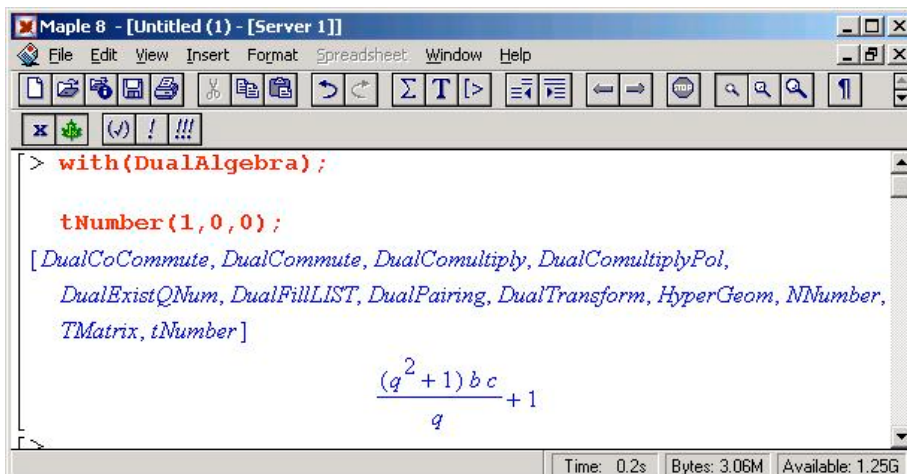


obrázek 27: Použití funkce „NNumber“

11. „tNumber“

Funkce $tNumber(l, i, j)$ provádí výpočet koeficientů koreprezentační matice definované v kapitole 7.2.1. U této funkce je nutné dát pozor na to, že $l \in \frac{1}{2}\mathbf{N}_0$.

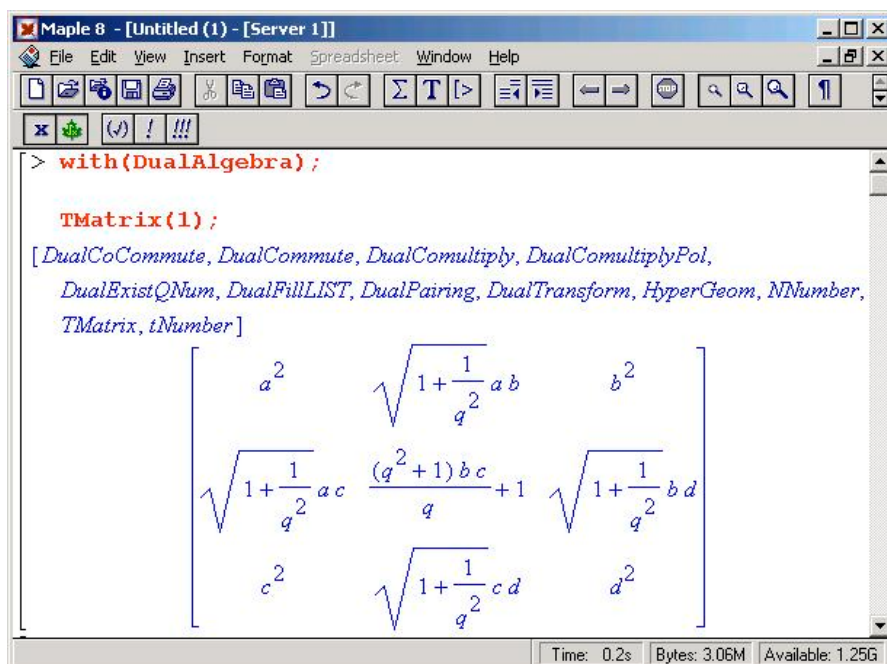
Funkce „tNumber“ využívá funkci „NNumber“.



obrázek 28: Použití funkce „tNumber“

12. „TMatrix“

Funkce $TMatrix(l)$ vrací koreprezentační matici, definovanou v kapitole 7.2.1, pro zadaný parametr l . Funkce „TMatrix“ využívá funkci „tNumber“.



obrázek 29: Použití funkce „TMatrix“

8.3.3.2 Vypracovaná identita využívající *DualAlgebra.mws*

Soubor *DualIdentity.mws* obsahuje vypracovanou identitu týkající se Hopfovy algebry $SL_q(2)$. Na začátku tohoto souboru je vložen hlavičkový soubor *DualAlgebra.mws*.

- $with(DualAlgebra)$

Vložení souboru *DualAlgebra* zpřístupní použití funkcí definovaných v 8.3.3.1.

Dále soubor *DualIdentity.mws* obsahuje následující funkce:

- $GetElement(parametr1, parametr2, T, dim)$
- $ComparePol(polynom1, polynom2)$
- $MatrixControl(T, l)$.

Souboru *DualIdentity.mws* ověřuje koreprezentační matice definované v kapitole 7.2.1. Celý postup ověření, zda je daná matice skutečně koreprezentací Hopfovy algebry, je popsán v kapitole 7.2.1.

Pro ověření vlastnosti koreprezentace je nutné umět konásobit libovolný prvek matice koreprezentace \mathbf{T} . To můžeme učinit pomocí funkce „DualComultiply“ popsané v kapitole 8.3.3.1. Výsledek získaný funkcí „DualComultiply“ můžeme dále upravit na kanonický tvar pomocí funkce „DualCoCommute“. Tímto postupem získáme první část pro ověření vlastnosti koreprezentace, tzn. výpočet konásobení libovolného prvku koreprezentační matice \mathbf{T} pomocí definičních vztahů konásobení generátorů Hopfovy algebry $SL_q(2)$.

Další částí je získání konásobení prvku matice \mathbf{T} pomocí tenzorového součinu prvků z odpovídajícího řádku a sloupce matice \mathbf{T} . K tomu slouží funkce „GetElement“ v souboru *DualIdentity.mws*. Parametry funkce „GetElement“ mají následující význam:

- první parametr - řádkový index i matice \mathbf{T}
- druhý parametr - sloupcový index j matice \mathbf{T}
- třetí parametr - matice \mathbf{T}
- čtvrtý parametr - dimenze matice \mathbf{T} .

Funkce „GetElement“ tedy vrátí výraz $\sum_{k=1}^n T_{ik} \otimes T_{kj}$. K tomuto tenzorovému součinu využívá funkci „DualComultiplyPol“ popsanou v kapitole 8.3.3.1.

Pokud se dané výrazy rovnají, tak je rovnost pro tento prvek ověřena. Celý postup je zobrazen na následujících dvou obrázcích. Na těchto obrázcích se ověřuje první prvek matice $\mathbf{T}_{1/2}$ stejně jako v příkladu v kapitole 7.2.1.

```

> with(DualAlgebra);

GetElement:=proc(p1,p2,T,dim)
  description "vraci prvek na dane adrese matice T":
  local i,vysledek:

  vysledek:=0:
  for i from 1 by 1 to dim do
    vysledek:=vysledek+DualComultiplyPol(T[p1,i],T[i,p2]):
  end do:
  return vysledek:
end proc:

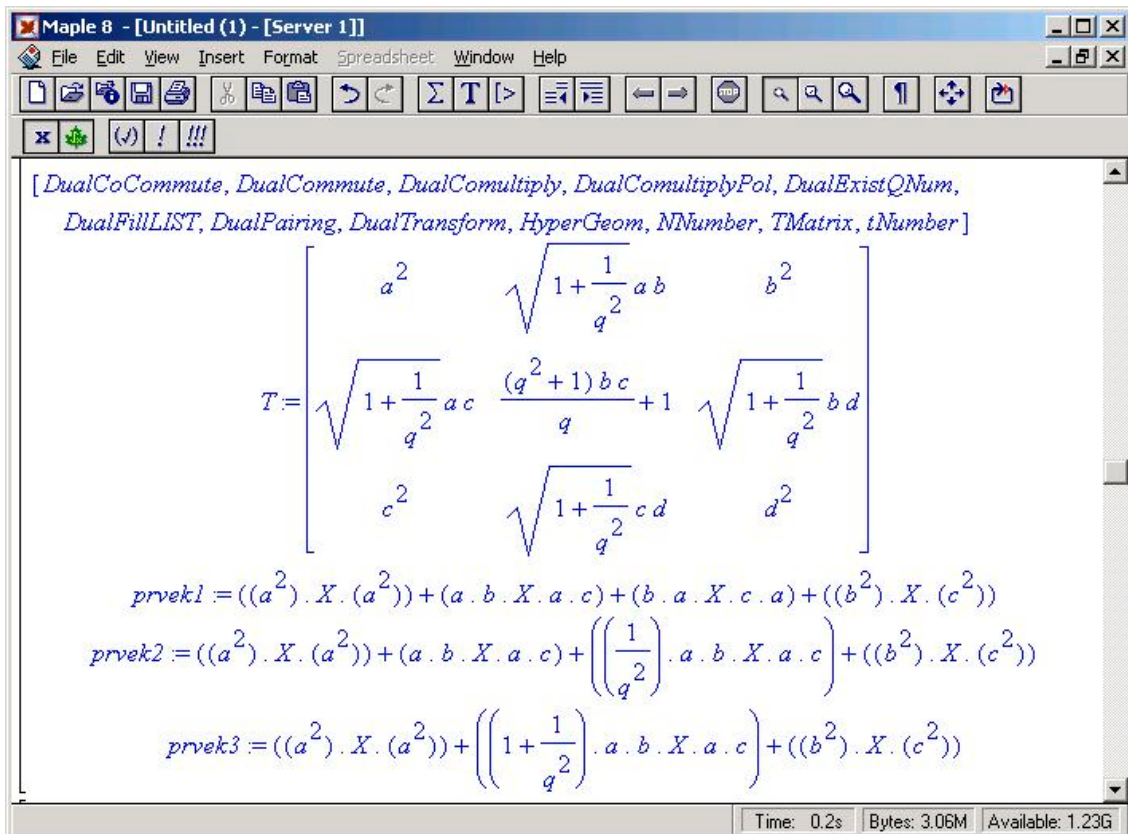
T:=TMatrix(1);

prvek1:=DualComultiply(T[1,1]);
prvek2:=DualCoComutate(prvek1);

prvek3:=GetElement(1,1,T,3);
  
```

Time: 4.6s Bytes: 3.06M Available: 166M

obrázek 30: Postup ověřování vlastnosti koreprezentační matice \mathbf{T} - použití funkcí



obrázek 31: Postup ověřování vlastnosti koreprezentační matice \mathbf{T} - výsledek

Na obrázcích je vidět, že „prvek2“ a „prvek3“ jsou totožné, čímž je první prvek matice \mathbf{T} ověřen.

Pro úplné ověření celé matice \mathbf{T} je nutné ověřit všechny prvky. K tomu slouží funkce „MatrixControl“. Tato funkce provádí popsany postup pro všechny prvky matice \mathbf{T} . Jejími parametry jsou matice \mathbf{T} a parametr l , se kterým byla matice \mathbf{T} vytvořena. K tomu, aby porovnávala vždy dva výsledné polynomy pro každý prvek matice, využívá funkci „ComparePol“, která porovná dva polynomy a vrátí 0, pokud jsou stejné. Výsledkem funkce „MatrixControl“ je tedy matice. Pokud matice obsahuje samé nuly, tak to značí, že daná matice je skutečně koreprezentační maticí Hopfovy algebry $SL_q(2)$. Na tomto místě je nutné zmínit, že výrazy, které při tomto postupu vznikají, jsou natolik složité, že vypracovaný program umí zpracovat koreprezentační matice dimenze maximálně sedm. Potom už algoritmus trvá velice dlouho a výsledné výrazy již Maple správně nezpracuje.

8.3.4 Duální párování Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$

Poslední vypracovanou identitou je ověření duality Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$ pomocí duálního párování definovaného v kapitole 7.3. Tato identita se nachází v adresáři *Program \ Pairing* v souboru *Pairing.mws*.

Na začátku tohoto souboru jsou vloženy hlavičkové soubory *HopfAlgebra.mws*, *DualAlgebra.mws* a *LinearAlgebra.mws*.

- $with(HopfAlgebra)$
- $with(DualAlgebra)$
- $with(LinearAlgebra)$

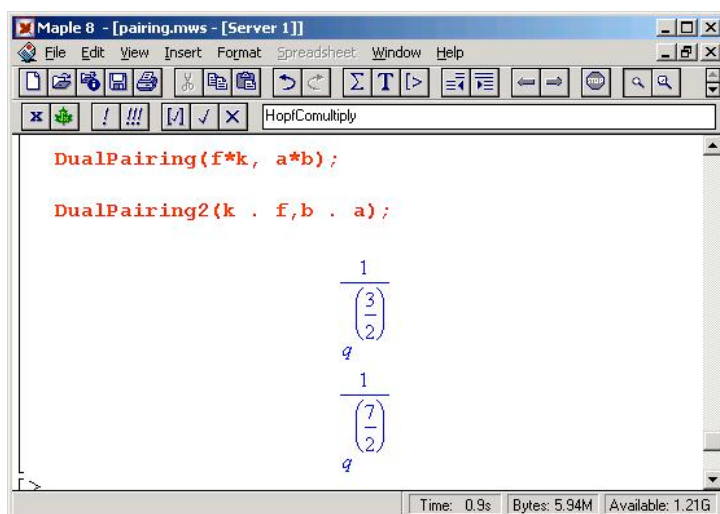
Vložení souboru *HopfAlgebra.mws* je zpřístupněno použitím všech funkcí definovaných v kapitole 8.3.2.1, vložení souboru *DualAlgebra.mws* jsou zpřístupněny funkce definované v kapitole 8.3.3.1. Hlavičkový soubor *LinearAlgebra* je standardně implementován v Maplu a slouží například k práci s maticemi.

Dále soubor *Pairing.mws* obsahuje následující funkce:

- $DualPairing2(parametr1, parametr2)$
- $MultiplyMatrix(Matrix1, Matrix2)$
- $ProvePairing(parametr, Matrix)$
- $ProveMatrixPairing(Matrix1, Matrix2)$.

1. Funkce „DualPairing2“

Funkce „DualPairing2“ je obdobná jako funkce „DualPairing“ popsaná v kapitole 8.3.3.1, s tím rozdílem, že provádí duální párování ještě nekomutovaných výrazů. Funkce „DualPairing“ předpokládá, že oba výrazy jsou již v kanonickém tvaru a obsahují tedy normální násobení označené v Maplu symbolem „*“. Naopak funkce „DualPairing2“ pracuje s výrazy, na které ještě nebyly použity funkce „DualCommute“, resp. „HopfCommute“ a funkce „DualTransform“, resp. „HopfTransform“ a vyskytuje se v nich nekomutativní násobení označené v Maplu symbolem „.“ Na následujícím obrázku je vidět rozdíl použití a výpočtu obou funkcí.



obrázek 32: Rozdíl použití a výpočtu funkcí „DualPairing“ a „DualPairing2“

Po převedení $k.f = \frac{1}{q}.f.k$ a $b.a = \frac{1}{q}.a.b$ je zřejmé, že oba výsledky jsou správně.

2. Funkce „MultiplyMatrix“

Tato funkce provádí obyčejné násobení matic. Parametry funkce jsou dvě čtvercové matice stejného rozměru.

3. Funkce „ProvePairing“

Tato funkce provádí ověření duality Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$. Parametry funkce $ProvePairing(parametr, Matrix)$ mají následující význam:

- *parametr* - libovolný prvek Hopfovy algebry $U_q(sl_2)$. Tento prvek musí být zapsán pomocí nekomutativního násobení značeného symbolem „.“.
- *Matrix* - koreprezentační matice Hopfovy algebry $SL_q(2)$.

Postup ověření duality Hopfových algeber $U_q(sl_2)$ a $SL_q(2)$ je popsán v kapitole 7.3 na posledním příkladu této kapitoly. Funkce „ProvePairing“ provádí přesně postup popsáný na tomto příkladu. Použití této funkce je uvedeno na následujícím obrázku.

```

matice:=TMatrix(1):
aa:=ProvePairing(f . e,matice);
aaa:=ProvePairing(HopfCommute(f . e),matice);
simplify(evalm(aa-aaa));

```

$$aa = \begin{bmatrix} \left(1 + \frac{1}{q^2}\right)q & 0 & 0 \\ 0 & \left(1 + \frac{1}{q^2}\right)q & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$aaa = \frac{\begin{bmatrix} -\frac{1}{q^2} & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -q^2 \end{bmatrix}}{q - \frac{1}{q}} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \left(1 + \frac{1}{q^2}\right)q & 0 \\ 0 & 0 & \left(1 + \frac{1}{q^2}\right)q \end{bmatrix} + \frac{\begin{bmatrix} q^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{q^2} \end{bmatrix}}{q - \frac{1}{q}}$$

obrázek 33: Použití funkce „ProvePairing“

Z obrázku je zřejmé, že se ověřuje koreprezentační matice, kde $l = 1$ s prvkem $f \cdot e$ Hopfovy algebry $U_q(sl_2)$.

První výsledná matice aa je maticí, kde je provedeno duální párování výrazu $f \cdot e$ s koreprezentační maticí \mathbf{T}_1 . Druhá výsledná matice aaa je duální párování výrazu $f \cdot e$, převedeném na kanonický tvar použitím funkce „HopfCommute“, s maticí \mathbf{T}_1 . Vzhledem k definici relace mezi f a e je zřejmé, že matice aaa se skládá ze třech matic. Pokud se matice v aaa sečtou a následně odečtou od matice aa , dostane se nulová matice, což značí, že matice aaa a aa jsou identické, čímž je dualita ověřena.

4. Funkce „ProveMatrixPairing“

Poslední funkcí souboru *Pairing.mws* je funkce *ProveMatrixPairing(Matrix1, Matrix2)*.

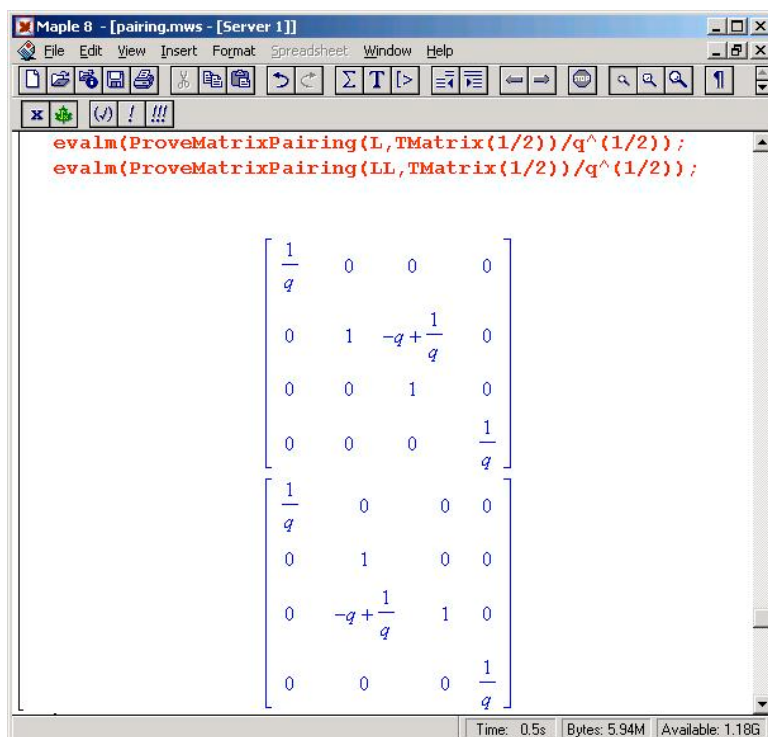
Funkce „ProveMatrixPairing“ ověřuje následující identitu.

Mějme matice \mathbf{L} , \mathbf{L}^* a \mathbf{R} definované v kapitole 8.3.2.2 a mějme koreprezentační matici $\mathbf{T}_{1/2}$.

Potom platí $\langle \mathbf{L}, \mathbf{T}_{1/2} \rangle = \mathbf{R}_{21}^{-1}$ a $\langle \mathbf{L}^*, \mathbf{T}_{1/2} \rangle = \mathbf{R}_{12}^{-1}$, kde $\mathbf{R}_{12}^{-1} = \mathbf{R}^{-1}$ a $\mathbf{R}_{21}^{-1} = (\mathbf{R}^{-1})^T$.

Matice $\langle \mathbf{L}, \mathbf{T}_{1/2} \rangle$ se získá obdobně jako tenzorový součin dvou matic, tzn. provede se duální párování prvku \mathbf{L}_{ij} s celou maticí $\mathbf{T}_{1/2}$ a výsledná matice 2×2 se vloží na místo prvku \mathbf{L}_{ij} . To se provede pro všechna i, j . Výsledná matice je typu 4×4 .

Použití funkce „ProveMatrixPairing“ je na následujícím obrázku.



obrázek 34: Použití funkce „ProveMatrixPairing“

9 ZÁVĚR

9.1 Získané výsledky

Cílem této diplomové práce bylo ukázat využití moderních počítačových algebraických systémů v teorii kvantových grup.

Na začátku celé práce byly naprogramovány identity týkající se nejjednodušší možné nekomutativní algebry se dvěma generátory x, y provázanými vztahem $yx = qxy$. Jako příklad pro ověření platných identit v této algebře byl vypracován program ověřující platnost zobecněné binomické věty a dále ověření vztahu pro násobení q -deformovaných exponenciálních funkcí.

V další části práce byly vytvořeny sady funkcí pro práci s konkrétními kvantovými grupami $U_q(sl_2)$ a $SL_q(2)$. Naprogramované funkce umožňují převádět libovolné polynomy v těchto algebrách na kanonický tvar. Poté je možné upustit od nekomutativního násobení a dále s těmito polynomy pracovat. Například je možné mezi sebou polynomy nekomutativně násobit.

Daleko složitějším problémem bylo vypracování funkcí pro konásobení prvků obou algeber. Symbol „ \otimes “ pro tenzorový součin u konásobení byl pro účely programování nahrazen symbolem „ X “. Výrazy obsahující konásobení (symbol „ X “) je možné „komutovat“, tzn. převádět na kanonický tvar pomocí definičních relací v obou algebrách.

Použitelnost těchto funkcí byla ukázána na ověření platných identit v kvantových grupách $U_q(sl_2)$ a $SL_q(2)$. Pro kvantovou grupu $U_q(sl_2)$ byla ověřena identita týkající se přeformulování definičních vztahů mezi generátory grupy pomocí maticových identit typu $\mathbf{R} \mathbf{L}_1 \mathbf{L}_2 = \mathbf{L}_2 \mathbf{L}_1 \mathbf{R}$, které zavedli Faddeev, Reshetikhin a Takhtajan. V kvantové grupě $SL_q(2)$ byla ukázána použitelnost funkcí pro konásobení. V této grupě byl vypracován program pro ověření koreprezentačních matic \mathbf{T}_n .

V poslední části práce byly vypracovány funkce pro ukázkou duality (ve smyslu duálního párování) mezi kvantovými grupami $U_q(sl_2)$ a $SL_q(2)$.

9.2 Zjištěné problémy

Po vypracování programu, který ověřuje koreprezentační matice (kapitola 8.3.3.2) v kvantové grupě $SL_q(2)$ byl zjištěn následující problém. Při ověřování matice rozměru většího než šest, tj. matice větší než $\mathbf{T}_{5/2}$ už vypracovaný program nevrátí správný výsledek. Jedná se o to, že výrazy, které se vyskytují v maticích ve větších dimenzích už jsou natolik složité, že po aplikování konásobení na tyto výrazy a dále jejich porovnání, už systém Maple nevyhodnotí správně. Toto je dáno zvoleným postupem. Každý polynom obsahuje nekomutativní násobení reprezentované symbolem „ \otimes “ a dále každý člen polynomu obsahuje tenzorový součin (symbol „ X “). Tento přístup ke konásobení je z uživatelského hlediska velmi výhodný, neboť zobrazuje polynomy vzniklé konásobením generátorů grupy, přesně tak jak je popsáno v literatuře o kvantových grupách (pouze symbol „ \otimes “ je nahrazen symbolem „ X “). Ovšem z hlediska časové náročnosti jsou tyto výrazy natolik složité, že jejich zpracování v Maplu trvá velice dlouho a pro složitější výrazy již nefunguje správně.

Řešením by bylo konásobení vypracovat tak, že místo tenzorového součinu se bude pracovat s uspořádanými dvojicemi, tzn. rozdělit prvky podle toho zda jsou před, nebo za symbolem „ \otimes “. Z uživatelského hlediska by toto řešení nebylo tak intuitivní, nicméně by vedlo k lepším výsledkům. Řešení zvolené v této práci je navrhuto s ohledem na postup, kterým se ověřuje vlastnost koreprezentační matice v kapitole 7.2.1. Naprogramovaný postup potom věrně kopíruje důkaz v této kapitole. Pro demonstraci využití počítačových algebraických systémů v teorii kvantových grup je dle mého názoru plně dostačující.

9.3 Další možnosti využití práce

Celý program je vytvořen tak, že sady funkcí jsou uspořádány do jednotlivých modulů. Tyto moduly je tedy možné vkládat do libovolných programů a vypracované funkce využívat. Tím se otevírá možnost vypracování dalších platných identit týkajících se kvantových grup $U_q(sl_2)$ a $SL_q(2)$, nebo q - calculu pomocí funkcí vypracovaných v této práci. Dále je možné dle této práce vypracovat podobné moduly pro další kvantové grupy. Veškeré vypracované funkce a identity se týkají úplného základu teorie kvantových grup a vzhledem k náročnosti a šířce celé této teorie může být tato práce odrazovým můstkem pro další práci a výzkum.

10 LITERATURA

- [1] S. Mac Lane - G. Birkhoff: *Algebra*, VTEL, Bratislava, 1973
- [2] A. Klimyk - K. Schmudgen: *Quantum Groups and Their Representations*, Berlin 1997
- [3] Hopf H.: *Über die Topologie der Gruppen-mannigfaltigkeiten und ihre Verallgemeinerungen*, 1941
- [4] S. Wolfram: *The Mathematica Book*, Cambridge University Press, 1999
- [5] V. Chari, A. Pressley: *A Guide to Quantum Groups*, Cambridge University Press, 1994
- [6] Jan Mareš: *Úvod do obecné algebry*, ČVUT, Praha, 1999
- [7] Čestmír Burdík: *Autoreferát doktorské disertace*, ČVUT, Praha, 1999