МАТЕМАТИЧКИ ФАКУЛТЕТ
УНИВЕРЗИТЕТ У БЕОГРАДУ

Владан Радивојевић

Класификација латинских квадрата

Магистарски рад

Ментор: др Миодраг Живковић

Београд, 2009.

Предговор

Појам латинског квадрата потиче од игара са картама и ређања фигура на шаховској табли, а први писани материјали, као и сам назив се везују за Ојлера (L. Euler) и 18. век.

Латински квадрат реда n је матрица димензија $n \times n$ чија свака врста и свака колона садржи пермутацију скупа од n симбола. Без смањења општости може се узети да је скуп симбола $\{0, 1, 2...n-1\}$.

Латински квадрати спадају у широку класу комбинаторних објеката, па су као такви пре свега предмет проучавања дискретне математике, а примена у статистици, криптографији и теорији кодова дала је на значају овим објектима и у другим сродним областима.

За два латинска квадрата се каже да су у истој изотопној класи ако се један може добити од другог пермутацијом врста, колона и симбола.

Проблем који се решава у овој тези је генерисање представника изотопних класа латинских квадрата што је могуће већих димензија. Овакво генерисање, по једног квадрата из сваке класе се назива *класификација*.

Највећа димензија за коју је класификација успела је 8 и она је једнака са најбољим резултатом који се тренутно може наћи у литератури. При томе метода којом је тај резултат добијен је нешто другачија од оних које су коришћене у другим радовима.

Рад се састоји од 6 поглавља.

Прво поглавље садржи уводне појмове и примере, као и везе латинских квадрата и правоугаоника са другим математичким објектима. У овом поглављу је такође дат и преглед неких примена латинских квадрата.

Тема другог поглавља је пребројавање латинских квадрата и правоугаоника. У њему је најпре дат историјски преглед резултата, а затим су детаљније обрађене методе којима су добијени тренутно најбољи резултати.

У трећем поглављу се дефинишу класе латинских квадрата и правоугаоника и разматрају методе за одређивање броја ових класа.

Четврто поглавље је централни део рада. Тема овог поглавља је генерисање представника класа латинских квадрата, односно сама класификација. Излагање је осмишљено тако да је сваки алгоритам за класификацију најпре описан на општем нивоу комбинаторних објеката, а затим примењен на латинске квадрате. Због овакве концепције разматрања дата у овом делу могу бити од користи и за класификацију других комбинаторних објеката.

Пето поглавље садржи опис имплементације неких од алгоритама из четвртог поглавља, као и резултате добијене уз помоћ развијених програма.

У последњем, шестом поглављу сумирају се постојећи резултати и указује се на могуће правце у даљем раду.

Осим понављања тренутно најбољег резултата у класификацији квадрата, у резултате ове тезе се могу уврстити и особине канонских правоугаоника које су дате у тачки 4.1.2.

Такође, оригиналан резултат овог рада су бројеви изотопних класа чији су канонски представници редуковани правоугаоници (четврта колона табеле 5.1). Овај резултат је прихваћен и објављен у "Енциклопедији целобројних низова" (The On-Line Encyclopedia of Integer Sequences) под бројем $A162545$.
(http://www.research.att.com/ njas/sequences/A162545)

Садржај

1

Основни појмови и тврђења

На почетку ће бити дефинисани појмови латинског квадрата и правоугаоника, као и размотрени неки примери.

1.1 Уводне дефиниције и примери

Нека су k и n два позитивна цела броја таква да је $k \leq n$.

Дефиниција 1.1 Латински правоугаоник *димензија* $k \times n$ *је матрица* $L = (L_{i,j})$ *са* k *врста и* n *колона, чији елементи су симболи азбуке од* n *елемената, таква да су елементи у свакој врсти и елементи у свакој колони различити.*

Латински правоугаоник за који је $k = n$ *се назива* латински квадрат*. У овом случају димензија се назива и* ред латинског квадрата*.*

Придев *латински* потиче од симбола латинске азбуке које је Ојлер користио као елементе квадрата. У наставку се овај придев подразумева у свим случајевима уз речи *квадрат* или *правоугаоник* и онда када није наведен.

Азбука симбола може бити произвољна, али се ради једноставнијег излагања у овом раду за латински правоугаоник са n колона користи азбука $\Sigma = \{0, 1, ..., n-1\}$. Такође, у раду се користи индексирање врста и колона од 0.

Латински правоугаоник може да има произвољне димензије. Наиме, важи следеће тврђење:

Теорема 1.1 *За свака два позитивна броја k и n таква да је $k \leq n$ постоји латински правоугаоник димензија $k \times n$.*

Доказ: Једноставан начин за конструкцију латинског правоугаоника је да се за свако i и j $(0 \leq i < k,\ 0 \leq j < n)$ елементи дефинишу на следећи начин:

$$L_{i,j} = i + j \pmod{n}.$$

Пример 1.1 *Латински квадрат реда 4 конструисан поступком из доказа претходне теореме је:*

$$L = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ \hline \end{array}$$

Латински правоугаоник димензија $k \times n$ се може представити и својом тзв. *ортогоналном репрезентацијом.* То је матрица димензија $3 \times k \cdot n$, чије колоне су облика $[i, j, L_{i,j}]^T$, $(0 \leq i < k,\ 0 \leq j < n)$.

Пример 1.2 *Ортогонална репрезентација квадрата из примера 1.1 је:*

$$OR(L) = \begin{array}{|cccc|cccc|cccc|cccc|} \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 & 0 & 2 & 3 & 0 & 1 & 3 & 0 & 1 & 2 \\ \hline \end{array}$$

Редослед колона у овој матрици није важан, па се она може посматрати и као скуп колона. Свака колона матрице $OR(L)$ је облика $[r, c, s]^T$, где је r врста, c колона, а s одговарајући симбол латинског правоугаоника. Сви уређени парови (r, c) који се јављају у колоном су различити. Исто важи за све парове (c, s) и све парове (r, s).

Дефиниција 1.2 *За латински квадрат чија ортогонална репрезентација се може добити пермутацијом врста у ортогоналној репрезентацији квадрата L се каже да је* конјугат *квадрата L.*

Непосредно из дефиниције следи да сваки квадрат има највише $3! = 6$ конјугата.

Пример 1.3 *Конјугат квадрата L који се добија заменом прве две врсте у матрици $OR(L)$ је транспонован квадрат L^T.*

Дефиниција 1.3 *За латински правоугаоник димензија $k \times n$ се каже да је* редукован *(*нормализован*) ако је његова прва врста $[0, 1, 2, ..., n-1]$, а његова прва колона $[0, 1, 2, ..., k-1]^T$.*

Пример 1.4 *Латински квадрат из примера 1.1 је редукован.*

Ако је правоугаоник (квадрат) редукован онда су елементи његове почетне врсте и почетне колоне поређани у растућем поретку. Треба нагласити да за квадрате важи и обратно тврђење, али за произвољан правоугаоник не, као што показује наредни пример.

Пример 1.5 *Елементи у почетној врсти и почетној колони правоугаоника*

$$P = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \\ \hline \end{array}$$

су поређани у растућем поретку, али он није *редукован.*

Важан појам у вези са латинским квадратима су и *међусобно ортогонални* квадрати.

Дефиниција 1.4 *Нека су L' и L'' латински квадрати реда n и нека је S скуп уређених парова*

$$S = \{(L'_{i,j}, L''_{i,j}), 0 \leq i, j < n\}.$$

За квадрате L' и L'' се каже да су међусобно ортогонални *ако скуп S садржи n^2 различитих уређених парова.*

Пример 1.6 *Латински квадрати*

$$L' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \\ \hline \end{array} \text{ и } L'' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \\ \hline \end{array}$$

су ортогонални. Заиста, сви уређени парови у скупу

$$S = \{(0,0), (1,1), (2,2), (1,2), (2,0), (0,1), (2,1), (0,2), (1,0)\}$$

су различити.

Појам ортогоналности се дефинише и за скуп који садржи више од два латинска квадрата.

Дефиниција 1.5 *Скуп латинских квадрата $\{L_1, L_2, L_3, ...\}$ се назива ортогоналним (MOLS - muthual orthogonal latin squares) ако су свака два квадрата из овог скупа ортогонална.*

Одређивање највећег броја међусобно ортогоналних квадрата за произвољну вредност n је отворен проблем који није предмет овог рада. Следећа теорема (која ће бити наведена без доказа), је илустрација неких резултата везаних за овај проблем, а више резултата се може наћи на пример у [12].

Теорема 1.2 *Ако је $N(n)$ највећи број међусобно ортогоналних квадрата реда n важи:*

- $N(n) \leq n-1$. *Дакле, скуп из примера 1.6 се не може проширити.*
- $N(6) = 1$. *Другим речима, не постоје два међусобно ортогонална квадрата реда 6.*

1.2 Еквивалентни објекти

У овој тачки се разматра еквивалентност латинских правоугаоника (квадрата) са неким другим математичким објектима, што је од важности за наставак излагања.

1.2.1 Квазигрупе и латински квадрати

Теорема 1.3 *Таблица множења коначне групе реда n је латински квадрат.*

Доказ: Претпоставимо супротно, односно да постоји група $(G, \cdot)$ таква да у некој врсти њене мултипликационе таблице постоје два иста елемента. Ово би значило да за неке елементе групе a, b и $c, (b \neq c)$ важи: $a \cdot b = a \cdot c$. Како a, као и сваки елемент групе има инверзни елемент, множењем горње једнакости са a^{-1} са леве стране се добија $b = c$, што је контрадикција. Слична аргументација важи и за колоне.

Наредни пример показује да обратно не важи, па не постоји еквиваленција између група и латинских квадрата.

Пример 1.7 *Латински квадрат*

0	2	1
2	1	0
1	0	2

не садржи врсту која је идентичка пермутација, па алгебраска структура чија је таблица множења представљена овим квадратом нема неутрални елемент, тј. није *група.*

Матрица димензија $n \times n$ је латински квадрат ако и само ако је таблица множења неке квазигрупе.

Дефиниција 1.6 Квазигрупа *је уређени пар* $(G, \cdot)$, *где је* G *скуп, а* $\cdot$ *бинарна операција за које важи:*

1. *ако* $x, y \in G$, *тада је* $x \cdot y \in G$.
2. *за свака два елемента* $a, b \in G$ *једначина* $a \cdot x = b$ *има јединствено решење по* x.
3. *за свака два елемента* $a, b \in G$ *једначина* $x \cdot a = b$ *има јединствено решење по* x.

Квазигрупа за разлику од групе не мора да има неутрални елемент, па еквивалентност са латинским квадратима непосредно следи.

1.2.2 Графови и латински правоугаоници

Пре разматрања односа латинских правоугаоника и графова биће дате дефиниције одређених појмова и наведена нека тврђења из теорије графова неопходна за наставак излагања.

Дефиниција 1.7 *Граф* G *је* бипартитан *ако се скуп његових чворова може поделити на два дисјунктна подскупа* U *и* V *тако да свака грана графа* G *повезује један чвор из скупа* U *са једним чвором из скупа* V. *Овакав граф се уобичајено обележава са* $G = (U \cup V, E)$, *где је* E *скуп грана.*

Наредни појмови могу бити дефинисани и за произвољне графове, али је за наставак довољно ограничити се само на бипартитне графове.

Дефиниција 1.8 *За бипартитни граф* $G = (U \cup V, E)$ *се каже да је* k-регуларан, *ако је сваки његов чвор из скупа* U *повезан са тачно* k *чворова из скупа* V *и обратно.*

Последица дефиниције је да за k-регуларан граф $G = (U \cup V, E)$ важи $|E| = k|U| = k|V|$, односно $|U| = |V|$.

Дефиниција 1.9 *Нека је* $G = (U \cup V, E)$ *бипартитни граф.* 1-фактор *графа* G *је 1-регуларан бипартитни граф* $G' = (U \cup V, E')$, *где је* $E' \subseteq E$.

Из дефиниција 1.8 и 1.9 следи да је $|U| = |V|$ потребан услов да граф $G = (U \cup V, E)$ има 1-фактор.

Довољан услов важан за наставак излагања, дат је у наредној теореми.

Теорема 1.4 *Ако је* $G = (U \cup V, E)$ *бипартитни граф у којем је* $|U| = |V|$ *и сваки скуп чворова* $S \subseteq U$ *је повезан са бар* $|S|$ *чворова из скупа* V*, тада постоји 1-фактор графа* G.

Доказ: Тврђење се може доказати индукцијом по броју чворова у скупу U*. Ако је* $|U| = 1$*, једини чвор из скупа* U *је према претпоставци теореме повезан са јединим чвором из скупа* V*. Ова грана је тражени 1-фактор.*

Претпоставимо да тврђење важи за $|U| \leq n$ *и размотримо тврдјење када је* $|U| = n + 1$*. Према претпоставци теореме сваки чвор из скупа* $|U|$ *је повезан са бар једним чвором из скупа* V*. Треба размотрити два случаја везана за скуп* U*:*

Постоји чвор $u_0 \in U$ *који је повезан са тачно једним чвором* $v_0 \in V$*.* *Нека је* $G' = (U' \cup V', E')$ *граф који се добија уклањањем чворова* u_0 *и* v_0 *и свих грана повезаних са њима из графа* G*. Докажимо да* G' *испуњава услов теореме. Претпоставимо супротно, тј. да постоји скуп чворова* $S' \subseteq U'$ *који је у графу* G' *повезан са* $|S'| - k$ *чворова, где је* $k > 0$*. У том случају би скуп* $S = S' \cup \{u_0\}$ *био повезан са* $|S'| - k + 1$ *чворова у графу* G *. Како граф* G *испуњава услов теореме, то је* $|S| = |S'| + 1 \leq |S'| - k + 1$*, тј.* $k \leq 0$*, што је контрадикција.*

Граф G' *дакле испуњава услов теореме и важи* $|U'| = n$*, па по индукцијској претпоставци постоји 1-фактор* F' *графа* G'*. Додавањем гране* (u_0, v_0) *скупу* F' *се добија 1-фактор графа* G*.*

Не постоји чвор у скупу U *који је повезан са тачно једним чвором из скупа* V*.* *Нека је* (u_0, v_0) *произвољна грана у графу* G *и нека је* $G' = (U' \cup V', E')$ *граф који се добија уклањањем чворова* u_0 *и* v_0 *и свих грана повезаних са њима из графа* G*. Докажимо да* G' *испуњава услов теореме. Претпоставимо супротно, тј. да постоји скуп чворова* $S' \subseteq U'$ *који је у графу* G' *повезан са* $|S'| - k$ *чворова, где је* $k > 0$*. Како је* S' *у графу* G *повезан са бар* $|S'|$ *чворова, следи да је* $k = 1$*, тј. да је скуп* S' *у графу* G *повезан са тачно* $|S'|$ *чворова. Ово је контрадикција са претпоставком да у графу* G *не постоји чвор из скупа* U *који је повезан са тачно једним чвором из скупа* V*.*

Граф G' *дакле испуњава услов теореме и важи* $|U'| = n$*, па по индукцијској претпоставци постоји 1-фактор* F' *графа* G'*. Додавањем гране* (u, v) *скупу* F' *се добија 1-фактор графа* G*.*

Дефиниција 1.10 1-факторизација *бипартитног графа* $G = (U \cup V, E)$ *је скуп* $\Phi = \{F_0, F_1, ...\}$ *1-фактора графа* G*, таквих да су скупови њихових грана дисјунктни и да је унија њихових грана једнака скупу* E*.*

Број могућих 1-факторизација графа G ће у наставку бити означен са $M(G)$. Ова вредност се може израчунати рекурзивно, на основу наредног тврђења из теорије графова које наводимо без доказа:

Теорема 1.5 *Нека је G k-регуларан, бипартитни граф и нека је e његова произвољна грана. Ако је Φ скуп свих 1-фактора графа G који садрже грану e, тада је:*

$$M(G) = \sum_{F \in \Phi} M(G - F). \tag{1.1}$$

Још један појам који има важну улогу у пребројавању и класификацији латинских правоугаоника и квадрата је изоморфизам графова. Изоморфизам се може дефинисати за произвољне графове, али је за ово излагање довољно ограничити се само на бипартитне.

Дефиниција 1.11 *Бипартитни графови $G = (U_1 \cup V_1, E_1)$ и $H = (U_2 \cup V_2, E_2)$ су* изоморфни *ако постоје бијекције*

$$f_1 : U_1 \to U_2, \quad f_2 : V_1 \to V_2$$

или бијекције

$$f_1 : U_1 \to V_2, \quad f_2 : V_1 \to U_2$$

такве да је (u, v) грана графа G ако и само ако је $(f_1(u), f_2(v))$ грана графа H. Пар пресликавања (f_1, f_2) се назива изоморфизам графова G и H.

Неформално, два графа су изоморфна, ако садрже исти број чворова повезаних на исти начин. Изоморфизам који пресликава граф у себе се назива *аутоморфизам.* Скуп свих аутоморфизама графа G са операцијом композиције функција чини групу $Aut(G)$.

Изоморфност графова је релација еквиваленције која скуп графова дели у класе изоморфних. Тврђење које је непосредна последица еквивалентности изоморфизама графова и одговарајућих пермутација чворова је:

Теорема 1.6 *Број бипартитних графова изоморфних са графом $G = (U \cup V, E)$ у којем је $|U| = |V| = n$ је*

$$\frac{2(n!)^2}{|Aut(G)|}. \tag{1.2}$$

Тренутно најефикаснији програм који за задате графове G и H проналази све њихове изоморфизме је NAUTY, чији је аутор

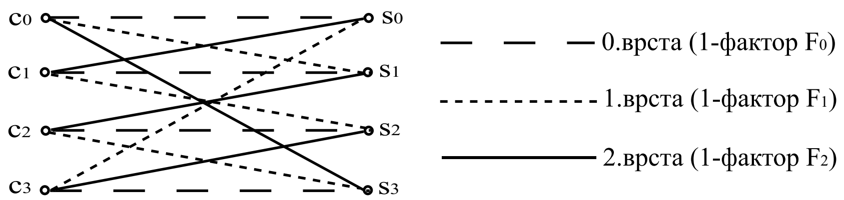
Мекеј(B. McKay). Ако је $G = H$, NAUTY решава проблем одређивања групе аутоморфизама $Aut(G)$. То је и мотив за избор назива програма ("No AUTomorphisms, Yes?").

Вратимо се односу између латинских правоугаоника и графова. Сваком латинском правоугаонику L димензија $k \times n$ може се придружити јединствен бипартитни граф $\gamma(L)$ на следећи начин:

- Скуп чворова графа $\gamma(L)$ је $C \cup S$, где је $C = \{c_0, ..., c_{n-1}\}$ и $S = \{s_0, ..., s_{n-1}\}$.
- (c_i, s_j) је грана графа $\gamma(L)$ ако и само ако колона i правоугаоника L садржи симбол j.

Овако дефинисан граф $\gamma(L)$ је k-регуларан. Наиме, свака колона правоугаоника садржи по тачно k различитих симбола, па је сваки чвор из скупа C повезан са тачно k чворова из скупа S. Са друге стране, сваки симбол се у латинском правоугаонику димензија $k \times n$ појављује тачно k пута (по једном у свакој врсти), па је сваки чвор из скупа S повезан са k различитих чворова из скупа C.

Пример 1.8 *На слици је дат 3-регуларан граф који одговара латинском правоугаонику из примера 1.5.*



Слика 1.1: граф $\gamma(L)$

Обратно, произвољан $k-$регуларан бипартитни граф G са укупно $2n$ чворова не одређује једнозначно правоугаоник L, јер не узима у обзир редослед симбола у његовим колонама. Број правоугаоника који му одговарају је дат у наредној теореми.

Теорема 1.7 *За сваки k-регуларан бипартитни граф G са укупно $2n$ чворова постоји тачно $k!M(G)$ различитих латинских правоугаоника L димензија $k \times n$ таквих да је $\gamma(L) = G$.*

Доказ: Сваком 1-фактору датог графа одговара једна врста правоугаоника, па једној 1-факторизацији одговара скуп *од k врста правоугаоника. Како се ове врсте могу поређати на $k!$ начина, следи да једној 1-факторизацији графа G одговара $k!$ различитих латинских правоугаоника, одакле непосредно следи тврђење теореме.*

1.3 Парцијални латински квадрати

Под парцијалним латинским квадратом се подразумева квадратна матрица која задовољава ограничења из дефиниције латинског квадрата, али чија сва поља нису дефинисана.

Пример 1.9 *Квадрат*

$$L = \begin{array}{|cccc|} \hline 0 & x & x & 3 \\ 1 & x & x & 0 \\ x & x & 0 & x \\ x & 0 & x & 2 \\ \hline \end{array}$$

је парцијални латински квадрат реда 4. Поља која садрже симбол x нису дефинисана.

Питање које се природно намеће је да ли је могуће одредити вредност недефинисаних поља тако да се од парцијалног добије потпун латински квадрат. У случају квадрата из претходног примера одговор је потврдан - једно од решења је латински квадрат из примера 1.1.

У општем случају овај проблем одлучивања је NP-комплетан [2]. Без обзира на ову чињеницу, постоје инстанце проблема које јесу решиве. На пример, наредну теорему је Еванс (T.Evans) исказао као претпоставку 1960. године [5], а доказана је након 21 године [23]:

Теорема 1.8 *Произвољан парцијални латински квадрат реда n са мање од n дефинисаних поља се може допунити до латинског квадрата.*

Од значаја за наставак овог рада су они парцијални квадрати чија дефинисана поља чине латински правоугаоник. Наиме, може се доказати да је сваки латински правоугаоник могуће допунити до латинског квадрата [12]. За то су неопходне наредне две теореме.

Теорема 1.9 *Нека је S скуп који садржи r произвољних колона правоугаоника L димензија $k \times n \quad (k < n)$. Постоји бар r симбола који нису у свакој колони из скупа S.*

Доказ: Нека је за свако $i \in \{0, 1, ...n-1\}$ S_i *скуп симбола који не постоје у* $i-$*тој колони правоугаоника* L. *Ако се изабере произвољних* r *скупова* S_i, *они укупно имају* $r(n-k)$ *симбола. Претпоставимо супротно тврђењу теореме, да је међу њима мање од* r *различитих. Из ове претпоставке следи да постоји симбол који се јавља више од* $n-k$ *пута у скуповима* S_i. *Ово је контрадикција, јер се сваки симбол у правоугаонику јавља тачно* k *пута (по једном у свакој врсти), односно постоји у тачно* $n-k$ *скупова* S_i.

Теорема 1.10 *Сваки латински правоугаоник димензија* $k \times n$ *се може проширити додавањем нове врсте у правоугаоник димензија* $(k+1) \times n$.

Доказ: За дати правоугаоник L *конструишимо одговарајући бипартитни граф* $\gamma(L) = (C \cup S, E)$ *на начин описан у тачки 1.2.2. Нека је* G *граф комплементаран са* $\gamma(L)$, *то јест у графу* G *постоји грана* (c_i, s_j) *ако и само ако* $i-$*та колона правоугаоника* L не садржи *симбол* j. *Према тврђењу претходне теореме* G *је бипартитни граф у којем је сваки скуп чворова* $C' \subseteq C$ *повезан са бар* $|C'|$ *чворова из скупа* S. *Дакле, граф* G *испуњава услов теореме 1.4, па је могуће конструисати његов 1-фактор, што је еквивалентно додавању нове врсте правоугаонику* L.

Непосредна последица претходне теореме је:

Теорема 1.11 *Сваки латински правоугаоник се може допунити до латинског квадрата.*

На крају овог дела, наводимо без доказа тврђење везано за још једну врсту парцијалних квадрата које је дао Рајзер (H.J.Ryser) [19].

Теорема 1.12 *Нека је* L *парцијални латински квадрат реда* n *такав да је поље* $L_{i,j}$ *дефинисано ако и само ако је* $0 \leq i < r$ *и* $0 \leq j < s$ *и нека је* $N(i)$ *ознака за број елемената квадрата једнаких* i. *Квадрат* L *може бити допуњен до латинског квадрата реда* n *ако и само ако за свако* i, $(0 \leq i < n)$ *важи:*

$$N(i) \geq r + s - n.$$

Пример 1.10 *Парцијални латински квадрат*

$$L = \begin{array}{|ccccc|} \hline 0 & 1 & 2 & x & x \\ 1 & 2 & 0 & x & x \\ 2 & 0 & 1 & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ \hline \end{array}$$

не задовољава Рајзеров услов, јер је $N(3) < 3+3-5$, *па не може бити допуњен до латинског квадрата.*

1.4 Неке примене латинских квадрата

За крај овог поглавља наводимо преглед неких примена латинских квадрата, као и веза са другим математичким објектима. Више о овде наведеним, као и о другим применама се може наћи на пример у [12].

Примена у статистици

Оригинална мотивација за проучавање латинских квадрата лежи добрим делом у примени у креирању статистичких експеримената.

Нека је дат процес $P(A, B)$ дефинисан са два параметра A и B који могу да имају редом, вредности $a_0, a_1, ..., a_{n-1}$ и $b_0, b_1, ..., b_{n-1}$ и нека је потребно тестирати дејство променљиве S која може да има вредности $s_0, s_1, ..., s_{n-1}$, на тај процес. Експеримент који најефикасније и систематски врши тражено тестирање подразумева креирање латинског квадрата реда n чији елементи су вредности $s_i, 0 \leq i < n$, а чије врсте и колоне редом, представљају вредности параметара A и B

За тестирање истовременог дејства две променљиве S и T на исти процес погодна метода је конструкција ортогоналних квадрата и њихово "спајање" у један квадрат чији су елементи одговарајући парови (s_i, t_j).

Следећи пример је илустрација описане технике за дизајнирање статистичког експеримента у области пољопривреде.

Пример 1.11 *Нека је потребно испитати дејство хемијске супстанце S која може да има вредности s_0, s_1 и s_2 на принос одређене културе у три годишња доба Γ_0, Γ_1 и Γ_2. То се може учинити поделом земљишта на делове Δ_0, Δ_1 и Δ_2 и затим тестирањем у складу са следећим распоредом:*

	Δ_0	Δ_1	Δ_2
Γ_0	s_0	s_1	s_2
Γ_1	s_0	s_1	s_2
Γ_2	s_0	s_1	s_2

Лоша особина оваквог тестирања је чињеница да се свака вредност променљиве S тестира увек на истом земљишту. Експеримент у складу са латинским квадратом

	Δ_0	Δ_1	Δ_2
Γ_0	s_0	s_1	s_2
Γ_1	s_1	s_2	s_0
Γ_2	s_2	s_0	s_1

даје тачније резулате о предностима и манама коришћења супстанце S у сваком годишњем добу, уз минимизирање утицаја земљишта на резултат.

Кодови за исправљање грешака

Тачан пренос информација је важан при преносу података кроз зашумљене канале као што су на пример телефонске линије или при снимању података на компакт дискове. Циљ је, не само да се евентуалне грешке при преносу и запису информација пронађу, већ и да се могу ефикасно исправити.

Скуп речи којима се кодирају информације се назива *код*. Формалније, тај скуп се уводи наредном дефиницијом.

Дефиниција 1.12 *$q-$нарни код дужине n је скуп стрингова дужине n над азбуком $S_q = \{0, 1, 2, ...q - 1\}$.*

Пример 1.12 *Скуп $C_2 = \{00000, 10000, 10110\}$ је бинарни код дужине 5, а $C_3 = \{0000, 0111, 0222, 1012, 1120, 1201, 2021, 2102, 2210\}$ је тернарни код дужине 4.*

За два стринга x и y се Хемингово (R.Hamming) одстојање $d(x, y)$ дефинише као број њихових различитих координата. Тако у коду C_2 из примера 1.12 важи $d(00000, 10000) = 1$, а у коду C_3 за свака два стринга x и y важи $d(x, y) = 3$. На основу Хеминговог одстојања се може дефинисати минимално одстојање кода.

Дефиниција 1.13 *Минимално одстојање кода C је:*

$$d(C) = min\{d(x, y)|x, y \in C\}.$$

На пример, непосредно се проверава да за кодове из примера 1.12 важи $d(C_2) = 1$ и $d(C_3) = 3$.

Ако је вредност $d(C)$ већа, проналажење и исправљање грешки у кодираним порукама кодом C је једноставније.

Ово се може приметити и у случају кодова из примера 1.12. Грешка у једном симболу у поруци кодираној кодом C_2 не мора бити откривена, док се таква грешка у поруци кодираној кодом C_3 може увек открити и исправити.

Заиста, ако се при кодирању кодом C_2 почетни симбол у стрингу 00000 грешком замени са 1 то неће бити регистровано, већ ће стринг са грешком бити протумачен као стринг 10000 који такође постоји у коду C_2.

У случају кода C_3 оваква ситуација се неће десити, јер се стринг са једном грешком разликује од свих осталих стрингова.

У овом случају грешка се може и исправити јер се стринг са једном грешком од оригиналног стринга разликује у једном симболу, а од осталих у два симбола.

Осим минималног одстојања, битне карактеристике кода су и број речи у њему као и њихова дужина. Већи број речи доприноси већој "изражајности" кода, а мања дужина речи значи ефикаснији пренос информација.

Могуће су конструкције кодова засноване на примени латинских квадрата. Наредна теорема се односи на такву конструкцију.

Теорема 1.13 *Постоји n-арни код са n^2 речи дужине s и минималним одстојањем $s-1$ ако и само ако постоји скуп од $s-2$ међусобно ортогоналних латинских квадрата реда n.*

Наредни пример илуструје значење претходне теореме за $n = 3$ и $s = 4$. У њему се конструише тернарни код са 9 речи дужине 4 и минималним одстојањем 3 на основу 2 ортогонална латинска квадрата реда 3.

Пример 1.13 *Нека су, као у примеру 1.6 дати међусобно ортогонални латински квадрати*

$$L' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \\ \hline \end{array} \text{ и } L'' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \\ \hline \end{array}.$$

Код који садржи стрингове дефинисане колонама матрице

$$M = [i, j, L'_{i,j}, L''_{i,j}]^T, 0 \le i, j < 3$$

има тражене особине. Заиста, стрингови дефинисани са 9 колона матрице

$$M = \begin{array}{|ccccccccc|} \hline 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 \\ \hline \end{array}$$

због ортогоналности квадрата L' и L'' имају највише 1 једнак симбол на истој позицији, то јест њихово минимално одстојање је 3.

Примена у криптографији

Криптографија се бави конструкцијом система за шифровање.

Један од начина креирања таквог система подразумева генерисање тајног кључа K и функције E_K којом се порука M трансформише у шифрован текст C, као и генерисање функције D_K која шифрован текст поново трансформише у поруку M. У наставку ће бити дат једноставан пример генерисања кључа K коришћењем латинских квадрата.

Кључ за шифровање K се може дефинисати као пар међусобно ортогоналних квадрата (L', L''). Ако је (i, j) порука коју треба шифровати и функција E_K се дефинише са

$$E_K(i, j) = (L'_{i,j}, L''_{i,j}) = (\alpha, \beta),$$

дешифровање добијене поруке (α, β) се врши проналажењем координата (i, j) у квадратима L' и L'' које задовољавају услове $L'_{i,j} = \alpha$ и $L''_{i,j} = \beta$. Међусобна ортогоналност квадрата L' и L'' обезбеђује јединственост тако дешифроване поруке.

Пример 1.14 *Ако кључ чине квадрати L' и L'' из примера 1.13, важи*

$$E_K(1, 2) = (L'_{1,2}, L''_{1,2}) = (0, 1)$$

Постоје случајеви када кључ треба да буде подељен између више учесника. Циљ је поделити кључ на n делова тако да се може реконструисати само у присуству више од k делова, где су n и k унапред дефинисани бројеви такви да је $0 < k \leq n$.

Један од начина за конструкцију оваквог кључа се базира на појму *критичног скупа* латинског квадрата.

Дефиниција 1.14 *Критични скуп $C(L)$ латинског квадрата L је скуп уређених тројки (i, j, k) за који важи:*

1. *L је једини латински квадрат за који је $L_{i,j} = k$ за сваку тројку (i, j, k) из скупа $C(L)$*
2. *Не постоји прави подскуп скупа $C(L)$ који има својство 1.*

Критични скуп $C(L)$ је другим речима, парцијални латински квадрат са минималним бројем дефинисаних поља, који се може допунити само до квадрата L.

Пример 1.15 *Скуп* $C(L) = \{(0, 0, 0), (0, 1, 1), (1, 3, 2), (2, 1, 3), (3, 2, 1)\}$ *је критични скуп квадрата*

$$L = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \\ \hline \end{array}$$

Одговарајући парцијални квадрат је

$$C(L) = \begin{array}{|cccc|} \hline 0 & 1 & x & x \\ x & x & x & 2 \\ x & 3 & x & x \\ x & x & 1 & x \\ \hline \end{array}$$

Наредни пример илуструје примену технике подељеног кључа на основу појма критичног скупа латинског квадрата.

Пример 1.16 *Нека је кључ K латински квадрат L из претходног примера и нека је сваком од 5 учесника који треба да дешифрују поруку додељена по једна тројка из скупа $C(L)$. У присуству свих учесника се може конструисати јединствен кључ L, а мање од 5 учесника то не могу да ураде.*

Магични квадрати и судоку

Магични квадрат реда n је матрица димензија $n \times n$ чији елементи су различити бројеви из скупа $\{0, 1, 2, ...n^2 - 1\}$ тако распоређени да је збир бројева у свакој врсти, свакој колони и у обе главне дијагонале једнак.

Пример 1.17 *Матрица*

$$M = \begin{array}{|cccc|} \hline 0 & 7 & 9 & 14 \\ 10 & 13 & 3 & 4 \\ 15 & 8 & 6 & 1 \\ 5 & 2 & 12 & 11 \\ \hline \end{array}$$

је магични квадрата реда 4.

Веза између латинских и магичних квадрата је дата у наредној теореми. Придев *дијагонални* који постоји у њој се односи на квадрат чије обе дијагонале садрже међусобно различите елементе.

Теорема 1.14 *Нека су L' и L'' два дијагонална, међусобно ортогонална латинска квадрата реда n. Матрица*

$$M = n \cdot L' + L'' \tag{1.3}$$

је магични квадрат.

Доказ: За свако i, $0 \leq i < n$ збир елемената у $i-$тој врсти квадрата M је

$$\sum_{j=0}^{n-1} n \cdot L'_{i,j} + \sum_{j=0}^{n-1} L''_{i,j} = n \cdot \frac{n(n-1)}{2} + \frac{n(n-1)}{2} = \frac{n(n^2-1)}{2}$$

Исти резултат се добија и за сваку колону, као и обе дијагонале квадрата M.

Пример 1.18 *Нека су дати квадрати*

$$L' = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ 3 & 2 & 1 & 0 \\ 1 & 0 & 3 & 2 \\ \hline\end{array} \text{ и } L'' = \begin{array}{|cccc|}\hline 0 & 3 & 1 & 2 \\ 2 & 1 & 3 & 0 \\ 3 & 0 & 2 & 1 \\ 1 & 2 & 0 & 3 \\ \hline\end{array}.$$

Применом једнакости 1.3 на L' и L'' добија се магични квадрат M из претходног примера.

Најпознатији латински квадрат је *судоку* (Јап. *Suuji wa dokushin ni kagiru* - "број се сме појавити само једном").

Судоку је латински квадрат реда 9 подељен на 9 квадрата реда 3, такав да су сви елементи у мањим квадратима различити.

Пример 1.19 *Латински квадрат*

$$L = \begin{array}{|ccc|ccc|ccc|}\hline 4 & 2 & 3 & 5 & 6 & 7 & 8 & 0 & 1 \\ 5 & 6 & 1 & 0 & 8 & 4 & 2 & 3 & 7 \\ 0 & 8 & 7 & 2 & 3 & 1 & 4 & 5 & 6 \\ \hline 7 & 4 & 8 & 6 & 5 & 0 & 3 & 1 & 2 \\ 3 & 1 & 5 & 7 & 4 & 2 & 6 & 8 & 0 \\ 6 & 0 & 2 & 8 & 1 & 3 & 7 & 4 & 5 \\ \hline 8 & 5 & 0 & 4 & 2 & 6 & 1 & 7 & 3 \\ 1 & 7 & 6 & 3 & 0 & 8 & 5 & 2 & 4 \\ 2 & 3 & 4 & 1 & 7 & 5 & 0 & 6 & 8 \\ \hline\end{array}$$

је судоку квадрат.

Око $0,00012\%$ свих латинских квадрата реда 9 јесу судоку квадрати (Фелгенауер (B.Felgenhauer) и Јарвис (F.Jarvis), 2005. године).

2

Број латинских правоугаоника

У овом поглављу се разматра укупан број латинских правоугаоника, као и веза овог броја са бројем редукованих правоугаоника. Први од ова два броја ће у наставку бити означен са $\Lambda_{k,n}$, а други са $R_{k,n}$. Поред тога, користе се и ознаке $\Lambda_n \stackrel{\text{def}}{=} \Lambda_{n,n}$ и $R_n \stackrel{\text{def}}{=} R_{n,n}$.

Пребројавање се овде односи на све правоугаонике који се разликују на начин уобичајен за разликовање матрица, то јест за два латинска правоугаоника L и L' истих димензија се каже да су различити ако и само ако постоје i и j такви да је $L_{i,j} \neq L'_{i,j}$.

2.1 Експлицитне формуле и неке процене

Размотримо најпре латинске правоугаонике димензија $2 \times n$.

Прва врста оваквог правоугаоника је било која од $n!$ пермутација скупа $\{0, 1, ..., n-1\}$. Друга врста може бити пермутација која се на свакој позицији разликује од прве врсте.

Нека је $k \in \{0, 1, ..., n\}$. Број пермутација које имају бар k симбола на истим позицијама као прва врста је $\binom{n}{k}(n-k)! = \frac{n!}{k!}$. Ако се фиксира прва врста, примењујући формулу укључивања и искључивања, добија се да је број начина на који се може конструисати друга врста

$$\sum_{k=0}^{n}(-1)^k \frac{n!}{k!} = n! - \frac{n!}{1!} + ... + (-1)^n \frac{n!}{n!}.$$

Следи је укупан број латинских правоугаоника димензија $2 \times n$

$$\Lambda_{2,n} = (n!)^2 (1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + ... + \frac{(-1)^n}{n!}).$$

Израз у другој загради представља првих $n+1$ сабирака развоја функције e^x за $x = -1$, па за $n \to \infty$ важи апроксимација:

$$\Lambda_{2,n} \sim \frac{(n!)^2}{e}.$$

За сваки позитиван цео број n грешка ове апроксимације је мања од $\frac{1}{(n+1)!}$, па се и за релативно мале вредности n добијају резултати веома блиски тачним.

Нешто сложенијим разматрањем (видети нпр. [12]) се за број латинских правоугаоника димензија $3 \times n$ може добити:

$$\Lambda_{3,n} \sim \frac{(n!)^3}{e^3}, \quad n \to \infty.$$

Под претпоставком да је $k < \sqrt[3]{n}$, Јамамото (K.Yamamoto) је доказао и аналогно опште тврђење [26]:

$$\Lambda_{k,n} \sim \frac{(n!)^k}{e^{\binom{k}{2}}}, \quad n \to \infty.$$

Претпоставка под којом важи претходна апроксимација говори да та процена није прецизна када су у питању квадрати. Бољу процену за број квадрата даје апроксимација која важи за $k = o(n^{6/7})$, а коју су дали Годсил (C.D.Godsil) и Мекеј [7]:

$$\Lambda_{k,n} \sim (n!)^k (\frac{n(n-1)...(n-k+1)}{n^k})^n (1 - \frac{k}{n})^{-n/2} e^{-k/2}, \quad n \to \infty.$$

За крај, наводимо још две процене за број различитих латинских квадрата реда n [24]:

$$(n!)^{2n} n^{-n^2} \leq \Lambda_n \leq \prod_{k=1}^{n} (k!)^{\frac{n}{k}}$$

$$\sqrt[n^2]{\Lambda_n} \sim \frac{n}{e^2}, \quad n \to \infty$$

2.2 Број редукованих правоугаоника

Овај део излагања разматра однос између укупног броја латинских квадрата и броја редукованих латинских квадрата истог реда. За утврђивање датог односа неопходна је следећа теорема.

Теорема 2.1 *Нека је L произвољан латински квадрат реда n. Постоје пермутација $c \in S_n$ његових колона и пермутација $r \in S_n$ његових врста која фиксира почетну врсту, које квадрат L трансформишу у редукован квадрат.*

Доказ: Нека је c пермутација колона која сортира у растућем поретку елементе почетне врсте квадрата L дајући квадрат L'. Нека је даље, r пермутација врста која сортира у растућем поретку елементе почетне колоне квадрата L' дајући квадрат L''. Непосредно следи да пермутација r фиксира почетну врсту, као и да је L'' редукован квадрат.

Следећи пример је илустрација претходне теореме.

Пример 2.1 *Пермутацијом колона, којом колоне 0, 1 и 2 постају редом колоне 2, 0 и 1 квадрат*

$$L = \begin{array}{|ccc|} \hline 2 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 2 \\ \hline \end{array}$$

се трансформише у квадрат

$$L' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \\ \hline \end{array}\,.$$

Пермутацијом врста, која размењује садржај врста 1 и 2 квадрат L' се трансформише у редукован квадрат

$$L'' = \begin{array}{|ccc|} \hline 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \\ \hline \end{array}\,.$$

Треба напоменути да за латинске правоугаонике *не важи* аналогна теорема. На пример, правоугаоник којег чине почетне две врсте квадрата L из претходног примера се не може пермутацијом колона и пермутацијом врста која фиксира почетну врсту, трансформисати у редукован правоугаоник.

Између укупног броја латинских квадрата Λ_n и броја редукованих квадрата R_n постоји однос дат у наредној теореми.

Теорема 2.2

$$\Lambda_n = n!(n-1)!R_n.$$

Доказ: Ако је дат произвољан латински квадрат реда n, *његове колоне се могу пермутовати на* $n!$ *начина. Свака два од овако добијених квадрата су различита. Након пермутација колона аналогно се може извршити* $(n-1)!$ *пермутација врста, изузимајући почетну. Сви овако добијени квадрати су поново међусобно различити, а различити су и од квадрата добијених примењујући само пермутације колона (јер је почетна врста фиксирана).*

Дакле, кренувши од произвољног редукованог квадрата реда n *се на овај начин може добити* $n!(n-1)!$ *међусобно различитих квадрата. Од њих је, због фиксираности почетне врсте, тачно један (почетни) редукован, тј. наведеним трансформацијама се редуковани квадрат не може трансформисати у други редуковани квадрат.*

За доказ теореме довољно је још доказати да се сваки латински квадрат може на описани начин добити од неког редукованог квадрата, као и да се два редукована квадрата на описан начин не могу трансформисати у исти квадрат.

Први од ових услова је непосредна последица теореме 2.1 - сваки квадрат се може добити од неког редукованог квадрата примењујући инверзне пермутације пермутација r *и* c *из теореме 2.1.*

Претпоставимо да други услов не важи тј. да постоје одговарајуће пермутације колона и врста c_i *и* r_i, $i \in \{1,2\}$ *такве да за два редукована квадрата* $L' \neq L''$ *важи*

$$r_1(c_1(L')) = r_2(c_2(L'')).$$

У том случају би, међутим важило

$$L' = r_1^{-1}(r_2(c_1^{-1}(c_2(L'')))),$$

тј. било би могуће један редукован квадрат трансформисати у други, што је контрадикција.

Аналогно се може доказати и опште тврђење, везано за латинске правоугаонике:

Теорема 2.3

$$\Lambda_{k,n} = \frac{n!(n-1)!}{(n-k)!}R_{k,n}. \tag{2.1}$$

Пребројавање латинских квадрата има дугу историју. Већина радова је за циљ имала одређивање броја квадрата, али је то обично подразумевало и пребројавање правоугаоника. Односи наведени у последње две теореме су разлог због којег се сви аутори баве одређивањем броја редукованих квадрата (правоугаоника).

Број квадрата реда $n \leq 5$ је одредио Ојлер [4] 1782. године. До броја квадрата реда 6 је дошао Фролов (M.Frolov), 1890. године [6]. У истом раду је дат и погрешан број квадрата реда 7, као и у раду Нортона (H.W.Norton) из 1939. године [17]. Тачан број квадрата реда 7 је дао Шаде (A.Sade), 1948. године [21]. Број квадрата реда 8 је одредио Велс (M.B.Wells) 1967. године [25], а број квадрата реда 9 су објавили Бамел (S.E.Bammel) и Ротштајн (J.Rothstein) 1975. године [18].

Број квадрата реда 10 је први одредио Роговски (E.Rogoyski) 1990. године, а тај број је потврдио и Мекеј следеће године. Ови резултати су објављени у [13]. Тадашњи ниво развоја хардвера није био довољан да се дође до броја квадрата реда 11, али је 2005. године применом скоро идентичног алгоритма то успело Мекеју и Ванлесу (I.M.Wanless) [15]. Резултати као и алгоритам наведен у овом раду ће бити размотрени у наставку.

У тачки 1.2.2 је разматран однос између бипартитних графова и латинских правоугаоника. Еквиваленција правоугаоника и 1-факторизације одговарајућег графа је у [15] послужила као основа за одређивање броја латинских правоугаоника.

Из теореме 1.7 непосредно следи да је укупан број латинских правоугаоника димензија $k \times n$ једнак:

$$\Lambda_{k,n} = \sum_{G} k! M(G). \tag{2.2}$$

Овде се сумирање врши по свим k-регуларним бипартитиним графовима G са укупно $2n$ чворова, а $M(G)$ је број могућих 1-факторизација графа G.

Како је $M(G)$ инваријанта за изоморфизме графа G, односно сви изоморфни графови имају једнак број 1-факторизација, следи да се претходно сумирање може вршити по мањем скупу $\beta(k,n)$ који садржи по тачно једног представника из сваке изоморфне класе $k-$регуларних бипартитиних графова са укупно $2n$ чворова, односно важи:

Теорема 2.4 *Број редукованих латинских правоугаоника димензија $k\times n$ је:*

$$R_{k,n} = 2nk!(n-k)! \cdot \sum_{G\in\beta(k,n)} \frac{M(G)}{|Aut(G)|}. \tag{2.3}$$

Доказ: Из (1.2) број бипартитиних графова изоморфних са са сваким графом G из скупа $\beta(k,n)$ је $\frac{2(n!)^2}{|Aut(G)|}$, па је на основу (2.2) $\Lambda_{k,n} = \sum_{G\in\beta(k,n)} \frac{2(n!)^2}{|Aut(G)|}k!M(G)$. Одавде се заменом (2.1), непосредно добија тражено тврђење.

Специјално за $k = n$ се за број редукованих латинских квадрата реда n добија

$$R_n = \frac{M(K_n)}{(n-1)!}.$$

Овде је K_n ознака за *потпун* бипартитни граф са укупно $2n$ чворова.

За пребројавање латинских правоугаоника Мекеј и Ванлес су у [15] применили следећи поступак:

- Користећи програм NAUTY, за све парове (k,n) $1 \le k \le n \le 11$ одређени су скупови $\beta(k,n)$.
- На основу (1.1) одређена је вредност $M(G)$ за сваки граф $G \in \beta(k,n)$.
- На основу (2.3) израчунате су вредности $R_{k,n}$.

Када је у питању први корак претходног поступка, треба напоменути да је довољно одредити скупове $\beta(k,n)$ само за $k \le \frac{n}{2}$. За $k > \frac{n}{2}$ графови у скупу $\beta(k,n)$ су комплементи оних графова који су у скуповима $\beta(n-k,n)$.

Главну тешкоћу у целом алгоритму представља велики број графова у скуповима $\beta(k,n)$. На пример, највећи број графова за $n = 11$ се добија када је $k = 5$ и он је око $8 \cdot 10^7$. Време потребно да би се дошло до објављених резултата је, према ауторима, било око 2 године (на процесору Pentium 3, који је радио на фреквенцији од 1 GhZ).

Мало је вероватно да ће ускоро бити могуће израчунати број R_{12}, јер је на пример број графова у скупу $\beta(6,12)$ већи од 10^{11}.

Резултати добијени за $n \le 11$ приказани су у табели 2.1. На крају табеле су процене броја редукованих латинских квадрата реда $12 \le n \le 15$ [13].

k	n	$R_{k,n}$	k	n	$R_{k,n}$
1	1	1	1	9	1
			2	9	16.687
1	2	1	3	9	103.443.808
2	2	1	4	9	207.624.560.256
			5	9	112.681.643.083.776
1	3	1	6	9	12.952.605.404.381.184
2	3	1	7	9	224.382.967.916.691.456
3	3	1	8	9	377.597.570.964.258.816
			9	9	377.597.570.964.258.816
1	4	1			
2	4	3	1	10	1
3	4	4	2	10	148.329
4	4	4	3	10	8.154.999.232
			4	10	147.174.521.059.584
1	5	1	5	10	746.988.383.076.286.464
2	5	11	6	10	870.735.405.591.003.709.440
3	5	46	7	10	177.144.296.983.054.185.922.560
4	5	56	8	10	4.292.039.421.591.854.273.003.520
5	5	56	9	10	7.580.721.483.160.132.811.489.280
			10	10	7.580.721.483.160.132.811.489.280
1	6	1			
2	6	53	1	11	1
3	6	1.064	2	11	1.468.457
4	6	6.552	3	11	798.030.483.328
5	6	9.408	4	11	143.968.880.078.466.048
6	6	9.408	5	11	7.533.492.323.047.902.093.312
			6	11	96.299.552.373.292.505.158.778.880
1	7	1	7	11	240.123.216.475.173.515.502.173.552.640
2	7	309	8	11	86.108.204.357.787.266.780.858.343.751.680
3	7	35.792	9	11	2.905.990.310.033.882.693.113.989.027.594.240
4	7	1.293.216	10	11	5.363.937.773.277.371.298.119.673.540.771.840
5	7	11.270.400	11	11	5.363.937.773.277.371.298.119.673.540.771.840
6	7	16.942.080			
7	7	16.942.080			
1	8	1			
2	8	2.119			
3	8	1.673.792			
4	8	420.909.504			
5	8	27.206.658.048	12	12	$\sim 1,62 \cdot 10^{44}$
6	8	335.390.189.568	13	13	$\sim 2,51 \cdot 10^{56}$
7	8	535.281.401.856	14	14	$\sim 2,33 \cdot 10^{70}$
8	8	535.281.401.856	15	15	$\sim 1,50 \cdot 10^{86}$

Табела 2.1: број редукованих латинских правоугаоника

3

Класе латинских квадрата

Пермутацијама врста и колона, као и преименовањем (пермутацијом) симбола латинског квадрата се добија нови латински квадрат, еквивалентан полазном. На овај начин, у зависности од броја и врста пермутација се на скупу латинских квадрата може уочити више класа еквиваленције, о којима ће у наставку бити речи.

Прва класификација која ће бити описана (класе изотопије) се може применити на скуп латинских правоугаоника, док су наредне две (изоморфне класе и главне класе) везане искључиво за латинске квадрате.

Симбол S_n који ће се при овим дефиницијама често користити је уобичајена ознака за симетричну групу (групу свих пермутација) скупа $\{0, 1, ..., n-1\}$.

За пермутацију

$$\begin{pmatrix} 0 & 1 & 2 & ... & n-1 \\ p(0) & p(1) & p(2) & ... & p(n-1) \end{pmatrix}$$

у наставку ће бити коришћен запис

$$\begin{pmatrix} p(0) & p(1) & p(2) & ... & p(n-1) \end{pmatrix}.$$

3.1 Класе изотопних латинских правоугаоника

Дефиниција 3.1 *За два латинска правоугаоника L и L' димензија $k \times n$ се каже да су* изотопни *ако постоје пермутације $r \in S_k$ и $c, s \in S_n$, такве да важи:*

$$L'_{i,j} = s(L_{r(i),c(j)}), \quad 0 \leq i < k, 0 \leq j < n.$$

Тројка (r, c, s) се у овом случају назива изотопизам. *У случају када је $k = n$ говори се о изотопизму латинских квадрата.*

Пример 3.1 *Правоугаоници*

$$L = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ \hline \end{array} \quad \text{и } L' = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 3 & 0 & 2 \\ 2 & 0 & 3 & 1 \\ \hline \end{array}$$

су изотопни. Одговарајући изотопизам, као што показује наредна табела је $(r = (120),\ \ c = (3021),\ \ s = (0132))$.

i	j	$r(i)$	$c(j)$	$L_{r(i),c(j)}$	$s(L_{r(i),c(j)})$
0	0	1	3	0	0
0	1	1	0	1	1
0	2	1	2	3	2
0	3	1	1	2	3
1	0	2	3	1	1
1	1	2	0	2	3
1	2	2	2	0	0
1	3	2	1	3	2
2	0	0	3	3	2
2	1	0	0	0	0
2	2	0	2	2	3
2	3	0	1	1	1

Изотопизам који правоугаоник L пресликава у себе се назива *аутотопизам* правоугаоника L.

Непосредно се проверава да је *бити изотопан* релација еквиваленције на скупу латинских правоугаоника одређених димензија. Класе еквиваленције ове релације се називају *изотопним класама.*

3.2 Класе изоморфних латинских квадрата

Дефиниција 3.2 *За два латинска квадрата* L *и* L' *реда* n *се каже да су* изоморфни *ако постоји пермутација* $f \in S_n$*, таква да важи:*

$$L'_{i,j} = f(L_{f(i),f(j)}), \quad 0 \le i, j < n.$$

Пермутација f *се у овом случају назива* изоморфизам*.*

Пример 3.2 *Квадрати*

$$L = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 1 & 0 \\ 3 & 2 & 0 & 1 \\ \hline \end{array} \quad \text{и } L' = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ \hline \end{array}$$

су изоморфни. Одговарајући изоморфизам, као што показује наредна табела је $f = (0213)$.

i	j	$f(i)$	$f(j)$	$L_{f(i),f(j)}$	$f(L_{f(i),f(j)})$
0	0	0	0	0	0
0	1	0	2	2	1
0	2	0	1	1	2
0	3	0	3	3	3
1	0	2	0	2	1
1	1	2	2	1	2
1	2	2	1	3	3
1	3	2	3	0	0
2	0	1	0	1	2
2	1	1	2	3	3
2	2	1	1	0	0
2	3	1	3	2	1
3	0	3	0	3	3
3	1	3	2	0	0
3	2	3	1	2	1
3	3	3	3	1	2

Изоморфизам $f \in S_n$ је изотопизам (r, c, s) за који важи $r = c = s = f$.

Изоморфизам који квадрат L пресликава у себе се назива *аутоморфизам* квадрата L.

Релација *бити изоморфан* је релација еквиваленције на скупу латинских квадрата одређене димензије. Одговарајуће класе еквиваленције се називају *изоморфним класама*.

3.3 Главне класе латинских квадрата

Ако се осим пермутација врста и колона, као и применовања (пермутација) симбола разматра и пермутација значења ових појмова, добија се још једна класа еквиваленције латинских квадрата.

Дефиниција 3.3 *За два латинска квадрата L и L' реда n се каже да су* паратопни *ако је квадрат L изотопан са неким конјугатом квадрата L'. Одговарајуће пресликавање се у овом случају назива* паратопизам.

Другим речима, паратопизам је четворка (r, c, s, t), где је (r, c, s) изотопизам, а $t \in S_3$ одговарајућа пермутација врста ортогоналне репрезентације.

Пример 3.3 *Квадрати*

$$L = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 1 & 0 \\ 3 & 2 & 0 & 1 \\ \hline \end{array} \quad \text{и} \quad L'' = \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 3 & 0 & 1 & 2 \\ 2 & 3 & 0 & 1 \\ 1 & 2 & 3 & 0 \\ \hline \end{array}$$

су паратопни. Одговарајући паратопизам је $(r = (0213),\ \ c = (0213),\ \ s = (0213),\ \ t = (021))$.

Заиста, као што је показано у претходном примеру, квадрат L *се изотопизмом* $(r = (0213),\ c = (0213),\ s = (0213))$ *пресликава у квадрат* L' *чија је ортогонална репрезентација (видети пример 1.2):*

$$OR(L') = \begin{array}{|cccccccccccccccc|} \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 & 0 & 2 & 3 & 0 & 1 & 3 & 0 & 1 & 2 \\ \hline \end{array}$$

Пермутацијом $t = (021)$, *која размењује садржај врста 1 и 2 матрице* $OR(L')$, *добија се матрица:*

$$OR(L'') = \begin{array}{|cccccccccccccccc|} \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 & 0 & 2 & 3 & 0 & 1 & 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ \hline \end{array}$$

која је ортогонална репрезентација квадрата L''.

Паратопизам који квадрат L пресликава у себе се назива *аутопаратопизам* квадрата L.

Релација *бити паратопан* је релација еквиваленције. Одговарајуће класе еквиваленције се називају *главним класама*.

Из претходних дефиниција следи да између класа дефинисаних у овој тачки важи импликација:

$$\begin{gathered} L \text{ и } L' \text{ су изоморфни} \\ \Rightarrow \\ L \text{ и } L' \text{ су изотопни} \\ \Rightarrow \\ L \text{ и } L' \text{ су у истој главној класи.} \end{gathered}$$

Дакле, свака главна класа се састоји из једне или више класа изотопије, које се састоје из једне или више изоморфних класа.

3.4 Број класа

Главна тема радова о којима је било речи на страни 27 је укупан број латинских квадрата. Резултати у вези са бројем класа за исти ред квадрата су се по правилу појављивали неколико деценија касније

Тачан број главних и изотопних класа до реда 6 је дао Шонхарт (E.Schonhardt) 1930. године [22].

Први приближан резултат везан за квадрате реда 7, у којем су недостајале по једна главна и изотопна класа, дао је 1939. године Нортон [17], а недостајуће класе је 1951. године дао Шаде [21].

Скоро тачан резултат за ред 8 је објавио Браун (J.W.Brown) 1968. године [1]. У овом раду је недостајало 10 класа изотопије. Тачан број главних класа и класа изотопије квадрата до реда 8 дали су Колесова (G.Kolesova), Лам (C.W.H.Lam) и Тил (L.Thiel) 1990. године [11].

Последњи добијен резултат је број класа квадрата реда 9 и реда 10, дат 2005. године у раду [16] чији су аутори Мекеј, Мејнерт (A.Meynert) и Мирволд (W.Myrvold).

Пре разматрања резултата из [16], биће уведени неки појмови са којима се на другачији, формалнији начин могу дефинисати класе из претходних делова.

Најпре, од важности је појам *дејства групе на скуп.*

Дефиниција 3.4 *Нека је дата група* $(G, \circ)$ *и скуп* S. *Пресликавање* $d : G \times S \to S$, *у ознаци* $d : (g, X) \to gX$ *за које важи:*

1. $1X = X$
2. $(g(hX) = (g \circ h)X$

се назива дејство групе G на скуп S. *Овде су* g *и* h *произвољни елементи групе* G, *а* 1 *је јединични елемент исте групе.*

Ако је X елемент скупа S на који делује група $(G, \circ)$, скуп

$$\{gX | g \in G\}$$

се назива *орбита* елемента X у односу на групу G.

Са друге стране, скуп

$$\{g \in G | gX = X\}$$

се назива стабилизатор елемента X у односу на групу G.

Нека је S_n симетрична група скупа $\{0, 1, 2, ..., n-1\}$ и нека је скуп $\langle S_n^3, S_3 \rangle$ дефинисан као:

$$\langle S_n^3, S_3 \rangle = S_n \times S_n \times S_n \times S_3.$$

Ако је $\cdot$ операција композиције пермутација задата са $(f \cdot g)(X) = f(g(X))$, за два елемента (r, c, s, t) и (r', c', s', t') из скупа $\langle S_n^3, S_3 \rangle$ дефинишимо операцију $\circ$ на следећи начин:

$$(r, c, s, t) \circ (r', c', s', t') = (r \cdot r', c \cdot c', s \cdot s', t \cdot t').$$

Непосредно се проверава да је пар $\Gamma = (\langle S_n^3, S_3 \rangle, \circ)$ група реда $6(n!)^3$.

Нека је даље, Δ_n скуп свих латинских квадрата реда n. *Дејство* призвољног елемента (r, c, s, t) групе Γ, на квадрат $L \in \Delta_n$, чија је ортогонална репрезентација $OR(L) = [i, j, L_{i,j}]^T_{0 \le i,j < n}$ дефинишимо на следећи начин:

$$OR((r, c, s, t)L) = [t(r(i), c(j), s(l_{r(i),c(j)}))]^T_{0 \le i,j < n}.$$

На овај начин је дефинисано дејство групе Γ на скуп Δ_n.

Орбите овако дефинисаног дејства групе на скуп су главне класе латинских квадрата, односно квадрати који су у истој орбити овог дејства су паратопни. Стабилизатор

$$Par(L) = \{(r, c, s, t) \in \Gamma | (r, c, s, t)L = L\}$$

се назива *аутопаратопна група* квадрата L. Њени елементи су аутопаратопизми квадрата L.

Класе изотопних квадрата су орбите дефинисане деловањем групе $\Gamma' \le \Gamma$, чији су елементи оне четворке (r, c, s, t) у којима је $t = id_3$ идентичка пермутација. Стабилизатор

$$Is(L) = \{(r, c, s, id_3) \in \Gamma' | (r, c, s, id_3)L = L\}$$

се назива *аутотопна група* квадрата L. Њени елементи су аутотопизми квадрата L.

Класе изоморфних квадрата су орбите дефинисане деловањем групе $\Gamma'' \le \Gamma'$, чији су елементи оне четворке (r, c, s, id_3) у којима је $r = c = s$. Стабилизатор

$$Aut(L) = \{(r, r, r, id_3) \in \Gamma'' | (r, r, r, id_3)L = L\}$$

се назива *аутоморфна група* квадрата L. Њени елементи су аутоморфизми квадрата L.

Тврђење следеће теореме последица је познатих својстава, када су у питању дејства група на скупове.

Теорема 3.1 *Нека је L латински квадрат реда n. Тада важи:*

1. *Број квадрата у истој главној класи као квадрат L је $\frac{6(n!)^3}{|Par(L)|}$.*

2. *Број изотопних класа у главној класи квадрата L је $\frac{6|Is(L)|}{|Par(L)|}$.*

3. *Број изоморфних класа у изотопној класи квадрата L је $\frac{(n!)^2|Aut(L)|}{|Is(L)|}$.*

Нека је M_n скуп који садржи по један квадрат са нетривијалним аутопаратопизмом из сваке главне класе у којој такав квадрат постоји. Тада важи:

Теорема 3.2 1. *Број главних класа латинских квадрата реда n је*

$$\frac{R_n}{6nn!} + \sum_{L \in M_n} \frac{|Par(L)| - 1}{|Par(L)|}.$$

2. *Број изотопних класа латинских квадрата реда n је*

$$\frac{R_n}{nn!} + 6 \cdot \sum_{L \in M_n} \frac{|Is(L)| - 1}{|Par(L)|}.$$

3. *Број изоморфних класа латинских квадрата реда n је*

$$(n-1)!R_n + 6(n!)^2 \cdot \sum_{L \in M_n} \frac{|Aut(L)| - 1}{|Par(L)|}.$$

Доказ:

Нека је Γ_1 скуп главних класа које имају представника у скупу M_n, а Γ_2 скуп главних класа које немају представника у скупу M_n.

Непсредно следи да је $|\Gamma_1| = |M_n| = \sum_{L \in M_n} 1$.

Одредимо број елемената у скупу Γ_2. Из тврђења 1 теореме 3.1 следи да је број квадрата чије главне класе немају представника у скупу M_n једнак $\Lambda_n - \sum_{L \in M_n} \frac{6(n!)^3}{|Par(L)|}$. Сви такви квадрати имају тривијалне аутопаратопне групе, па се налазе у главним класама реда $6(n!)^3$. Одавде је $|\Gamma_2| = \frac{1}{6(n!)^3}(\Lambda_n - \sum_{L \in M_n} \frac{6(n!)^3}{|Par(L)|})$, па је

$$|\Gamma_1| + |\Gamma_2| = \sum_{L \in M_n} 1 + \frac{1}{6(n!)^3}(\Lambda_n - \sum_{L \in M_n} \frac{6(n!)^3}{|Par(L)|}).$$

Заменом $\Lambda_n = n!(n-1)!R_n$ у последњу једнакост, директно следи тражено тврђење. Тврђења 2 и 3 се доказују аналогно уз примену теореме 3.1.

n	број главних класа	број изотопних класа
1	1	1
2	1	1
3	1	1
4	2	2
5	2	2
6	12	22
7	147	564
8	283.657	1.676.267
9	19.270.853.541	115.618.721.533
10	34.817.397.894.749.939	208.904.371.354.363.006

n	број изоморфних класа
1	1
2	1
3	5
4	35
5	1.411
6	1.130.531
7	12.198.455.835
8	2.697.818.331.680.661
9	15.224.734.061.438.247.321.497
10	2.750.892.211.809.150.446.995.735.533.513

Табела 3.1: бројеви класа латинских квадрата

Да би дошли до бројева класа, датих у горњој табели, Мекеј и Мирволд [16] су применили следећи поступак:

- Одређене су вредности R_n из табеле 2.1
- Користећи програм NAUTY одређен је скуп M_n и израчунате су вредности $|Par(L)|, |Is(L)|$ и $|Aut(L)|$ за сваки квадрат L из скупа M_n [16].
- Извршена су сумирања из теореме 3.2

4

Алгоритми за класификацију квадрата

На крају претходног поглавља у табели 3.1 су дати бројеви класа који су израчунати, *без генерисања* представника класа. Генерисање одговарајућих представника, тј. класификација латинских квадрата је тема овог поглавља. Излагање је конципирано тако да је свака техника класификације најпре описана на општем нивоу комбинаторних објеката, а затим примењена на латинске квадрате.

Пре саме формулације проблема која је дата у другој тачки, у првој тачки су уведене поједине ознаке и неки математички појмови неопходни за наставак. У трећој тачки се разматра генерисање стабла претраге које садржи објекте које треба класификовати, а четврта тачка садржи алгоритме за класификацију и њихову анализу.

4.1 Канонско пресликавање

Нека је дат скуп Γ и група $(G, \circ)$ која делује на Γ у смислу дефиниције 3.4. На скупу Γ се може дефинисати релација $\sim_G$ на следећи начин:

$$X \sim_G Y \quad \overset{\text{def}}{\Leftrightarrow} \quad (\exists g \in G)\ gX = Y. \tag{4.1}$$

За релацију $\sim_G$ се на основу дефиниције дејства групе на скуп, непосредно може проверити да је релација еквиваленције, односно да на скупу Γ одређује класе еквиваленције.

Пример 4.1 *Нека је* Γ_n *скуп свих латинских квадрата реда* n. *У поглављу 3 су разматране групе* $(S_n^3 \times S_3, \circ)$, $(S_n^3, \circ)$ *и* $(S_n, \circ)$ *и њихово деловање на скуп* Γ_n, *а затим су на основу тог деловања дефинисане редом главне, изотопне и изоморфне класе латинских квадрата.*

За објекте $X, Y \in \Gamma$ за које важи $X \sim_G Y$ каже се да су изоморфни, а одговарајуће пресликавање $g \in G$ се назива изоморфизам.

Изоморфност се може проширити и на парове објеката на следећи начин:

$$(X, Z) \sim_G (Y, W) \overset{\text{def}}{\Leftrightarrow} (\exists g \in G)\ gX = Y\ i\ gZ = W. \qquad (4.2)$$

Непосредно се доказује да из $(X, Z) \sim_G (Y, W)$ следи $X \sim_G Y$ као и $Z \sim_G W$, али обратно не важи.

Појам неопходан у наставку излагања, а који се дефинише на основу овако уведене релације $\sim_G$ је *канонско пресликавање.*

Дефиниција 4.1 Канонско пресликавање скупа Γ у односу на релацију $\sim_G$ *је функција* $\rho : \Gamma \to \Gamma$ *за коју важи:*

1. *за сваки објекат* $X \in \Gamma$ *је* $X \sim_G \rho(X)$.

2. *за сваки објекат* $X \in \Gamma$ *и свако* $g \in G$ *је* $\rho(X) = \rho(gX)$.

Објекат $\rho(X)$ *се назива* канонски облик *(*канонска слика*) објекта* X.

У наставку се подразумева да је *канонско пресликавање* у односу на релацију $\sim_G$ и када то није наведено у називу пресликавања. Takoђе, индекс ће бити изостављен у запису релације $\sim_G$, то јест ако се не нагласи другачије подразумева се да је $\sim$ релација дефинисана изразом (4.1), односно изразом (4.2).

Пример канонског пресликавања је дат у наредној теореми.

Теорема 4.1 *Нека је дата релација потпуног поретка* $\prec$ *на скупу* Γ *на којем делује група* G. *Функција* $\rho : \Gamma \to \Gamma$ *дефинисана са:*

$$\rho(X) = min_{\prec}\{gX | g \in G\}$$

је канонско пресликавање.

Доказ: Функција је добро дефинисана, јер су орбите $\{gX | g \in G\}$ *коначни скупови, па минимуми у односу на релацију* $\prec$ *постоје. Непосредно се доказује да функција задовољава оба услова из дефиниције 4.1.*

Аналогна теорема важи и ако се уместо минималне вредности изабере максимална вредност.

Следеће тврђење непосредно следи из дефиниције канонског пресликавања.

Теорема 4.2 *Ако је* $\rho : \Gamma \rightarrow \Gamma$ *канонско пресликавање, тада за свака два објекта* $X, Y \in \Gamma$ *важи:*

$$X \sim Y \quad \Leftrightarrow \quad \rho(X) = \rho(Y)$$

Доказ:
$\Rightarrow$: *Ако је* $X \sim Y$, *тада постоји* $g \in G$ *тако да је* $Y = gX$. *Из услова 2 дефиниције 4.1 је* $\rho(X) = \rho(gX)$, *па следи* $\rho(X) = \rho(Y)$.
$\Leftarrow$: *Ако је* $\rho(X) = \rho(Y)$, *тада из услова 1 дефиниције 4.1 следи да је* $X \sim \rho(X) = \rho(Y) \sim Y$.

Нека је γ произвољна класа еквиваленције одређена релацијом $\sim$ на скупу Γ. Из претходне теореме следи да сви објекти у класи γ имају јединствену канонску слику γ_ρ одређену пресликавањем ρ. Ова слика је такође, елемент класе γ.

Дефиниција 4.2 *Јединствена канонска слика* γ_ρ *објеката из класе* γ *је* канонски представник класе γ.

Непосредно следи да је објекат X у канонском облику ако и само ако важи

$$X = \rho(X),$$

што је и начин за проверу да ли је неки објекат канонски представник класе.

Следећа тачка садржи пример канонског пресликавања дефинисаног на скупу латинских правоугаоника.

4.1.1 Канонски облик латинског правоугаоника у односу на релацију изотопије

Нека је $\prec$ релација потпуног поретка, дефинисана на скупу вектора димензије n на следећи начин:

$$x \prec y \;\overset{\text{def}}{\Leftrightarrow}\; x_0 < y_0 \;\vee (\exists i, 0 < i < n)(x_i < y_i \wedge (\forall j < i)(x_i = y_i))$$

Овде је $<$ уобичајена релација лексикографског поретка на скупу симбола $\{0, 1, 2, ...\}$ дефинисана са $0 < 1 < 2 < ....$

Ако је $x \prec y$ каже се да је вектор x *лексикографски мањи* од вектора y.

Ако се сваком латинском правоугаонику L димензија $k \times n$ додели јединствена $k \cdot n$-орка v^L на следећи начин:

$$(\forall i, j,\ \ 0 \le i < k,\ \ 0 \le j < n) \qquad v^L_{n \cdot i + j} \stackrel{\text{def}}{=} L_{i,j},$$

релација лексикографског упоређивања се може дефинисати и на скупу латинских правоугаоника:

$$L \prec L' \ \stackrel{\text{def}}{\Leftrightarrow}\ v^L \prec v^{L'}.$$

Ако је $L \prec L'$ каже се да је правоугаоник L *лексикографски мањи* од правоугаоника L'.

Специјално за $k = n$, говори се о лексикографском упоређивању латинских квадрата.

Пример 4.2 *Нека су дати квадрати*

$$L = \begin{array}{|ccc|}\hline 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \\ \hline\end{array} \ \text{и}\ L' = \begin{array}{|ccc|}\hline 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \\ \hline\end{array}$$

Како је $v^L = (012120201)$ *и* $v^{L'} = (012201120)$, *важи* $L \prec L'$.

Нека је G група изотопија која делује на скуп латинских правоугаоника $\Gamma_{k,n}$, тј. група чије деловање омогућава пермутације врста, колона и симбола у латинском правоугаонику (видети дефиницију 3.1).

Како је $\prec$ релација тоталног поретка дефинисана на скупу $\Gamma_{k,n}$, у складу са теоремом 4.1 се може дефинисати канонско пресликавање ρ у односу на релацију изотопије $\sim$ на следећи начин:

$$\rho(X) = Y \ \stackrel{\text{def}}{\Leftrightarrow}\ X \sim Y \wedge (\forall g \in G)\ \ Y \preceq gX.$$

Пресликавање ρ дакле, сваки правоугаоник X слика у лексикографски минималан правоугаоник који је изотопан са X.

4.1.2 Неке особине канонских правоугаоника

Тврђења дата у наставку ове тачке садрже неопходне услове да би правоугаоник био канонски у односу на релацију изотопије. До ових услова је аутор дошао у току рада на овој тези.

Треба напоменути да се и у наставку, као и раније користи индексирање од 0, односно да се почетна врста означава као *врста 0*, а наредна као *врста 1.* Исто важи и за колоне.

Најпре, размотримо почетну врсту и почетну колону.

Теорема 4.3 *Почетна врста и почетна колона канонског правоугаоника садрже елементе поређане у растућем поретку.*

Доказ:

Претпоставимо супротно, да постоји i $(0 \leq i < n-1)$ *тако да у канонском правоугаонику* L *важи* $L_{0,i+1} < L_{0,i}$. *Пермутацијом колона која размењује садржај* i*-те и* $(i+1)$*-ве колоне се добија правоугаоник* L' *који је лексикографски мањи од* L *што је контрадикција. Доказ је аналоган и ако су у питању елементи почетне колоне.*

Специјално, за $k = n$ претходна теорема има следеће значење:

Теорема 4.4 *Канонски представник сваке изотопне класе латинских квадрата је редукован квадрат.*

Наредна теорема даје неопходне услове за врсту 1.

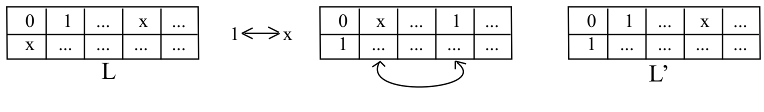
Теорема 4.5 *Нека је* L *правоугаоник димензија* $k \times n$ *који је канонски у односу на релацију изотопије. Тада важи:*

1. $L_{1,0} = 1.$
2. *За свако* j $(0 \leq j < n-1)$ *важи* $L_{1,j} \leq j+1.$
3. *Ако је* $poz(j)$ *позиција на којој се елемент* j *налази у врсти 1 правоугаоника* L*, тада за свако* j $(0 < j \leq n-1)$ *важи*

$$j - 1 \leq poz(j) \leq n - 1.$$

Доказ:

1. *Из теореме 4.3 следи да је* $L_{0,0} = 0$*, па је* $L_{1,0} \neq 0$. *Претпоставимо да је* $L_{1,0} = x > 1$. *Нека је* L' *правоугаоник који се од* L *добија пермутацијом симбола која размењује симболе 1 и* x *и пермутацијом колона која размењује садржај колоне 1 и* x*-те колоне.*



Непосредно следи да је почетна врста правоугаоника L' једнака са почетном врстом правоугаоника L. Takoђe, важи

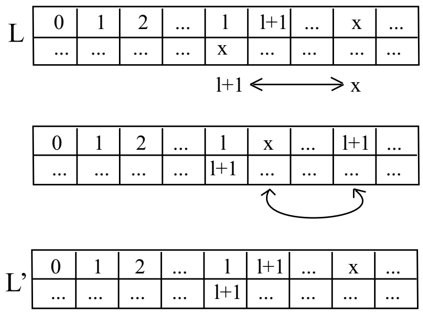
$$L'_{1,0} = 1 < x = L_{1,0},$$

па је $L' < L$ што је контрадикција са претпоставком да је L канонски.

2. *Доказ се може извести индукцијом по j.*

   За $j = 0$ тврђење је доказано у претходном делу ове теореме.

   Нека је $0 < l < n - 1$ и нека тврђење важи за свако $j < l$. Докажимо да тада тврђење важи и за $j = l$. Претпоставимо супротно, да је $L_{1,l} = x > l + 1$. Нека је L' правоугаоник који се од L добија пермутацијом симбола која размењује симболе x и $l + 1$ и пермутацијом колона која размењује садржај x-те и $(l + 1)$-ве колоне.



   Непосредно следи да је почетна врста правоугаоника L' једнака са почетном врстом правоугаоника L. Из индукцијске претпоставке следи да су у врсти 1 правоугаоника L на местима $0, 1, ..., l-1$ елементи мањи од $l+1$, па се овај део правоугаоника наведеним трансформацијама не мења. Takoђe, важи

$$L'_{1,l} = l + 1 < x = L_{1,l},$$

   па је $L' < L$ што је контрадикција са претпоставком да је L канонски.

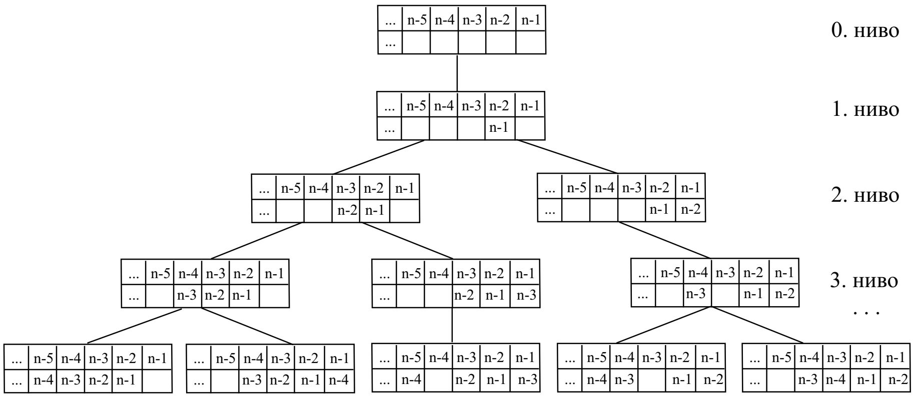
3. *Претпоставимо супротно, то јест да постоји j, $(0 < j \leq n-1)$ тако да је $poz(j) = x < j - 1$, односно да је $L_{1,x} = j$ и $j > x + 1$. Одавде, међутим следи да је $L_{1,x} > x + 1$ што је контрадикција са 2. делом ове теореме.*

Последица теореме 4.3 је да врста 0 канонског правоугаоника садржи идентичку пермутацију, тј. да се врста 0 канонског правоугаоника може конструисати на јединствен начин.

Размотримо на основу теореме 4.5, на колико начина се може конструисати врста 1.

Конструкција може почети од "празне" врсте, тј. пермутације у којој није дефинисан садржај ни на једној позицији. Ова врста се у првом кораку проширује дефинисањем позиције за елемент $n-1$, на све начине допуштене теоремом 4.5. Затим се све добијене врсте проширују дефинисањем позиције за елемент $n-2$, на све начине допуштене теоремом 4.5 итд. У претпоследњем кораку се одређује позиција за елемент 1, а у последњем (n-том кораку) за елемент 0.

Ова конструкција се може представити стаблом τ датим на наредној слици.



Слика 4.1: конструкција врсте 1 канонског правоугаоника

На 0. нивоу стабла τ је "празна" врста 1, а на k-том нивоу ($1 \le k \le n$) се налазе све врсте 1 које испуњавају услове из теореме 4.5 и у којима је дефинисана позиција за елементе $n-1, n-2, ..., n-k$.

Размотримо колико синова има произвољан чвор у овом стаблу.

Теорема 4.6 *Нека је* $0 \le k < n-2$ *и* X *произвољан чвор на* k*-том нивоу стабла* τ*. Тада важи:*

- *Ако на месту* $X_{1,n-k-1}$ није дефинисан *садржај, тада* X *има тачно једног сина.*
- *Ако на месту* $X_{1,n-k-1}$ јесте дефинисан *садржај, тада* X *има тачно два сина.*

Доказ: Према начину конструкције стабла τ *у врсти 1 чвора* X *су дефинисане позиције за тачно* k *елемената:* $n-1,\ n-2,\ ...,\ n-k$.

Из 3. дела теореме 4.5 следи да за позиције тих елемената у врсти 1 важи

$$poz(n-1) \geq n-2, \;\; poz(n-2) \geq n-3, \;\; ..., \;\; poz(n-k) \geq n-k-1.$$

Дакле, свих k дефинисаних позиција у врсти 1 чвора X се налазе у интервалу $[n-k-1, \;\; n-1]$, односно X у том интервалу има тачно једну недефинисану позицију. Нека је та позиција $X_{1,p}$. Показаћемо да ако је $p = n-k-1$, чвор X има тачно једног сина, а ако је $p \neq n-k-1$ чвор X има тачно два сина.

Син чвора X се добија тако што се у врсти 1 чвора X одреди позиција за елемент $n-k-1$. За тражену позицију према 3. делу теореме 4.5 важи

$$poz(n-k-1) \geq n-k-2.$$

Како је $X_{1,p}$ једина позиција у интервалу $[n-k-1, \;\; n-1]$ која није дефинисана следи

$$poz(n-k-1) \in \{n-k-2, \;\; p\}.$$

Дакле, чвор X може имати највише два сина:

чвор Y *у којем је* $Y_{1,n-k-2} = n-k-1$

чвор Z *у којем је* $Z_{1,p} = n-k-1$

Чвор Y се на описан начин увек може дефинисати.
Чвор Z се на описан начин може дефинисати само у случају да је $p \neq n-k-1$, јер би у супротном важило $Z_{1,n-k-1} = n-k-1$, што је контрадикција са чињеницом да је $Z_{0,n-k-1} = n-k-1$, а која следи из теореме 4.3. Овиме је теорема доказана.

Нека је Y син чвора X у стаблу τ. За чвор који је син чвора Y казаћемо да је *унук* чвора X. Последица претходне теореме је наредно тврђење.

Теорема 4.7 *Нека је $0 \leq k < n-3$ и X произвољан чвор на k-том нивоу стабла τ. тада важи:*

- *Ако X има тачно једног сина, тада X има тачно два унука.*
- *Ако X има тачно два сина, тада X има тачно три унука.*

Доказ: Ако је Y једини син чвора X, тада је према доказу претходне теореме садржај на позицији $Y_{1,n-k-2}$ дефинисан. Како је Y на $(k+1)$-ом нивоу стабла τ и важи $k+1 < n-2$, следи да Y према претходној теореми има тачно два сина. Дакле X има тачно два унука.

Ако X осим чвора Y има и сина Z, тада према доказу претходне теореме садржај на позицији $Z_{1,n-k-2}$ није дефинисан. Како је Z на $(k+1)$-ом нивоу стабла τ и важи $k+1 < n-2$, следи да Z према претходној теореми има тачно једног сина. Дакле X у овом случају има тачно три унука (два сина чвора Y и једног сина чвора Z).

На основу претходне две теореме може се одредити рекурентна формула за број чворова на нивоима стабла τ.

Нека је $k \geq 0$ и нека су α_k и β_k бројеви чворова на k-том нивоу стабла τ који редом имају тачно једног сина и тачно два сина. Ако је U_k укупан број чворова на k-том нивоу стабла τ, тада важи:

1. $U_0 = 1$ (из дефиниције стабла τ)

2. $U_1 = 1$ (из 1. дела теореме 4.6 за $k = 0$)

3. За свако $k \leq n$ је $U_k = \alpha_k + \beta_k$ (из дефиниције бројева α_k и β_k)

4. За свако $k \leq n-1$ је $U_{k+1} = \alpha_k + 2\beta_k$ (из дефиниције бројева α_k и β_k)

5. За свако $k < n-3$ је $U_{k+2} = 2\alpha_k + 3\beta_k$ (из теореме 4.7)

6. $U_{n-2} = U_{n-1} = U_n$
   (Ако је X чвор на $(n-2)$-ом нивоу стабла τ, његови синови се добијају дефинисањем позиције за елемент 1. Како је $X_{1,0} = 1$ (1. део теореме 4.5), та позиција је једнозначно одређена, па сваки чвор на $(n-2)$-ом нивоу има тачно једног сина, односно важи $U_{n-2} = U_{n-1}$.

   Ако је X чвор на $(n-1)$-ом нивоу стабла τ, он има $n-1$ дефинисаних позиција, па је преостала позиција за елемент 0 једнозначно одређена. Дакле, сваки чвор на $(n-1)$-ом нивоу има тачно једног сина, односно важи $U_{n-1} = U_n$.)

Сабирањем десних страна треће и четврте једнакости добија се десна страна пете једнакости, па следи да за свако $k < n-3$ важи

$$U_{k+2} = U_k + U_{k+1}.$$

Из претходног следи да су бројеви U_0, $U_1, ..., U_{n-2}$ елементи Фибоначијевог (L.Fibonacci) низа, односно низа дефинисаног на следећи начин:

- $F_0 = F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$ (за $n > 1$).

На основу свега изложеног следи:

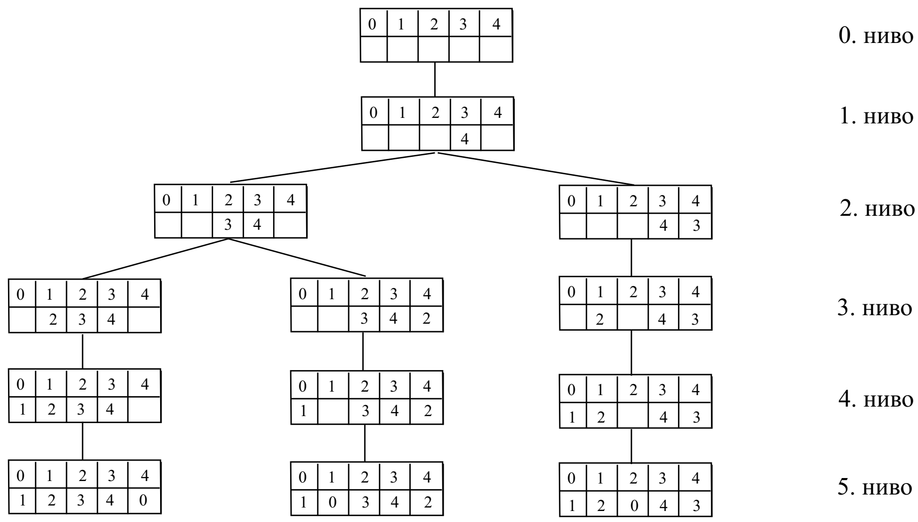
Теорема 4.8 *У латинском правоугаонику са n колона, који је канонски у односу на релацију изотопије, врста 1 се може конструисати на највише F_{n-2} начина.*

Доказ: Врста 1 канонског правоугаоника задовољава услове из теореме 4.5, па је тражени број врста једнак укупном броју чворова на последњем (n-том) нивоу стабла τ. Како за тај број (U_n) важи $U_n = U_{n-1} = U_{n-2} = F_{n-2}$, тврђење је доказано.

Специјално за $k = 2$ претходна теорема има следеће значење:

Теорема 4.9 *Број правоугаоника димензија $2 \times n$ који су канонски у односу на релацију изотопије је највише F_{n-2}.*

Пример 4.3 *На слици је дато стабло чији су листови сви латински правоугаоници димензија 2×5 који испуњавају услове из теорема 4.3 и 4.5.*



Број добијених правоугаоника на 5. нивоу стабла одговара 3. члану Фибоначијевог низа ($F_0 = 1,\ F_1 = 1,\ F_2 = 2,\ F_3 = 3$).

Неопходни услови за каноничност наведени у претходним теоремама могу знатно да допринесу ефикасности алгоритма за одређивање канонског облика задатог правоугаоника. Ефикасност овог алгоритма суштински утиче на алгоритме за класификацију у наредној тачки.

Ако је дат правоугаоник димензија $k \times n$ елементарна метода за одређивање његовог канонског облика захтева $(n!)^2k!$ трансформација. Она подразумева да се правоугаоник трансформише свим пермутацијама симбола, врста и колона и да се запамти лексикографски минималан правоугаоник.

Алгоритам *канонски_правоугаоник* садржи знатно ефикаснију процедуру која непосредно следи из теореме 4.3.

АЛГОРИТАМ 4.1 [канонски_правоугаоник(правоугаоник $L_{k,n}$)]

---

1 мин:= L;
2 for all $p \in S_n$
3   нека је L' добијен од L пермутацијом симбола p;
4   for $(i := 0; i < k; i++)$
5   изврши перм. колона у L' тако да врста i буде растућа;
6   изврши перм. врста у L' тако да колона 0 буде растућа;
7   if ($L' \prec$ мин) мин:= L';
8 return мин;

Овај алгоритам такође захтева разматрање $n!$ пермутација симбола, што се постиже петљом у 2. линији алгоритма. Од 4. до 7. линије се реализује конструкција лексикографски минималног правоугаоника који се од L' може добити пермутацијама врста и колона. У 4. линији се врши избор врсте која ће бити почетна у резултујућем правоугаонику, а затим се у 5. линији елементи изабране врсте сортирају, чиме се одређује пермутација колона. Елементи почетне колоне у добијеном правоугаонику се сортирају у 6. линији алгоритма, што одређује пермутацију врста. Целокупна процедура захтева $kn!$ пермутација у линијама 5 и 6.

Наведени алгоритам се може једноставно трансформисати у *процедуру одлучивања* која даје одговор на питање да ли је дати правоугаоник L канонски. Ова процедура је дата у наредном алгоритму.

АЛГОРИТАМ 4.2 [канон_правоугаоник(правоугаоник $L_{k,n}$)]

---

if L не задовољава неопходне услове из теорема 4.3 и 4.5
   return 0;
for all $p \in S_n$
   нека је L' добијен од L пермутацијом симбола p;
   for $(i := 0; i < k; i++)$
      изврши перм. колона у L' тако да врста i буде растућа;
      изврши перм. врста у L' тако да колона 0 буде растућа;
      if ($L' \prec L$) return 0;
return 1;

Уз помоћ претходног алгоритма се може показати да теорема 4.4 важи само када су у питању квадрати, односно да латински правоугаоник који је канонски у односу на релацију изотопије не мора да буде редукован.

Пример 4.4 *Латински правоугаоници*

$$L_1 = \begin{array}{|cccccc|} \hline 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 & 3 \\ 3 & 4 & 5 & 1 & 2 & 0 \\ \hline \end{array} \text{ и } L_2 = \begin{array}{|cccccc|} \hline 0 & 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 0 & 4 & 5 & 3 \\ 3 & 5 & 4 & 1 & 0 & 2 \\ \hline \end{array}$$

су канонски, иако нису редуковани.

Може се показати и да су L_1 и L_2 правоугаоници најмањих димензија који имају ову особину.

Наиме, из 1. дела теореме 4.5 следи да за свако n, правоугаоник димензија $2 \times n$ који је канонски јесте и редукован.

Размотримо правоугаонике димензија $3 \times 3,\ 3 \times 4,\ 3 \times 5$ и 4×5.

- *У првом случају у питању су квадрати, а за њих по теореми 4.4 важи да ако су канонски, јесу и редуковани.*

- *Од свих правоугаоника димензија 3×4 неопходне услове из теорема 4.3 и 4.5 испуњавају правоугаоници облика*

$$\begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ x & x & x & x \\ \hline \end{array} \text{ и } \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ x & x & x & x \\ \hline \end{array}.$$

Њихов број је 2 и одговара члану F_2 Фибоначијевог низа. Постоје 3 нередукована правоугаоника који су овог облика:

$$\begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 3 & 2 & 0 & 1 \\ \hline \end{array}, \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ \hline \end{array} \text{ и } \begin{array}{|cccc|} \hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 3 & 0 & 1 & 2 \\ \hline \end{array}.$$

Процедура дата у алгоритму 4.2 показује да ниједан од њих није канонски. Изотопизми (записани у складу са дефиницијом 3.1) који наведене правоугаонике трансформишу у лексикографски мање су, редом:

$$\begin{array}{l} (r = (012),\ \ c = (0132),\ \ s = (0132)), \\ (r = (012),\ \ c = (0132),\ \ s = (0132)), \\ (r = (120),\ \ c = (3102),\ \ s = (0213)). \end{array}$$

- *Од свих правоугаоника димензија 3×5 неопходне услове из теорема 4.3 и 4.5, као што је показано у примеру 4.3 испуњавају правоугаоници облика*

$$\begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ x & x & x & x & x \\ \hline \end{array}\,, \quad \begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 & 2 \\ x & x & x & x & x \\ \hline \end{array} \text{ и } \begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 0 & 4 & 3 \\ x & x & x & x & x \\ \hline \end{array}\,.$$

  Њихов број је 3 и одговара члану F_3 Фибоначијевог низа. Постоји 25 нередукованих правоугаоника који су овог облика. Процедура дата у алгоритму 4.2 показује да ниједан од њих није канонски.

- *Од свих правоугаоника димензија 4×5 неопходне услове из теорема 4.3 и 4.5, као што је показано у примеру 4.3 испуњавају правоугаоници облика*

$$\begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ x & x & x & x & x \\ x & x & x & x & x \\ \hline \end{array}\,, \quad \begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 0 & 3 & 4 & 2 \\ x & x & x & x & x \\ x & x & x & x & x \\ \hline \end{array} \text{ и } \begin{array}{|ccccc|}\hline 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 0 & 4 & 3 \\ x & x & x & x & x \\ x & x & x & x & x \\ \hline \end{array}\,.$$

  Њихов број је 3 и одговара члану F_3 Фибоначијевог низа. Постоји 38 нередукованих правоугаоника са растућом почетном колоном који су овог облика. Процедура дата у алгоритму 4.2 показује да ниједан од њих није канонски.

4.2 Класификација комбинаторних објеката

Уопштено речено, класификација комбинаторних објеката се односи на скуп објеката Γ и релацију еквиваленције $\sim$ дефинисану на њему. Проблем класификације за $(\Gamma, \sim)$ се може формулисати на следећи начин:

Генерисати скуп Σ који садржи тачно један објекат из сваке класе еквиваленције дефинисане релацијом $\sim$ на скупу Γ или закључити да је Γ празан скуп.

Релација $\sim$ из претходне формулације може бити произвољна релација еквиваленције, али се у наставку подразумева да је то релација дата изразом (4.1) која је дефинисана дејством групе G на скуп Γ.

Када су у питању латински квадрати, проблем класификације којим се бави ово поглавље се односи на релацију изотопије и може се формулисати на следећи начин:

Генерисати скуп Σ *који садржи тачно један латински квадрат из сваке класе изотопије скупа* Γ_n.

Елементарна и неефикасна техника за класификацију је да се најпре генерише скуп Γ, а затим да се из њега уклони изоморфност примењујући следећи алгоритам:

АЛГОРИТАМ 4.3 [неизоморфни(скуп објеката Γ)]

```
1  Σ = ∅;
2  for all X ∈ Γ
3      if у скупу Σ не постоји Y тако да је X ~ Y
4          Σ := Σ ∪ X;
```

Дакле, за сваки објекат X се испитује да ли је изоморфан са неким објектом који је већ у скупу Σ. Ако то није случај објекат X се додаје скупу Σ, а у супротном се то не ради. Скуп Σ на крају садржи тачно једног представника сваке класе.

Ако се у 2. линији алгоритма елементима скупа Γ приступа по лексикографском поретку, скуп Σ на крају садржи канонског представника сваке класе.

Наставак овог поглавља садржи ефикасније технике за класификацију произвољних објеката, а затим се разматра примена ових техника на проблем класификације латинских квадрата у односу на релацију изотопије.

Алгоритми који се разматрају састоје се од две фазе:

- *генерисање стабла претраге* које садржи све објекте из скупа Γ које треба класификовати.
- *уклањање изоморфизма из стабла претраге*, односно генерисање скупа $\Sigma \subseteq \Gamma$ без међусобно изоморфних објеката, тако да је сваки елемент скупа Γ изоморфан са неким елементом из Σ.

Први део је тема наредне тачке. У њој су дакле, разматрања која се тичу само конструкције објеката, без испитивања изоморфности.

4.3 Конструкција комбинаторних објеката

Алгоритам за конструкцију свих комбинаторних објеката неког типа подразумева постепену процедуру, која креће од иницијалног објекта, проширује га на све могуће начине и преко парцијалних (непотпуних) објеката доводи до потпуних (завршних) објеката.

Структура најпогоднија за развој овакве процедуре је стабло. Корен R стабла може да буде празан или да садржи неки иницијални објекат. Између чворова X и Y постоји грана ако се Y добија у једном кораку проширењем објекта X. Листови на последњем нивоу стабла садрже потпуне објекте, а листови на осталим нивоима непотпуне објекте који се не могу даље проширити.

За сваки чвор X, скуп његових непосредних наследника (синова) у наставку ће бити означен са $C(X)$, а његов непосредни претходник (отац), ако постоји, са $p(X)$.

Конструкција стабла се може реализовати наредним рекурзивним алгоритмом.

АЛГОРИТАМ 4.4 [стабло(чвор X)]

```
1  if X је потпун објекат
2      return;
3  C(X) :=прошири(X);
4  for all Y ∈ C(X)
5      стабло(Y);
```

Стабло је дакле одређено кореном R који је аргумент првог позива процедуре и функцијом *прошири()*. Она за објекат X израчунава скуп његових синова $C(X)$ који се добијају у једном кораку проширујући X на све могуће начине. Алгоритам се затим рекурзивно примењује на сваки објекат из скупа $C(X)$.

Скуп $C(X)$ добијен у 3. линији алгоритма је празан ако је X парцијални објекат који се не може даље проширити, односно ако не постоји потпуни објекат који се добија проширивањем објекта X. У том случају, ток се предаје позивајућој функцији, односно алгоритам се враћа корак уназад (backtracking) и прелази се на нови парцијални објекат.

У наредној тачки се разматра примена управо описане процедуре за конструкцију стабла које на последњем нивоу садржи све латинске квадрате одређеног реда.

4.3.1 Стабло претраге за латинске квадрате

Техника приказана у алгоритму *стабло* у претходној тачки ће најпре бити примењена за проширење латинског правоугаоника новом врстом.

Генерисање скупа Σ свих латинских правоугаоника димензија $(k+1)\times n$, који се могу добити додавањем нове врсте правоугаонику L димензија $k \times n$ се може реализовати позивом рекурзивног алгоритма *новаврста*, ако се као први аргумент наведе правоугаоник L, а као колона од које се врши конструкција нове врсте наведе $c = 0$. Скуп Σ је пре почетка рада процедуре празан, а нова врста се формира постепено, елемент по елемент.

АЛГОРИТАМ 4.5 [новаврста(правоугаоник $L_{k,n}$, колона c)]

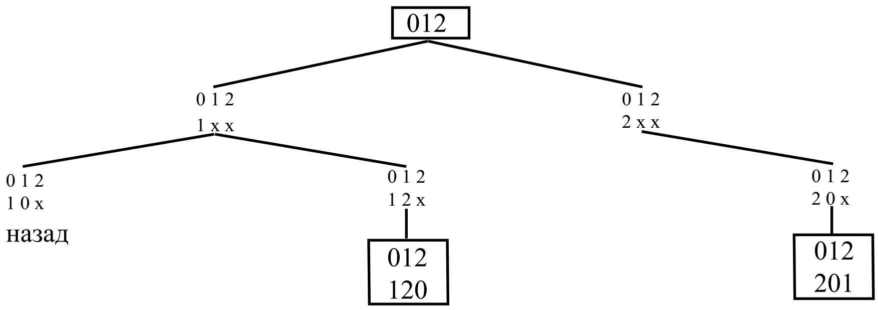
---

1 if $(c = n)$
2     $\Sigma := \Sigma \cup L$;
3 S :=скуп кандидата за позицију c у новој врсти;
4 for all $x \in S$
5     $L_{k,c} := x$;
6   новаврста$(L, c+1)$;

Питање у 1. линији одговара питању "if X је потпун објекат" у 1. линији алгоритма *стабло*. Скуп S у 3. линији се формира у складу са дефиницијом латинских правоугаоника, односно у том скупу су они елементи који у тренутном правоугаонику не постоје у врсти k и колони c. Реализација 3, 4 и 5. линије овог алгоритма одговара позиву функције *прошири()* у алгоритму *стабло*. Скуп Σ на крају садржи све правоугаонике са последњег нивоа стабла, које се имплицитно формира у току рада алгоритма. То су сви правоугаоници димензија $(k+1)\times n$ који се добијају проширивањем правоугаоника L новом врстом.

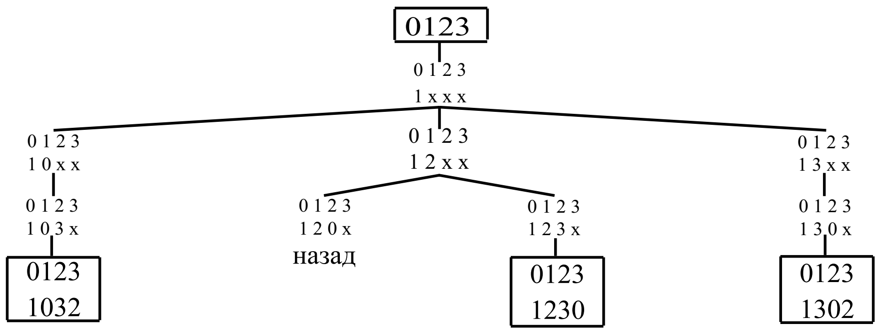
Алгоритам који додаје нову врсту *редукованом* правоугаонику је истог облика, али се у случају када је $c = 0$ скуп S дефинише као једночлан, односно тада је $S = \{k\}$. Овај алгоритам ће у наставку бити означен са *новаврстаP*.

Пример 4.5 *Додавање нове врсте латинском правоугаонику чија је једина врста* 012*, применом алгоритма* новаврста *је представљено стаблом на слици 4.2.*



Слика 4.2: додавање нове врсте латинском правоугаонику

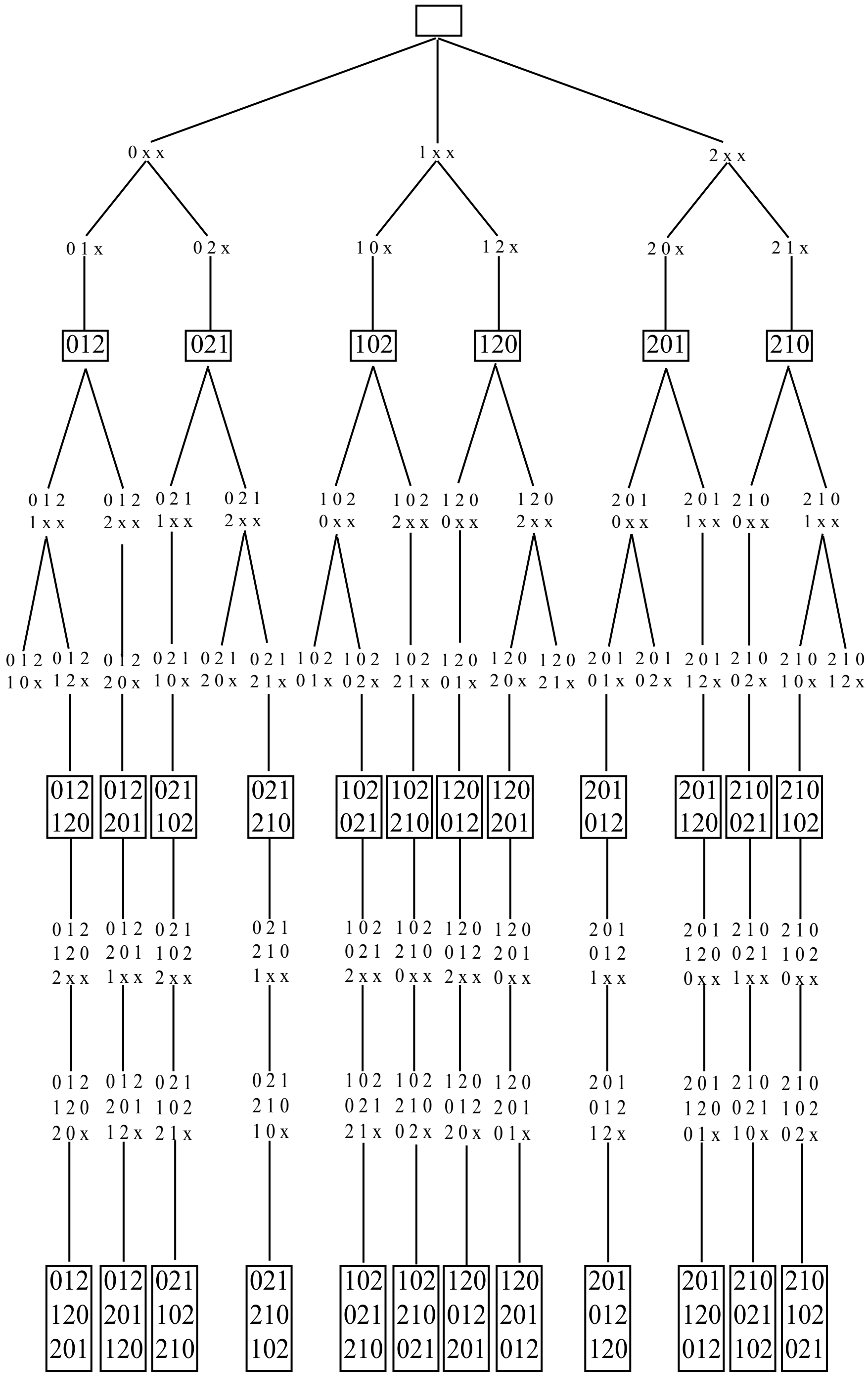
Пример 4.6 *Додавање нове врсте* редукованом *латинском правоугаонику чија је једина врста* 0123*, применом алгоритма* новаврстаР *је представљено стаблом на слици 4.3.*



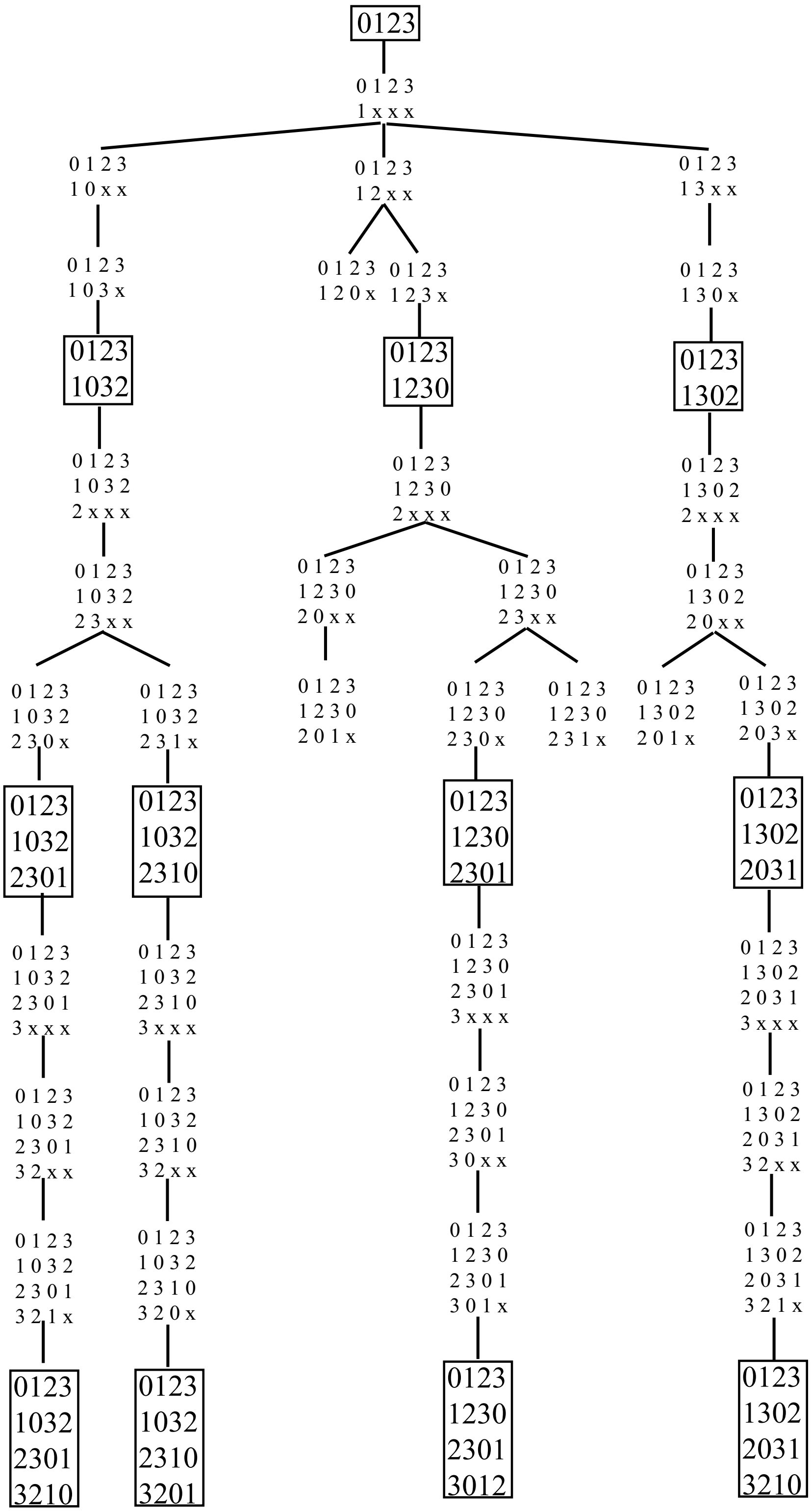
Слика 4.3: додавање нове врсте редукованом правоугаонику

Скуп свих латинских квадрата реда n се може генерисати применом алгоритма *стабло* из претходне тачке на празан корен, ако улогу функције *прошири()* има алгоритам *новаврста*. Стабло које се добија на овај начин на свом k-том нивоу $(1 \leq k \leq n)$ садржи све латинске правоугаонике димензија $k \times n$. У наставку ће се ознака T_n односити на овакво стабло.

Пример 4.7 *На слици 4.4 је дато стабло* T_3. *Број добијених правоугаоника на сваком нивоу је једнак одговарајућим вредностима у табели 2.1 помноженим са* $\frac{n!(n-1)!}{(n-k)!}$.



Слика 4.4: стабло T_3



Слика 4.5: стабло $\overline{T_4}$

Скуп свих *редукованих* латинских квадрата реда n се може генерисати применом алгоритма *стабло* из претходне тачке на корен који садржи врсту 0123...$n-1$, ако улогу функције *прошири()* има алгоритам *новаврстаP*. Стабло које се добија на овај начин на свом k-том нивоу ($1 \leq k \leq n$) садржи све редуковане латинске правоугаонике димензија $k \times n$. У наставку ће се ознака $\overline{T_n}$ односити на овакво стабло.

Пример 4.8 *На слици 4.5 је дато стабло $\overline{T_4}$. Број добијених редукованих правоугаоника на сваком нивоу је једнак одговарајућим вредностима у табели 2.1.*

У оба примера на "спољашњем" нивоу претраге (додавање целих нових врста) нема враћања (backtracking-a), јер се према теореми 1.10 сваком латинском правоугаонику може додати нова врста. Зато је сваки чвор који је потпун правоугаоник корен подстабла чији је бар један лист потпун латински квадрат.

На нивоу "унутрашње" претраге, односно при реализацији алгоритама *новаврста* и *новаврстаP* ситуација је другачија. Овде се дешава да парцијално решење не води потпуном решењу и у тим случајевима ток се предаје позивајућој функцији.

4.4 Алгоритми за класификацију

У претходној тачки смо размотрили алгоритам за конструкцију стабла претраге, а у овој прелазимо на класификацију објеката добијених у том стаблу.

4.4.1 Техника чувања објеката

Нека је T стабло чија свака два изоморфна чвора имају изоморфне синове, то јест стабло за чија свака два чвора X и Y важи:

C 1 $\quad X \sim Y \Rightarrow (\forall Z \in C(X))(\exists W \in C(Y))\ \ Z \sim W$

Показаћемо да алгоритам који следи, када се примени на корен стабла које задовољава услов $C1$, врши класификацију у односу на релацију $\sim$. Алгоритам захтева чување скупа Δ неизоморфних објеката који су обрађени до тренутка позива процедуре. Пре првог позива скуп Δ је празан.

АЛГОРИТАМ 4.6 [record(чвор X)]

```
if (∃Y ∈ Δ тако да је X ∼ Y)
    return;
Δ := Δ ∪ {X};
if(X је потпун објекат)
    Σ := Σ ∪ X;
else
    for all Y ∈ C(X)
        record(Y);
```

Дакле, када претрага дође до објекта X, тестира се постојање објекта Y у скупу Δ који је изоморфан са X. У случају да такав објекат постоји, из даље претраге се одбацује не само X, већ и цело подстабло чији је X корен.

Како стабла претраге на својим вишим нивоима садрже мање (парцијалне) објекте, а потпуни објекти су на најнижем нивоу, на овај начин се постиже уклањање изоморфности већ на нивоу парцијалних објеката, то јест избегава се пролазак кроз велики број подстабала која на својим крајевима имају изоморфне потпуне објекте.

За доказ коректности алгоритма *record*, неопходно је следеће тврђење:

Теорема 4.10 *Нека је γ_X ознака за класу еквиваленције у којој је објекат X. У стаблу претраге које задовољава услов C1 се може дефинисати релација поретка $<$ на класама еквиваленције чворова, таква да за сваки чвор X различит од корена важи $\gamma_{p(X)} < \gamma_X$.*

Доказ:

Нека је G граф конструисан на основу стабла T на следећи начин:

1. *Чворови графа G су класе еквиваленције објеката из стабла T.*

2. *У графу G за сваки чвор X који није корен стабла T постоји грана усмерена од класе $\gamma_{p(X)}$ до класе γ_X.*

Треба доказати да граф G не садржи циклус.

Претпоставимо супротно, односно да постоји циклус у којем су класе еквиваленције $\gamma_0, \gamma_1, ..., \gamma_{m-1}$ тако да за свако $i \in 0, 1, ..., m-1$ постоји грана од γ_i до $\gamma_{i+1(mod\ m)}$. За $j \in 0, 1, ...$ уочимо низ чворова $X_j \in \gamma_j$. За дати објекат X_j постоје објекти $Y \in \gamma_j$ и $Z \in \gamma_{j+1(mod\ m)}$ такви да је $X \sim Y$ и $Y = p(Z)$. Према услову C1 у класи $\gamma_{j+1(mod\ m)}$

постоји објекат X_{j+1} *такав да је* $X_j = p(X_{j+1})$ *и да је* $X_{j+1} \sim Z$*. На овај начин се може конструсиати пут произвољне дужине у стаблу* T *индукован са* $X_0, X_1, \ldots$*. Ово је међутим контрадикција, јер је стабло* T *коначно.*

Дакле, граф G *не садржи циклус, па тополошко сортирање његових класа еквиваленције даје тражену релацију поретка на њима.*

Сада се уз помоћ претходне теореме може доказати коректност алгоритма *record*.

Теорема 4.11 *Када се примени на корен стабла које задовољава услов C1 процедура дата у алгоритму* record *формира скуп* Σ *који садржи тачно један потпун објекат из сваке класе еквиваленције одређене релацијом* $\sim$*.*

Доказ: Према структури алгоритма непосредно следи да скуп Σ *на крају садржи највише један потпун објекат из сваке класе еквиваленције. Треба доказати и обратно, односно да за сваки потпун објекат* X *на крају рада алгоритма* record *у скупу* Σ *постоји објекат* Y *такав да је* $X \sim Y$*.*

Најпре, треба приметити да се објекат из корена стабла налази у скупу Δ*, као и да се сваки потпун објекат који је у скупу* Δ *налази и у скупу* Σ*.*

Претпоставимо да постоји непразан скуп S *потпуних објеката који немају изоморфне објекте у скупу* Σ*, а тиме ни у скупу* Δ*. Из претходне теореме следи да постоји минимална класа чворова из скупа* S*. Нека је* X *елемент те класе. Како* X *није корен стабла, то постоји чвор* $p(X)$*. Чвор* $p(X)$ *није у скупу* S*, јер би у том случају он, а не* X *био минималан у њему. Дакле, постоји објекат* $Y \in \Delta$ *изоморфан са* $p(X)$*. Ово је, међутим контрадикција, јер би тада због испуњености услова C1 у скупу* Δ *постојао и објекат* $Z \in C(Y)$ *изоморфан са* X*.*

У реализацији 1. линије алгоритма *record* се обично користе канонска пресликавања из дефиниције 4.1. Наиме, у скупу Δ се не чувају објекти већ њихови канонски облици, па се испитивање изоморфности реализује упоређивањем канонских облика објеката. Претраживање велике колекције објеката у скупу Δ се врши користећи хеш табелу или неку другу структуру која га убрзава.

Најзначајнија мана технике која је описана у овој тачки је потреба за чувањем великог броја објеката у скупу Δ. Друга тешкоћа је немогућност паралелизовања алгоритма, јер процес претраге мора све време да има приступ свим дотадашњим резултатима.

Потенцијални проблем је и чињеница да испитивање изоморфности, па и упоређивањем канонских облика различитих објеката може да буде веома захтевно што се тиче и временских и просторних ресурса.

У наредној тачки се алгоритам *record* примењује на латинске квадрате и релацију изотопије.

4.4.2 Класификација латинских квадрата техником чувања објеката

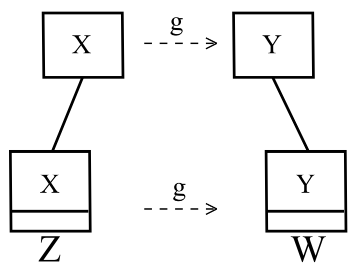
Нека је T_n стабло претраге дефинисано у поглављу 4.3.1, које на свом $k-$том нивоу ($1 \leq k \leq n$) садржи све латинске правоугаонике димензија $k \times n$.

Важи следеће тврђење:

Теорема 4.12 *Стабло T_n испуњава услов C1 у којем је $\sim$ ознака за релацију изотопије.*

Доказ: Нека су X и Y чворови стабла T_n који су изотопни и нека је g одговарајућа изотопија. По конструкцији стабла следи са су X и Y на истом нивоу стабла. Ако је то $k-$ти ниво, тада су X и Y правоугаоници димензија $k \times n$.

Ако је $Z \in C(X)$, према начину конструкције стабла (радом алгоритма новаврста*) првих k врста правоугаоника Z је једнако са првих k врста правоугаоника X.*



Слика 4.6: изотопни правоугаоници X и Y имају изотопне синове Z и W у стаблу претраге

Нека је W правоугаоник који се добија применом изотопије g на Z (слика 4.6). Првих k врста правоугаоника W је једнако са првих k врста правоугаоника Y. Како се скуп $C(Y)$ добија додавањем нове врсте правоугаонику Y на све могуће начине, следи да је $W \in C(Y)$.

Из претходне теореме следи да се процедура дата у алгоритму 4.6 - *record* може применити на стабло T_n.

На тај начин се генеришу представници класа, тј. извршава се класификација латинских квадрата реда n у односу на релацију изотопије.

Нека је T'_n подстабло стабла T_n чије чворове обрађује алгоритам *record*. Број листова на последњем нивоу стабла T'_n је једнак броју класа изотопије квадрата реда n, а број чворова који нису листови на k-том нивоу ($1 \leq k < n$) стабла T'_n, одговара броју класа изотопије правоугаоника димензија $k \times n$.

Пример 4.9 *У горњем делу слике 4.7 је приказано стабло T_3 генерисано у примеру 4.7, а у доњем делу је подстабло T'_3 које обрађује алгоритам* record.

Сви чворови на првом нивоу стабла T_3 се могу добити пермутацијом симбола у чвору P_1, па алгоритам даље разматра само подстабло чији је корен P_1. У преосталом стаблу чворови P_2 и P_3 су изотопни (изотопија која P_2 слика у P_3 је $(r = (10), c = (201), s = (012))$), па се даље разматра само подстабло чији је корен P_2.

Број чворова на трећем нивоу стабла T'_3 одговара резултату из табеле 3.1, односно постоји тачно једна класа изотопије латинских квадрата реда 3.

Техника која је описана састоји се из две фазе:

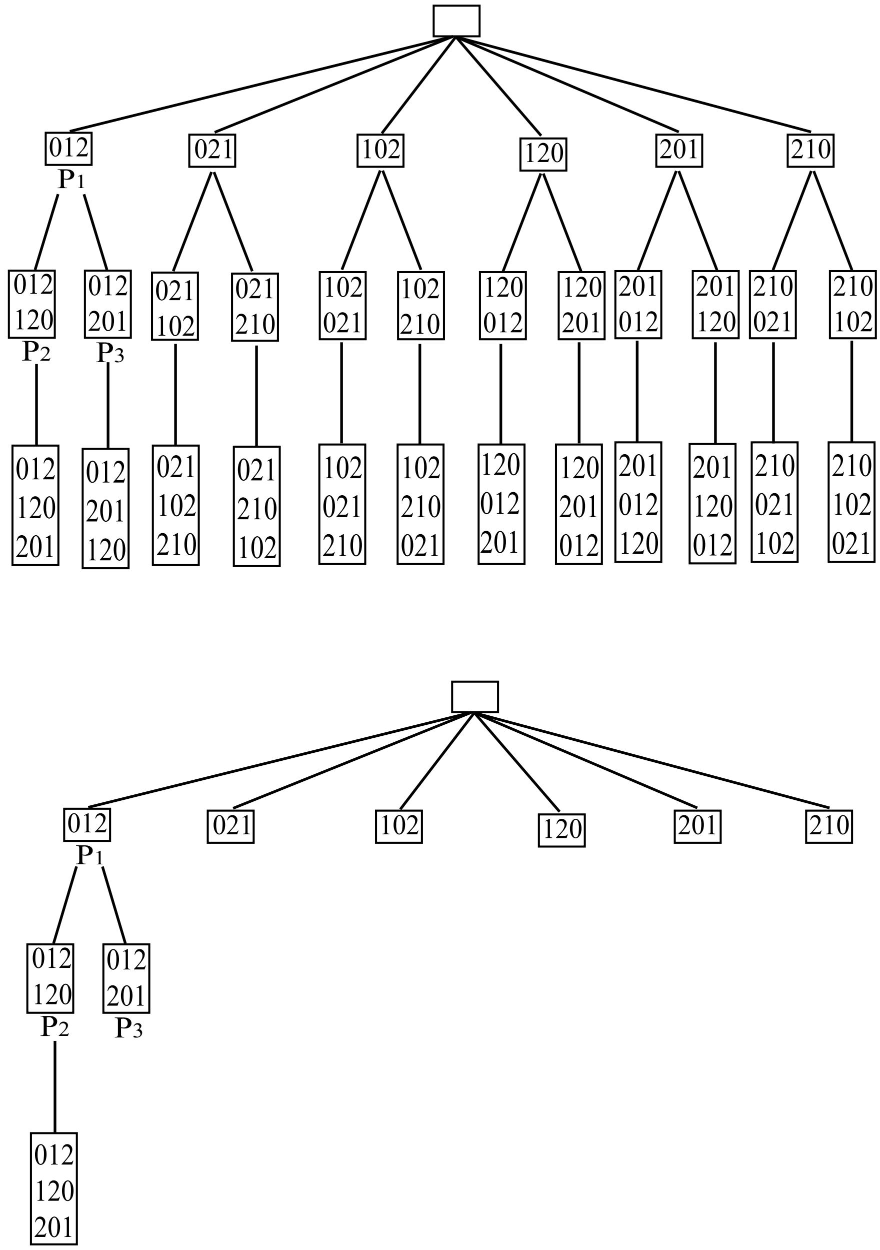
- генерисање стабла претраге (алгоритам 4.4 - *стабло*)
- класификација квадрата (алгоритам 4.6 - *record*)

Знатно убрзање, као и уштеда меморије се постиже спајањем ова два алгоритма, чиме се одсецање подстабала са изотопним квадратима врши још у току самог генерисања стабла.

Алгоритам који ово реализује је *record1*. Он се дакле, не примењује на стабло T_n, већ директно генерише стабло T'_n.

АЛГОРИТАМ 4.7 [record1(чвор X)]

```
if(канонски_правоугаоник(X) ∈ Δ)
    return;
Δ := Δ ∪ канонски_правоугаоник(X);
if(X је потпун квадрат)
    Σ := Σ ∪ X;
else
    C(X) :=новаврста(X, 0);
    for all Y ∈ C(X)
        record1(Y);
```



Слика 4.7: стабло T_3 и подстабло T_3'

Функције *канонски_правоугаоник()* у 1. линији и *новаврста()* у 7. линији се реализују истоименим алгоритмима 4.1 и 4.5.

Коректност алгоритма *record1* је непосредна последица коректности алгоритама *стабло*(4.4) и *record*(4.6), то јест скуп Σ на крају садржи тачно један квадрат из сваке класе изотопије. Једноставним мењањем упита у 4. линији се може постићи да Σ садржи правоугаонике одређених димензија.

4.4.3 Канонска класификација

Техника која се описује у овој тачки има важну особину да се одлука о томе да ли даље разматрати подстабло чији је корен чвор X доноси локално, само на основу својстава чвора X, без испитивања његове изоморфности са другим чворовима. Оваква претрага може бити ефикасно паралелизована, јер се одвојена подстабла могу претраживати независно. Добра особина овог метода је и непостојање потребе за меморисањем парцијалних објеката.

Нека је Γ скуп објеката које треба класификовати у односу на релацију $\sim$ и $\rho : \Gamma \to \Gamma$ канонско пресликавање у односу на исту релацију (дефиниција 4.1). Нека је даље, T стабло са кореном R које садржи све објекте из Γ и које задовољава следеће услове:

C 2 Канонски облик сваког потпуног објекта је чвор у стаблу T.

C 3 За сваки чвор $X \neq R$ који је у канонском облику, чвор $p(X)$ је у канонском облику.

Размотримо наредни алгоритам:

АЛГОРИТАМ 4.8 [канонска_класификација(чвор X)]

```
if(X ≠ ρ(X))
    return;
if(X је потпун објекат)
    Σ := Σ ∪ X;
else
    for all Y ∈ C(X)
        канонска_класификација(Y);
```

Важи тврђење:

Теорема 4.13 *Када се на корен стабла које задовољава услове C2 и C3 примени алгоритам* канонска_класификација *формира се скуп* Σ *који садржи тачно један потпун објекат из сваке класе одређене релацијом* $\sim$.

Доказ: Из 1. и 2. линије алгоритма следи да процедура разматра само објекте који су у канонском облику, па непосредно следи да ће највише један потпун објекат из сваке класе еквиваленције бити додат скупу Σ. *Треба показати и обратно, односно да ће из сваке класе потпуних објеката бар један елемент бити додат скупу* Σ.

Нека је X *произвољан потпун објекат. Према услову C2 његов канонски облик* $\rho(X)$ *је чвор стабла. Из услова C3 следи да су сви претходници објекта* $\rho(X)$ *укључујући и корен стабла, такође у канонском облику. Одавде следи да алгоритам* канонска_класификација *обрађује цело подстабло*

$$R, ..., p(p(\rho(X))), p(\rho(X)), \rho(X)$$

и на крају $\rho(X)$ *додаје у скуп* Σ, *јер је* $\rho(X)$ *потпун објекат.*

У наредној тачки се управо описана процедура примењује на латинске квадрате и релацију изотопије.

4.4.4 Канонска класификација латинских квадрата

Нека су T_n и $\overline{T_n}$ стабла претраге дефинисана у поглављу 4.3.1, која на свом $k-$том нивоу $(1 \leq k \leq n)$ садрже редом, све латинске правоугаонике димензија $k \times n$ и све редуковане латинске правоугаонике истих димензија.

Услове неопходне за примену алгоритма *канонска_класификација* у односу на релацију изотопије испуњавају и стабло T_n и стабло $\overline{T_n}$. Због тога се алгоритам може применити на стабло $\overline{T_n}$ које има мање чворова, па се може ефикасније обрадити.

Важи следеће тврђење:

Теорема 4.14 *Стабло* $\overline{T_n}$ *испуњава услове C2 и C3 у којима је* $\sim$ *ознака за релацију изотопије.*

Доказ:

Услов C2 је испуњен, јер алгоритам новаврстаP() *при генерисању стабла* $\overline{T_n}$ *на последњем нивоу генерише све редуковане квадрате, а канонски представник сваке класе изотопије латинских квадрата је према теореми 4.4 редукован.*

Услов C3 је такође испуњен. Претпоставимо супротно, то јест да у стаблу $\overline{T_n}$ *постоји правоугаоник* L *који је канонски, а да његов непосредни претходник* $p(L)$ *није канонски. У том случају би постојала изотопија* g *таква да је* $gp(L) \prec p(L)$. *Међутим, тада би непосредно следило и* $gL \prec L$, *што је контрадикција.*

Из претходне теореме следи да се процедура дата у алгоритму 4.8 - *канонска_класификација* може применити на стабло $\overline{T_n}$ и на тај начин извршити класификација редукованих (а тиме и свих) латинских квадрата реда n у односу на релацију изотопије.

Нека је $\overline{T'_n}$ подстабло стабла $\overline{T_n}$ чије чворове обрађује алгоритам *канонска_класификација*.

На последњем нивоу стабла $\overline{T'_n}$ су сви редуковани квадрати реда n који су канонски објекти, то јест који су канонски представници сопствених класа изотопије. Како је према теореми 4.4 канонски представник сваке класе изотопије латинских квадрата редукован, то број чворова на последњем нивоу стабла $\overline{T'_n}$ одговара броју класа изотопије латинских квадрата реда n.

Размотримо остале нивое стабла $\overline{T'_n}$. Објекти на k-том нивоу $(1 \leq k < n)$ овог стабла који нису листови су редуковани правоугаоници димензија $k \times n$ који су канонски (лексикографски минимални) у својој класи изотопије.

Пример 4.4 показује да канонски представник класе изотопије латинских правоугаоника не мора да буде редукован, па је број правоугаоника на k-том нивоу стабла $\overline{T'_n}$ који нису листови мањи од броја класа изотопије латинских правоугаоника димензија $k \times n$. Он је једнак броју оних класа чији канонски представник јесте редукован.

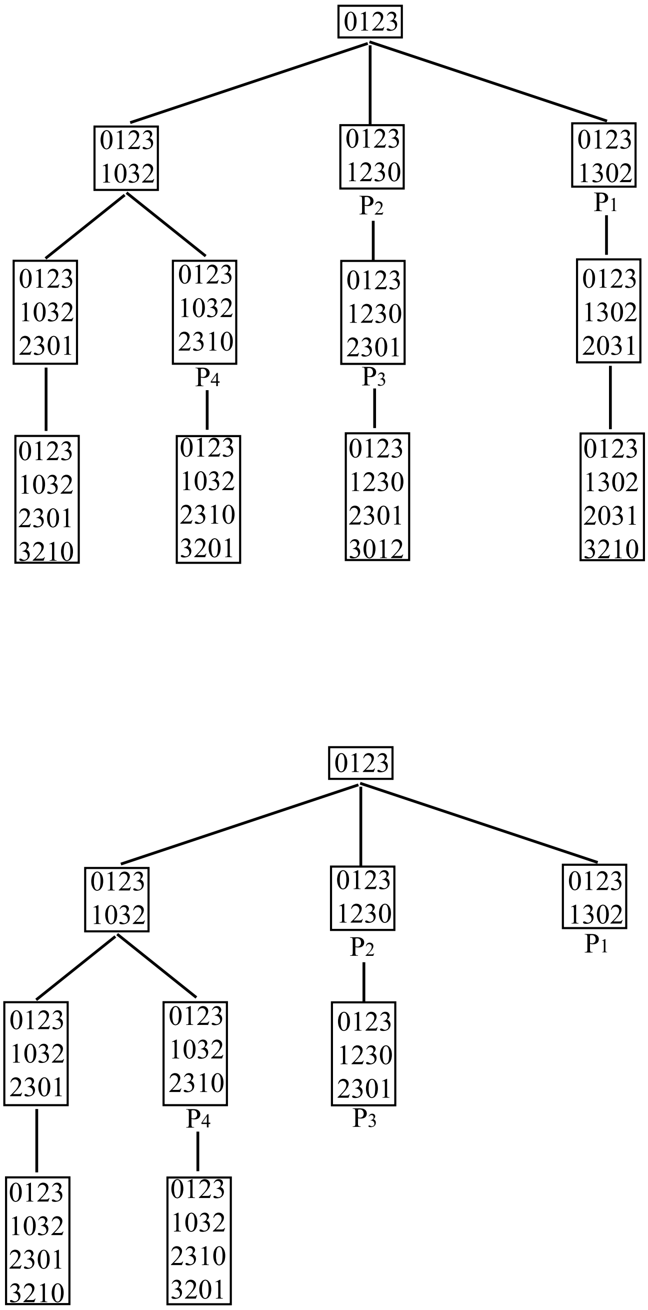
Пример 4.10 *У горњем делу слике 4.8 је приказано стабло* $\overline{T_4}$ *генерисано у примеру 4.8, а у доњем делу је одговарајуће подстабло* $\overline{T'_4}$.

На првом нивоу стабла $\overline{T_4}$ *правоугаоник* P_1 *није канонски јер важи* $P_1 \neq \rho(P_1) = P_2$, *па долази до одсецања подстабла чији је корен* P_1.

У преосталом стаблу правоугаоник P_3 *није канонски зато што је* $P_3 \neq \rho(P_3) = P_4$, *па долази до одсецања и подстабла чији је корен* P_3.

Број чворова на четвртом нивоу стабла $\overline{T'_4}$ *одговара резултату из табеле 3.1, то јест постоје две класе изотопије латинских квадрата реда 4.*

Број чворова на првом, другом и трећем нивоу стабла $\overline{T'_4}$ *који нису листови, одговара броју оних класа изотопије правоугаоника димензија* 1×4, 2×4 *и* 3×4 *чији канонски представници су редуковани правоугаоници.*



Слика 4.8: стабло $\overline{T_4}$ и подстабло $\overline{T_4'}$

Техника која је описана састоји се из две фазе:

- генерисање стабла претраге (алгоритам 4.4 - *стабло*)
- класификација (алгоритам 4.8 - *канонска_класификација*)

Као и у преходној тачки, описана техника се може знатно побољшати спајањем ова два алгоритма, чиме се одсецање подстабала са неканонским правоугаоницима врши још у току самог генерисања стабла.

Алгоритам који ово реализује је *канонска_класификација_1*. Он се дакле, не примењује на стабло $\overline{T_n}$, већ директно генерише стабло $\overline{T'_n}$.

АЛГОРИТАМ 4.9 [канонска_класификација_1(чвор X)]

```
1 if(¬канон_правоугаоник(X))
2     return;
3 if(X је потпун квадрат)
4     Σ := Σ ∪ X;
5 else
6     C(X) :=новаврстаP(X, 0);
7     for all Y ∈ C(X)
8         канонска_класификација_1(Y);
```

Функције *канон_правоугаоник()* у 1. линији и *новаврстаP()* у 6. линији се реализују истоименим алгоритмима 4.2 и 4.5.

Коректност алгоритма *канонска_класификација_1* је непосредна последица коректности алгоритама *стабло*(4.4) и *канонска_класификација*(4.8), то јест скуп Σ на крају садржи по један квадрат из сваке класе изотопије.

Ако се 3. линија замени линијом

if(X је правоугаоник димензија $k \times n$)

скуп Σ на крају садржи редуковане правоугаонике димензија $k \times n$ који су канонски.

4.4.5 Слабо канонско проширење

У овој тачки се разматра техника канонског проширења коју је дефинисао Мекеј [14]. Главно својство ове технике је да се објекти генеришу на карактеристичан начин, одакле потиче и назив поступка.

Нека је T стабло претраге са кореном R, које садржи објекте које треба класификовати у односу на релацију $\sim$. Нека је даље, ω функција која сваком објекту $X \neq R$ додељује објекат $\omega(X)$, такав да за свака два чвора X и Y у стаблу T важи:

$$X \sim Y \Rightarrow \omega(X) \sim \omega(Y). \tag{4.3}$$

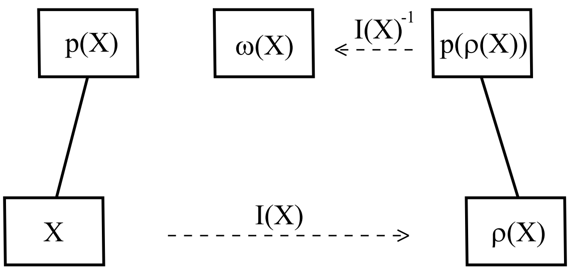
Објекат $\omega(X)$ се назива *слаб канонски отац објекта* X.

Ово пресликавање је увек могуће дефинисати. Тривијално се то може учинити тако што ω слика сваки објекат у корен стабла, а пресликавање које се уобичајено користи при класификацији је дато у наредној теореми.

Теорема 4.15 *Нека је $I(X)$ изоморфизам који објекат X пресликава у његов канонски облик $\rho(X)$ у односу на релацију $\sim$. Пресликавање ω дефинисано са:*

$$\omega(X) = I(X)^{-1}(p(\rho(X))) \tag{4.4}$$

задовољава услов (4.3), тј. $\omega(X)$ је слаб канонски отац објекта X.



Слика 4.9: конструкција слабог канонског оца $\omega(X)$ објекта X

Доказ: Нека су X и Y два изоморфна објекта. Докажимо да тада важи и $\omega(X) \sim \omega(Y)$, то јест да постоји изоморфизам који пресликава објекат $\omega(X)$ у објекат $\omega(Y)$.

Нека је $g = I(Y)^{-1}I(X)$. Непосредно из дефиниције пресликавања ω и I, као и чињенице да због $X \sim Y$ важи $\rho(X) = \rho(Y)$, следи:

$$g\omega(X) = I(Y)^{-1}I(X)I(X)^{-1}(p(\rho(X))) = I(Y)^{-1}(p(\rho(Y))) = \omega(Y).$$

Дакле, g је тражени изоморфизам који пресликава објекат $\omega(X)$ у објекат $\omega(Y)$.

У наредном примеру је размотрена конструкција слабог канонског оца латинског правоугаоника.

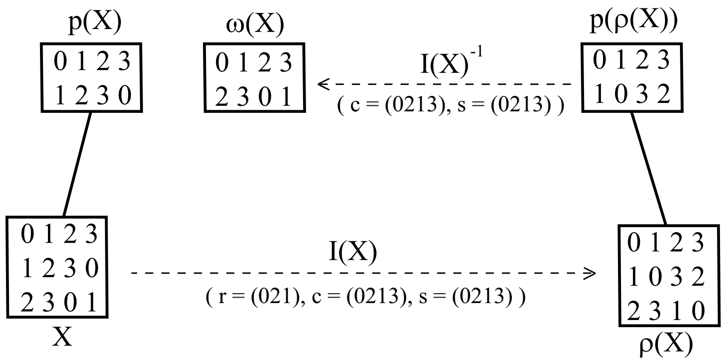
Пример 4.11 *Нека је дат латински правоугаоник*

$$X = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ \hline\end{array}.$$

Применом програма који је имплементација алгоритма 4.1 за канонску слику правоугаоника X се добија

$$\rho(X) = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \\ 2 & 3 & 1 & 0 \\ \hline\end{array}.$$

Одговарајући изотопизам $I(X)$ је $(r = (021), c = (0213), s = (0213))$. Према теореми 4.3 елементи у почетној врсти и почетној колони канонског правоугаоника су поређани у растућем поретку, па је пермутација врста у изотопизму $I(X)$ једнозначно одређена пермутацијама симбола и колона. Ово нам омогућава да инверзни изотопизам $I(X)^{-1}$ дефинишемо само на основу инверзних пермутација симбола и колона и затим га на начин дат у претходној теореми применимо на $p(\rho(X))$. Описана конструкција је дата на наредној слици.



Слика 4.10: конструкција слабог канонског оца канонског правоугаоника

Дакле, слаб канонски отац правоугаоника X је

$$\omega(X) = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ \hline\end{array}.$$

Слаб канонски отац $\omega(X)$ и "обичан" отац $p(X)$ могу да буду у различитим односима. Ситуација која је значајна је када су ова два објекта изоморфна.

За објекат X за који важи:

$$p(X) \sim \omega(X) \tag{4.5}$$

се каже да је генерисан *слабим канонским проширењем.*

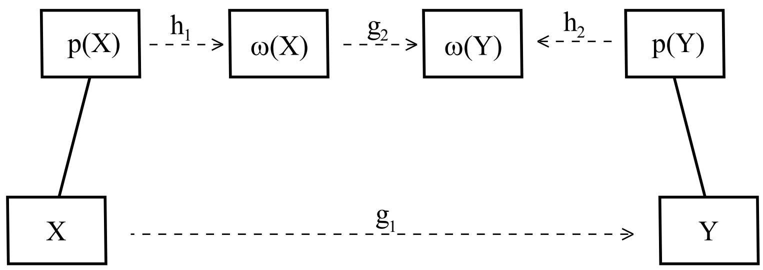
Пример 4.12 *Правоугаоник X из претходног примера (слика 4.10)* није *генерисан слабим канонским проширењем, јер његов отац* $p(X)$ *и његов канонски отац* $\omega(X)$ *нису изотопни.*

Правоугаоник $\rho(X)$ *из претходног примера* јесте *генерисан слабим канонским проширењем. У овом случају пресликавање* $I(\rho(X))$ *је идентичко, па је отац објекта* $\rho(X)$ *истовремено и његов слаб канонски отац.*

За два изоморфна чвора X и Y генерисана слабим канонским проширењем на основу (4.3) и (4.5) важи:

$$p(X) \sim \omega(X) \sim \omega(Y) \sim p(Y). \tag{4.6}$$

Другим речима, изоморфни чворови генерисани слабим канонским проширењем имају изоморфне очеве (слика 4.11).



Слика 4.11: изоморфни чворови X и Y генерисани слабим канонским проширењем

Претпоставимо да на неком нивоу стабла нема изоморфних чворова и да су X и Y објекти генерисани слабим канонским проширењем на следећем нивоу стабла. Из претходног разматрања следи да, ако су X и Y изоморфни, они имају заједничког оца. Дакле, уклањање изоморфизма међу чворовима који испуњавају услов (4.5) је довољно спровести *само међу синовима истог оца.* Ова особина се може искористити за алгоритам класификације.

Нека је T стабло претраге са кореном R и нека за његове чворове X и Y важе услови:

$C\,1$ $\quad X \sim Y \Rightarrow (\forall Z \in C(X))(\exists W \in C(Y))\ \ Z \sim W.$

$C\,4$ $\quad$ *За сваки $X \neq R$, постоји Y такав да је $X \sim Y$ и $p(Y) \sim \omega(Y)$.*

Као што је и раније напоменуто, значење услова $C1$ је да изоморфни чворови имају изоморфне синове. Смисао услова $C4$ је да у свакој класи еквиваленције постоји објекат који је генерисан слабим канонским проширењем.

Размотримо следећи алгоритам:

АЛГОРИТАМ 4.10 [слабо_канонско_проширење(чвор X)]

---

1 $\Sigma := \Sigma \cup X$;
2 $\Delta := \emptyset$;
3 for all $Z \in C(X)$
4   if $p(Z) \sim \omega(Z)$
5   if у скупу Δ не постоји Y такав да је $Z \sim Y$
6             $\Delta := \Delta \cup Z$;
7 for all $Z \in \Delta$
8   слабо_канонско_проширење(Z);

Неформално, алгоритам врши претрагу стабла T разматрајући подстабло T' које садржи само чворове генерисане слабим канонским проширењем. Ово се постиже у линијама 3 и 4 тако што се за синове истог оца испитује важење услова (4.5). У разматрању које непосредно претходи алгоритму показано је да само међу таквим објектима треба тражити изоморфне, па се у 5. и 6. линији алгоритма постиже јединственост представника сваке класе у скупу Δ. Егзистенција представника сваке класе у скупу Σ је обезбеђена због важења услова $C4$.

Ако је $\prec$ релација поретка на скупу чворова стабла T и ако се 3. линија алгоритма реализује тако што се објекти претражују од "мањих" ка "већима" у односу на $\prec$, тада скуп Σ на крају садржи канонског представника сваке класе.

Доказ наредне теореме је формалан доказ коректности алгоритма *слабо_канонско_проширење*.

Теорема 4.16 *Када се алгоритам* слабо_канонско_проширење *примени на стабло претраге T у којем важе услови $C1$ и $C4$, генерише се скуп Σ који садржи тачно један објекат из сваке изоморфне класе објеката из стабла T.*

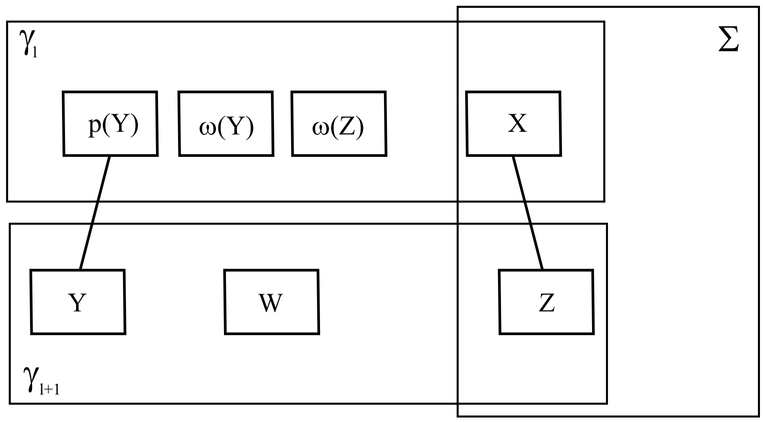
Доказ: Нека су изоморфне класе чворова поређане на начин који следи из теореме 4.10. Доказ се може извршити индукцијом по редном броју изоморфне класе. Како је једини елемент у првој изоморфној класи корен стабла, а аргумент првог позива процедуре је управо корен, непосредно следи да је испуњена почетна индукцијска претпоставка.

Да би доказали јединственост елемента из сваке изоморфне класе у скупу Σ, претпоставимо да јединственост важи за првих l класа, као и да постоје два објекта Z_1 и Z_2 из $(l+1)-$ве класе у скупу Σ. Ова два објекта су прошла тест у 4. линији алгоритма пре него што су у наредном позиву процедуре додати скупу Σ, па важи:

$$p(Z_1) \sim \omega(Z_1) \;\; i \;\; p(Z_2) \sim \omega(Z_2)$$

Како су Z_1 и Z_2 из исте класе, на основу (4.3) важи $\omega(Z_1) \sim \omega(Z_2)$, па је $p(Z_1) \sim p(Z_2)$. Због индукцијске претпоставке и чињенице да су $p(Z_1)$ и $p(Z_2)$ у класи реда мањег од $l+1$, следи да је $p(Z_1) = p(Z_2)$, а како се у 5. линији алгоритма изоморфизам међу синовима истог оца уклања, следи да је $Z_1 = Z_2$.

Треба доказати и егзистенцију елемента из сваке изоморфне класе у скупу Σ.



Претпоставимо да она важи за првих l класа и нека је W објекат из $(l+1)-$ве класе, тј. $W \in \gamma_{l+1}$ (горња слика). Доказаћемо да у скупу Σ постоји објекат изоморфан са W.

Према услову $C4$ постоји објекат Y изоморфан са W такав да је $p(Y) \sim \omega(Y)$, а из начина на који су поређане класе непосредно следи $Y \in \gamma_{l+1}$ и $p(Y), \omega(Y) \in \gamma_l$.

Према индукцијској претпоставци у скупу Σ постоји чвор X, такав да је $X \sim p(Y)$, а из услова $C1$ следи да постоји $Z \in C(X)$ такав да је $Z \sim Y$. Из дефиниције слабог канонског родитеља следи $\omega(Z) \sim \omega(Y)$, односно

$$p(Z) = X \sim p(Y) \sim \omega(Y) \sim \omega(Z).$$

Дакле, објекат Z задовољава услов у 4. линији алгоритма па се налази у скупу Σ.

Важно је напоменути да скуп Σ на крају рада описане процедуре садржи представнике класа *свих* објеката у стаблу, односно и парцијалне и потпуне објекте. Алгоритам се може једноставно модификовати да би се из скупа Σ издвајали само потпуни објекти.

4.4.6 Класификација латинских квадрата слабим канонским проширењем

У овој тачки се разматра примена технике слабог канонског проширења, дате у претходној тачки, на латинске квадрате.

Нека је T_n стабло претраге дефинисано у тачки 4.3.1, које на свом $k-$том нивоу ($1 \leq k \leq n$) садржи све латинске правоугаонике димензија $k \times n$. Важи следеће тврђење:

Теорема 4.17 *Стабло T_n испуњава услове $C1$ и $C4$ у којима је $\sim$ ознака за релацију изотопије, а $\omega(X) = I(X)^{-1}(p(\rho(X)))$ пресликавање дефинисано као у теореми 4.15.*

Доказ: Испуњеност услова $C1$ за стабло T_n је доказана у теореми 4.12. Услов $C1$ је у том делу био неопходан за примену технике чувања објеката (алгоритам 4.6 - record*).*

Докажимо да је испуњен и услов $C4$, то јест да у свакој изотопној класи постоји латински правоугаоник који је генерисан слабим канонским проширењем.

Такав правоугаоник је канонски правоугаоник. Заиста, ако је Y канонски правоугаоник тада важи $Y = \rho(Y)$, односно пресликавање $I(Y)$ је идентичко, па је

$$\omega(Y) = I(Y)^{-1}(p(\rho(Y))) = I(Y)(p(Y)) = p(Y).$$

Дакле, у свакој класи постоји чвор Y такав да је $p(Y) \sim \omega(Y)$, то јест испуњен је услов $C4$.

Из претходне теореме следи да се процедура дата у алгоритму 4.10 - *слабо_канонско_проширење* може применити на стабло T_n и на тај начин извршити класификација латинских квадрата реда n у односу на релацију изотопије.

Нека је T''_n подстабло стабла T_n чије чворове обрађује алгоритам *слабо_канонско_проширење.* Број чворова на последњем нивоу стабла T''_n одговара броју класа изотопије квадрата реда n, а број чворова на $k-$том нивоу $(1 \leq k < n)$ стабла T''_n који нису листови, одговара броју класа изотопије правоугаоника димензија $k \times n$.

Техника која је описана, као и технике у претходним тачкама, састоји се из две фазе:

- генерисање стабла претраге (алгоритам 4.4 - *стабло*)
- класификација (алгоритам 4.10 - *слабо_канонско_проширење*)

И у овом случају може се постићи знатно побољшање спајањем ова два алгоритма. Алгоритам којим се то реализује је *слабо_канонско_проширење_1.* Он се дакле, не примењује на стабло T_n, већ директно генерише стабло T''_n.

АЛГОРИТАМ 4.11 [слабо_канонско_проширење_1(чвор X)]

```
1  if(X је потпун квадрат)
2      Σ := Σ ∪ X;
3  else
4      Δ := ∅;
5      C(X) :=новаврста(X, 0);
6      for all Z ∈ C(X)
7          if p(Z) ∼ ω(Z)
8              if у скупу Δ не постоји Y такав да је Z ∼ Y
9                  Δ := Δ ∪ Z;
10     for all Z ∈ Δ
11         слабо_канонско_проширење_1(Z);
```

Функција *новаврста()* у 5. линији се реализује истоименим алгоритмом 4.5. У линијама 6 и 7 се из даљег разматрања уклањају сви правоугаоници који нису генерисани слабим канонским проширенјем, а у 8. линији се међу браћом уклањају изоморфни.

Коректност алгоритма *слабо_канонско_проширење_1* је последица коректности алгоритама *стабло*(4.4) и *слабо_канонско_проширење* (4.10), то јест скуп Σ на крају садржи по један квадрат из сваке класе.

4.4.7 Канонско проширење

Техника представљена у тачки 4.4.5 може се побољшати, ако се услови неопходни за њену реализацију учине строжијим. У овој тачки, осим тога да је чвор X генерисан слабим канонским проширењем, тј. да испуњава услов $p(X) \sim \omega(X)$, захтева се и да је X добијен проширењем објекта $p(X)$ на карактеристичан начин.

И у овом случају разматрамо класификацију објеката који се налазе у стаблу претраге T са кореном R.

Нека је m функција која сваком објекту $X \neq R$ додељује објекат $m(X)$, тако да за свака два изоморфна чвора X и Y исти изоморфизам који постоји између њих постоји и између $m(X)$ и $m(Y)$. Односно, нека је m пресликавање за које важи:

$$X \sim Y \Rightarrow (X, m(X)) \sim (Y, m(Y)). \tag{4.7}$$

Објекат $m(X)$ се назива *канонски отац објекта* X. Услов (4.7) је строжија верзија услова (4.3), па следи да канонски отац јесте и слаб канонски отац, али обратно не важи.

Као и код слабог канонског оца ово пресликавање је увек могуће дефинисати. Тривијално се то може учинити тако што m слика сваки објекат у себе, а пресликавање за које је у теореми 4.15 доказано да је слаб канонски отац задовољава и услов (4.7). Наиме, важи следеће тврђење:

Теорема 4.18 *Нека је $I(X)$ изоморфизам који објекат X пресликава у његов канонски облик $\rho(X)$ у односу на релацију $\sim$. Пресликавање m дефинисано са:*

$$m(X) = I(X)^{-1}(p(\rho(X))) \tag{4.8}$$

задовољава услов (4.7), то јест $m(X)$ је канонски отац објекта X.

Доказ: Нека је $X \sim Y$. У теореми 4.15 је доказано да је $g = I(Y)^{-1}I(X)$ изоморфизам који пресликава објекат $m(X)$ у објекат $m(Y)$. Докажимо да исти изоморфизам пресликава и објекат X у објекат Y, то јест да је $(X, m(X)) \sim (Y, m(Y))$.

Заиста, непосредно из дефиниције пресликавања m и I, као и чињенице да због $X \sim Y$ важи $\rho(X) = \rho(Y)$, следи:

$$gX = I(Y)^{-1}I(X)X = I(Y)^{-1}\rho(X) = I(Y)^{-1}\rho(Y) = Y.$$

Пример 4.13 *Из претходне теореме следи да је слаб канонски отац латинског правоугаоника из примера 4.11 истовремено и његов канонски отац, односно важи:*

Канонски отац правоугаоника

$$X = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ \hline\end{array}$$

је правоугаоник

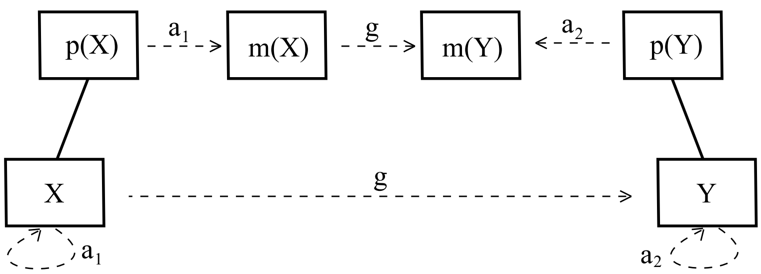
$$m(X) = \begin{array}{|cccc|}\hline 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \\ \hline\end{array}\ .$$

Аналогно као у тачки 4.4.5, може се дефинисати појам канонског проширења. За објекат X за који важи:

$$(X, p(X)) \sim (X, m(X)) \tag{4.9}$$

каже се да је генерисан *канонским проширењем*. Објекат X је дакле, генерисан канонским проширењем ако постоји његов аутоморфизам који пресликава $p(X)$ у $m(X)$.

За објекат који је генерисан канонским проширењем тј. испуњава услов (4.9) следи да је генерисан и слабим канонским проширењем тј. да испуњава услов (4.5), али обратно не важи.



Слика 4.12: изоморфни чворови X и Y генерисани канонским проширењем

За два изоморфна чвора X и Y генерисана канонским проширењем на основу (4.7) и (4.9) важи:

$$(X, p(X)) \sim (X, m(X)) \sim (Y, m(Y)) \sim (Y, p(Y)). \tag{4.10}$$

Другим речима, изоморфни чворови генерисани канонским проширењем не само да имају изоморфне очеве, већ и између очева постоји исти изоморфизам као између њихових синова (слика 4.12).

Претпоставимо да на неком нивоу стабла нема изоморфних чворова и да су X и Y објекти генерисани канонским проширењем на следећем нивоу. Из претходног разматрања следи да ако су X и Y изоморфни, они имају заједничког оца, као и да је изоморфизам који пресликава X у Y *аутоморфизам* њиховог заједничког оца.

Дакле, уклањање изоморфизма међу чворовима који испуњавају услов (4.9) је довољно спровести *само међу синовима истог оца и то испитујући да ли између њих постоји изоморфизам g који је аутоморфизам оца.* Ова особина се може искористити за алгоритам класификације.

Нека је T стабло претраге са кореном R и нека важе услови:

$C\ 5$ $\quad X \sim Y \Rightarrow (\forall Z \in C(X))(\exists W \in C(Y))\ \ (Z, X) \sim (W, Y).$

$C\ 6$ $\quad$ *За сваки чвор $X \neq R$, постоји чвор Y такав да је*

$$X \sim Y \ \text{и}\ (Y, p(Y)) \sim (Y, m(Y)).$$

Услови $C5$ и $C6$ су строжије верзије услова $C1$ и $C4$, а у већини случајева, стабло које задовољава услов $C1$, задовољава и услов $C5$.

Значење услова $C5$ је да изоморфни чворови имају изоморфне синове између којих важи исти изоморфизам као између очева. Смисао услова $C6$ је да у свакој класи еквиваленције постоји објекат који је генерисан канонским проширењем.

Размотримо наредни алгоритам:

АЛГОРИТАМ 4.12 [канонско_проширење(чвор X)]

```
1 Σ := Σ ∪ X;
2 Δ := ∅;
3 for all Z ∈ C(X)
4     if (Z, p(Z)) ~ (Z, m(Z))
5         if Δ ∩ {aZ : a ∈ Aut(X)} = ∅
6             Δ := Δ ∪ Z;
7 for all Z ∈ Δ
8     канонско_проширење(Z);
```

Главна разлика између управо наведеног алгоритма и алгоритма 4.10 - *слабо_канонско_проширење* је у линији 5. Ова линија се у алгоритму 4.10 реализује тако што се испитује постојање изоморфног објекта са Z у скупу Δ, разматрајући све изоморфизме. У алгоритму *канонско_проширење* то се чини разматрајући само аутоморфизме објекта X, што знатно убрзава претрагу. Као што смо показали у разматрању које непосредно претходи алгоритму *канонско_проширење*, ово је омогућено условом у 4. линији.

Наредна теорема и њен доказ су еквивалентни теореми 4.16 о коректности алгоритма 4.10 - *слабо_канонско_проширење* и њеном доказу.

Теорема 4.19 *Када се алгоритам* канонско_проширење *примени на стабло претраге T у којем важе услови $C5$ и $C6$ генерише се скуп Σ који садржи тачно један објекат из сваке изоморфне класе објеката из стабла T.*

Доказ: Нека су изоморфне класе чворова поново поређане на начин који следи из теореме 4.10. Доказ се може извршити индукцијом по редном броју изоморфне класе. Како је једини елемент у првој изоморфној класи корен дрвета, непосредно следи да је испуњена почетна индукцијска претпоставка.

Да би доказали јединственост елемента из сваке изоморфне класе у скупу Σ, претпоставимо да јединственост важи за првих l класа, као и да постоје два објекта Z_1 и Z_2 из $(l+1)-$ве класе у скупу Σ. Ова два објекта су прошла тест у 4. линији алгоритма пре него што су у наредном позиву процедуре додати скупу Σ, па важи:

$$(Z_1, p(Z_1)) \sim (Z_1, m(Z_1)) \;\; i \;\; (Z_2, m(Z_2)) \sim (Z_2, p(Z_2)).$$

Како су Z_1 и Z_2 из исте класе, на основу (4.7) важи

$$(Z_1, p(Z_1)) \sim (Z_1, m(Z_1)) \sim (Z_2, m(Z_2)) \sim (Z_2, p(Z_2)),$$

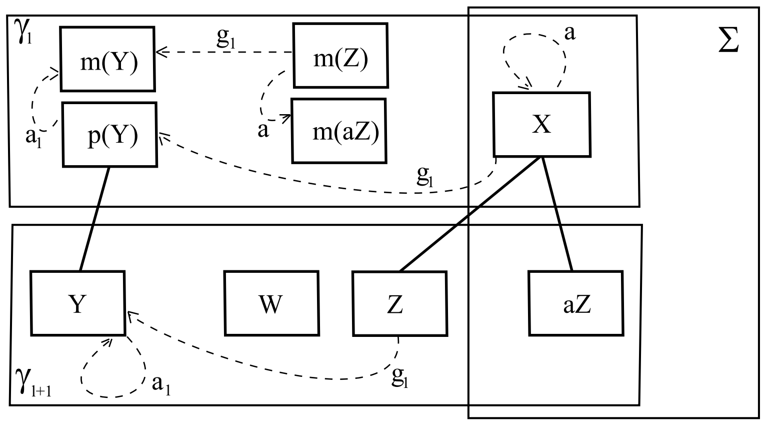
одакле следи $p(Z_1) \sim p(Z_2)$.

Како су објекти $p(Z_1)$ и $p(Z_2)$ у класи реда мањег од $l+1$, следи да је $p(Z_1) = p(Z_2)$, па из $(Z_1, p(Z_1)) \sim (Z_2, p(Z_2))$ следи да постоји $a \in Aut(p(Z_2))$ такав да је $aZ_2 = Z_1$.

Одавде следи да ако $Z_1 \in \Delta$, тада скуп $\Delta \cap \{aZ_2 : a \in Aut(p(Z_2)\}$ у 5. линији није празан, па се Z_2 у 6. линији не додаје *скупу Δ. Ово је контрадикција са претпоставком да се и Z_1 и Z_2 налазе у скупу Σ.*

Треба доказати и егзистенцију елемента из сваке класе у скупу Σ. Претпоставимо да она важи за првих l класа и нека је W објекат из $(l+1)-$ве класе. Доказаћемо да у скупу Σ постоји објекат изоморфан са W.

Према услову $C6$ постоји објекат Y изоморфан са W такав да је $(Y, p(Y)) \sim (Y, m(Y))$. Из индукцијске претпоставке следи да у скупу Σ постоји X, такав да је $X \sim p(Y)$, па из услова $C5$ следи да постоји $Z \in C(X)$ такав да је $(X, Z) \sim (p(Y), Y)$.



Нека је $a \in Aut(X)$ тако да је $aZ \in C(X)$. Непосредно следи да је $p(aZ) = X = p(Z)$ и $(aZ, X) \sim (Z, X)$, па из (4.7), даље следи да је

$$(aZ, m(aZ)) \sim (Z, m(Z)) \sim (Y, m(Y)) \sim (Y, p(Y)) \sim (Z, p(Z)) \sim (aZ, p(aZ)).$$

Дакле, aZ пролази тест у 4. линији алгоритма. Следи да скуп Σ на крају садржи објекат изоморфан са $Z \sim Y \sim W$.

Као и у алгоритму 4.10, скуп Σ на крају садржи представнике класа *свих* објеката у стаблу, то јест и парцијалне и потпуне објекте. Алгоритам се може једноставно модификовати да би се из скупа Σ издвајали само потпуни објекти.

Процедура дата у алгоритму *канонско_проширење* се може ефектно паралелизовати, јер се испитивање главног услова (4.9) односи само на објекат и његовог оца. Такође, изоморфност се испитује само између синова истог оца, па у датом тренутку знање о другим објектима осим њих није потребно.

4.4.8 Класификација латинских квадрата канонским проширењем

У овој тачки се разматра примена технике канонског проширења, дате у претходној тачки, на латинске квадрате.

Нека је T_n стабло претраге дефинисано у тачки 4.3.1, које на свом $k-$том нивоу ($1 \leq k \leq n$) садржи све латинске правоугаонике димензија $k \times n$. Важи следеће тврђење:

Теорема 4.20 *Стабло* T_n *испуњава услове* $C5$ *и* $C6$ *у којима је* $\sim$ *ознака за релацију изотопије, а* $m(X) = I(X)^{-1}(p(\rho(X)))$ *пресликавање дефинисано као у теореми 4.18.*

Доказ: Значење услова $C5$ *у овом случају је да изотопни правоугаоници у стаблу имају изотопне синове између којих важи исти изотопизам као између очева. Испуњеност овог услова у стаблу* T_n *је већ доказана у доказу теореме 4.12 (слика 4.6).*

Докажимо да је испуњен и услов $C6$*, то јест да у свакој изотопној класи постоји латински правоугаоник који је генерисан канонским проширењем.*

Такав правоугаоник је канонски правоугаоник. Заиста, ако је Y *канонски правоугаоник тада важи* $Y = \rho(Y)$*, односно пресликавање* $I(Y)$ *је идентичко, па је*

$$m(Y) = I(Y)^{-1}(p(\rho(Y))) = I(Y)(p(Y)) = p(Y).$$

Дакле, у свакој класи постоји чвор Y *такав да је* $(Y, p(Y)) \sim (Y, m(Y))$*, то јест испуњен је услов* $C6$*.*

Из претходне теореме следи да се процедура дата у алгоритму 4.12 - *канонско_проширење* може применити на стабло T_n и на тај начин извршити класификација латинских квадрата реда n у односу на релацију изотопије.

Нека је T''_n подстабло стабла T_n чије чворове обрађује алгоритам *канонско_проширење*. Број чворова на последњем нивоу стабла T''_n одговара броју класа изотопије квадрата реда n, а број чворова на $k-$том нивоу $(1 \leq k < n)$ стабла T''_n који нису листови, одговара броју класа изотопије правоугаоника димензија $k \times n$.

Техника која је описана, као и технике у претходним тачкама, састоји се из две фазе:

- генерисање стабла претраге (алгоритам 4.4 - *стабло*)
- класификација (алгоритам 4.12 - *канонско_проширење*)

И у овом случају спајање ова два алгоритма резултира знатним убрзањем целе процедуре.

Алгоритам у којем је спајање реализовано је *канонско_проширење_1*. Он се дакле, не примењује на стабло T_n, већ директно генерише стабло T''_n.

АЛГОРИТАМ 4.13 [канонско_проширење_1(чвор X)]

```
1  if(X је потпун квадрат)
2      Σ := Σ ∪ X;
3  else
4      Δ := ∅;
5      C(X) :=новаврста(X, 0);
6      for all Z ∈ C(X)
7          if (Z, p(Z)) ~ (Z, m(Z))
8              if Δ ∩ {aZ : a ∈ Aut(X)} = ∅
9                  Δ := Δ ∪ Z;
10     for all Z ∈ Δ
11         канонско_проширење(Z);
```

Функција *новаврста()* у 5. линији се реализује истоименим алгоритмом 4.5. У линијама 6 и 7 се из даљег разматрања уклањају сви правоугаоници који нису генерисани канонским проширењем, а у 8. линији се међу браћом уклањају изоморфни.

Коректност алгоритма *канонско_проширење_1* је последица коректности алгоритама *стабло*(4.4) и *канонско_проширење*(4.12), то јест Σ на крају садржи по један квадрат из сваке класе изотопије.

4.4.9 Преглед наведених алгоритама

На крају поглавља размотримо још једном наведене алгоритме.

Најпре ће укратко бити описани *сви* алгоритми, а затим је дата табела 4.1 која се односи на алгоритме за класификацију. Она садржи услове које стабло треба да задовољава да би одговарајући алгоритам за класификацију био примењен, као и нека позитивна и негативна својства алгоритама који се примењују.

- АЛГОРИТАМ 4.1
  [канонски_правоугаоник(правоугаоник $L_{k,n}$)]

  За дати латински правоугаоник генерише његов канонски облик у односу на релацију изотопије. (стр. 49)

- АЛГОРИТАМ 4.2
  [канон_правоугаоник(правоугаоник $L_{k,n}$)]

  Одлучујућа верзија претходног алгоритма. За дати латински правоугаоник испитује да ли је канонски у односу на релацију изотопије. (стр. 49)

- АЛГОРИТАМ 4.3
  [неизоморфни(скуп објеката Γ)]

  Елементаран алгоритам за класификацију скупа комбинаторних објеката. (стр. 52)

- АЛГОРИТАМ 4.4
  [стабло(чвор X)]

  Генерише стабло које на свом последњем нивоу садржи све комбинаторне објекте неког типа. (стр. 53)

- АЛГОРИТАМ 4.5
  [новаврста(правоугаоник $L_{k,n}$, колона c)]

  Дати латински правоугаоник на све начине проширује новом врстом. Одговарајући алгоритам који се односи на *редуковане* правоугаонике је означен са *новаврстаР*. (стр. 54)

- АЛГОРИТАМ 4.6
  [record(чвор X)]

  Врши класификацију произвољних комбинаторних објеката методом чувања објеката. (стр. 59)

- АЛГОРИТАМ 4.7
  [record1(чвор X)]

  Претходни алгоритам примењен на латинске квадрате и релацију изотопије. (стр. 62)

- АЛГОРИТАМ 4.8
  [канонска_класификација(чвор X)]

  Врши класификацију произвољних комбинаторних објеката канонском методом. (стр. 64)

- АЛГОРИТАМ 4.9
  [канонска_класификација_1(чвор X)]

  Претходни алгоритам примењен на латинске квадрате и релацију изотопије. (стр. 68)

- АЛГОРИТАМ 4.10
  [слабо_канонско_проширење(чвор X)]

  Врши класификацију произвољних комбинаторних објеката методом слабог канонског проширења. (стр. 72)

- АЛГОРИТАМ 4.11
  [слабо_канонско_проширење_1(чвор X)]

  Претходни алгоритам примењен на латинске квадрате и релацију изотопије. (стр. 75)

- АЛГОРИТАМ 4.12
  [канонско_проширење(чвор X)]

  Врши класификацију произвољних комбинаторних објеката методом канонског проширења. (стр. 78)

- АЛГОРИТАМ 4.13
  [канонско_проширење_1(чвор X)]

  Претходни алгоритам примењен на латинске квадрате и релацију изотопије. (стр. 82)

$C1$	Изоморфни чворови у стаблу имају изоморфне синове.
$C2$	Канонски облик сваког потпуног објекта постоји у стаблу.
$C4$	Отац сваког канонског чвора је у канонском облику.
$C4$	За сваки чвор, у стаблу постоји њему изоморфан чвор који је генерисан слабим канонским проширењем.
$C5$	Изоморфни чворови у стаблу имају изоморфне синове између којих постоји исти изоморфизам као између очева.
$C6$	За сваки чвор, у стаблу постоји њему изоморфан чвор који је генерисан канонским проширењем.

услови	алгоритам	позитивна својства алгоритма	негативна својства алгоритма
$C1$	АЛГОРИТАМ 4.6 record	једноставна имплементација	чување парцијалних објеката, испитивање изоморфности, немогућност паралелизовања
$C2$ $C3$	АЛГОРИТАМ 4.8 канонска_ класификација	једноставна имплементација, не испитивање изоморфности, могућност паралелизовања	испитивање каноничности великог броја објеката
$C1$ $C4$	АЛГОРИТАМ 4.10 слабо_канонско_ проширење	испитивање изоморфности само међу браћом	чување парцијалних објеката, имплементација слабог канонског родитеља
$C5$ $C6$	АЛГОРИТАМ 4.12 канонско_ проширење	испитивање малог броја изоморфизама само међу браћом	конструкција групе аутотопизама $Aut()$, имплементација канонског родитеља

Табела 4.1: алгоритми за класификацију

5

Имплементација алгоритама

Тренутно најбољи резултат у класификацији латинских квадрата у односу на релацију изотопије је дао Мекеј [14] и односи се на квадрате реда $n \leq 8$. Овај резултат је добијен имплементацијом алгоритма 4.13 - *канонско_проширење_1* датог у тачки 4.4.8. Осим техника описаних у тој тачки, за реализацију линије

8 if $\Delta \cap \{aZ : a \in Aut(X)\} = \emptyset$

овог програма је било потребно одредити групу аутотопизама за квадрат X, што је урађено трансформисањем квадрата X у одговарајући граф $\gamma(X)$ (као у тачки 1.2.2) и затим одређивањем групе $Aut(\gamma(X))$ уз помоћ програма NAUTY.

Као међурезултат (у стаблу претраге пре последњег нивоа) добијени су и представници класа изотопије латинских правоугаоника димензија $k \times n$ за $n \leq 8$ и $1 \leq k \leq n$, а додатно је извршена и класификација правоугаоника димензија $k \times 9$ за $1 \leq k \leq 4$.

Резултати Мекеја из [14] су наведени у 3. колони табеле 5.1.

5.1 Имплементација канонске класификације

Програм који извршава канонску класификацију латинских квадрата у односу на релацију изотопије је саставни део овог рада. Њиме је као и у [14] извршена класификација за квадрате до реда $n \leq 8$. Како је коришћена друга техника, одговарајући међурезултати имају други смисао, али је коначан резултат, то јест број добијених квадрата идентичан као у раду Мекеја.

Као што је наведено у тачки 4.4.4, добијени међурезултати у стаблу претраге на нивоима пре последњег, у овом случају представљају бројеве оних изотопних класа чији су канонски представници редуковани правоугаоници.

Алгоритам *канонска_класификација_2* који је имплементиран је нешто другачији од алгоритма 4.9 - *канонска_класификација_1* наведеног у тачки 4.4.4. Измене нису суштинске и урађене су да би се омогућила реализација програма "део по део", односно да би се стабло претраге обрађивало "ниво по ниво". Ово је било важно због дугог времена потребног за класификацију, као и велике количине радне меморије која би била неопходна за класификацију из једног дела.

Програм је написан у програмском језику C, а експерими су извршени на рачунару са процесором који ради на фреквенцији од 3,06 GhZ и има 2GB РАМ-а.

Програм је реализован тако да је његов улаз фајл F са свим редукованим правоугаоницима димензија $k \times n$ $(1 \leq k < n)$ који су канонски. Ови правоугаоници се проширују новом врстом, а у излазни фајл G се уписују они новодобијени правоугаоници који су такође канонски.

АЛГОРИТАМ 5.1 [канонска_класификација_2(фајл F)]

```
фајл G = ∅;
for all X ∈ F
    C(X) :=новаврстаP(X, 0);
    for all Y ∈ C(X)
        if(канон_правоугаоник(Y))
            G := G ∪ Y;
```

Да би се извршила класификација квадрата реда n $(1 < n \leq 8)$ примењен је следећи поступак:

- програм је примењен на датотеку `file_1xn` која садржи само правоугаоник $[012...n-1]$, чиме је генерисана датотека `file_2xn` са свим редукованим правоугаоницима димензија $2 \times n$ који су канонски.

- програм је примењен на датотеку `file_2xn` која садржи све правоугаонике димензија $2 \times n$ који су канонски, чиме је генерисана датотека `file_3xn` са свим редукованим правоугаоницима димензија $3 \times n$ који су канонски.
- ...
- програм је примењен на датотеку `file_(n-1)xn` која садржи све правоугаонике димензија $(n-1) \times n$ који су канонски, чиме је генерисана датотека `file_nxn` са свим квадратима реда n који су канонски.

Датотеке `file_kxn` ($1 \leq k \leq n \leq 8$) које представљају одговарајуће нивое стабла претраге су такође, саставни део овог рада.

Резултати програма су дати у четвртој колони табеле 5.1. Ови резултати су прихваћени и објављени у "Енциклопедији целобројних низова" (The On-Line Encycopledia of Integer Sequences) под бројем $A162545$. (http://www.research.att.com/ njas/sequences/A162545)

У последњој колони табеле је укупно време потребно за генерисање одговарајућих квадрата.

k	n	број изотопних класа [14]	број изотопних класа чији су канонски представници редуковани правоугаоници	укупно време потребно за генерисање
1	1	1	1	
1	2	1	1	
2	2	1	1	
1	3	1	1	
2	3	1	1	
3	3	1	1	
1	4	1	1	
2	4	2	2	
3	4	2	2	
4	4	2	2	
1	5	1	1	
2	5	2	2	
3	5	3	3	
4	5	3	3	
5	5	2	2	
1	6	1	1	
2	6	4	4	
3	6	16	14	
4	6	56	34	
5	6	40	31	
6	6	22	22	1 секунд
1	7	1	1	
2	7	4	4	
3	7	56	54	
4	7	1.398	427	
5	7	6.941	1.410	
6	7	3.479	1.096	
7	7	564	564	12 секунди
1	8	1	1	
2	8	7	7	
3	8	370	330	
4	8	93.561	20.259	
5	8	4.735.238	509.027	
6	8	29.163.047	3.144.797	
7	8	13.302.311	2.847.673	
8	8	1.676.267	1.676.267	82 сата
1	9	1	1	
2	9	8	8	
3	9	2.877	2.409	52 сата
4	9	8.024.046		

Табела 5.1: добијени резултати алгоритама за класификацију латинских квадрата

6

Правци даљег рада

Главно својство које разликује програм коришћен у овом раду од програма који су раније дали исте резултате је да он не трансформише латинске правоугаонике (квадрате) у графове, па се класификација извршава без алата за манипулацију графовима као што је NAUTY.

Битно је такође нагласити и разматрање мањег броја правоугаоника који су међурезултати. Наиме, алгоритам канонске класификације генерише, као што је раније истакнуто, само редуковане латинске правоугаонике. Ако је циљ само класификација квадрата, а не и правоугаоника ово је бољи приступ од алгоритама који генеришу све правоугаонике.

Програм, којим је извршена класификација и који је саставни део овог рада, реализован је на рачунару са процесором који ради на фреквенцији од 3,06 GhZ и има 2GB РАМ-а.

Укупно време потребно за класификацију квадрата реда 8 је било 82 сата (табела 5.1).

Како је за генерисање 2.409 редукованих правоугаоника димензија 3×9 који су канонски било потребно око 52 сата и како постоји 115.618.721.533 изотопних класа квадрата реда 9 (табела 3.1), процењује се да би за класификацију квадрата реда 9 истом програму на истом рачунару било потребно више хиљада година.

Са друге стране, време потребно за класификацију квадрата реда 7 је само 12 секунди, па се може проценити да је најважнији узрок немогућности класификације квадрата реда 9 њихов велики број, а не само евентуални недостаци алгоритма. Ово потврђује и чињеница да ни другим ауторима није успела класификација за ред 9.

Време потребно за класификацију не увећава само недовољно велика процесорска брзина, већ и недостатак радне меморије. Наиме, велики број класа има за последицу да се њихови представници не могу чувати у радној меморији до краја рада програма, већ се морају смештати у спољашњу меморију што знатно успорава процес. Ово би се у будућности могло решити коришћењем умрежених рачунара од којих би сваки обрађивао засебно подстабло претраге и тиме у својој радној меморији чувао део скупа решења.

Што се самог алгоритма тиче, највише времена се троши на проверавање каноничности великог броја правоугаоника на сваком нивоу стабла, од којих врло мали проценат јесте канонски. Знатно убрзање би се постигло смањењем броја "кандидата", што би се могло остварити евентуалним проналажењем нових неопходних услова за каноничност правоугаоника који имају $k \geq 3$ врста, као што је у тачки 4.1.2 урађено за правоугаонике који имају 2 врсте.

Са друге стране, напредак применом других алгоритама (пре свега алгоритма 4.13 - *канонско проширење*) поред наведене идеје умрежавања може се остварити и побољшањем особина програма NAUTY, што подразумева и известан напредак у теорији графова.

Литература

[1] J. W. Brown, *Enumeration of latin squares with application to order 8*, Journal of Combinatorial Theory 5, 177 – 184, Phoenix, 1968.

[2] C. J. Colbourn, *The complexity of completing partial latin squares*, Discrete Applied Mathematics 8, 25 – 30, St. Louis, 1984.

[3] C. J. Colbourn, J. H. Dinitz, *Handbook of Combinatorial Designs*, Discrete Mathematics and its applications - CRC press, New York, 2007.

[4] L. Euler, *Recherches sur une nouvelle espece de quarres magiques*, Verh. v. h. Zeeuwsch Genoottsch. der Wetensch., 85 – 239, Vlissingen, 1782.

[5] T. Evans, *Embedding incomplete latin squares*, American Mathematical Monthly 67, 958 – 961, 1960.

[6] M. Frolov, *Sur les permutations carres*, Journal de Mathhematiques spec. 4, 8 – 11, 25 – 30, 1890.

[7] C. D. Godsil, B. D. McKay, *Asymptotic enumeration of Latin rectangles*, Journal of Combinatorial Theory series B N48, 1986.

[8] P. Kaski, *Isomorph-free exhaustive generation of combinatorial designs*, Helsinki University of Technology Laboratory for Theoretical Computer Science - Research Reports 70, Helsinki, 2002.

[9] P. Kaski, *Algorithms for classification of combinatorial objects*, Helsinki University of Technology Laboratory for Theoretical Computer Science - Research Reports 94, Helsinki, 2005.

[10] P. Kaski, *Classification alghorithms for codes and designes*, Springer Publishing Company, New York, 2006.

[11] G. Kolesova, C. W. H. Lam, L. Thiel, *On the number of* 8×8 *Latin squares*, Journal of Combinatorial Theory series A N54, 143 – 148, Phoneix, 1990.

[12] C. F. Laywine, G. L. Mullen, *Discrete Mathematics using Latin Squares*, A Wiley Interscience Publication, New York, 1998.

[13] B. D. McKay, E. Rogoyski, *Latin Squares of order 10*, The Electronic Journal of Combinatorics N3, 1995.

[14] B. D. Mckay, *Isomorph-free exhaustive generation*, Department of Computer Science Australian National University, Canberra, 1998.

[15] B. D. McKay, I. M. Wanless, *On the number of Latin Squares*, Department of Computer Science Australian National University, Canberra, 2005.

[16] B. D. McKay, A. Meynert, W. Myrvold, *Small Latin Squares, Quasigroups and Loops*, Department of Computer Science Australian National University, Canberra, 2005.

[17] H. W. Norton, *The* 7×7 *squares*, Annals of Eugenics 9, $269 - 307$, 1939.

[18] J. Rothstein, S. E. Bammel, *The number of* $9{\times}9$ *Latin squares*, Discrete Mathematics 11, $93 - 95$, 1975.

[19] H. J. Ryser, *A combinatorial theorem with an application to latin rectangles*, Proceedings of the American Mathematical Society 2, $550{-}552$, Providence, 1951.

[20] A. Sade, *Enumeration des carres latins. Application au 7'eme ordre*, Conjectures pour les ordres superieurs, privately published, Marseille, 1948.

[21] A. Sade, *An omission in Nortons list of* 7×7 *squares*, The Annals of Mathematical Statistics 22, $306 - 307$, Marseille, 1951.

[22] E. Schonhardt, *Uber lateinische Quadrate und Unionen*, Journal fur die reine und angewandte Mathematik 163, $183 - 230$, Berlin, 1930.

[23] B. Smetaniuk, *A new construction on latin squares I: A proof of the Evans conjecture*, Ars Combinatoria 11, $155 - 172$, Winnipeg, 1981.

[24] J. H. Van Lint, R. M. Wilson, *A Course in Combinatorics*, Cambridge University Press, Cambridge, 2001.

[25] M. B. Wells, *The number of Latin squares of order eight*, Journal of Combinatorial Theory 3, $98 - 99$, Phoenix, 1967.

[26] K. Yamamoto, *On the asymptotic number of Latin rectangles*, Japanese Journal of Mathematics 21, $113 - 119$, Tokyo, 1951.