

AUTOREFERAT

Włodzimierz Moczurad

2016

1 Imię i nazwisko: Włodzimierz Moczurad

2 Posiadane dyplomy i stopnie naukowe

- Tytuł magistra informatyki, Uniwersytet Jagielloński, Wydział Matematyki i Fizyki, 1992 (z wyróżnieniem).
- Stopień doktora nauk matematycznych w zakresie informatyki, Uniwersytet Jagielloński, Wydział Matematyki i Fizyki, 2000 (z wyróżnieniem), rozprawa „Algebraiczne i algorytmiczne własności kodów klockowych”.

3 Informacje o dotychczasowym zatrudnieniu

- Od października 2014: starszy wykładowca w Instytucie Informatyki i Matematyki Komputerowej, Wydział Matematyki i Informatyki, Uniwersytet Jagielloński.
- 2005–2014: adiunkt w Instytucie Informatyki, Wydział Matematyki i Informatyki, Uniwersytet Jagielloński.
- 1992–2005: asystent w Instytucie Informatyki, Wydział Matematyki i Informatyki (do 2001: Wydział Matematyki i Fizyki; do 2003: Wydział Matematyki, Fizyki i Informatyki), Uniwersytet Jagielloński.
- 1991–1992: asystent stażysta w Instytucie Informatyki, Wydział Matematyki i Fizyki, Uniwersytet Jagielloński (w czasie ostatniego roku studiów).

4 Wskazanie osiągnięcia naukowego

(zgodnie z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki, Dz. U. nr 65, poz. 595 ze zm.)

4.1 Tytuł: Kody złożone z figur skierowanych w \mathbb{Z}^2

4.2 Lista publikacji wchodzących w skład osiągnięcia naukowego

- [H1] Michał Kolarz i Włodzimierz Moczurad. “Directed figure codes are decidable”. W: *Discrete Mathematics and Theoretical Computer Science* 11.2 (2009). IF=0.414, s. 1–14.
- [H2] Włodzimierz Moczurad. “Directed Figure Codes with Weak Equality”. W: *Intelligent Data Engineering and Automated Learning IDEAL 2010*. T. 6283. Lecture Notes in Computer Science. Springer, 2010, s. 242–250.
- [H3] Włodzimierz Moczurad. “Plane-Filling Properties of Directed Figures”. W: *FAW-AAIM 2011*. T. 6681. Lecture Notes in Computer Science. Springer, 2011, s. 255–263.
- [H4] Michał Kolarz i Włodzimierz Moczurad. “Multiset, set and numerically decipherable codes over directed figures”. W: *IWOCA 2012*. T. 7643. Lecture Notes in Computer Science. Springer, 2012, s. 224–235.
- [H5] Włodzimierz Moczurad. “Domino Graphs and the Decipherability of Directed Figure Codes”. W: *IWOCA 2013*. T. 8288. Lecture Notes in Computer Science. Springer, 2013, s. 453–457.

- [H6] Włodzimierz Moczurad. “Defect property in \mathbb{Z}^2 ”. W: *Proceedings of WORDS 2013*. T. 20. TUCS Lecture Notes. TUCS, 2013, s. 64–70.
- [H7] Małgorzata Moczurad i Włodzimierz Moczurad. “Testing decipherability of directed figure codes with domino graphs”. W: *Schedae Informaticae 22* (2014), s. 27–40.

4.3 Omówienie osiągnięcia naukowego

Kody, czyli podzbiory X monoidu takie, że każdy iloczyn ma jednoznaczny rozkład nad X , są powszechnym przedmiotem badań. Typowe kody o zmiennej długości (podzbiory wolnego monoidu Σ^* , gdzie Σ jest skończonym, niepustym alfabetem) są najbardziej znanym przykładem, a ich własności są powszechnie znane; zob. klasyczna monografia Berstela i Perina [10]. Omawiane własności obejmują defekt, rozstrzygalność deszyfrowalności (własności bycia kodem), jak również okresowość, maksymalność, synchronizację i powiązanie między kodami a automata — by wymienić tylko kilka przykładów. Różni autorzy rozszerzali kody na inne struktury, w tym drzewa, np. Karhumäki, Mantaci i Restivo [26, 35], oraz poliomina, np. Aigrain i Beauquier [3, 8]; dwuwymiarowe obrazy prostokątne były badane przez Giammarresi i Restivo [19]. Wielu autorów, jak np. Costagliola i in. [16, 17], zajmowało się jeszcze innymi strukturami w dwóch wymiarach. Zainteresowanie to bierze się z wielorakich zastosowań obrazów w rozmaitych dyscyplinach. Co więcej, obrazy można traktować jako naturalne rozszerzenie słów. Warto zauważyć, że dziedzina ta przeżywa w ostatnich latach wyraźny rozkwit; zob. dla przykładu prace [4–7, 18, 25, 29].

Choć niektóre z rozszerzeń zachowują większość własności związanych z deszyfrowalnością, wiele własności tracimy w przypadku poliomina lub figur określonych jako poliomina z etykietami. Dla przykładu, działa wówczas jedynie bardzo ograniczona wersja twierdzenia o defekcie, a deszyfrowalność jest rozstrzygalna tylko dla niewielkiej klasy figur; zob. rozległe opracowanie [22], a także [P1, P6].

W naszej pracy staramy się rozszerzyć typową teorię kodów zmiennej długości na dwa wymiary w taki sposób, by zachować przynajmniej część własności związanych z deszyfrowalnością. Badania nasze są pierwszą obszerniejszą próbą takiego uogólnienia, wychodzącą poza dość restrykcyjną strukturę drzew lub obrazów prostokątnych. W tym celu definiujemy *figury skierowane* jako etykietowane poliomina z określonym punktem początkowym i końcowym. Układ taki pozwala na naturalną definicję konkatenacji i jest podobny do jednego z modeli, symbolic pixel pictures, opisanych w [16]. Używamy słowa „skierowane”, by podkreślić sposób konkatenacji figur; jest to inne znaczenie tego słowa niż np. w określeniu „poliomina skierowane”.

W pracy [H1] zajmujemy się testowaniem deszyfrowalności i własnością defektu dla figur skierowanych. Dowodzimy, że sprawdzenie, czy dany skończony zbiór figur jest kodem, jest rozstrzygalne i podajemy konstruktywny algorytm (konstruktywny w tym sensie, że dla zbioru nie będącego kodem znajduje podwójny rozkład). Jest to istotna zmiana w stosunku do wspomnianych wcześniej modeli obrazów. Z drugiej strony pokazujemy kilka kontrprzykładów dla twierdzenia o defekcie.

W pracy [H2] rozszerzamy wcześniejsze wyniki poprzez dopuszczenie ogólniejszej definicji kodu. Prowadzi to do czterech rodzajów kodów, z których pewne mogą okazać się bardziej przydatne w zastosowaniach takich jak np. indeksowanie obrazów lub „obrazkowe kody kreskowe”. Pokazujemy, że kody te tworzą nietrywialną hierarchię ze względu na inkluzję. Algorytmy znajdujące podwójny rozkład figury dla zbioru nie będącego kodem można prosto wywieść z dowodu twierdzenia.

Samoorganizacja to proces, w którym proste, autonomiczne elementy tworzą bardziej złożone struktury bez żadnego zewnętrznego sterowania. Jest to powszechne zjawisko w świecie fizyki i biologii, które można obserwować w rozmaitych zakresach — od cząsteczek, poprzez

komórki (w sensie biologicznym), po zjawiska makroskopowe. Do opisu samoorganizacji zaproponowano różne modele matematyczne, w szczególności znany Tile Assembly Model Erica Winfree’ego, oparty na płytkach Wang’a na płaszczyźnie. W [H3] proponujemy model oparty na figurach skierowanych, wyposażonych w konkatenację częściową, określoną dla figur nienakładających się. Udowadniamy kilka własności dotyczących rozstrzygalności, związanych z wypełnianiem płaszczyzny przez dany zbiór figur i z tworzeniem nieskończonych i semi-nieskończonych zipperów. Choć nie są to zupełnie nowe rezultaty z perspektywy płytek Wang’a, pozwalają jednak na jednolite podejście do samoorganizacji, które może okazać się użyteczne w dalszych badaniach.

Klasyczne pojęcie kodu wymaga, by zakodowana wiadomość została jednoznacznie odcodowana, tzn. należy odtworzyć pierwotny ciąg słów kodowych. Jednakże w niektórych sytuacjach wystarczające może być odtworzenie jedynie multisetu, zbioru lub zaledwie liczby słów kodowych. Prowadzi to do trzech rodzajów deszyfrowalności, znanych jako, odpowiednio, deszyfrowalność *multisetowa* (multiset decipherability, MSD), *zbiorowa* (set decipherability, SD) i *liczbowa* (numeric decipherability, ND). Oryginalna pełna deszyfrowalność zwana jest *deszyfrowalnością jednoznaczną* (unique decipherability, UD). W pracy [H4] rozszerzamy wcześniejsze rezultaty, rozważając nie tylko kody UD, lecz także MSD, SD i ND dla figur skierowanych. Pokazujemy rozstrzygalność lub nierozstrzygalność dla poszczególnych kombinacji niezależnych kryteriów: typ konkatenacji (z funkcją łączącą lub bez niej), rodzaj deszyfrowalności (UD, MSD, SD, ND) oraz geometria kodu (kilka klas wyznaczonych przez wzajemne położenie punktów początkowych i końcowych figur). Dwie kombinacje pozostają jednak wciąż otwarte.

W pracach [H5, H7] definiujemy wariant grafów dominowych, co pozwala na rozstrzygnięcie niektórych rodzajów deszyfrowalności poprzez poszukiwanie pewnych ścieżek w grafie. Jest to koncept oparty na pomysłach Heada i Webera [24]. Główny wynik charakteryzuje rozstrzygalność deszyfrowalności za pomocą własności grafowych.

W pracy [H6] zajmujemy się trzema modelami figur na płaszczyźnie \mathbb{Z}^2 w oparciu o definicje sformułowane w [H1], [P1] i [31]. Pokazujemy, że nie da się w rozsądny sposób uzyskać własności defektu dla figur skierowanych, zarówno z funkcją łączącą, jak i bez niej.

4.3.1 Preliminaria

Niech Σ będzie skończonym, niepustym alfabetem. *Przesunięcie* o wektor u w \mathbb{Z}^2 i jego naturalne rozszerzenia oznaczamy przez tr_u . Niech dist będzie metryką miejską na \mathbb{Z}^2 . Zbiór $U \subseteq \mathbb{Z}^2$ jest *spójny*, jeżeli dla dowolnych $x, y \in U$ istnieje ciąg $x = x_1, x_2, \dots, x_n, x_{n+1} = y$ zawarty w U taki, że $\text{dist}(x_i, x_{i+1}) = 1$ dla $i \in \{1, \dots, n\}$.

Definicja 1 (Figura skierowana) Niech $D \subseteq \mathbb{Z}^2$ będzie skończony i spójny, $b \in D$, $e \in D$ lub $\text{dist}(e, D) = 1$ oraz $\ell : D \rightarrow \Sigma$. Czwórkę $f = (D, b, e, \ell)$ nazywamy figurą skierowaną (nad Σ) o

dziedzynie	$\text{dom}(f)$	$= D,$
punkcie początkowym	$\text{begin}(f)$	$= b,$
punkcie końcowym	$\text{end}(f)$	$= e,$
etykietowaniu	$\text{label}(f)$	$= \ell.$

Wektor przesunięcia figury f określony jest jako $\text{tran}(f) = \text{end}(f) - \text{begin}(f)$. Ponadto pusta figura skierowana ε to $(\emptyset, (0, 0), (0, 0), \{\})$, gdzie $\{\}$ oznacza funkcję o pustej dziedzinie.

Zbiór wszystkich figur skierowanych nad Σ oznaczamy przez Σ^\diamond . Dwie figury skierowane x, y są *równe* (co zapisujemy jako $x = y$), jeżeli istnieje $u \in \mathbb{Z}^2$ taki, że

$$y = (\text{tr}_u(\text{dom}(x)), \text{tr}_u(\text{begin}(x)), \text{tr}_u(\text{end}(x)), \text{tr}_u(\text{label}(x))).$$

Figury rozważamy zatem z dokładnością do przesunięcia.

Przykład 1 *Figura skierowana i jej reprezentacja graficzna. Każdy punkt dziedziny, (x, y) , reprezentowany jest przez kwadrat jednostkowy w \mathbb{R}^2 , którego lewy dolny róg ma współrzędne (x, y) . Kółko oznacza punkt początkowy, zaś kwadracik — punkt końcowy figury. Figury rozważamy z dokładnością do przesunięcia, przeto nie zaznaczamy współrzędnych.*

$$(\{(0, 0), (1, 0), (1, 1)\}, (0, 0), (2, 1), \{(0, 0) \mapsto a, (1, 0) \mapsto b, (1, 1) \mapsto c\}) \quad \begin{array}{|c|c|} \hline & c \\ \hline a & b \\ \hline \end{array} \diamond$$

Definicja 2 (Konkatenacja (z funkcją łączącą), m -konkatenacja)

Niech $x = (D_x, b_x, e_x, \ell_x)$ i $y = (D_y, b_y, e_y, \ell_y)$ będą figurami skierowanymi. Konkatenację figur x i y z funkcją łączącą $m : \Sigma \times \Sigma \rightarrow \Sigma$ określamy jako

$$x \circ_m y = (D_x \cup \text{tr}_{e_x - b_y}(D_y), b_x, \text{tr}_{e_x - b_y}(e_y), \ell),$$

gdzie

$$\ell(z) = \begin{cases} \ell_x(z) & \text{dla } z \in D_x \setminus \text{tr}_{e_x - b_y}(D_y), \\ \text{tr}_{e_x - b_y}(\ell_y)(z) & \text{dla } z \in \text{tr}_{e_x - b_y}(D_y) \setminus D_x, \\ m(\ell_x(z), \text{tr}_{e_x - b_y}(\ell_y)(z)) & \text{dla } z \in D_x \cap \text{tr}_{e_x - b_y}(D_y). \end{cases}$$

Przykład 2

$$\begin{array}{|c|c|} \hline & c \\ \hline a & b \\ \hline \end{array} \diamond \circ_{\pi_1} \begin{array}{|c|c|} \hline a & \\ \hline b & c \\ \hline \end{array} \diamond = \begin{array}{|c|c|} \hline a & \\ \hline c & c \\ \hline a & b \\ \hline \end{array} \diamond$$

gdzie π_1 oznacza rzutowanie na pierwszy argument.

Obserwacja 1 $\Sigma_m^\diamond = (\Sigma^\diamond, \circ_m)$ jest monoidem wtedy i tylko wtedy, gdy m jest łączna.

Od tej pory niech m będzie dowolną łączną funkcją łączącą. Piszemy Σ_m^\diamond na oznaczenie zarówno monoidu, jak i samego zbioru figur. Nadużywając tej notacji, piszemy również X_m^\diamond na oznaczenie zbioru wszystkich figur, które można uzyskać za pomocą konkatenacji figur ze zbioru $X \subseteq \Sigma_m^\diamond$. Indeks m zwykle pomijamy, gdy m jest nieistotne lub oczywiste (np. gdy $|\Sigma| = 1$), pisząc także xy zamiast $x \circ_m y$.

Obserwacja 2 Monoid Σ_m^\diamond nie jest wolny, gdyż jego baza musi zawierać „figury jednostkowe” (figury typu

$$a_E = \begin{array}{|c|} \hline a \\ \hline \end{array} \diamond, a_N = \begin{array}{|c|} \hline a \\ \hline \end{array}, a_W = \begin{array}{|c|} \hline \diamond \\ \hline \end{array}, a_S = \begin{array}{|c|} \hline \diamond \\ \hline \end{array},$$

dla wszystkich $a \in \Sigma$), co prowadzi do sprzeczności z założeniem, że Σ_m^\diamond jest wolny.

Przykład 3 Niezależnie od funkcji łączącej, podana figura rozkłada się na dwa sposoby, np. jako $a_N a_N a_E a_S a_S a_E a_N a_N a_E$ oraz $a_E a_E a_N a_W a_W a_N a_E a_E a_E$.

$$\begin{array}{|c|c|c|} \hline a & a & a \\ \hline a & a & a \\ \hline a & a & a \\ \hline \end{array} \diamond$$

Definicja 3 (Kod (z funkcją łączącą), m -kod) $X \subseteq \Sigma_m^\diamond$ jest kodem, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość

$$x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$$

pociąga $k = l$ oraz $x_i = y_i$ dla $i \in \{1, \dots, k\}$.

Przykład 4 $X = \{ \begin{array}{|c|} \hline a \\ \hline \end{array} \diamond, \begin{array}{|c|} \hline a \\ \hline \end{array} \} \subseteq \{a\}_{\pi_1}^\diamond$ w oczywisty sposób jest kodem.

Przykład 5 $X = \{w = \begin{array}{|c|c|} \hline a & a \\ \hline a & a \\ \hline \end{array} \diamond, x = \begin{array}{|c|c|} \hline a & a \\ \hline a & a \\ \hline \end{array} \diamond, y = \begin{array}{|c|} \hline a \\ \hline \end{array} \diamond, z = \begin{array}{|c|} \hline a \\ \hline \end{array} \} \subseteq \{a\}_{\pi_1}^\diamond$ nie jest kodem, gdyż

$$wx = yz = \begin{array}{|c|c|} \hline a & a \\ \hline a & a \\ \hline \end{array} \diamond.$$

4.3.2 Deszyfrowalność na figurach skierowanych jest rozstrzygalna

W pracy [H1] zajmujemy się sprawdzaniem deszyfrowalności i własnością defektu dla figur skierowanych. Pokazujemy, że sprawdzenie, czy dany skończony zbiór figur jest kodem, jest rozstrzygalne, oraz podajemy konstruktywny algorytm (konstruktywny w tym sensie, że znajduje podwójny rozkład dla niekodu). Jest to istotna zmiana w stosunku do wspomnianych poprzednio modeli. Z drugiej strony, wskazujemy kontrprzykłady do twierdzenia o defekcie.

Warunek konieczny deszyfrowalności

Twierdzenie 1 (Warunek konieczny) *Niech $X = \{x_1, \dots, x_n\} \subseteq \Sigma_m^\circ$. Jeżeli istnieją $\alpha_{i_1}, \dots, \alpha_{i_k} \in \mathbb{Z}_+$, gdzie $\{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$, takie że*

$$\sum_{j=1}^k \alpha_{i_j} \text{tran}(x_{i_j}) = (0, 0),$$

to X nie jest kodem.

Definicja 4 (Zbiory jedno- i dwustronne) *Zbiór, który spełnia warunek z twierdzenia 1, nazywamy dwustronnym; w przeciwnym razie nazywamy go jednostronnym.*

Rozstrzygalność sprawdzania deszyfrowalności

Pokazujemy, że sprawdzenie, czy dany zbiór figur jest kodem, jest rozstrzygalne. Dowód ten ma dla naszej pracy zasadnicze znaczenie, nie tylko jako główny wynik artykułu, ale również jako narzędzie w dalszych badaniach. Zaczynamy od obserwacji, które pozwalają skonstruować „obszary ograniczające” dla figur. Przechodzimy następnie do własności gwarantujących skończoność zbiorów możliwych konfiguracji i w konsekwencji rozstrzygalność omawianego problemu.

Niech $X = \{x_1, \dots, x_n\} \subseteq \Sigma_m^\circ$. Albo istnieje wektor $\tau_E \in \mathbb{Z}^2$ taki, że $\tau_E \cdot \text{tran}(x) > 0$ dla $x \in X$, albo na mocy twierdzenia 1 zbiór X nie jest kodem. Jeżeli τ_E istnieje, można wziąć wektor na tyle długi, by dla dowolnego $x \in X$

$$\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \text{HP}(\tau_E, \text{begin}(x)),$$

gdzie, dla danych $v, w \in \mathbb{R}^2$, $\text{HP}(v, w)$ oznacza półpłaszczyznę $\{u \in \mathbb{Z}^2 \mid v \cdot (u - (w + v)) \leq 0\}$. Zauważmy, że zatem

$$\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \text{HP}(\tau_E, \text{end}(x))$$

dla każdego $x \in X$. Bez straty ogólności możemy założyć, że

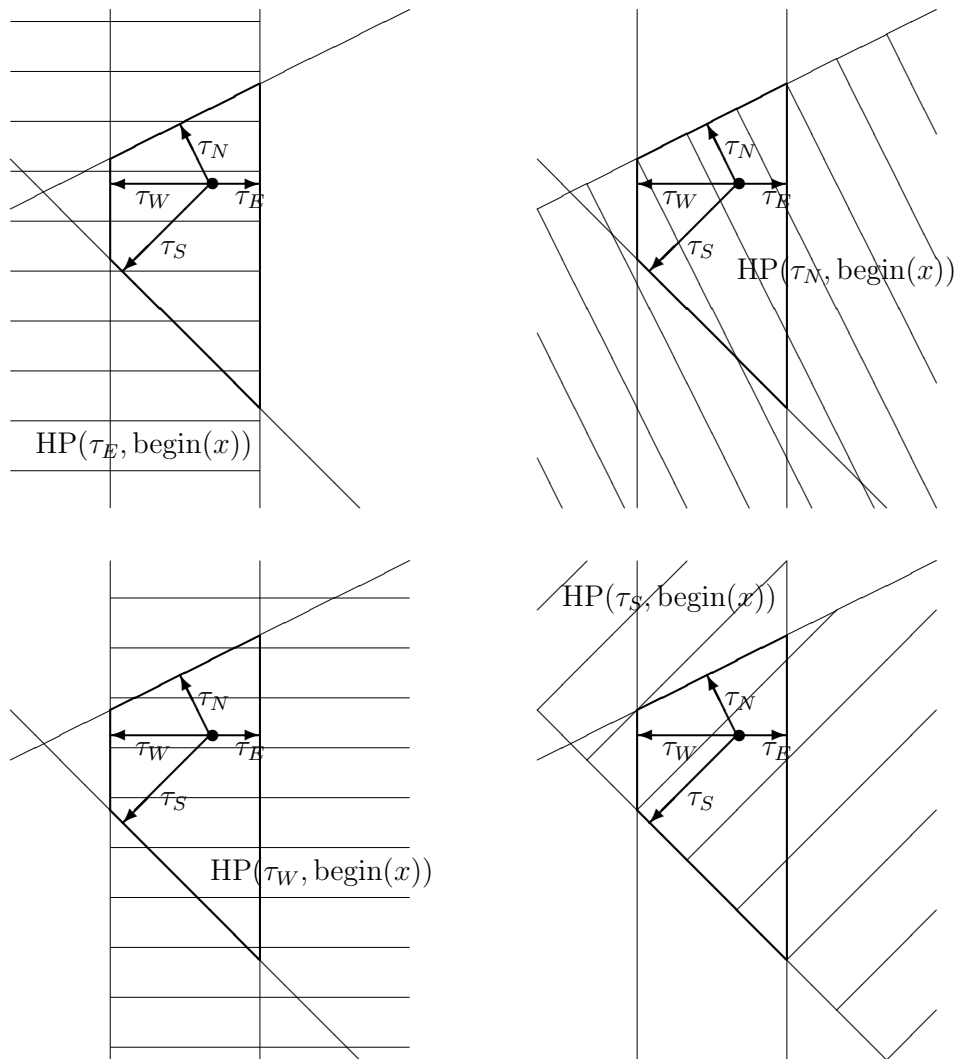
$$\angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_1)) \leq \angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_2)) \leq \dots \leq \angle(R_{-\frac{\pi}{2}}(\tau_E), \text{tran}(x_n))$$

(gdzie R_ϕ oznacza obrót o kąt ϕ , a \angle to kąt rozpięty przez dwa wektory) oraz $\text{begin}(x) = (0, 0)$ dla $x \in X$. Dobierzmy teraz stałe $r_S, r_N, r_W > 0$ takie, że wektory

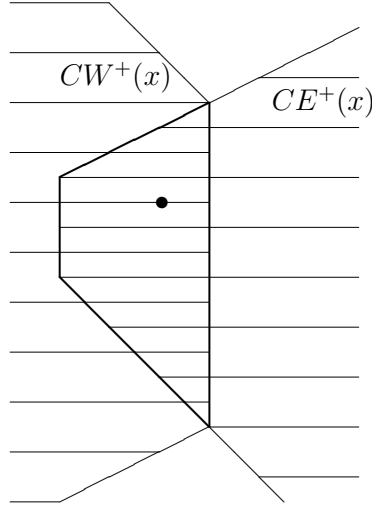
$$\begin{aligned} \tau_N &= r_N R_{\frac{\pi}{2}}(\text{tran}(x_n)), \\ \tau_W &= -r_W \tau_E, \\ \tau_S &= r_S R_{-\frac{\pi}{2}}(\text{tran}(x_1)) \end{aligned}$$

wyznaczają „obszary ograniczające” dla figur z X , tzn. dla $x \in X$ zachodzi

$$\text{dom}(x) \cup \{\text{end}(x)\} \subseteq \bigcap_{\tau \in \{\tau_N, \tau_W, \tau_S\}} \{\text{HP}(\tau, \text{begin}(x))\}.$$



Rysunek 1: Półpłaszczyzny $HP(\tau, \text{begin}(x))$ dla $\tau \in \{\tau_E, \tau_N, \tau_W, \tau_S\}$; czarna kropka oznacza punkt początkowy figury x .



Rysunek 2: Obszary $CW^+(x)$ i $CE^+(x)$; czarna kropka oznacza punkt końcowy figury x .

Rysunek 1 przedstawia półpłaszczyzny $HP(\tau, \text{begin}(x))$ dla $\tau \in \{\tau_E, \tau_N, \tau_W, \tau_S\}$. Dla $x \in X_m^\diamond$ zdefiniujemy

$$\begin{aligned} CE^+(x) &= HP(\tau_S, \text{end}(x)) \cap HP(\tau_N, \text{end}(x)) \cap HP(\tau_W, \text{end}(x)), \\ CE^-(x) &= \mathbb{Z}^2 \setminus CE^+(\text{end}(x)), \\ CW^+(x) &= \bigcup_v \{v + (CE^+(\text{end}(x)) \cap HP(\tau_E, \text{end}(x)))\}, \\ CW^-(x) &= \mathbb{Z}^2 \setminus CW^+(\text{end}(x)), \end{aligned}$$

gdzie suma w definicji $CW^+(x)$ przebiega po $v \in \mathbb{Z}^2$ leżących wewnątrz kąta rozpiętego przez wektory $-\text{tran}(x_1)$ i $-\text{tran}(x_n)$.

Definiujemy *konfigurację* jako parę (x, y) , gdzie $x, y \in X_m^\diamond$. Mówimy, że $(x', y') \in (X_m^\diamond)^2$ jest *następnikiem* konfiguracji (x, y) , pisząc $(x, y) \prec (x', y')$, jeżeli

$$\begin{aligned} x' &= xx'' \text{ dla pewnego } x'' \in X \text{ oraz } y = y', \text{ lub} \\ y' &= yy'' \text{ dla pewnego } y'' \in X \text{ oraz } x = x'. \end{aligned}$$

Przez \prec^* oznaczamy przechodnie domknięcie \prec . Oczywiście X nie jest kodem wtedy i tylko wtedy, gdy istnieją $x, y \in X$ i $z \in X_m^\diamond$ takie, że $x \neq y$ i $(x, y) \prec^* (z, z)$. Naszym celem jest albo znaleźć konfigurację (x, y) taką, że $x \neq y$ oraz

$$(x, y) \prec \dots \prec (z, z)$$

(wówczas X nie jest kodem), albo pokazać, że taka konfiguracja nie istnieje (wówczas X jest kodem). Konfiguracja spełniająca powyższy warunek zwana jest konfiguracją *użyteczną*. Potencjalnie istnieje nieskończenie wiele konfiguracji, które należy sprawdzić; poniższe własności pozwalają zmniejszyć ich liczbę.

Własność 1 Jeżeli (x, y) jest użyteczna i $(x', y') \prec (x, y)$, to (x', y') jest użyteczna.

Własność 2 Jeżeli (x, y) jest użyteczna, to

$$\begin{aligned} \text{end}(x) &\in CW^+(y) \cup CE^+(y), \\ \text{end}(y) &\in CW^+(x) \cup CE^+(x). \end{aligned}$$

Własność 3 Jeżeli (x, y) jest użyteczna, to

$$\text{label}(x) \mid_{CE^-(x) \cap CE^-(y)} \equiv \text{label}(y) \mid_{CE^-(x) \cap CE^-(y)}.$$

Zauważmy, że nie jest nam potrzebna cała informacja zawarta w konfiguracjach, a jedynie te etykietowania, które mogą ulec zmianie wskutek dalszych konkatencji. Na mocy własności 3, zamiast (x, y) możemy rozważać konfigurację zredukowaną, określoną jako para $(\pi_{RC}(x, y), \pi_{RC}(y, x))$, gdzie

$$\pi_{RC}(z, z') = (\text{end}(z), \text{label}(z) \mid_{\text{dom}(z) \setminus (CE^-(z) \cap CE^-(z'))}).$$

Własność 1 oznacza, że wystarczy rozważać konfiguracje, w których „rozciągłość” (odległość skrajnych punktów) w kierunku τ_E jest ograniczona przez $|\tau_E|$, tzn. $|\tau_E \cdot (\text{end}(x) - \text{end}(y))| \leq |\tau_E|^2$, gdyż jedna figura przesuwana $\text{end}(x)$ lub $\text{end}(y)$ o co najwyżej $|\tau_E|$. Ponadto własność 2 ogranicza rozciągłość prostopadłą (w kierunku $R_{-\frac{\pi}{2}}(\tau_E)$). Stąd liczba konfiguracji zredukowanych, z dokładnością do przesunięcia, jest skończona. Prowadzi nas to do głównego twierdzenia artykułu:

Twierdzenie 2 Problem, czy dany skończony zbiór $X \subseteq \Sigma_m^\circ$ jest kodem, jest rozstrzygalny.

Algorytm

Z dowodu twierdzenia 2 otrzymujemy algorytm sprawdzający, czy dany zbiór figur skierowanych jest kodem. Następująca procedura zwraca **true**, jeżeli zbiór na wejściu jest kodem; w przeciwnym razie zwraca **false**. Ponadto, jeżeli zbiór nie jest kodem, algorytm znajduje $x_1, \dots, x_k, x'_1, \dots, x'_l$ takie, że $x_1 \cdots x_k = x'_1 \cdots x'_l$. Zauważmy, że z tego właśnie powodu główna pętla algorytmu przetwarza konfiguracje (a nie tylko konfiguracje zredukowane).

1. **input** $X = \{x_1, \dots, x_n\} \subseteq \Sigma_m^\circ$
2. **if** $\exists \alpha_{i_1}, \dots, \alpha_{i_k} \in \mathbb{Z}_+, \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\} : \sum_{j=1}^n \alpha_{i_j} \text{tran}(x_{i_j}) = (0, 0)$
then return false
3. **compute** $\tau_E, \tau_W, \tau_S, \tau_N$
4. $C := \emptyset$ (zbiór sprawdzonych konfiguracji),
 $RC := \emptyset$ (zbiór sprawdzonych konfiguracji zredukowanych)
5. **for each** $x, y \in \{\text{tr_begin}(x) \mid x \in X\}, x \neq y$ **do**
 - (a) **if** (x, y) może być użyteczna (spełnione są warunki z własności 2 i 3)
and $(\pi_{RC}(x, y), \pi_{RC}(y, x)) \notin RC$
then
 $C := C \cup \{(x, y)\}$,
 $RC := RC \cup \{(\pi_{RC}(x, y), \pi_{RC}(y, x))\}$
6. $RC_{TMP} := \emptyset$
7. **while** $RC \neq RC_{TMP}$ **do**
 - (a) $C_{TMP} := C, RC_{TMP} := RC, C := \emptyset$
 - (b) **for each** $(x, y) \in C_{TMP}$ **and** $z \in X$ **do**
 - i. **if** $xz = y$ **or** $x = yz$
then return false

- ii. **if** (xz, y) może być użyteczna
and $|\tau_E \cdot (\text{end}(xz) - \text{end}(y))| \leq |\tau_E|^2$
and $(\pi_{RC}(xz, y), \pi_{RC}(y, xz)) \notin RC$
then
 $C := C \cup \{(xz, y)\},$
 $RC := RC \cup \{(\pi_{RC}(xz, y), \pi_{RC}(y, xz))\}$
- iii. **if** (x, yz) może być użyteczna
and $|\tau_E \cdot (\text{end}(x) - \text{end}(yz))| \leq |\tau_E|^2$
and $(\pi_{RC}(x, yz), \pi_{RC}(yz, x)) \notin RC$
then
 $C := C \cup \{(x, yz)\},$
 $RC := RC \cup \{(\pi_{RC}(x, yz), \pi_{RC}(yz, x))\}$

8. **return true**

Zauważmy, że konfiguracje są rozpatrywane z dokładnością do przesunięcia; odnosi się to w szczególności do testu „... $\notin RC$ ”. Faktyczna implementacja może nie używać zbioru RC_{TMP} , a jako warunek pętli przyjąć $C \neq \emptyset$ (nowy element jest dodawany do C wtedy i tylko wtedy, gdy nowy element jest dodawany do RC). Zmiana taka poprawiłaby efektywność algorytmu, ale przesłoniłaby jego podobieństwo do algorytmu Sardinasa i Pattersona. Implementacja może również wykorzystywać symetryczną naturę konfiguracji.

Własność defektu

Przytaczamy tu kilka kontrprzykładów do twierdzenia o defekcie dla figur skierowanych. Pokazują one, że własność ta zawodzi nawet dla bardzo prostych zbiorów. Z drugiej strony, załączenie figur do „niby-słów”, z odpowiednio dobranymi punktami początkowymi i końcowymi, gwarantuje oczywiście zachowanie własności defektu.

Twierdzenie 3 (Klasyczna własność defektu) *Niech $X \subseteq \Sigma^+$ będzie niekodem. Istnieje $Y \subseteq \Sigma^+$ taki, że $|Y| < |X|$ oraz $X^+ \subseteq Y^+$.*

Rozważmy teraz następujące przykłady:

Przykład 6 *Niech $\Sigma = \{a\}$, $m = \{(a, a) \mapsto a\}$ oraz*

$$X = \{x = \boxed{a} \diamond, y = \diamond \boxed{a}\}.$$

Dla dowolnego $n \geq 1$

$$(xy)^n = \boxed{\diamond a}$$

zatem X nie jest kodem. Nie istnieje jednak Y taki, że $|Y| < 2$ i $X^+ \subseteq Y^+$.

Własność defektu nie zachodzi więc nawet dla kwadratów jednostkowych. Co więcej, istnieją singletony nie będące kodami:

Przykład 7 *Niech $\Sigma = \{a\}$, $m = \{(a, a) \mapsto a\}$ oraz*

$$X = \{\boxed{\diamond}\}$$

Dla dowolnego $n \geq 1$

$$x^n = x$$

zatem X nie jest kodem. Nie istnieje jednak Y taki, że $|Y| < 1$ i $X^+ \subseteq Y^+$.

Własność defektu nie zachodzi także dla niekodów, które nie pozwalają utworzyć figury o zerowym wektorze przesunięcia:

Przykład 8 Niech $\Sigma = \{a, b, c\}$, $m = \{(a, \cdot) \mapsto a, (\cdot, a) \mapsto a, \dots\}$ (pozostałe wartości dowolne) oraz

$$X = \{x = \boxed{a \diamond a}, y = \boxed{a \diamond a}\}.$$

Wówczas

$$xy = yx = \boxed{a \diamond a}$$

zatem X nie jest kodem. Nie istnieje jednak Y taki, że $|Y| < 2$ i $X^+ \subseteq Y^+$.

Uwagi

Warto zauważyć, że złożoność sprawdzania deszyfrowalności zależy od kąta rozpiętego przez wektory przesunięć figur. Większe kąty dają wyższą złożoność, gdyż należy wówczas sprawdzić więcej konfiguracji. Oczywiste ograniczenie górne zależy wykładniczo od rozmiaru „obszarów ograniczających”; ten z kolei rośnie proporcjonalnie do $\text{tg}(\alpha/2)$, gdzie α to wspomniany kąt. Jednakże dla $\alpha = \pi$ złożoność radykalnie spada ze względu na twierdzenie 1. Po drugiej stronie spektrum, gdy kąt zmierza do zera, figury stają się podobne do słów, a działanie algorytmu przypomina działanie algorytmu Sardinas i Pattersona.

4.3.3 Kody ze słabą równością

W pracy [H2] rozszerzamy wcześniejsze wyniki poprzez dopuszczenie ogólniejszej definicji kodu. Prowadzi to do czterech rodzajów kodów, z których pewne mogą okazać się bardziej przydatne w zastosowaniach takich jak np. indeksowanie obrazów lub „obrazkowe kody kreskowe”. Pokazujemy, że kody te tworzą nietrywialną hierarchię ze względu na inkluzję. Głównym rezultatem jest tu rozstrzygalność sprawdzania deszyfrowalności dla wszystkich czterech rodzajów kodów. Algorytmy znajdujące podwójny rozkład figury dla zbioru nie będącego kodem można prosto wywieść z dowodu twierdzenia.

Nasze rozszerzenie osłabia (jeden lub obydwa) warunki równości figur w klasycznej definicji kodu:

$$x_1 \cdots x_k = y_1 \cdots y_l \Rightarrow k = l \text{ oraz } x_i = y_i \text{ dla } i \in \{1, \dots, k\}.$$

Jest to podejście zbliżone do (R, S) -kodów określonych przez Halavę i in. w [21]. Słaba równość pomija punkty początkowe i końcowe figur, pozwalając na prostsze reprezentacje, którą mogą być bardziej odpowiednie we wspomnianych zastosowaniach.

Definiujemy cztery rodzaje kodów na figurach skierowanych, będące rezultatem ewentualnego użycia słabej równości w klasycznej definicji kodu, i pokazujemy ich hierarchię ze względu na inkluzję. Przypomnijmy, że dwie figury skierowane x, y są *równe* ($x = y$) jeżeli istnieje $u \in \mathbb{Z}^2$ taki, że

$$y = (\text{tr}_u(\text{dom}(x)), \text{tr}_u(\text{begin}(x)), \text{tr}_u(\text{end}(x)), \text{tr}_u(\text{label}(x))).$$

Dwie figury skierowane x, y są *słabo równe* (z dokładnością do punktów początkowych i końcowych; $x \simeq y$), jeżeli istnieje $u \in \mathbb{Z}^2$ taki, że

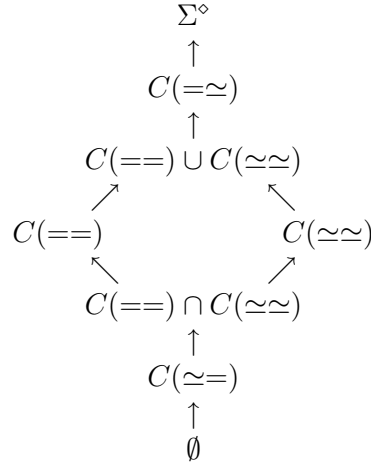
$$\text{dom}(y) = \text{tr}_u(\text{dom}(x)) \text{ oraz } \text{label}(y) = \text{tr}_u(\text{label}(x)).$$

Zauważmy, że dla dowolnych $x, y \in \Sigma^\circ$, $x = y$ implikuje $x \simeq y$.

Definicja 5 (Kody ze słabą równością) Niech $\stackrel{R}{\equiv}$ i $\stackrel{S}{\equiv}$ będą równościami na figurach: $=$ lub \simeq . Zbiór $X \subseteq \Sigma_m^\circ$ jest $(\stackrel{R}{\equiv}, \stackrel{S}{\equiv})$ -kodem, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \cdots x_k \stackrel{R}{\equiv} y_1 \cdots y_l$ implikuje $k = l$ oraz $x_i \stackrel{S}{\equiv} y_i$ dla $i \in \{1, \dots, k\}$.

Zbiór wszystkich $(\stackrel{R}{=} \stackrel{S}{=})$ -kodów oznaczamy przez $C(\stackrel{R}{=} \stackrel{S}{=})$. Dla przejrzystości zapisu pomijamy odniesienie do Σ_m° . Określamy zatem cztery rodziny kodów, $C(==)$, $C(=\simeq)$, $C(\simeq=)$ i $C(\simeq\simeq)$, gdzie $C(==)$ oznacza dotychczasowe kody na figurach skierowanych.

Własność 4 *Diagram przedstawia zawierania pomiędzy powyższymi rodzinami kodów. Wszystkie inkluzje są właściwe.*



Główny wynik omawianego artykułu to rozstrzygalność deszyfrowalności dla wszystkich czterech rodzajów kodów.

Twierdzenie 4 *Problem, czy dany skończony zbiór $X \subseteq \Sigma_m^\circ$ jest kodem danego rodzaju, $(==)$, $(=\simeq)$, $(\simeq=)$ lub $(\simeq\simeq)$, jest rozstrzygalny.*

4.3.4 Pokrywanie płaszczyzny figurami skierowanymi

Samoorganizacja to proces, w którym proste, autonomiczne elementy tworzą bardziej złożone struktury bez żadnego zewnętrznego sterowania. Jest to powszechne zjawisko w świecie fizyki i biologii, które można obserwować w rozmaitych zakresach — od cząsteczek, poprzez komórki (w sensie biologicznym), po zjawiska makroskopowe. Zainteresowanie samoorganizacją wynika z perspektywicznych zastosowań technologicznych m.in. w inżynierii materiałowej, medycynie, robotyce i obliczeniach opartych o DNA. Warto podkreślić, że interakcje pomiędzy elementami są lokalne; jest to podstawowa charakterystyka samoorganizacji.

Matematyczne badania samoorganizacji zostały zainicjowane przez Adlemana i Winfree'ego [1, 41] już w latach 90., a w ostatniej dekadzie stały się ponownie przedmiotem wzmożonego zainteresowania. Rozległą bibliografię tej tematyki można znaleźć np. w pracy [2].

Do opisu samoorganizacji zaproponowano różne modele matematyczne, w szczególności znany Tile Assembly Model Erica Winfree'ego, oparty na płytkach Wanga na płaszczyźnie. W [H3] proponujemy model oparty na figurach skierowanych, wyposażonych w konkatenację częściową, określoną dla figur nienakładających się.

Proponowany przez nas model jest elastyczny i pozwala na naturalny opis samoorganizujących się elementów, jak również np. na reprezentację obrazów, por. [H2]. Udowadnimy kilka własności dotyczących rozstrzygalności, związanych z pokrywaniem płaszczyzny przez dany zbiór figur i z tworzeniem nieskończonych i semi-nieskończonych zipperów. Choć nie są to zupełnie nowe rezultaty z perspektywy płytek Wanga, pozwalają jednak na jednolite podejście do samoorganizacji, które może okazać się użyteczne w dalszych badaniach.

Nowe definicje

Przedział całkowity $\{k \in \mathbb{Z} \mid i \leq k \leq j\}$ oznaczamy przez $[i, \dots, j]$. Nie wymagamy od tej pory, by dziedzina figury była spójna, i nie czynimy żadnych założeń co do wzajemnego położenia dziedziny, punktu początkowego i końcowego figury. Oprócz dotychczasowej konkatenacji z funkcją łączącą definiujemy wariant bez takiej funkcji.

Definicja 6 (Konkatenacja bez funkcji łączącej)

Niech $x = (D_x, b_x, e_x, \ell_x)$ i $y = (D_y, b_y, e_y, \ell_y)$ będą figurami skierowanymi. Jeżeli

$$D_x \cap \text{tr}_{e_x - b_y}(D_y) = \emptyset,$$

to konkatenacja bez funkcji łączącej figur x i y określona jest jako

$$x \circ y = (D_x \cup \text{tr}_{e_x - b_y}(D_y), b_x, \text{tr}_{e_x - b_y}(e_y), \ell),$$

gdzie

$$\ell(z) = \begin{cases} \ell_x(z) & \text{dla } z \in D_x, \\ \text{tr}_{e_x - b_y}(\ell_y)(z) & \text{dla } z \in \text{tr}_{e_x - b_y}(D_y). \end{cases}$$

Jeżeli $D_x \cap \text{tr}_{e_x - b_y}(D_y) \neq \emptyset$, to $x \circ y$ nie jest określona.

Przykład 9

Przykład 10

Powyższa konkatenacja nie jest określona, gdyż figury nie spełniają warunku z definicji: punkt z etykietą c koliduje z punktem e .

Zippery nieskończone i semi-nieskończone

Nieformalnie, ciąg figur $(f_i)_{i \in \mathbb{Z}}$ nazywamy *zipperem nieskończonym*, jeżeli określona jest nieskończona konkatenacja $\dots f_{-1} \circ f_0 \circ f_1 \dots$, tzn. nie ma kolizji dziedzin. Formalnie:

Definicja 7 (Zipper nieskończony) Niech $F \subseteq \Sigma^\diamond$. Ciąg figur $(f_i)_{i \in \mathbb{Z}}$, gdzie $f_i \in F$, nazywamy zipperem nieskończonym (nad F), jeżeli dla dowolnych $i, j \in \mathbb{Z}$, $i < j$,

$$\text{dom}(f_i) \cap \text{tr}_\Delta(\text{dom}(f_j)) = \emptyset,$$

gdzie

$$\Delta = \text{end}(f_i) + \sum_{k=i+1}^{k=j-1} \text{tran}(f_k) - \text{begin}(f_j).$$

Obraz zippera nieskończonego $(f_i)_{i \in \mathbb{Z}}$ określamy jako $\text{zip}((f_i)_{i \in \mathbb{Z}}) = (D, L)$, gdzie

$$\begin{aligned} D &= \bigcup_{i=\dots, -2, -1} \text{tr}_{-\text{end}(f_i) - \sum_{k=i+1}^{k=-1} \text{tran}(f_k) + \text{begin}(f_0)}(\text{dom}(f_i)) \\ &\cup \text{dom}(f_0) \\ &\cup \bigcup_{i=1, 2, \dots} \text{tr}_{\text{end}(f_0) + \sum_{k=1}^{k=i-1} \text{tran}(f_k) - \text{begin}(f_i)}(\text{dom}(f_i)), \\ L &= \bigcup_{i=\dots, -2, -1} \text{tr}_{-\text{end}(f_i) - \sum_{k=i+1}^{k=-1} \text{tran}(f_k) + \text{begin}(f_0)}(\text{label}(f_i)) \\ &\cup \text{label}(f_0) \\ &\cup \bigcup_{i=1, 2, \dots} \text{tr}_{\text{end}(f_0) + \sum_{k=1}^{k=i-1} \text{tran}(f_k) - \text{begin}(f_i)}(\text{label}(f_i)). \end{aligned}$$

Warunek pustości przecięć dziedzin gwarantuje, że wszystkie konkatencje są określone. Zauważmy, że obraz nieskończonego zippera nie ma punktu początkowego ani końcowego.

W podobny sposób definiujemy *zipper semi-nieskończony* $(f_i)_{i \in \mathbb{N}}$:

Definicja 8 (Zipper semi-nieskończony) Niech $F \subseteq \Sigma^\circ$. Ciąg figur $(f_i)_{i \in \mathbb{N}}$, gdzie $f_i \in F$, nazywamy zipperem semi-nieskończonym (nad F), jeżeli dla dowolnych $i, j \in \mathbb{N}$, $i < j$,

$$\text{dom}(f_i) \cap \text{tr}_\Delta(\text{dom}(f_j)) = \emptyset,$$

gdzie

$$\Delta = \text{end}(f_i) + \sum_{k=i+1}^{k=j-1} \text{tran}(f_k) - \text{begin}(f_j).$$

Obraz zippera semi-nieskończonego $(f_i)_{i \in \mathbb{N}}$ określamy jako $\text{zip}((f_i)_{i \in \mathbb{N}}) = (D, L, b)$, gdzie

$$\begin{aligned} D &= \text{dom}(f_0) \cup \bigcup_{i=1,2,\dots} \text{tr}_{\text{end}(f_0) + \sum_{k=1}^{k=i-1} \text{tran}(f_k) - \text{begin}(f_i)}(\text{dom}(f_i)), \\ L &= \text{label}(f_0) \cup \bigcup_{i=1,2,\dots} \text{tr}_{\text{end}(f_0) + \sum_{k=1}^{k=i-1} \text{tran}(f_k) - \text{begin}(f_i)}(\text{label}(f_i)), \\ b &= \text{begin}(f_0). \end{aligned}$$

Definiujemy także zippery skończone:

Definicja 9 (Zipper skończony) Niech $F \subseteq \Sigma^\circ$. Ciąg figur $(f_l)_{l \in [i, \dots, j]}$, gdzie $f_l \in F$, nazywamy zipperem skończonym (nad F), jeżeli $f = f_i \circ \dots \circ f_j$ jest określona. Obraz zippera skończonego $(f_l)_{l \in [i, \dots, j]}$ o środku w k -tym elemencie, gdzie $k \in [i, \dots, j]$, określamy jako

$$\text{zip}_k((f_l)_{l \in [i, \dots, j]}) = (\text{tr}_\Delta(\text{dom}(f)), \text{tr}_\Delta(\text{label}(f)), \text{tr}_\Delta(\text{begin}(f)), \text{tr}_\Delta(\text{end}(f))),$$

gdzie

$$\Delta = -\text{begin}(f_i) - \sum_{l=i}^{l=k-1} \text{tran}(f_l) + \text{begin}(f_k).$$

Długość zippera skończonego to liczba elementów ciągu, czyli $j - i + 1$.

Obraz o środku w k -tym elemencie zachowuje pierwotną pozycję f_k , w odróżnieniu od zwykłej konkatencji, która zachowuje położenie pierwszej figury.

Zauważmy, że dla nieskończonych i semi-nieskończonych zipperów, odpowiednio:

$$\begin{aligned} \text{zip}((f_i)_{i \in \mathbb{Z}}) &= \left(\bigcup_{i=0}^{\infty} \text{dom}(\text{zip}_0((f_j)_{j \in [-i, \dots, i]})), \bigcup_{i=0}^{\infty} \text{label}(\text{zip}_0((f_j)_{j \in [-i, \dots, i]})) \right), \\ \text{zip}((f_i)_{i \in \mathbb{N}}) &= \left(\bigcup_{i=0}^{\infty} \text{dom}(\text{zip}_0((f_j)_{j \in [0, \dots, i]})), \bigcup_{i=0}^{\infty} \text{label}(\text{zip}_0((f_j)_{j \in [0, \dots, i]})), \right. \\ &\quad \left. \text{begin}(f_0) \right) \end{aligned}$$

Pokazujemy teraz, że istnienie nieskończonego zippera nad danym zbiorem figur $F \subseteq \Sigma^\circ$ jest nierozstrzygalne. W tym celu kodujemy za pomocą figur skierowanych skierowany układ płytek (directed tile system) (T, d) z pracy Adlemana i in. [2], w której w twierdzeniu 4.1 dowiedziono, że istnienie nieskończonego (T, d) -skierowanego zippera jest nierozstrzygalne.

Niech $A = \{1, \dots, n\}$ będzie skończonym, niepustym alfabetem. Płytką Wang'a to czwórka $(t_N, t_E, t_S, t_W) \in A^4$, interpretowana geometrycznie jako kwadrat jednostkowy, którego krawędzie „pokolorowane” są elementami t_N, t_E, t_S i t_W (krawędź północna, wschodnia, południowa

i zachodnia). *Skierowany układ płytek* to para (T, d) , gdzie T jest zbiorem płytek Wanga, a $d : T \rightarrow \{N, E, S, W\}$ jest funkcją, przypisującą płytkom kierunki.

Nieskończony (T, d) -skierowany zipper to, nieformalnie, ciąg płytek $(t_i)_{i \in \mathbb{Z}}$ ($t_i \in T$), odwzorowanych w \mathbb{Z}^2 , taki że $d(t_i)$ określa pozycję płytki t_{i+1} , a płytki zajmujące sąsiednie pozycje mają taki sam kolor na wspólnej krawędzi. Sąsiednie pozycje określone są jako $[(x, y)$ i $(x+1, y)]$ lub $[(x, y)$ i $(x, y+1)]$, dla $x, y \in \mathbb{Z}$.

Twierdzenie 5 *Niech F będzie skończonym zbiorem figur. Istnienie nieskończonego zippera nad F jest nierozstrzygalne.*

Powyższy rezultat można łatwo rozszerzyć na zippery semi-nieskończone. Zauważmy najpierw następujące równoważności:

Twierdzenie 6 *Dla danego $F \subseteq \Sigma^\circ$ następujące warunki są równoważne:*

1. *Istnieje nieskończony zipper nad F .*
2. *Istnieje semi-nieskończony zipper nad F .*
3. *Dla dowolnego $n \in \mathbb{N}$ istnieje skończony zipper nad F o długości n .*

Wniosek 1 *Niech F będzie skończonym zbiorem figur. Istnienie semi-nieskończonego zippera nad F jest nierozstrzygalne.*

Pokrywanie płaszczyzny figurami

Analizujemy tu możliwość pokrywania płaszczyzny figurami skierowanymi, pokazując jego nierozstrzygalność. Dowód stanowi redukcja z problemu pokrywania płytkami Wanga. Dla danego zbioru płytek Wanga T konstruujemy zbiór figur F_T taki, że T pokrywa płaszczyznę wtedy i tylko wtedy, gdy F_T pokrywa płaszczyznę.

Mówimy, że skończony zbiór płytek Wanga T *pokrywa płaszczyznę*, jeżeli istnieje odwzorowanie $\tau : \mathbb{Z}^2 \rightarrow T$ takie, że płytki na sąsiednich pozycjach mają ten sam kolor na wspólnej krawędzi. Następujące znane twierdzenie pochodzi od Bergera [9].

Twierdzenie 7 *Niech T będzie skończonym zbiorem płytek Wanga. Problem, czy T pokrywa płaszczyznę, jest nierozstrzygalny.*

Niech teraz $F \subseteq \Sigma^\circ$ będzie skończonym zbiorem figur. Mówimy, że F *pokrywa płaszczyznę*, jeżeli istnieje odwzorowanie $\tau : \mathbb{N} \rightarrow F$ takie, że $(\tau(i))_{i \in \mathbb{N}}$ jest semi-nieskończonym zipperem i dziedziną obrazu $\text{zip}((\tau(i))_{i \in \mathbb{N}})$ jest \mathbb{Z}^2 . Zauważmy, że problem pokrywania płaszczyzny przez F jest zawężeniem problemu z poprzedniego rozdziału i jego nierozstrzygalność nie jest prostą konsekwencją tamtej nierozstrzygalności.

Twierdzenie 8 *Niech F będzie skończonym zbiorem figur. Problem, czy F pokrywa płaszczyznę, jest nierozstrzygalny.*

Dowód wykorzystuje redukcję z problemu pokrywania płytkami Wanga. Zasadniczą ideą jest zamiana płytek na figury w taki sposób, by ułożenie figur obok siebie było możliwe tylko wtedy, gdy pierwotne płytki mają zgodne kolory na wspólnych krawędziach i mogą być umieszczone na spirali wypełniającej płaszczyznę. Konstrukcja ta przypomina *snaked tile system* Chena i Goela [15].

Zauważmy, że podobną redukcję można skonstruować dla zippera nieskończonego, gdy pokrywanie zdefiniowane jest w kategoriach odwzorowania $\tau : \mathbb{Z} \rightarrow F$. Konieczne jest wówczas użycie większej liczby figur, by mogły utworzyć podwójną spiralę z dwoma równoległymi włóknami.

4.3.5 Deszyfrowalność multisetowa, zbiorowa i liczbowa

Klasyczne pojęcie kodu wymaga, by zakodowana wiadomość została jednoznacznie odkodowana, tzn. należy odtworzyć pierwotny ciąg słów kodowych. Jednakże w niektórych sytuacjach wystarczające może być odtworzenie jedynie multisetu, zbioru lub zaledwie liczby słów kodowych. Prowadzi to do trzech rodzajów deszyfrowalności, znanych jako, odpowiednio, deszyfrowalność *multisetowa* (multiset decipherability, MSD), *zbiorowa* (set decipherability, SD) i *liczbowa* (numeric decipherability, ND). Oryginalna pełna deszyfrowalność zwana jest *deszyfrowalnością jednoznaczną* (unique decipherability, UD).

Deszyfrowalność multisetową wprowadził Lempel w pracy [32], natomiast deszyfrowalność liczbowa pochodzi od Heada i Webera [23]. Ci sami autorzy w [24] rozwijają „grafy dominowe”, dające użyteczną technikę sprawdzania deszyfrowalności. Guzman [20] zdefiniował deszyfrowalność zbiorową i przedstawił jednolite podejście do różnych typów deszyfrowalności poprzez rozmaitości monoidów. Prace Restivo [38] oraz Blanchet-Sadri i Morgana [12] rozstrzygają hipotezy Lempela dla pewnych kodów MSD i SD. Blanchet-Sadri [11] charakteryzuje deszyfrowalność kodów złożonych z trzech słów, zaś Burderi i Restivo wiążą deszyfrowalność z nierównością Krafta [14] i z podziałami kodowymi [13]. Salomaa i in. [39], mimo że nie zajmuje się wprost deszyfrowalnością, bada rozkłady języków za pomocą kodów ND (zwanym tu *length codes*).

Jak dowiedziono w [H1], sprawdzenie, czy dany skończony zbiór figur skierowanych jest kodem UD, jest rozstrzygalne. Jest to wciąż prawda w nieco ogólniejszym przypadku kodów ze słabą równością, por. [H2]. Z drugiej strony, figury skierowane z konkatenacją bez funkcji łączącej (określoną tylko wówczas, gdy figury nie nakładają się) mają ponownie nierozstrzygalne sprawdzanie deszyfrowalności UD [30, 31].

W pracy [H4] rozszerzamy wcześniejsze rezultaty, rozważając nie tylko kody UD, lecz także MSD, SD i ND dla figur skierowanych. Pokazujemy rozstrzygalność lub nierozstrzygalność dla poszczególnych kombinacji niezależnych kryteriów: typ konkatenacji (z funkcją łączącą lub bez niej), rodzaj deszyfrowalności (UD, MSD, SD, ND) oraz geometria kodu (kilka klas wyznaczonych przez wzajemne położenie punktów początkowych i końcowych figur). Dwie kombinacje pozostają jednak wciąż otwarte.

Kody

Definiujemy łącznie osiem rodzajów kodów na figurach skierowanych, będących rezultatem użycia czterech typów deszyfrowalności i dwóch typów konkatenacji. Stwierdzenia formułowane dla obydwóch typów konkatenacji \circ i \circ_m zapisujemy za pomocą symbolu \bullet , zatem „ $x \bullet y$ ” należy czytać jako „ $x \circ_m y$ (odpowiednio $x \circ y$)”. Podobnie „ $x \in X_\bullet$ ” należy czytać jako „ $x \in X_m^\diamond$ (odpowiednio $x \in X^\diamond$)”. By rozróżnić kody związane z konkatenacją z funkcją łączącą i bez niej, nazywamy je odpowiednio m -kodami i η -kodami. Ogólne odniesienia do kodów należy rozumieć jako „ m -kody (odpowiednio η -kody)”.

Definicja 10 (m -kod UD) Niech $X \subseteq \Sigma^\circ$. X jest m -kodem jednoznacznie deszyfrowalnym, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$ implikuje równość ciągów (x_1, \dots, x_k) i (y_1, \dots, y_l) , tzn. $k = l$ oraz $x_i = y_i$ dla $i \in \{1, \dots, k\}$.

Definicja 11 (η -kod UD) Niech $X \subseteq \Sigma^\circ$. X jest η -kodem jednoznacznie deszyfrowalnym, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \circ \dots \circ x_k = y_1 \circ \dots \circ y_l$ implikuje równość ciągów (x_1, \dots, x_k) i (y_1, \dots, y_l) .

W pozostałych definicjach stosujemy oczywiste skróty.

Definicja 12 (m -kody i η -kody MSD) Niech $X \subseteq \Sigma^\circ$. X jest kodem deszyfrowalnym multisetowo, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implikuje równość multisetów $\{\{x_1, \dots, x_k\}\}$ i $\{\{y_1, \dots, y_l\}\}$.

Definicja 13 (m -kody i η -kody SD) Niech $X \subseteq \Sigma^\circ$. X jest kodem deszyfrowalnym zbiorowo, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implikuje równość zbiorów $\{x_1, \dots, x_k\}$ i $\{y_1, \dots, y_l\}$.

Definicja 14 (m -kody i η -kody ND) Niech $X \subseteq \Sigma^\circ$. X jest kodem deszyfrowalnym liczbowo, jeżeli dla dowolnych $x_1, \dots, x_k, y_1, \dots, y_l \in X$ równość $x_1 \bullet \dots \bullet x_k = y_1 \bullet \dots \bullet y_l$ implikuje $k = l$.

Własność 5 Jeżeli X jest m -kodem UD (odpowiednio MSD, SD, ND), to X jest η -kodem UD (odpowiednio MSD, SD, ND).

Zauważmy, że implikacja odwrotna nie jest prawdziwa. Zbiór może nie spełniać warunku w definicji m -kodu UD, mając $x_1 \circ_m \dots \circ_m x_k = y_1 \circ_m \dots \circ_m y_l$, lecz wciąż być η -kodem UD po prostu dlatego, że niektóre konkatenacje spośród $x_1 \circ \dots \circ x_k$ i $y_1 \circ \dots \circ y_l$ są nieokreślone.

Przykład 11 Weźmy $X = \{x = \textcircled{0}\}$. X nie jest m -kodem UD, gdyż $x \circ_m x = x$. Jest on jednak trywialnym η -kodem UD, ponieważ $x \circ x$ nie jest określona.

Własność 6 Każdy η -kod UD jest η -kodem MSD; każdy η -kod MSD jest η -kodem SD i η -kodem ND. Każdy m -kod UD jest m -kodem MSD; każdy m -kod MSD jest m -kodem SD i m -kodem ND.

Podsumowujemy teraz wszystkie nietrywialne rezultaty dotyczące sprawdzania deszyfrowalności. Naszym celem jest udowodnienie rozstrzygalności lub nierozstrzygalności dla poszczególnych kombinacji niezależnych kryteriów: typ konkatenacji (z funkcją łączącą lub bez niej), rodzaj deszyfrowalności (UD, MSD, SD, ND) oraz geometria kodu (jednostronne, dwustronne, dwustronne z równoległymi wektorami przesunięcia). Dwie kombinacje pozostają jednak wciąż otwarte. Zauważmy, że we wszystkich nietrywialnych przypadkach rozstrzygalnych istnieją algorytmy sprawdzające rozważaną deszyfrowalność. Algorytmy te znajdują podwójny rozkład figury, gdy odpowiedź jest przecząca.

Pozytywne wyniki dotyczące rozstrzygalności

Przypomnijmy przytoczone już wcześniej rezultaty:

Twierdzenie 9 (zob. [H1], rozdział 4) Niech X będzie jednostronnym zbiorem nad Σ . Problem, czy X jest m -kodem UD, jest rozstrzygalny.

Twierdzenie 10 (zob. [31], rozdział 3) Niech X będzie jednostronnym zbiorem nad Σ . Problem, czy X jest η -kodem UD, jest rozstrzygalny.

Uogólniając twierdzenia 9 i 10, otrzymujemy podobne wyniki dla jednostronnych kodów MSD, SD i ND.

Twierdzenie 11 Niech X będzie jednostronnym zbiorem nad Σ . Problem, czy X jest $\{m$ -kodem lub η -kodem} $\{UD, MSD, SD$ lub $ND\}$, jest rozstrzygalny.

W powiązaniu z twierdzeniem 1, dowodzi to rozstrzygalności dla wszystkich m -kodów UD, MSD i ND. Otwarty pozostaje jednak przypadek dwustronnych m -kodów SD.

Dwustronne η -kody z równoległymi wektorami przesunięcia stanowią ciekawy przypadek szczególny.

Twierdzenie 12 (zob. [30], rozdział 4) Niech $X = \{x_1, x_2, \dots, x_n\}$ będzie dwustronnym zbiorem nad Σ z równoległymi wektorami przesunięcia, tzn. istnieje wektor $\tau \in \mathbb{Z}^2$ taki, że $\text{tran}(x_i) = \alpha_i \tau$ dla pewnych $\alpha_i \in \mathbb{Z}$ ($i = 1, 2, \dots, n$), przy czym α_i nie są wszystkie dodatnie ani wszystkie ujemne. Problem, czy X jest η -kodem UD, jest rozstrzygalny.

Tu ponownie możliwe jest uogólnienie na dwustronne η -kody MSD, SD i ND z równoległymi wektorami przesunięcia.

Twierdzenie 13 Niech X będzie dwustronnym zbiorem nad Σ z równoległymi wektorami przesunięcia. Problem, czy X jest η -kodem UD, MSD, SD lub ND, jest rozstrzygalny.

Negatywne wyniki dotyczące rozstrzygalności

Twierdzenie 14 (zob. [31], rozdział 2) Niech X będzie dwustronnym zbiorem nad Σ . Problem, czy X jest η -kodem UD, jest nierozstrzygalny.

Ten rezultat rozszerzamy na inne typy deszyfrowalności:

Twierdzenie 15 Niech X będzie dwustronnym zbiorem nad Σ . Problem, czy X jest η -kodem UD, MSD, SD lub ND, jest nierozstrzygalny.

Podsumowanie wyników dotyczących rozstrzygalności

Poniższa tabela przedstawia stan naszej wiedzy o rozstrzygalności deszyfrowalności. Przypadki rozstrzygalne i nierozstrzygalne oznaczone są odpowiednio znakami + i -. Kombinacje, które są wciąż otwarte, oznaczone są znakiem zapytania.

		UD	MSD	ND	SD
1	Jednostronne η -kody	+	+	+	+
2	Jednostronne m -kody	+	+	+	+
3	Dwustronne η -kody	-	-	-	-
4	Dwustronne m -kody	+	+	+	?
5	Dwustronne η -kody z równoległymi wektorami	+	+	+	+
6	Dwustronne m -kody z równoległymi wektorami	+	+	+	?

Zauważmy, że przypadki pozytywne w wierszach 4 i 6 są trywialne, gdyż nie istnieją jednostronne m -kody UD, MSD lub ND. Dla pozostałych kombinacji rozstrzygalnych odpowiednie dowody prowadzą do efektywnych algorytmów sprawdzania. Z drugiej strony, przypadek dwustronnych m -kodów SD jest nietrywialny; istnieją zarówno m -kody SD, jak i zbiory nie będące kodami tego rodzaju. Żadna z dotychczasowych technik dowodu nie pozwala jednak rozstrzygnąć tego przypadku.

4.3.6 Grafy dominowe a deszyfrowalność

W pracach [H5] i [H7] definiujemy wariant grafów dominowych, który pozwala nam rozstrzygnąć niektóre typy deszyfrowalności (określone wcześniej w [H4]) poprzez poszukiwanie pewnych ścieżek w grafie. Pomysł ten bazuje na idei Head'a i Webera [23, 24]. Główny wynik to zatem charakteryzacja deszyfrowalności za pomocą własności grafowych.

Konfiguracje zredukowane, których będziemy tu używać, a w konsekwencji graf dominowy, są zależne od typu konkatenacji. Ustalmy zatem ów typ — z funkcją łączącą m lub bez funkcji łączącej. Zakładamy ponadto, że wszystkie zbiory są jednostronne, w przeciwnym bowiem razie

konfiguracje zredukowane nie są określone. Definicje *konfiguracji* oraz *konfiguracji zredukowanej* pochodzą z dowodów twierdzeń 2 i 11.

Niech $rc(C)$ oznacza konfigurację zredukowaną związaną z konfiguracją C . Dla danej figury $z \in X$ definiujemy *rozszerzenie konfiguracji zredukowanej* $rc((x_1, \dots, x_k), (y_1, \dots, y_l))$ o (z, ε) jako nową konfigurację zredukowaną $rc((x_1, \dots, x_k, z), (y_1, \dots, y_l))$. Rozszerzenie to jest dobrze określone, gdyż $rc((x_1, \dots, x_k), (y_1, \dots, y_l)) = rc((x'_1, \dots, x'_k), (y'_1, \dots, y'_l))$ pociąga $rc((x_1, \dots, x_k, z), (y_1, \dots, y_l)) = rc((x'_1, \dots, x'_k, z), (y'_1, \dots, y'_l))$. Analogicznie definiujemy rozszerzenie o (ε, z) . Zauważmy, że w przypadku konkatencji bez funkcji łączącej konkretne rozszerzenie może nie być określone.

Konfiguracja zredukowana, zdefiniowana w twierdzeniu 11, to para $((e_L, l_L), (e_R, l_R))$ z punktami końcowymi $e_L, e_R \in \mathbb{Z}^2$ i etykietowaniami l_L, l_R , które są odwzorowaniami częściowymi $\mathbb{Z}^2 \rightarrow \Sigma$. Nieformalnie, rozszerzenie $((e_L, l_L), (e_R, l_R))$ o (z, ε) to konfiguracja zredukowana $((e'_L, l'_L), (e_R, l_R))$, gdzie $e'_L = e_L + \text{tran}(z)$, a l'_L uzyskano przez „konkatencję” l_L z z i odpowiednie zawężenie dziedziny. Konfiguracja zredukowana zwana jest *ostateczną*, jeżeli jest postaci $((e, l), (e, l))$, tzn. jej lewy i prawy składnik są równe. Zauważmy, że $rc(C)$ jest ostateczna wtedy i tylko wtedy, gdy $L_\bullet(C) = R_\bullet(C)$. Określenie „kryteria RC” odnosi się do warunków wymienionych w dowodzie twierdzenia 11.

Niech $RC(X)$ będzie zbiorem wszystkich konfiguracji zredukowanych nad X , które spełniają kryteria RC, tzn. $RC(X) = \{rc((x_i), (y_j)) \mid (x_i), (y_j)\}$, gdzie (x_i) i (y_j) przebiegają wszystkie skończone, niepuste ciągi elementów X , spełniające kryteria RC. Na mocy twierdzenia 11, $RC(X)$ jest skończony dla każdego jednostronnego kodu X .

Definicja 15 (Graf dominowy) Niech X będzie jednostronnym zbiorem nad Σ . Graf dominowy zbioru X to graf skierowany (V, E) , gdzie $V = RC(X) \cup \{0\}$, $E = E_0 \cup E_1$ oraz

- E_0 zawiera wszystkie krawędzie $(0, v)$ takie, że $v \in RC(X)$ i $v = rc((x), (y))$ dla pewnych $x, y \in X$, $x \neq y$,
- E_1 zawiera wszystkie krawędzie (v_1, v_2) takie, że $v_1, v_2 \in RC(X)$, v_1 nie jest ostateczna i v_2 jest rozszerzeniem v_1 o (z, ε) lub (ε, z) , dla pewnego $z \in X$.

Definiujemy ponadto funkcję dominową $d : E \rightarrow \wp((X \cup \{\varepsilon\}) \times (X \cup \{\varepsilon\}))$, przypisującą krawędziom etykiety:

$$\begin{aligned} d(0, v) &= \{(x, y) \in X \times X \mid v = rc((x), (y))\} \\ d(v_1, v_2) &= \{(x, y) \in (X \times \{\varepsilon\}) \cup (\{\varepsilon\} \times X) \mid v_2 \text{ jest rozszerzeniem } v_1 \text{ o } (x, y)\}. \end{aligned}$$

Zauważmy, że dla krawędzi (v_1, v_2) z $v_1 \neq 0$, $d(v_1, v_2)$ zawiera pary postaci albo (z, ε) , albo (ε, z) , ale nie jedne i drugie. Co więcej, jeżeli przykładowo (z, ε) i $(z', \varepsilon) \in d(v_1, v_2)$, to $\text{tran}(z) = \text{tran}(z') \neq (0, 0)$, gdyż X jest jednostronny, a dwie konfiguracje zredukowane v_1 i v_2 wyznaczają dokładnie jeden wektor przesunięcia, potrzebny do rozszerzenia v_1 do v_2 . Stąd $d(0, v)$ są jedynymi wartościami d , zawierającymi pary z dwiema niepustymi figurami.

Funkcję dominową można rozszerzyć na ścieżki w grafie dominowym G : mając daną ścieżkę $p = (e_1, e_2, \dots, e_n)$, gdzie e_i są krawędziami w G , definiujemy

$$d(p) = d(e_1) \bullet d(e_2) \bullet \dots \bullet d(e_n),$$

gdzie \bullet oznacza oczywiste rozszerzenie konkatencji figur na pary figur.

Dla danej ścieżki $p = (e_1, e_2, \dots, e_n)$ definiujemy również *realizację* p jako dowolny ciąg par figur $((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$ taki, że $(x_i, y_i) \in d(e_i)$. Zauważmy, że $x_i, y_i \in X \cup \{\varepsilon\}$.

Dla ścieżki p rozpoczynającej się w wierzchołku 0, $d(p)$ opisuje próbę skonstruowania dwóch różnych rozkładów pewnej figury. Jeżeli uda się doprowadzić p do wierzchołka ostatecznego,

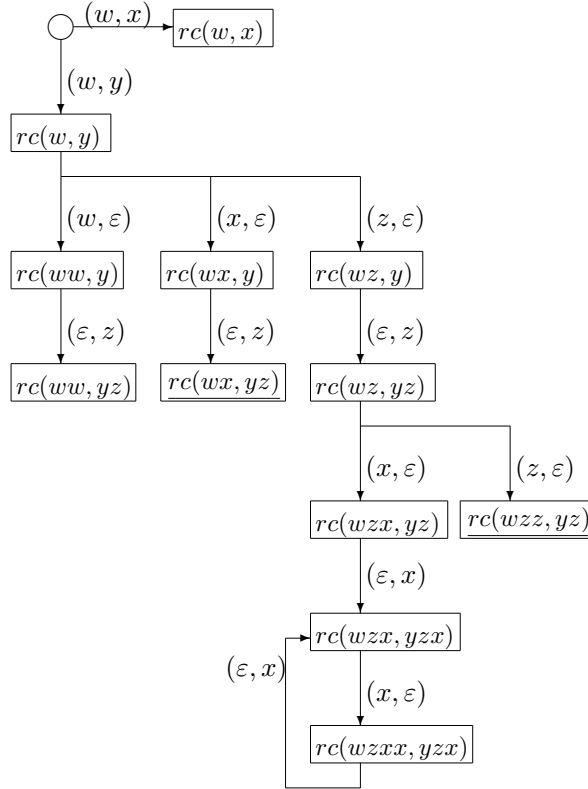
próba ta kończy się sukcesem (ścieżka p jest „pomyślna”); wiemy wówczas, że X nie jest kodem UD. Sprawdzenie innych typów deszyfrowalności wymaga sprawdzenia wszystkich pomyślnych ścieżek pod kątem szczególnych własności, podobnych do warunków w dowodzie twierdzenia 11. Mówi o tym następujące twierdzenie:

Twierdzenie 16 *Niech X będzie jednostronnym zbiorem nad Σ .*

1. X nie jest kodem UD wtedy i tylko wtedy, gdy jego graf dominowy zawiera ścieżkę od 0 do wierzchołka ostatecznego.
2. X nie jest kodem MSD wtedy i tylko wtedy, gdy jego graf dominowy zawiera ścieżkę p od 0 do wierzchołka ostatecznego taką, że istnieje realizacja $p, ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$, dla której $\{\{x_1, \dots, x_n\}\}$ i $\{\{y_1, \dots, y_n\}\}$ są różnymi multisetami.
3. X nie jest kodem SD wtedy i tylko wtedy, gdy jego graf dominowy zawiera ścieżkę p od 0 do wierzchołka ostatecznego taką, że istnieje realizacja $p, ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$, dla której $\{x_1, \dots, x_n\}$ i $\{y_1, \dots, y_n\}$ są różnymi zbiorami.
4. X nie jest kodem ND wtedy i tylko wtedy, gdy jego graf dominowy zawiera ścieżkę p od 0 do wierzchołka ostatecznego taką, że istnieje realizacja $p, ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$, dla której liczba niepustych x_i jest różna od liczby niepustych y_i .

Następujący przykład pokazuje graf dominowy zbioru, który nie jest kodem. Załóżmy, że $\Sigma = \{a\}$ i $m = \{(a, a) \mapsto a\}$. Etykiety krawędzi to wartości funkcji dominowej d ; zauważmy że dla wszystkich krawędzi jest $|d(e)| = 1$. Dla zwięzłości diagramu zapis konfiguracji zredukowanych pomija wewnętrzne nawiasy i przecinki. Wierzchołki ostateczne są podkreślone.

Przykład 12 *Rozważmy $X = \{w = \begin{array}{c} \diamond \\ \boxed{a} \end{array} a, x = \begin{array}{c} \boxed{a} \end{array} \boxed{a} \diamond, y = \begin{array}{c} \boxed{a} \\ \boxed{a} \end{array} \diamond, z = \begin{array}{c} \boxed{a} \\ \boxed{a} \end{array} \}$ i ustalmy $\tau_E = (1, 1)$, $\tau_W = (-\frac{1}{2}, -\frac{1}{2})$, $\tau_S = (0, -1)$, $\tau_N = (-\frac{1}{2}, 0)$. Pomijamy pary, które można uzyskać z innej pary przez zamianę elementów; nie tracimy jednak w ten sposób możliwości odkrycia żadnej z własności scharakteryzowanych w twierdzeniu 16. Zauważmy, że graf zawiera dwie pomyślne ścieżki, $0 \rightarrow rc(w, y) \rightarrow rc(wx, y) \rightarrow rc(wx, yz)$ oraz $0 \rightarrow rc(w, y) \rightarrow rc(wz, y) \rightarrow rc(wz, yz) \rightarrow rc(wzz, yz)$. Pierwsza obala deszyfrowalność UD, MSD i SD (ale nie ND); druga obala wszystkie cztery typy deszyfrowalności zbioru X .*



4.3.7 Własność defektu w \mathbb{Z}^2

Własności związane z deszyfrowalnością rzadko przenoszą się na kody w \mathbb{Z}^2 . W szczególności dotyczy to twierdzenia o defekcie, będącego jednym z fundamentalnych wyników kombinatoryki na słowach. Szczegóły można znaleźć w książkach Lothaire'a [33, 34] oraz w wyczerpującej pracy Harju i Karhumäkiego [22].

W swej klasycznej wersji twierdzenie o defekcie mówi, że jeżeli $X \subseteq \Sigma^*$ jest skończonym niekodem (tzn. istnieje słowo w X^* mające dwa różne rozkłady nad X), to istnieje kod $Y \subseteq \Sigma^*$ taki, że $X \subseteq Y^*$ oraz $|Y| < |X|$. Ścisłej, za Y można wziąć wolną otoczkę X , czyli najmniejszy wolny podmonoid Σ^* zawierający X . Podczas gdy własność defektu można łatwo rozszerzyć na drzewa, nie jest ona spełniona w wielu prostych przypadkach figur w \mathbb{Z}^2 .

W [H6] dokonujemy przeglądu trzech modeli figur w \mathbb{Z}^2 , w oparciu o definicje sformułowane w [H1], [P1] i [31]. Pokazujemy, że w żadnym z tych przypadków nie da się w rozsądny sposób uzyskać własności defektu.

Przytaczamy kontrprzykłady, które przeczą własności defektu dla figur skierowanych. Pokazują one, że własność ta zawodzi nawet dla bardzo prostych zbiorów, w tym singletonów, zbiorów jednostronnych i zbiorów „całkowicie nienakładających się” (takich, że konkatencja \circ istnieje dla dowolnego ciągu figur, gdyż nie ma kolizji dziedzin).

5 Pozostałe osiągnięcia naukowo-badawcze

- [P1] Włodzimierz Moczurad. “Brick codes: families, properties, relations”. W: *International Journal of Computer Mathematics* 74 (2000). IF=0.121, s. 133–150.
- [P2] Małgorzata Moczurad i Włodzimierz Moczurad. “Decidability of simple brick codes”. W: *Mathematics and Computer Science III (Algorithms, Trees, Combinatorics and Probabilities)*. Trends in Mathematics. Birkhäuser, 2004, s. 541–542.

- [P3] Małgorzata Moczurad i Włodzimierz Moczurad. “Some open problems in decidability of brick (labelled polyomino) codes”. W: *Computing and Combinatorics COCOON 2004*. T. 3106. Lecture Notes in Computer Science. IF=0.513. Springer, 2004, s. 72–81.
- [P4] Małgorzata Moczurad i Włodzimierz Moczurad. “Asymptotic density of brick and word codes”. W: *Ars Combinatoria* 83 (2007). IF=0.234, s. 169–177.
- [P5] Małgorzata Moczurad i Włodzimierz Moczurad. “How many figure sets are codes?” W: *Language and Automata Theory and Applications LATA 2008*. T. 5196. Lecture Notes in Computer Science. Springer, 2008, s. 385–396.
- [P6] Włodzimierz Moczurad. “Defect theorem in the plane”. W: *Theoretical Informatics and Applications RAIRO* 41.4 (2007), s. 403–409.

[P1] wprowadza wielowymiarowe słowa i kody. Są to w istocie etykietowane poliomina, które nazywamy *klockami* (bricks) lub *figurami*, por. [3, 8].

Zaczynamy od podstawowych definicji i własności związanych z deszyfrowalnością. Przedstawiamy między innymi kilka ważnych rodzin kodów figurowych. Badamy następnie, jakie rodziny kodów można określić za pomocą n -argumentowych relacji. Pokazujemy, że niektóre nietrywialne rodziny nie pozwalają na taką charakteryzację. Analizujemy również strukturę zbioru relacji, które na taką charakteryzację pozwalają. Wreszcie wprowadzamy pojęcie liczby chromatycznej, czyli najmniejszej liczby etykiet pozwalającej uczynić dany zbiór kodem, i badamy jej własności w zależności od rozmiaru i „ziarnistości” zbioru figur.

W [P2] i [P3] rozważamy rozstrzygalność deszyfrowalności zbioru figur. Problem, czy dany zbiór figur (bądź nawet poliomin) jest kodem, jest w ogólności nierozstrzygalny. Można tego dowieść przez redukcję z problemu pokrywania płytkami Wanga (zob. [8]). Problem ten jest otwarty dla zbiorów dwuelementowych. W omawianych pracach rozważamy zbiory złożone wyłącznie z figur kwadratowych. Pokazujemy, że w takim układzie deszyfrowalność małych zbiorów (dwa elementy) jest rozstrzygalna, natomiast 15 elementów wystarcza, by uczynić problem nierozstrzygalnym.

Zauważmy, że deszyfrowalność jest w trywialny sposób rozstrzygalna, jeżeli kwadraty używają tylko jednej etykiety, czyli gdy w istocie są poliominami. Wówczas kodami są jedynie singletony. Zauważmy też, że z wyjątkiem przypadku z jedną etykietą, rozstrzygalność deszyfrowalności dla kwadratów nie zależy od wielkości alfabetu, gdyż większy alfabet można „symulować” alfabetem binarnym za cenę rozmiaru kwadratów. Gdy rozważamy dowolne kształty, etykiety można symulować kształtami, co powoduje, że rozstrzygalność deszyfrowalności dla figur jest równoważna rozstrzygalności dla poliomin.

W końcowej części pracy badamy rozstrzygalność dla zbiorów z kluczem, czyli dla kwadratów mających w jednym z rogów „prefiks-klucz” ustalonego rozmiaru.

Prace [P4] i [P5] poświęcone są zliczaniu kodów figurowych. Zliczanie poliomin lub figur ustalonego rozmiaru jest trudne; przykładowo, nie jest znany dokładny wzór bądź funkcja tworząca dla ciągu (p_n) , opisującego liczbę poliomin rozmiaru n ; por. [40]. Z drugiej strony, problem ten jest trywialny dla słów. Zliczanie kodów poliominowych lub figurowych jest co najmniej tak trudne, jak zliczanie poliomin. Deszyfrowalność jest własnością „semantyczną”, która nie pozwala na prostą charakteryzację „syntaktyczną”. Stąd poważne narzędzia kombinatoryczne, jak np. funkcje tworzące, nie są tu pomocne.

Liczmy gęstość asymptotyczną w następującym sensie, por. [37, 42]: mając dany zbiór obiektów, wybieramy obiekty o rozmiarze n i liczymy stosunek tych, które mają żądaną własność (w tym przypadku deszyfrowalność), do wszystkich, przy n dążącym do nieskończoności. Miarę rozmiaru definiujemy oddzielnie dla różnych klas figur.

Głównym rezultatem pracy jest twierdzenie, że dla pewnych klas figur powyższa proporcja zmierza do 1, gdy rozmiar figur zmierza do nieskończoności. Wynik ten można interpretować probabilistycznie następująco: prawdopodobieństwo tego, że losowo wybrany zbiór figur jest kodem, zbliża się do 1, gdy figury są duże. Ubocznym efektem jest wskazanie klas figur o gęstości równej 1 wśród wszystkich kodów. Pokazujemy także nieoczywisty związek między rozmiarem alfabetu a postacią kodów złożonych z domin.

W pracy [P6] ustalamy status własności defektu dla kodów figurowych. Twierdzenie o defekcie jest jednym z fundamentalnych rezultatów kombinatoryki na słowach, który dla zbiorów słów określa pewnego rodzaju wymiar (por. Lothaire [33, 34]). Różne warianty oraz próby rozszerzenia na inne struktury, w tym drzewa i dwuwymiarowe figury, można znaleźć w pracach [22, 26–28, 35, 36].

Podczas gdy rozszerzenie na drzewa jest łatwe, własność nie przenosi się prosto na figury na płaszczyźnie. W pracy podajemy kontrprzykłady dla kwadratów, prostokątów i domin. Pokazujemy, że dwuelementowe zbiory takich figur własność defektu posiadają. Pozwala nam to wskazać graniczne wielkości zbiorów, przy których twierdzenie przestaje obowiązywać.

Udowadniamy następnie twierdzenie o defekcie dla trzech domin (prostokątów $n \times 1$ or $1 \times n$), rozwiązując tym samym problem postawiony w [22] i ustalając dokładnie graniczną wielkość zbioru, przy której własność zawodzi.

Bibliografia

- [1] Leonard Adleman. “Molecular computation of solutions to combinatorial problems”. W: *Science* 266.5187 (1994), s. 1021–1024.
- [2] Leonard Adleman i in. “The Undecidability of the Infinite Ribbon Problem: Implications for Computing by Self-Assembly”. W: *SIAM Journal on Computing* 38.6 (2009), s. 2356–2381.
- [3] Philippe Aigrain i Danièle Beauquier. “Polyomino tilings, cellular automata and codicity”. W: *Theoretical Computer Science* 147.1–2 (1995), s. 165–180.
- [4] Marcella Anselmo, Dora Giammarresi i Maria Madonia. “Strong Prefix Codes of Pictures”. W: *Algebraic Informatics*. T. 8080. Lecture Notes in Computer Science. Springer, 2013, s. 47–59.
- [5] Marcella Anselmo, Dora Giammarresi i Maria Madonia. “Two Dimensional Prefix Codes of Pictures”. W: *Developments in Language Theory*. T. 7907. Lecture Notes in Computer Science. Springer, 2013, s. 46–57.
- [6] Marcella Anselmo, Dora Giammarresi i Maria Madonia. “Picture codes with finite deciphering delay”. W: *Language and Automata Theory and Applications LATA 2014*. T. 8370. Lecture Notes in Computer Science. Springer, 2014, s. 88–100.
- [7] Marcella Anselmo, Dora Giammarresi i Maria Madonia. “Prefix picture codes: A decidable class of two-dimensional codes”. W: *International Journal of Foundations of Computer Science* 25.8 (2014), s. 1017–1031.
- [8] Danièle Beauquier i Maurice Nivat. “A codicity undecidable problem in the plane”. W: *Theoretical Computer Science* 303.2–3 (2003), s. 417–430.
- [9] Robert Berger. “The undecidability of the domino problem”. W: *Memoirs of the American Mathematical Society* 66 (1966), s. 1–72.
- [10] Jean Berstel i Dominique Perrin. *Theory of Codes*. Academic Press, 1985.

- [11] Francine Blanchet-Sadri. “On unique, multiset, set decipherability of three-word codes”. W: *IEEE Transactions on Information Theory* 47.5 (2001), s. 1745–1757.
- [12] Francine Blanchet-Sadri i C. Morgan. “Multiset and set decipherable codes”. W: *Computers and Mathematics with Applications* 41.10–11 (2001), s. 1257–1262.
- [13] Fabio Burderi i Antonio Restivo. “Coding Partitions”. W: *Discrete Mathematics and Theoretical Computer Science* 9.2 (2007), s. 227–240.
- [14] Fabio Burderi i Antonio Restivo. “Varieties of Codes and Kraft Inequality”. W: *Theory of Computing Systems* 40.4 (2007), s. 507–520.
- [15] Ho-Lin Chen i Ashish Goel. “Error Free Self-assembly Using Error Prone Tiles”. W: *10th International Workshop on DNA Computing, DNA10, Milan*. T. 3384. Lecture Notes in Computer Science. Springer, 2005, s. 62–75.
- [16] Gennaro Costagliola, Filomena Ferrucci i Carmine Gravino. “Adding symbolic information to picture models: definitions and properties”. W: *Theoretical Computer Science* 337 (2005), s. 51–104.
- [17] Gennaro Costagliola i in. “On regular drawn symbolic picture languages”. W: *Information and Computation* 187 (2003), s. 209–245.
- [18] Guilhem Gamard i Gwenaël Richomme. “Coverability in Two Dimensions”. W: *Language and Automata Theory and Applications*. T. 8977. Lecture Notes in Computer Science. Springer, 2015, s. 402–413.
- [19] Dora Giammarresi i Antonio Restivo. “Two-Dimensional Finite State Recognizability”. W: *Fundamenta Informaticae* 25.3 (1996), s. 399–422.
- [20] Fernando Guzmán. “Decipherability of codes”. W: *Journal of Pure and Applied Algebra* 141.1 (1999), s. 13–35.
- [21] Vesa Halava, Tero Harju i Tomi Kärki. “Relational codes of words”. W: *Theoretical Computer Science* 389.1–2 (2007), s. 237–249.
- [22] Tero Harju i Juhani Karhumäki. “Many aspects of defect theorems”. W: *Theoretical Computer Science* 324.1 (2004), s. 35–54.
- [23] Tom Head i Andreas Weber. “The Finest Homophonic Partition and Related Code Concepts”. W: *Mathematical Foundations of Computer Science MFCS 1994*. T. 841. Lecture Notes in Computer Science. Springer, 1994, s. 618–628.
- [24] Tom Head i Andreas Weber. “Deciding multiset decipherability”. W: *IEEE Transactions on Information Theory* 41.1 (1995), s. 291–297.
- [25] Mari Huova. “A note on defect theorems for 2-dimensional words and trees”. W: *Journal of Automata, Languages and Combinatorics* 14.3 (2009), s. 203–209.
- [26] Juhani Karhumäki i Sabrina Mantaci. “Defect Theorems for Trees”. W: *Fundamenta Informaticae* 38.1–2 (1999), s. 119–133.
- [27] Juhani Karhumäki i Ján Maňuch. “Multiple factorizations of words and defect effect”. W: *Theoretical Computer Science* 273 (2002), s. 81–97.
- [28] Juhani Karhumäki, Ján Maňuch i Wojciech Plandowski. “A defect theorem for bi-infinite words”. W: *Theoretical Computer Science* 292 (2003), s. 237–243.
- [29] Jarkko Kari i Michal Szabados. “An Algebraic Geometric Approach to Nivat’s Conjecture”. W: *ICALP 2015*. T. 9135. Springer, 2015, s. 273–285.
- [30] Michał Kolarz. “Directed Figure Codes: Decidability Frontier”. W: *COCOON 2010*. T. 6196. Lecture Notes in Computer Science. Springer, 2010, s. 530–539.

- [31] Michał Kolarz. “The code problem for directed figures”. W: *Theoretical Informatics and Applications RAIRO* 44.4 (2010), s. 489–506.
- [32] Abraham Lempel. “On multiset decipherable codes”. W: *IEEE Transactions on Information Theory* 32.5 (1986), s. 714–716.
- [33] M. Lothaire. *Combinatorics on Words*. Cambridge University Press, 1997.
- [34] M. Lothaire. *Algebraic Combinatorics on Words*. Cambridge University Press, 2002.
- [35] Sabrina Mantaci i Antonio Restivo. “Codes and equations on trees”. W: *Theoretical Computer Science* 255 (2001), s. 483–509.
- [36] Ján Maňuch. “Defect Effect of Bi-infinite Words in the Two-element Case”. W: *Discrete Mathematics and Theoretical Computer Science* 4.2 (2001), s. 273–290.
- [37] Małgorzata Moczurad, Jerzy Tyszkiewicz i Marek Zaionc. “Statistical properties of simple types”. W: *Mathematical Structures in Computer Science* 10 (2000), s. 575–594.
- [38] Antonio Restivo. “A note on multiset decipherable code”. W: *IEEE Transactions on Information Theory* 35.3 (1989), s. 662–663.
- [39] Arto Salomaa, Kai Salomaa i Sheng Yu. “Variants of codes and indecomposable languages”. W: *Information and Computation* 207.11 (2009), s. 1340–1349.
- [40] Herbert Wilf. *generatingfunctionology*. Academic Press, 1994.
- [41] Erik Winfree. “Algorithmic Self-Assembly of DNA”. Prac. dokt. California Institute of Technology, 1998.
- [42] Karen Yeats. “Asymptotic Density in Combined Number Systems”. W: *New York Journal of Mathematics* 8 (2002), s. 62–83.

Włodzisław Moczurad