

# Konstrukcje PDL w logice ze strażnikami

Jakub Michaliszyn

19 czerwca 2008

## Streszczenie

W tej pracy spróbujemy rozszerzyć logikę ze strażnikami  $GF^2$  o możliwość używania pewnych konstrukcji z logiki modalnej PDL. Najpierw pokażemy, że dodanie możliwości stosowania złożenia, nawet po ograniczeniu do relacji występujących jedynie w strażnikach, prowadzi do nierozstrzygalności problemu spełnialności. Następnie rozważymy stosowanie operacji przechodniego domknięcia relacji atomowych. Z wcześniejszych wyników wiadomo, że zezwolenie na używanie przechodniego domknięcia w dowolnym miejscu formuły prowadzi do nierozstrzygalności problemu spełnialności. Okazuje się jednak, że jeśli zezwalamy na stosowanie przechodniego domknięcia tylko dla relacji, które występują jedynie w strażnikach, to taka logika jest rozstrzygalna w czasie podwójnie wykładniczym, więc, na mocy wcześniejszych wyników,  $2EXPSPACE$ -zupelna.

# 1 Wprowadzenie

Logiki modalne zostały wprowadzone w celu poszerzenia rachunku zdań o możliwość wyrażania *możliwości* ( $\diamond$ ) i *konieczności* ( $\square$ ). Początkowo te logiki były wykorzystywane głównie przez filozofów. Wraz z rozwojem komputerów zaczęto szukać sposobu, aby wykorzystywać te logiki do weryfikacji programów komputerowych oraz sprzętu. Ze względu na stosunkowo małą siłę wyrazu klasyczne logiki modalne okazują się jednak często niewystarczające, więc zaproponowano wiele rozszerzeń tych logik.

Zdaniowa logika dynamiczna (PDL), wprowadzona w [7], jest jednym z najbardziej znanych rozszerzeń logik modalnych. Szczególnie dobrze nadaje się do opisywania własności programów zapisanych w imperatywnych językach programowania, gdyż pozwala elegancko wyrażać pętle oraz składanie instrukcji. Rok po wprowadzeniu logiki PDL w pracy [16] udowodniono, że problem spełnialności dla tej logiki jest EXPTIME-zupełny. Dla porównania, problem spełnialności dla zwykłych logik modalnych jest PSPACE-zupełny.

Strzeżony fragment logiki pierwszego rzędu, zwany również logiką ze strażnikami (GF), został po raz pierwszy wprowadzony w pracy [1]. Główną motywacją przy definiowaniu tej logiki była chęć zanurzenia logiki modalnej w logice pierwszego rzędu w taki sposób, by nie utracić rozstrzygalności podstawowych problemów decyzyjnych. W odróżnieniu od zwykłej logiki pierwszego rzędu, w logice ze strażnikami wymagamy, by po każdym kwantyfikatorze pojawił się *strażnik*, to znaczy formuła, która ogranicza wszystkie zmienne wolne występujące w podformule, którą wiąże ten kwantyfikator. Pełną definicję GF można znaleźć w rozdziale 2.

W pracy, w której zdefiniowano GF ([1]), pokazano również dowód rozstrzygalności problemu spełnialności dla wariantu GF bez równości. W ogólnym przypadku problem spełnialności okazał się być 2EXPTIME-zupełny ([12]), jednak przy ograniczonej liczbie zmiennych w formule problem staje się jedynie EXPTIME-zupełny. Dla porównania, fragment logiki pierwszego rzędu z dwiema zmiennymi ( $\text{FO}^2$ ) jest NEXPTIME-zupełny ([9]).

Okazuje się, że zdania standardowej logiki modalnej można w prosty sposób przełożyć na zdania logiki ze strażnikami z dwiema zmiennymi, zastępując zmienne przez unarne predykaty oraz operatory modalne przez kwantyfikatory ([3], [4]). Taki przekład zwiększa jednak złożoność obliczeniową problemu spełnialności z PSPACE do EXPTIME, jednak w  $\text{GF}^2$  można wyrazić więcej własności.

*Przykład.* Formułę  $\square\diamond(p \wedge \square q)$  można przełożyć na formułę  $\forall y(xRy \Rightarrow \exists x(yRx \wedge P(x) \wedge \forall y(xRy \Rightarrow Q(y))))$  należąca do GF. Warto zwrócić uwagę, że do przekładu dowolnej formuły wystarczają dwie zmienne. Formuła  $\forall x(x = x \Rightarrow P(x))$  nie ma odpowiednika wśród formuł logiki modalnej.

W roku 1997 w pracy [11] pokazano, że logika  $\text{FO}^2$  poszerzona o możliwość używania relacji przechodnich jest nierozstrzygalna. Dwa lata później w pracy [8] udowodniono również nierozstrzygalność dla  $\text{GF}^2$  z relacjami przechodnimi bez równości, jednocześnie pokazując, że jeśli zabronimy używania relacji o arności większej niż 1 poza strażnikami, to logika staje się rozstrzygalna. Aby uzyskać nieco lepsze wyniki, zaczęto definiować nieco bogatsze logiki. Nas będą interesować przede wszystkim logika z przechodnimi strażnikami ( $[\text{GF} + \text{TG}]$ ) oraz logika z przechodnio domykanyymi strażnikami ( $[\text{GF}^{++}]$ ). Wspomniemy też o logice z punktami stałymi ( $\mu\text{GF}$ ).

Logika  $\mu\text{GF}$  powstaje z GF przez dodanie możliwości definiowania punktów stałych. W roku 1999 w pracy [10] pokazano, że problem spełnialności dla strzeżonej logiki z punktami stałymi jest 2EXPTIME-zupełny. Zauważmy, że to implikuje, że  $\text{GF}^2$  z przechodnim domknięciem jest rozstrzygalna, gdy zabronimy używania relacji o arności większej niż 1 poza strażnikami.

Aby się o tym przekonać zauważmy najpierw, że dowolną formułę tego języka można zapisać w postaci normalnej (patrz: rozdział 2), wobec czego wystarczy pokazać, jak przełożyć formuły  $\forall y.xT^+y \Rightarrow \varphi(x, y)$  oraz  $\exists y.xT^+y \wedge \varphi(x, y)$ , gdzie w  $\varphi$  są tylko unarne symbole relacyjne, na formuły z języka  $\mu\text{GF}$ . Dla ustalonej formuły  $\varphi$  bez kwantyfikatorów, niech  $U$  będzie zbiorem symboli unarnych występujących w tej formule oraz niech  $t \subseteq U$ . Łatwo można zdefiniować formułę bez kwantyfikatorów  $ma - typ_t$  taką, że  $ma - typ_t(v)$  zachodzi wtedy i tylko wtedy, gdy  $v$  spełnia wszystkie predykaty z  $t$  i nie spełnia żadnego predykatu z  $U \setminus t$ . Można również zdefiniować formułę bez kwantyfikatorów  $\psi_t^\varphi$  taką, że dla dowolnego  $v$  jeśli  $ma - typ_t(v)$ , to dla każdego  $w$  zachodzi  $\psi_t^\varphi(w) \iff \varphi(v, w)$ , gdyż mając ustalone wartości wszystkich relacji dla  $v$  można sprawdzić, jakiego typu punkty spowodują, że formuła będzie spełniona. Mając te formuły możemy zdefiniować zbiór elementów, z których jest osiągalny ścieżką po  $T$  punkt spełniający  $\psi_t^\varphi$ :  $W_t^\varphi = [LFP Wx.\psi_t^\varphi(x) \vee (\exists y.xTy \wedge W(y))]$  oraz zbiór elementów, z których każdy element osiągalny relacją  $T$  spełnia  $\psi_t^\varphi$ :  $V_t^\varphi = [GFP Wx.\psi_t^\varphi(x) \wedge (\forall y.xTy \wedge W(y))]$ . Z tymi definicjami łatwo już zdefiniować przekład interesujących nas formuł:

- $\exists y.xT^+y \wedge \varphi(x, y)$  przekładamy na  $\exists y.xTy \wedge \bigwedge_{t \subseteq U} (ma - typ_t(x) \Rightarrow W_t^\varphi(y))$
- $\forall y.xT^+y \Rightarrow \varphi(x, y)$  przekładamy na  $\forall y.xTy \Rightarrow \bigwedge_{t \subseteq U} (ma - typ_t(x) \Rightarrow V_t^\varphi(y))$

Zauważmy, że ten przekład jest wykładniczy, co wobec EXPTIME-zupełności problemu spełnialności dla  $\mu\text{GF}$  z dwiema zmiennymi daje podwójnie wykładniczy algorytm rozstrzygający problem spełnialności dla tej logiki.

Logika  $[\text{GF} + \text{TG}]$  powstaje z  $\text{GF}$  przez dodanie możliwości definiowania pewnych relacji jako przechodnich, przy czym zabrania się występowania takich relacji poza strażnikami. W roku 2001 praca [18] pokazała, że  $[\text{GF} + \text{TG}]$  jest rozstrzygalna w czasie  $2\text{EXPTIME}$ , a zatem, wobec wcześniejszych wyników,  $2\text{EXPTIME}$ -zupełna.

W tej pracy uzupełnimy wyniki z [18] i [10] dowodząc, że  $\text{GF}^2$  z wyróżnionym zbiorem relacji binarnych, które mogą się pojawiać tylko w strażnikach oraz które można przechodnio domykać, ma  $2\text{EXPTIME}$ -zupełny problem spełnialności.

Aby zyskać motywację, porównajmy siłę wyrazu  $[\text{GF} + \text{TG}]$  z  $[\text{GF}^{++}]$ . Oczywiście, każda własność definiowana formułą z  $[\text{GF} + \text{TG}]$  jest wyrażalna w  $\text{GF}$  z przechodnim domknięciem, gdyż każdą relację przechodnią możemy zastąpić przez przechodnie domknięcie pewnej nowej relacji. Interesujące jest zatem pytanie, czy  $[\text{GF}^{++}]$  pozwala zdefiniować jakieś własności, które nie są wyrażalne w  $[\text{GF} + \text{TG}]$ . Zauważmy, że własność „dana relacja jest przechodnia” jest wyrażalna w logice pierwszego rzędu. Oznacza to, że wszystkie formuły wyrażalne przez logikę  $[\text{GF} + \text{TG}]$  można przełożyć na logikę FO. Znanym faktem ([2]) jest, że w FO nie można wyrazić własności „z pewnego punktu spełniającego  $S$  jest osiągalny ścieżką po  $R$  pewien punkt spełniający  $T$ ”. Własność tę można jednak wyrazić w  $[\text{GF}^{++}]$  zdaniem  $\exists x.S(x) \wedge \exists y.xR^+y \wedge T(y)$ . Zatem  $[\text{GF}^{++}]$  rzeczywiście zwiększa liczbę własności, które można wyrazić. Zauważmy też, że przechodnie domknięcie lepiej nadaje się do opisu programów niż relacje przechodnie, gdyż dzięki niemu można opisać relację przejścia w jednym kroku, a następnie sprawdzać pewne własności, które mogą zajść po dowolnie wielu krokach.

Niestety, formalny dowód faktu, że istnieją formuły w  $[\text{GF}^{++}]$ , których nie daje się wyrazić w  $\mu\text{GF}$ , nie jest obecnie znany. Wydaje się jednak, że w logice z punktami stałymi nie można powiedzieć, że bieżący element  $v$  jest w ustalony sposób połączony relacjami binarnymi z elementem osiągalnym z  $v$  pewną ścieżką po  $R$ , więc formuły  $\forall xy.xR^+y \Rightarrow (P(y) \Leftrightarrow xSy)$  nie da się wyrazić w  $\mu\text{GF}$ . Warto również zauważyć, że gdyby  $[\text{GF}^{++}]$  dała się wyrazić w  $\mu\text{GF}$ , to tym bardziej  $[\text{GF} + \text{TG}]$  by się dała, co by zmniejszyło wagę niektórych znanych wyników (np. [18]).

Ten dokument składa się z dwóch istotnych części: dowodu nierozstrzygalności  $\text{GF}^2$  ze złożeniem oraz dowodu rozstrzygalności  $\text{GF}^2$  z przechodnim domknięciem. Rozdział 2 wprowadza de-

finicje niezbędne do zrozumienia dalszych rozdziałów. Zakłada się, że czytelnik zna podstawowe pojęcia związane z logiką, więc definicje elementarnych pojęć pominięto. Rozdział 3 dowodzi, że logika ze strażnikami, w której dodatkowo zezwolimy na składanie relacji samej z sobą pod warunkiem, że ta relacja nie występuje poza strażnikami, ma nierozstrzygalny problem spełnialności. To implikuje, że nie da się łatwo przełożyć całego PDL na wariant GF. W kolejnych rozdziałach spróbujemy zbadać wariant, w którym do  $GF^2$  dodajemy z PDL jedynie operację przechodniego domknięcia. W rozdziale 4 zdefiniujemy pewną własność modelu i zobaczymy, że każda spełnialna formuła logiki  $[GF+^*]$  ma model o tej własności. Następnie w rozdziale 5 zobaczymy algorytm, który sprawdza, czy dana formuła  $[GF+^*]$  ma model o tej własności, a zatem rozstrzyga problem spełnialności dla tej logiki.

## 2 Preliminaria

### 2.1 Rodzaje logik

Przez FO będziemy oznaczać całą logikę pierwszego rzędu bez symboli funkcyjnych, stałych oraz relacji 0-arnych. Przez GF będziemy oznaczać logikę ze strażnikami, która jest fragmentem FO. Formuły dopuszczalne w logice GF to najmniejszy zbiór spełniający poniższe warunki:

- każda formuła atomowa należy do języka GF
- jeśli  $\psi$  oraz  $\varphi$  należą do języka GF, to  $\psi \vee \varphi$ ,  $\psi \wedge \varphi$ ,  $\neg\psi$  oraz  $\psi \Rightarrow \varphi$  należą do języka GF
- jeśli  $\psi$  należy do języka GF,  $\vec{x}, \vec{y}$  są ciągami zmiennych oraz  $\alpha(\vec{x}, \vec{y})$  jest formułą atomową zawierającą wszystkie zmienne z  $\vec{x}$  i  $\vec{y}$ , to  $\forall\vec{x}.\alpha(\vec{x}, \vec{y}) \Rightarrow \psi(\vec{x}, \vec{y})$  oraz to  $\exists\vec{y}.\alpha(\vec{x}, \vec{y}) \wedge \psi(\vec{x}, \vec{y})$  należą do języka GF

Formuła  $\alpha(\vec{x}, \vec{y})$  jest często nazywana strażnikiem. Warto zauważyć, że szczególnym przypadkiem strażnika jest formuła  $x = x$ .

Przez  $FO^k$  ( $GF^k$ ) będziemy oznaczać fragment logiki FO (GF), w którym dopuszcza się użycie jedynie  $k$  zmiennych.

Logikę z przechodnio domykanymi strażnikami, oznaczaną przez  $[GF+^+]$  definiujemy analogicznie do GF z tą różnicą, że wyróżniamy pewien podzbiór binarnych symboli relacyjnych  $T$  i zabraniamy używania symboli z  $T$  poza strażnikami. W strażnikach pozwalamy na użycie zarówno tych symboli, jak i ich przechodniego domknięcia, oznaczanego plusem. Podzbiór  $[GF+^+]$  dopuszczający użycie jedynie dwóch zmiennych będziemy oznaczać przez  $[GF^2+^+]$ .

Logikę ze złożeniem definiujemy analogicznie do  $[GF+^+]$ , jednak w strażnikach pozwalamy tym razem na używanie złożenia relacji z  $T$ , oznaczanego, podobnie jak w PDL, średnikiem.

*Przykład.* Formuła  $\forall xy.xRy$  jest formułą  $FO^2$ , ale nie należy do GF. Formuła  $\forall x.P(x) \Rightarrow \forall y.xR^+y \Rightarrow \neg S(y)$ , mówiąca, że z żadnego elementu spełniającego  $P$  nie można dojść relacją  $R$  do elementu spełniającego  $S$ , należy do  $[GF^2+^+]$ , ale nie należy do GF ani FO. Formuła  $\forall xy.xSy \Rightarrow xR^+y$  nie jest poprawna w żadnym z rozważanych języków. Formuła  $\forall xy.xR; R; Ry \Rightarrow x = y$  należy do logiki ze złożeniem i nie należy do żadnej z pozostałych logik.

### 2.2 Pojęcia związane z logikami

Dla danej sygnatury  $\Sigma$  możemy zdefiniować typy punktów oraz połączeń pojawiających się w modelach nad  $\Sigma$ . Pojęcie  $k$ -typu atomowego jest znane w logice jako maksymalny niesprzeczny

zbiór formuł atomowych z  $k$  zmiennymi. Nas będą głównie interesowały relacje unarne i binarne, więc zdefiniujemy sobie te pojęcia w nieco wygodniejszy sposób. Atomowym 1-typem (zwanym dalej 1-typem) elementu  $v$  z modelu  $\mathfrak{M}$  będziemy nazywali zbiór  $t$  takich symboli relacyjnych  $R$ , że  $\mathfrak{M} \models R(x, \dots, x)$ . Atomowym 2-typem (zwanym dalej 2-typem) pary  $v, w$  elementów modelu  $\mathfrak{M}$  nazywamy zbiór  $t$  składający się z par  $\langle 1, t_v \rangle$ , par  $\langle 2, t_w \rangle$ , gdzie  $t_v$  i  $t_w$  to odpowiednio 1-typy  $v$  i  $w$ , oraz par  $\langle 3, R \rangle$  dla symboli relacji binarnych  $R$  takich, że  $\mathfrak{M} \models vRw$ , oraz  $\langle 4, R \rangle$  dla symboli relacji binarnych  $R$  takich, że  $\mathfrak{M} \models wRr$ . Przycięciem 2-typu  $t$  do relacji ze zbioru  $B$  będziemy nazywali zbiór  $t|_B = t \cap \{\langle a, b \rangle \mid a < 3 \vee b \in B\}$ .

Często będą nas interesowały zarówno następniki, jak i poprzedniki pewnego elementu względem jakiejś relacji. Dla uproszczenia notacji przez  $T^{-1}$  dla danego zbioru  $T$  relacji binarnych lub symboli relacji binarnych będziemy oznaczać zbiór  $\{R^{-1} \mid R \in T\}$ .

Dla danej logiki, problem spełnialności formuł w tej logice definiujemy następująco: dana formuła  $\psi$  bez zmiennych wolnych, czy istnieje model  $\mathfrak{M}$  taki, że  $\mathfrak{M} \models \psi$  (model  $\mathfrak{M}$  spełnia  $\psi$ )?

### 2.3 Alternujące maszyny Turinga

Alternujące maszyny Turinga to uogólnienie maszyn niedeterministycznych. Stany alternującej maszyny Turinga podzielone są na cztery grupy: stany egzystencjalne ( $\exists$ ), stany uniwersalne ( $\forall$ ), stany akceptujące (*true*) i stany odrzucające (*false*). Stany egzystencjalne i uniwersalne, z których jest możliwe tylko jedno przejście, będziemy nazywać stanami deterministycznymi. Sytuacje, w których alternująca maszyna  $\mathfrak{M}$  akceptuje, można zdefiniować indukcyjnie – konfiguracja maszyny jest akceptująca, jeśli maszyna jest w stanie akceptującym, jeśli jest w stanie uniwersalnym i każda następna możliwa konfiguracja jest akceptująca lub gdy maszyna jest w stanie egzystencjalnym i pewna kolejna możliwa konfiguracja jest akceptująca. Maszyna akceptuje słowo  $w$ , jeśli konfiguracja składająca się ze stanu początkowego i słowa  $w$  na taśmie jest akceptująca. Więcej informacji dotyczących alternujących maszyn Turinga można znaleźć w [6].

### 2.4 Klasy złożoności obliczeniowej

Klasy złożoności obliczeniowej pozwalają na uszeregowanie problemów decyzyjnych według asymptotycznych czasów ich działania. Dla danej funkcji całkowitej  $f$  przez klasę (N)TIME( $f(n)$ ) oznaczamy klasę problemów rozstrzyganych przez (nie)deterministyczną maszynę Turinga w czasie ograniczonym przez  $f(n)$ , gdzie  $n$  jest rozmiarem wejścia. Podobnie, ASPACE( $f(n)$ ) oznacza klasę problemów rozstrzygalnych przez alternującą maszynę Turinga w pamięci  $f(n)$ . Niech *Poly* będzie zbiorem wszystkich wielomianów. W tej pracy będą nas szczególnie interesować klasy

$$2(\text{N})\text{EXPTIME} = \bigcup_{p \in \text{Poly}} (\text{N})\text{TIME}(2^{2^p(n)})$$

$$\text{AEXPSPACE} = \bigcup_{p \in \text{Poly}} \text{ASPACE}(2^{2^p(n)})$$

oraz znany fakt ([6]):

$$\text{AEXPSPACE} = 2\text{EXPTIME}$$

### 2.5 Postaci normalne

**Definicja 2.1.** Powiemy, że formuła  $\psi \in \text{FO}^2$  jest w postaci normalnej Scotta [17], jeśli jest koniunkcją formuł postaci

- (i)  $\exists x.\rho(x)$
- (ii)  $\forall xy.\delta(x, y)$
- (iii)  $\forall x.\exists y.\delta(x, y)$

gdzie  $\rho(x)$  i  $\delta(x, y)$  nie zawierają kwantyfikatorów oraz w  $\psi$  nie ma symboli relacyjnych o arności większej niż 2. Powiemy, że  $w$  jest *świadkiem* dla elementu  $v$ , jeśli dla pewnej formuły typu (iii) zachodzi  $\delta(v, w)$ .

Każdą formułę  $\text{FO}^2$  można w wielomianowym czasie przekształcić do formuły w postaci normalnej Scotta w taki sposób, aby otrzymana formuła była spełnialna wtedy i tylko wtedy, gdy wyjściowa formuła była spełnialna. Co więcej, formuła po przekształceniu ma modele tej samej mocy, co formuła wyjściowa. Postać normalna Scotta będzie dla nas szczególnie użyteczna ze względu na podobieństwo do postaci normalnej dla  $[\text{GF}^{2++}]$ .

**Definicja 2.2.** Powiemy, że formuła  $\psi \in [\text{GF}^{2++}]$  jest w postaci normalnej, jeśli jest koniunkcją formuł postaci

- (i')  $\exists x.\alpha(x) \wedge \rho(x)$
- (ii')  $\forall xy.\beta(x, y) \Rightarrow \delta(x, y)$
- (iii')  $\forall x.\alpha(x) \Rightarrow \exists y.\beta(x, y) \wedge \delta(x, y)$

gdzie  $\alpha(x)$  oraz  $\beta(x, y)$  są poprawnymi strażnikami, a  $\rho(x)$  i  $\delta(x, y)$  nie zawierają kwantyfikatorów.

**Lemat 2.3.** *Każdą formułę  $\varphi$  języka  $[\text{GF}^{2++}]$  nad sygnaturą  $\Sigma$  można efektywnie przekształcić do zbioru formuł  $\delta$  języka  $[\text{GF}^{2++}]$  nad sygnaturą  $\Sigma'$  w postaci normalnej tak, że*

- $\varphi$  jest spełnialna wtedy i tylko wtedy, gdy  $\bigvee \delta$  jest spełnialna
- $|\delta| = O(2^{|\varphi|})$  oraz  $|\Sigma'| = O(|\varphi|)$  oraz dla każdego  $\psi \in \delta$  zachodzi  $|\psi| = O(|\varphi| \log |\varphi|)$
- $|\delta|$  może zostać obliczona w wykładniczym czasie, a każda  $\psi \in \delta$  może zostać obliczona w czasie wielomianowym względem  $|\varphi|$

Dowód tego lematu jest bardzo podobny do dowodu lematu 3.2 z pracy [18] dotyczącego analogicznej własności dla  $\text{GF}^k$  z relacjami przechodnimi, gdyż w tym dowodzie w żadnym miejscu nie wnika się w postać strażników.

Dodatkowo będziemy zakładać, że koniunkt postaci (i') pojawia się w formule dokładnie 1 raz. Jeśli pojawia się więcej razy, każde kolejne wystąpienie  $\exists x.\alpha(x) \wedge \rho(x)$  można zastąpić przez  $\forall y.y = y \Rightarrow \exists x.\alpha(x) \wedge \rho(x)$ , natomiast jeśli nie pojawia się wcale oraz nie akceptujemy rozwiązania w postaci modelu nie zawierającego żadnych elementów, to można dodać formułę  $\exists x.x = x \wedge \top$ . Co więcej, używając techniki zaprezentowanej w [9] można wszystkie predykaty o arności większej niż 2 zastąpić predykatami binarnymi, nie wychodząc poza naszą postać normalną. Będziemy więc dalej zakładać, że w formule występują tylko predykaty unarne i binarne.

Analogicznie do przypadku  $\text{FO}^2$  powiemy, że  $w$  jest *świadkiem* dla elementu  $v$ , jeśli dla pewnej formuły typu (iii')  $v$  spełnia  $\alpha(v)$  oraz zachodzi  $\beta(v, w) \wedge \delta(v, w)$ .

### 3 Nierozstrzygalność złożenia

Dowód nierozstrzygalności problemu spełnialności dla wariantu logiki strzeżonej ze złożeniem jest implikowany przez wyniki z pracy [13]. Praca ta zawiera również dowód, że ta logika staje się rozstrzygalna, jeśli zabronimy używania symbolu równości. My zobaczymy, że wariant z równością nawet z dodatkowymi ograniczeniami ma nierozstrzygalny problem spełnialności. Okazuje się, że do pokazania nierozstrzygalności złożenia wystarczy składać relację z sobą samą. Dla ułatwienia będziemy stosować następujące oznaczenia:  $R^1 = R$  oraz  $R^{i+1} = R^i; R$ . Logikę, w której zezwalamy tylko na składanie relacji samej ze sobą, będziemy nazywać GF z potęgowaniem.

**Twierdzenie 3.1.** *Problem spełnialności GF<sup>2</sup> z potęgowaniem jest nierozstrzygalny.*

*Dowód.* Przez  $\mathfrak{G}_{\mathbb{N}}$  będziemy oznaczać model składający się ze zbioru wierzchołków  $\mathbb{N}^2$  oraz dwóch relacji:  $V = \{\langle x, n \rangle, \langle x, n + 1 \rangle \mid x, n \in \mathbb{N}\}$  oraz  $H = \{\langle n, y \rangle, \langle n + 1, y \rangle \mid n, y \in \mathbb{N}\}$ . Model  $\mathfrak{G}_{\mathbb{N}}$  jest nazywany kratą.

W celu pokazania nierozstrzygalności skorzystamy z poniższego lematu (sformułowanego w nieco większej ogólności w [15], na bazie techniki prezentowanej w [5]):

**Lemat 3.2.** *Niech  $L$  będzie rozszerzeniem GF<sup>2</sup> bez równości. Jeśli istnieje formuła  $\varphi \in L$  taka, że:*

- (1)  $\mathfrak{G}_{\mathbb{N}}$  może zostać rozszerzony do modelu  $\varphi$  przez określenie dodatkowych relacji
- (2) jeśli  $\mathfrak{M} \models \varphi$ , to istnieje homomorfizm z  $\mathfrak{G}_{\mathbb{N}}$  w  $\mathfrak{M}$

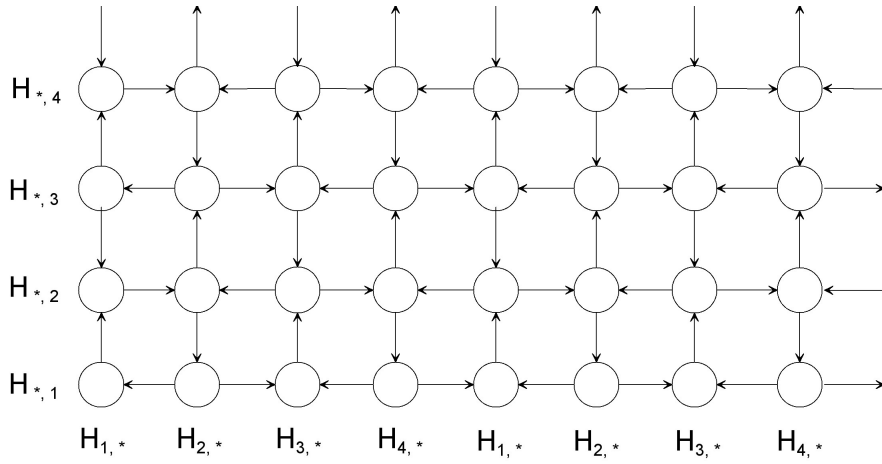
to problem spełnialności dla  $L$  jest nierozstrzygalny.

Wystarczy zatem zdefiniować taką formułę. Sygnatura dla naszej formuły będzie się składała z unarnych symboli  $H_{i,j}$  dla  $i, j \in \{1, 2, 3, 4\}$  oraz binarnych  $R, H, V$ . Dla uproszczenia notacji zdefiniujemy też następujące operacje:

- $\oplus : \{1, 2, 3, 4\}^2 \rightarrow \{1, 2, 3, 4\}$  taką, że  $i \oplus j = k$  wtedy i tylko wtedy, gdy  $i + j \equiv k \pmod{4}$
- $\ominus : \{1, 2, 3, 4\}^2 \rightarrow \{1, 2, 3, 4\}$  taką, że  $i \ominus j = k$  wtedy i tylko wtedy, gdy  $i - j \equiv k \pmod{4}$

Niech  $\varphi$  będzie koniunkcją następujących formuł:

- (1) istnieje element spełniający  $H_{1,1}$
- (2) każdy element spełnia dokładnie jeden z predykatów  $H_{i,j}$
- (3)  $\forall x. H_{i,j}(x) \Rightarrow (\exists y. xRy \wedge H_{i,j \oplus 1}(y)) \wedge (\exists y. yRx \wedge H_{i \oplus 1,j}(y))$  dla  $1 \leq i, j \leq 4$  takich, że  $i + j$  jest parzyste (wymusza istnienie odpowiednich następników i poprzedników dla tych elementów, które spełniają  $H_{i,j}$ )
- (4)  $\forall x. H_{i,j}(x) \Rightarrow (\exists y. xRy \wedge H_{i \oplus 1,j}(y)) \wedge (\exists y. yRx \wedge V_{i,j \oplus 1}(y))$  dla  $1 \leq i, j \leq 4$  takich, że  $i + j$  jest nieparzyste
- (5)  $\forall xy. xR; R; R; Ry \Rightarrow \bigvee_{i,j \in \{1,2,3,4\}} (H_{i,j}(x) \wedge H_{i,j}(y)) \Rightarrow x = y$  (jeśli elementy są odległe dokładnie o 4 i spełniają tę samą relację  $H_{i,j}$ , to są równe)
- (6) formuły definiujące relacje  $H$  i  $V$ :



Rysunek 1: Model  $K$  formuły  $\varphi$  (dla czytelności nie zaznaczono relacji  $V$  i  $H$ ).

- $\forall xy.xRy \Rightarrow H_{i,j}(x) \Rightarrow (H_{i\oplus 1,j}(y) \Rightarrow H(y,x)) \wedge (H_{i,j\oplus 1}(y) \Rightarrow V(x,y)) \wedge (H_{i\ominus 1,1}(y) \Rightarrow H(x,y)) \wedge (H_{1,j\ominus 1}(y) \Rightarrow V(y,x))$  dla  $i, j$  takich, że  $i + j$  parzyste
- $\forall xy.xRy \Rightarrow H_{i,j}(x) \Rightarrow (H_{i\oplus 1,j}(y) \Rightarrow H(x,y)) \wedge (H_{i,j\oplus 1}(y) \Rightarrow V(y,x)) \wedge (H_{i\ominus 1,1}(y) \Rightarrow H(y,x)) \wedge (H_{1,j\ominus 1}(y) \Rightarrow V(x,y))$  dla  $i, j$  takich, że  $i + j$  nieparzyste

Łatwo zauważyć, że model  $K$  z rysunku 1 jest modelem  $\varphi$ . Co więcej, dla każdego modelu  $\mathfrak{M}$  formuły  $\varphi$ , istnieje homomorfizm  $h$  z  $K$  w  $\mathfrak{M}$ , a więc także z  $\mathfrak{G}_{\mathbb{N}}$  w  $\mathfrak{M}$ . Dla ustalenia notacji przyjmijmy, że lewy dolny róg  $K$  nazywamy elementem  $(0,0)$ , a element odległy od  $(0,0)$  o  $k$  przejść w prawo i  $l$  w górę nazywa się  $(k,l)$ . Zdefiniujmy również porządek na tych elementach taki, że  $(k,l) \leq (m,n) \Leftrightarrow k+l \leq m+n$ . Zauważmy, że ten porządek jest dobrze ufundowany, więc można przeprowadzić indukcję zgodnie z nim.

Weźmy dowolny model  $\mathfrak{M}$  formuły  $\varphi$ . Zbudujemy teraz indukcyjnie homomorfizm  $h$  jednocześnie dowodząc, że dla dowolnego  $(k,l)$  po ustaleniu wartości  $h((k+1, l+1))$  istnieje w  $\mathfrak{M}$  odpowiednio połączony (tzn. zgodnie z rysunkiem 1) kwadrat złożony z elementów  $h((k,l))$ ,  $h((k+1, l))$ ,  $h((k, l+1))$ ,  $h((k+1, l+1))$ . Wobec tego, że w modelu  $K$  każdy z punktów jest połączony tylko z punktami odległymi o 1, to ta lokalna własność wystarcza, aby  $h$  było homomorfizmem.

Zauważmy na początek, że w punkcie 1 wymuszamy istnienie elementu  $h((0,0))$ . Weźmy dowolne  $k$  i  $l$ . Dla ustalenia uwagi założmy, że  $k, l > 0$  oraz że  $k \bmod 4 = l \bmod 4 = 0$  (pozostałe przypadki są podobne). Z założenia indukcyjnego wiemy, że istnieją punkty  $h((k,l))$ ,  $h((k, l+1))$  oraz  $h((k+1, l))$  takie, że  $h((k,l))Rh((k, l+1))$ ,  $h((k+1, l))Rh((k,l))$ , oraz te punkty mają odpowiednie typy:  $H_{2,1}(h((k+1, l)))$  i  $H_{1,2}(h((k, l+1)))$ . Z członu 4 dla  $h((k+1, l))$  wynika, że istnieje  $v_1 \in M$  takie, że  $H_{2,2}(v_1)$  oraz  $v_1Rh((k+1, l))$ , natomiast dla  $h((k, l+1))$  wynika, że istnieje  $v_2$  takie, że  $H_{2,2}(v_2)$  oraz  $h((k, l+1))Rv_2$ . Mamy zatem ścieżkę relacją  $R$ :  $v_1, h((k+1, l)), h((k, l)), h((k, l+1)), v_2$ , na której  $v_1$  i  $v_2$  spełniają ten sam predykat  $H_{2,2}$  oraz są odległe o 4. Z członu 5 dostajemy więc  $v_1 = v_2$  i ustalamy  $h((k+1, l+1)) = v_1$ .

Pozostaje tylko zauważyć, że punkt 6 rzeczywiście definiuje odpowiednie połączenia relacjami  $V$  i  $H$ .  $\square$

Używając bardzo podobnej techniki można pokazać, że  $\text{GF}^2$  ze złożeniem, w którym zabronimy składania relacji samej z sobą, ma nierozstrzygalny problem spełnialności – wystarczy zadbać o to, by relacje o różnych kierunkach i zwrotach miały różne nazwy i odpowiednio dostosować do tego wszystkie człony formuły  $\varphi$ . W analogiczny sposób można również wykazać,



że problem skończonej spełnialności dla  $\text{GF}^2$  ze złożeniem jest nierozstrzygalny. Przez problem skończonej spełnialności rozumiemy tu pytanie, czy dana formuła ma model skończony.

## 4 Własność modelu rozgałęzionego dla $\text{GF}^2++$

Dowód rozstrzygalności dla  $[\text{GF}^2++]$  bazuje na idei przedstawionej w [18]. Oba dowody tworzą najpierw model o pewnej użytecznej własności, opierając się na następujących pomysłach:

- (1) Bierzemy formułę w postaci normalnej i jej model.
- (2) Zauważamy, że wystarczy, by w tym modelu między dowolnymi dwoma punktami było połączenie najwyżej jedną relacją z tych, które są przechodnie/przechodnio domykane.
- (3) Wybieramy punkt z tego modelu i od niego zaczynamy budować nowy model.
- (4) Jeśli ten punkt jest w jakiejś klicie złożonej z relacji przechodnich/przechodnio domykanych, to bierzemy tę klikę, kompresujemy i dodajemy do budowanego modelu.
- (5) Jeśli element wymaga jeszcze jakichś świadków, to znajdujemy ścieżkę w modelu do tego świadka, kompresujemy ją i dodajemy ją do modelu.
- (6) Rekurencyjnie zapewniamy świadków dla nowododanych elementów.

Warto sobie uzmysłowić, dlaczego zajmujemy się klikami. Naszym celem jest stworzenie modelu, który przypomina drzewo. W logice  $[\text{GF}^2++]$  można napisać formułę  $\forall x(S(x) \Rightarrow \forall y(xR^+y \Rightarrow S_0(y) \Rightarrow x = y))$ , która mówi, że z elementu  $v$  spełniającego  $S_0$  nie jest osiągalny ścieżką relacją  $R$  żaden element różny od  $v$ , który spełnia  $S_0$ . Jeśli dodamy do tego formuły mówiące, że każdy element spełnia dokładnie jeden z predykatów  $S_0, \dots, S_{n-1}$ , istnieje element spełniający  $S_0$ , oraz formuły  $\forall x.S_i(x) \Rightarrow \exists y.xRy \wedge S_{(i+1) \bmod n}(y)$  dla  $i < n$ , to otrzymamy formułę, której każdy model zawiera cykl długości  $n$ . Nieco staranniejsze podejście pozwala wyrażać nawet cykle długości wykładniczej. W związku z tym struktura modelu lokalnie czasem nie może przypominać drzewa. Okazuje się, że w obu logikach każdy taki niedrzewiasty fragment można w pewnym sensie skompresować do fragmentu o jedynie wykładniczym rozmiarze.

Istotne różnice między dowodami występują w punktach 4 i 5 i wynikają głównie z tego, że tym razem pojawiające się klikki nie są po prostu połączone na zasadzie „każdy z każdym”, ale na zasadzie „z każdego punktu można dojść do każdego”. Jest możliwe, że pomiędzy pewnymi punktami w takiej klicie w ogóle nie może być bezpośredniego połączenia. Dlatego najpierw przystosujemy technikę kompresji klik tak, by pasowała do sytuacji, w której otrzymany model dopiero po przechodnim domknięciu jednej z relacji ma być kliką. W tym celu wykażemy pewną własność  $\text{FO}^2$ , która wydaje się być ciekawa sama w sobie.

**Definicja 4.1.** Powiemy, że model  $\mathfrak{M}$  jest *dwuspójny* względem relacji  $T$ , jeśli istnieje element modelu  $v$  taki, dla dowolnego elementu modelu  $w \neq v$  zachodzi  $\mathfrak{M} \models vT^+w \wedge wT^+v$ .

Zauważmy, że model jest dwuspójny względem relacji  $T$  wtedy i tylko wtedy, gdy cały model jest kliką względem przechodniego domknięcia  $T$ .

### 4.1 Własność modelu wykładniczego dla dwuspójnego $\text{FO}^2$

Niech  $U = \{U_1, U_2, \dots, U_u\}$  będzie zbiorem unarnych symboli relacyjnych,  $B = \{T, B_1, B_2, \dots, B_b\}$  – zbiorem binarnych symboli relacyjnych oraz  $\Sigma = B \cup U$ . Ustalmy formułę języka  $\text{FO}^2$

w postaci normalnej Scotta nad sygnaturą  $\Sigma$

$$\varphi = \exists x. \rho(x) \wedge \forall x \forall y \phi(x, y) \wedge \bigwedge_{i=1}^k \forall x \exists y \psi_i(x, y)$$

taką, że  $\mathfrak{M}$  jest dwuspójnym względem  $T$  modelem  $\varphi$  o uniwersum  $M$ . Pokażemy, że ten model możemy przebudować do modelu o rozmiarze wykładniczym względem  $|\Sigma|$ , nie tracąc dwuspójności.

#### Definicje 4.2.

- Elementami *królewskimi* nazywamy te elementy, które w modelu mają unikalny 1-typ.
- Ścieżką relacją  $R$  nazywamy ciąg wierzchołków  $v_1, v_2, \dots, v_k$  taki, że dla każdego  $i < k$  zachodzi  $\mathfrak{M} \models v_i R v_{i+1}$ .
- Ścieżkę  $s, v_1, \dots, v_n, t$  nazywamy *niekrólewską*, jeśli żaden z elementów  $v_1, \dots, v_n$  nie jest królewski.
- *Skrótem* ścieżki  $s, \dots, v_{pi}, v_i, \dots, v_j, v_{nj}, \dots, t$ , na której wierzchołki  $v_i$  oraz  $v_j$  mają takie same 1-typy, nazywamy ścieżkę  $s, \dots, v_{pi}, v_i, v_{nj}, \dots, t$ , na której wierzchołki  $v_i$  i  $v_{nj}$  są połączone dokładnie tak, jak  $v_j$  z  $v_{nj}$  na skracanej ścieżce.
- Powiemy, że ścieżka  $s$  jest *kompresją* ścieżki  $t$ , jeśli  $s$  nie ma żadnego skrótu oraz istnieje ciąg  $r_1, r_2, \dots, r_n$  taki, że  $r_1 = t$ ,  $r_n = s$  oraz  $r_{i+1}$  jest skrótem  $r_i$  dla  $1 \leq i \leq n$ .

Zauważmy, że jeśli sygnatura jest skończona, to każdą ścieżkę można skompresować do ścieżki o rozmiarze wykładniczym względem rozmiaru sygnatury.

**Lemat 4.3.** *Niech  $\varphi$  będzie formułą  $FO^2$  w postaci normalnej Scotta nad sygnaturą  $\Sigma$  z wyróżnionym binarnym symbolem relacyjnym  $T$ , składającą się z  $k$  koniunktów taką, że  $\mathfrak{M}$  jest dwuspójnym względem  $T$  modelem tej formuły. Wtedy formuła  $\varphi$  ma model dwuspójny względem relacji  $T$  o rozmiarze co najwyżej  $k \cdot 2^{2^{|\Sigma|+5}}$ , w którym występują wszystkie 1-typy z modelu  $\mathfrak{M}$ .*

*Dowód lematu.* Zbudujemy model  $\mathfrak{M}'$  o uniwersum  $M'$  taki, że  $|M'| \leq k \cdot 2^{2^{|\Sigma|+5}}$ . Dowód będzie przebiegał podobnie do dowodu własności modelu wykładniczego dla  $FO^2$ , zaprezentowanego w [9]. Model będzie składał się z następujących części: pałacu królewskiego  $V_k$ , dworu  $V_d$  i trzech miasteczek  $V_1, V_2$  i  $V_3$ , tworzonych następująco:

- (1) Do pałacu królewskiego wstawiamy wszystkie elementy królewskie z  $\mathfrak{M}$ , zachowując ich 1-typy. Jeśli żaden element królewski nie spełnia  $\rho$ , to dodatkowo pałacu umieszczamy jedną kopię elementu, który w  $\mathfrak{M}$  spełnia  $\rho$ . Wszystkie połączenia wewnątrz pałacu ustalamy tak, jak w  $\mathfrak{M}$ .
- (2) Do dworu wstawiamy w pierwszej kolejności kopie tych elementów, które są świadkami dla odpowiedników elementów z pałacu w  $\mathfrak{M}$ , wraz z odpowiednimi połączeniami. Będziemy te elementy nazywać rdzennymi dworzanami.
- (3) Dla każdej niepołączonej relacją  $T$  pary  $(u, v)$  z pałacu królewskiego, takiej, że w  $\mathfrak{M}$  jest niekrólewska ścieżka  $s$  relacją  $T$  z  $u$  do  $v$ , dodajemy kompresję  $s$  do dworu.
- (4) Dla każdego elementu dworu, z którego nie ma jeszcze ścieżki relacją  $T$  do żadnego elementu pałacu, znajdujemy w  $\mathfrak{M}$  ścieżkę  $t$  do możliwie bliskiego elementu pałacu i, po kompresji, dodajemy ją do dworu. Analogicznie postępujemy z elementami, do których nie ma ścieżki od żadnego z elementów pałacu.

- (5) Ustalamy relacje między niepołączonymi elementami dworu zgodnie z tym, jak były one połączone w  $\mathfrak{M}$ . W tym celu dla każdej pary wierzchołków niepołączonych znajdujemy w  $\mathfrak{M}$  parę elementów o tym samym 1-typie i kopiujemy połączenia między nimi do  $\mathfrak{M}'$ . Zauważmy, że takie pary zawsze istnieją, gdyż wszystkie elementy dworu są wzięte z  $\mathfrak{M}$ , a ponadto jeśli jakiś typ elementu w  $\mathfrak{M}$  występował tylko 1 raz, to w  $\mathfrak{M}'$  też został dodany dokładnie raz.
- (6) Budujemy trzy miasteczka. W pierwszej kolejności do każdego z miasteczek dodajemy po  $k$  kopii każdego niekrólewskiego elementu, który występuje w  $\mathfrak{M}$ . Te elementy będziemy zwali rdzennymi mieszczanami.
- (7) Dla każdego rdzennego mieszczanina  $v$  o 1-typie  $t_v$  znajdujemy w  $\mathfrak{M}$  możliwie krótką ścieżkę  $s_1$  od elementu tego typu  $t_v$  do pałacu królewskiego w  $\mathfrak{M}$  oraz możliwie krótką ścieżkę  $s_2$  z pałacu do elementu typu  $t_v$  i dodajemy kompresję  $s_1$  oraz kompresję  $s_2$  do miasteczka, z którego  $v$  pochodzi, zamieniając skrajne elementy ścieżki odpowiednio na element  $v$  i element z pałacu.
- (8) Zapewniamy niekrólewskich świadków wszystkim elementom: dla elementów z  $V_d$  oraz  $V_3$  znajdujemy świadków w  $V_1$ , dla elementów z  $V_1$  w  $V_2$ , zaś dla elementów z  $V_2$  w  $V_3$ . Świadków tworzymy kopiując odpowiednie połączenia z oryginalnego modelu. Zauważmy, że element może żądać najwyżej  $k$  różnych świadków, więc  $k$  kopii każdego niekrólewskiego 1-typu wystarczy.
- (9) Dla tych par elementów, między którymi połączenia nie były jeszcze ustalone, znajdujemy w  $\mathfrak{M}$  punkty o tych samych typach i kopiujemy połączenie między nimi do nowego modelu.

W ten sposób został zdefiniowany cały model  $\mathfrak{M}'$ . Zauważmy, że  $|V_k| \leq 2^{|\Sigma|}$ ,  $|V_d| \leq (1 + 2 \cdot 2^{|\Sigma|})k|V_k|$  oraz  $|V_i| \leq (1 + 2 \cdot 2^{|\Sigma|})k2^{|\Sigma|}$  dla  $i \in \{1, 2, 3\}$ , zatem  $|M| = O(k \cdot 2^{|\Sigma|})$ . Zauważmy ponadto, że:

- Formuła  $\exists x \rho(x)$  jest spełniona, gdyż element spełniający tę formułę został jawnie dodany w 1. kroku.
- Wszystkie 1-typy oraz 2-typy z  $\mathfrak{M}'$  występują również w  $\mathfrak{M}$ , ponadto elementy królewskie pozostają królewskimi (nie dodaliśmy żadnego elementu królewskiego poza pałacem dzięki temu, że wybieraliśmy zawsze ścieżki do najbliższego elementu z pałacu), więc formuła  $\phi$  jest spełniona.
- Każdy element posiada wszystkich potrzebnych świadków, więc wszystkie formuły  $\psi_j$  są spełnione w  $\mathfrak{M}'$ .
- Pałac królewski tworzy dwuspójną składową względem  $T$ .
- Każdy rdzenny dworzanin oraz mieszczanin jest osiągalny z pałacu relacją  $T$  oraz można z niego osiągnąć pałac relacją  $T$  – odpowiednie ścieżki zostały dodane w krokach 4 i 7.
- Każdy nierdzenny dworzanin oraz mieszczanin jest osiągalny z pałacu relacją  $T$  oraz można z niego osiągnąć pałac relacją  $T$ , gdyż znajduje się on na ścieżce relacją  $T$  z pewnego rdzennego elementu do pałacu lub z pałacu do rdzennego elementu.

Zatem model spełnia formułę  $\varphi$ , pałac tworzy dwuspójną składową i dla każdego elementu istnieją ścieżki od niego do pałacu i z pałacu do niego, więc otrzymany model jest dwuspójny względem relacji  $T$ . Wprost z konstrukcji wynika, że w otrzymanym modelu są zrealizowane wszystkie 1-typy, które były zrealizowane w  $\mathfrak{M}$ .  $\square$

Dokładne szacowanie rozmiaru modelu w lemacie było potrzebne, aby w algorytmie, który powstanie później, móc ograniczyć rozmiar tworzonych klik. Dla formuł, które nie są w postaci normalnej Scotta, dokładna liczba będzie zależała od technicznych detali przekładu do postaci normalnej, jednak dla wielomianowego przekładu pozostanie ona wykładnicza względem sygnatury. Możemy więc sformułować ogólne twierdzenie.

**Twierdzenie 4.4.** *Niech  $\varphi$  będzie formułą  $FO^2$  nad sygnaturą  $\Sigma$  z wyróżnioną relacją binarną  $T$  taką, że  $\mathfrak{M}$  jest dwuspójnym względem  $T$  modelem tej formuły. Wtedy formuła  $\varphi$  ma model dwuspójny względem relacji  $T$  o rozmiarze wykładniczym względem  $|\Sigma|$ .*

## 4.2 Konstrukcja modelu rozgałęzionego dla $[GF^{2++}]$

Weźmy dowolną formułę  $\varphi = \exists x.\rho(x) \wedge \bigwedge_{i=1}^j \forall xy.\delta_i(x,y) \wedge \bigwedge_{i=1}^k \forall x.\alpha_i(x) \Rightarrow \exists y.\psi_i(x,y)$  w postaci normalnej należącej do języka  $[GF^{2++}]$ . Niech  $U$  będzie zbiorem unarnych symboli relacji występujących w  $\varphi$ ,  $B$  oraz  $T = \{T_1, \dots, T_n\}$  będą zbiorami symboli relacji binarnych z  $\varphi$  takimi, że wszystkie relacje przechodnio domykane w  $\varphi$  należą do  $T$ , natomiast pozostałe należą do  $B$ . Przez  $\Sigma = U \cup B \cup T$  będziemy oznaczać sygnaturę  $\varphi$ .

Zdefiniujemy na początek pewne własności połączeń między elementami.

### Definicje 4.5.

- Niech  $t$  będzie właściwym 2-typem. Powiemy, że  $t$  jest *k-pozytywny*, gdy istnieje dokładnie  $k$  różnych relacji  $R$  należących do zbioru  $T$  takich, że dla pary  $x, y$  o 2-typie  $t$  w dowolnym modelu zachodzi  $xRy \vee yRx$ .
- Powiemy, że model jest *1-pozytywny*, gdy każdy 2-typ występujący w tym modelu jest 1-pozytywny lub 0-pozytywny.
- *Obcięciem 2-typu  $t$  do relacji  $R \in T$*  nazywamy 2-typ, który jest zgodny z  $t$  na relacjach  $S \in \{R\} \cup U \cup B$  i w którym dla żadnej relacji  $S \in T \setminus \{R\}$  nie zachodzi  $\langle 3, S \rangle \in t$  ani  $\langle 4, S \rangle \in t$ .
- *Rozszerzonym 1-typem* elementu  $v$  będziemy nazywać zbiór składający się z 1-typu tego elementu oraz par  $\langle R, t \rangle$ , takich, że  $R \in T \cup T^{-1}$  oraz dla pewnego  $w \neq v$  o 1-typie  $t$  zachodzi  $vR^+w$ .

Oczywiście każdy obcięty 2-typ jest co najwyżej 1-pozytywny. Tworzony przez nas model będzie się składał z samych obciętych 2-typów, więc będzie 1-pozytywny.

Aby panować nad sytuacją, w której między pewnymi punktami mamy połączenia w obie strony, wprowadźmy pojęcie klik i podstawowe operacje na niej.

### Definicje 4.6.

- $R^+$ -kliką w modelu  $\mathfrak{M}$  nazywamy podzbiór  $M$  elementów modelu  $\mathfrak{M}$  taki, że dla każdego  $v, w \in M$  zachodzi  $\mathfrak{M} \models vR^+w$ .
- Przez  $R^+$ -klikę( $v$ ) oznaczamy największą  $R^+$ -klikę zawierającą  $v$ .
- *Ścieżką  $R^+$ -klik* nazywamy ciąg  $\langle C_1, v_1^{in}, v_1^{out} \rangle, \langle C_2, v_2^{in}, v_2^{out} \rangle, \dots, \langle C_k, v_k^{in}, v_k^{out} \rangle$  taki, że dla każdego  $1 \leq i \leq k$   $C_i$  jest maksymalną  $R^+$ -kliką, do której należą wierzchołki  $v_i^{in}$  i  $v_i^{out}$ , oraz dla  $i < k$  wierzchołek  $v_i^{out}$  jest połączony relacją  $R$  z  $v_{i+1}^{in}$ .

- Powiemy, że wierzchołek  $v$  jest odległy od  $w$  o  $m$  klik, jeśli najkrótsza ścieżka klik z pewnej klikli zawierającej  $v$  do pewnej klikli zawierającej  $w$  ma długość  $m$ .
- *Skrótem* ścieżki  $R^+$ -klik  $\langle C_1, v_1^{in}, v_1^{out} \rangle, \dots, \langle C_z, v_z^{in}, v_z^{out} \rangle, \langle C_p, v_p^{in}, v_p^{out} \rangle, \dots, \langle C_d, v_d^{in}, v_d^{out} \rangle, \dots$  na której wierzchołki  $v_p^{in}$  oraz  $v_d^{in}$  mają takie same 1-typy, nazywamy ścieżkę  $\langle C_1, v_1^{in}, v_1^{out} \rangle, \dots, \langle C_z, v_z^{in}, v_z^{out} \rangle, \langle C_d, v_d^{in}, v_d^{out} \rangle, \dots$
- *Kompresją ścieżki  $R^+$ -klik  $s$*  nazywamy dowolną minimalną ścieżkę powstającą przez iterowanie operacji skracania ścieżki  $s$ .

Zauważmy, że dla dowolnej relacji  $R \in T$  pojedynczy wierzchołek tworzy  $R^+$ -klikę.

Zdefiniujemy teraz pewną operację na grafie, która pozwoli nam zdefiniować drzewiasty charakter modelu. Intuicyjnie, spłaszczenie modelu polega na zastąpieniu każdej klikli pojedynczym wierzchołkiem oraz połączeniu wierzchołków odpowiadających tym klikom, które miały wspólny wierzchołek, były połączone pewną relacją z  $T$  lub były połączone pewną relacją z  $B$ , a nie były połączone przechodnim domknięciem żadnej relacji z  $T$ . Formalna definicja jest dość techniczna.

**Definicja 4.7.** Powiemy, że nieskierowany graf  $G = \langle V, E \rangle$  jest *spłaszczeniem* modelu  $\mathfrak{M}$  z korzeniem  $K$ , jeśli

- $V = \{R^+$ -klik( $w$ ) $|R \in T \wedge w \in M\}$ , czyli każdy wierzchołek odpowiada jednej klikli z  $\mathfrak{M}$
- $E = \bigcup_{i=1}^{\infty} Y^i(\{K, K\}) \setminus \{K, K\}$ , gdzie  $Y(E)$  jest zbiorem zawierającym  $E$  oraz takie krawędzie  $\{T_i^+$ -klik( $w$ ),  $T_j^+$ -klik( $w'$ ) $\}$ , że  $T_i^+$ -klik( $w$ ) należy do pewnej krawędzi z  $E$ ,  $T_j^+$ -klik( $w'$ ) nie należy do żadnej krawędzi z  $E$  oraz zachodzi jedna z trzech okoliczności:
  - $w = w'$
  - dla pewnej  $S \in T \cup T^{-1}$  w  $\mathfrak{M}$  zachodzi  $wSw'$
  - dla pewnej relacji  $S \in B \cup B^{-1}$  w  $\mathfrak{M}$  zachodzi  $wBw'$  oraz dla żadnej relacji  $S \in T \cup T^{-1}$  nie zachodzi  $wT^+w'$

Intuicyjnie, operator  $Y$  do utworzonej już spójnej składowej grafu dodaje sąsiadujące z nią klikli, które nie są jeszcze w tej składowej. Operacja zaczyna się od wierzchołka  $K$ . Ze względu na to, że model może być nieskończony, zbiór krawędzi  $G$  wyrażamy jako nieskończoną sumę krawędzi dodawanych w kolejnych krokach.

Wprowadzimy teraz pewną własność modelu, która pozwoli na łatwe panowanie nad strukturą modelu przez algorytm rozstrzygający problem spełnialności.

**Definicja 4.8.** Powiemy, że model  $\mathfrak{M}$  formuły  $\varphi$  jest  *$r$ -rozgałęziony*, jeżeli

- (1)  $\mathfrak{M}$  jest 1-pozytywny.
- (2) Rozmiar każdej  $R^+$ -klikli w  $\mathfrak{M}$  dla  $R \in T$  jest ograniczony przez  $r$ .
- (3) Spłaszczenie  $\mathfrak{M}$  o dowolnym korzeniu jest drzewem (być może nieskończonym).

**Twierdzenie 4.9.** *Jeśli  $\varphi$  ma model, to ma model  $r$ -rozgałęziony dla  $r = |\varphi| \cdot 2^{2^{|\Sigma|+5}}$ , w którym każdy element ma wszystkich potrzebnych świadków odległych o co najwyżej  $2^{|\Sigma|}$  klik.*

*Dowód.* Niech  $\mathfrak{M}$  będzie modelem  $\varphi$ . Na podstawie  $\mathfrak{M}$  utworzymy  $|\varphi| \cdot 2^{2|\Sigma|+5}$ -rozgałęziony model  $\mathfrak{M}'$ . Równocześnie będziemy tworzyć funkcję *from*, która każdemu wierzchołkowi  $\mathfrak{M}'$  będzie przyporządkowywała element  $\mathfrak{M}$ , z którego ten wierzchołek powstał.

Na początku z modelu  $\mathfrak{M}$  wybieramy punkt  $r$ , który spełnia  $\rho$ , i dodajemy jego kopię  $v$  do zbioru wierzchołków  $\mathfrak{M}'$ , ustalając  $from(v) = r$ . Następnie dla tego wierzchołka wykonujemy następujące operacje:

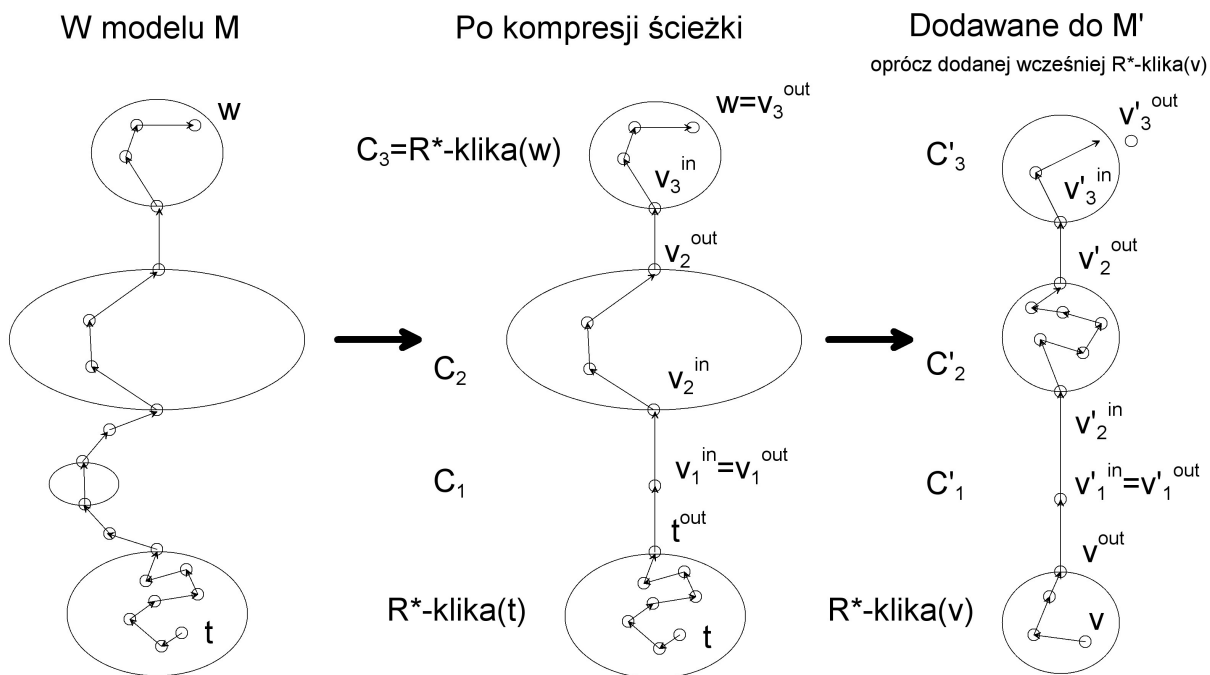
- (1) Kopiujemy z  $\mathfrak{M}$  kliki, w których występuje  $from(v)$ . W tym celu znajdujemy w  $\mathfrak{M}$  punkt  $t = from(v)$ , a następnie dla każdej relacji  $R \in T$  sprawdzamy, czy  $v$  znajduje się już wewnątrz  $R^+$ -kliki o rozmiarze co najmniej 2. Jeśli nie, to wybieramy z modelu  $\mathfrak{M}$  maksymalny zbiór wierzchołków  $H$  zawierający  $t$  oraz będący  $R^+$ -kliką i przycinamy 2-typy w nim występujące do relacji  $R$ . Otrzymany podmodel tworzy dwuspójną względem  $R$  składową. Jeśli ta składowa ma więcej niż 1 element, to korzystając z metody przedstawionej w rozdziale 4.1 dla formuły  $\exists x. \top \wedge \bigwedge_{i=1}^j \forall xy. \delta'_i(x, y)$  przekształcamy tę składową do grafu  $H'$  o wykładniczym rozmiarze, gdzie formuły  $\delta'_i$  powstają przez zamianę wszystkich strażników  $xR^+y$  na  $\top$  (bo model i tak ma być  $R$ -dwuspójny) oraz usunięcie wszystkich koniunktów zawierających  $xS^+y$  dla  $S \in T \setminus \{R\}$  (gdyż usunęliśmy z tej składowej wszystkie inne relacje z  $T$ ). Następnie z  $H'$  wybieramy wierzchołek  $w$  o 1-typie takim, jak  $v$ , i dołączamy  $H'$  do modelu  $\mathfrak{M}'$  poprzez utożsamienie  $v$  z  $w$ . Na koniec dla każdego wierzchołka  $r' \neq w$  z  $H'$  znajdujemy w  $H$  wierzchołek  $r$  o tym samym 1-typie i ustalamy  $from(r') = r$ .

- (2) Zapewniamy potrzebnych świadków dla  $v$ . W tym celu dla każdej formuły  $\psi_i = \gamma_i \wedge \delta_i$ , gdzie  $\gamma_i$  jest strażnikiem, punkt  $v$  spełnia  $\alpha_i$  oraz która jeszcze nie jest spełniona w  $\mathfrak{M}'$ , wybieramy z modelu  $\mathfrak{M}$  punkt  $w$ , który jest świadkiem dla  $from(v)$ . Zauważmy, że  $w$  nie leży w jednej klicie z  $from(v)$ . Następnie:
  - (a) Jeśli  $\gamma_i = xRy$  lub  $\gamma_i = yRx$  dla pewnego  $R \in B \cup T$ , to dołączamy kopię  $w'$  punktu  $w$  do modelu  $\mathfrak{M}'$  i ustalamy połączenia między  $v$  i  $w'$  takie, jakie były w  $\mathfrak{M}$ , ale po obciążeniu do  $R$ . Ponadto ustalamy  $from(w') = w'$ .
  - (b) Jeśli  $\gamma_i = xR^+y$  lub  $\gamma_i = yR^+x$  dla pewnego  $R \in T$ , to wybierzemy z  $\mathfrak{M}$  całą ścieżkę świadczącą o spełnieniu tej formuły wraz z pojawiającymi się na niej  $R^+$ -klikami. Skupmy się najpierw na sytuacji, gdy  $\gamma_i = xR^+y$ .  
Z modelu  $\mathfrak{M}$  wybieramy najkrótszą ścieżkę  $R^+$ -klik z  $R^+$ -kliki( $t$ ) do  $R^+$ -kliki( $w$ ), przycinamy wszystkie 2-typy występujące na niej do  $R$ , a następnie kompresujemy tę ścieżkę otrzymując ścieżkę  $\langle R^+$ -kliki( $t$ ),  $t$ ,  $t^{out} \rangle$ ,  $\langle C_1, v_1^{in}, v_1^{out} \rangle$ ,  $\dots$ ,  $\langle C_k, v_k^{in}, v_k^{out} \rangle$ , w której  $C_k = R^+$ -kliki( $w$ ) oraz  $v_k^{out} = w$ .  
Każdą z klik  $C_i$  kompresujemy do kliki  $C'_i$  tak, jak w punkcie 1, otrzymując kliki, w których punkty  $v_i^{in}$  i  $v_i^{out}$  mają odpowiednio takie 1-typy, jak  $v_i^{in}$  i  $v_i^{out}$ . W  $R^+$ -kliki( $v$ ) znajdujemy wierzchołek  $v^{out}$  o tym samym 1-typie, co  $t^{out}$ . Do modelu  $\mathfrak{M}'$  dodajemy ścieżkę  $\langle C'_1, v_1^{in}, v_1^{out} \rangle$ ,  $\dots$ ,  $\langle C'_k, v_k^{in}, v_k^{out} \rangle$  oraz ustalamy połączenia między  $v^{out}$  oraz  $v_1^{in}$  takie, jakie były między  $t^{out}$  a  $v_1^{in}$ . Przykładowe przekształcenie ścieżek można zobaczyć na rysunku 2.

Pozostaje ustalić wartości dla funkcji *from*. Dla każdego  $i$  ustalamy  $from(v_i^{in}) = v_i^{in}$  oraz  $from(v_i^{out}) = v_i^{out}$ , a ponadto dla każdego wierzchołka  $r' \in C'_i$ , dla którego funkcja *from* nie została jeszcze ustalona, znajdujemy w  $C_i$  wierzchołek  $r$  o tym samym 1-typie i ustalamy  $from(r') = r$ .

W przypadku, gdy formuła jest postaci  $\gamma_i = yR^+x$ , wystarczy powtórzyć rozumowanie dla  $R^{-1}$ .

- (3) Łączymy punkty w modelu  $\mathfrak{M}'$  relacjami z  $B$  tak, jak były połączone ich odpowiedniki w  $\mathfrak{M}$ . W tym celu bierzemy z modelu dwa punkty  $v, w$  oraz:



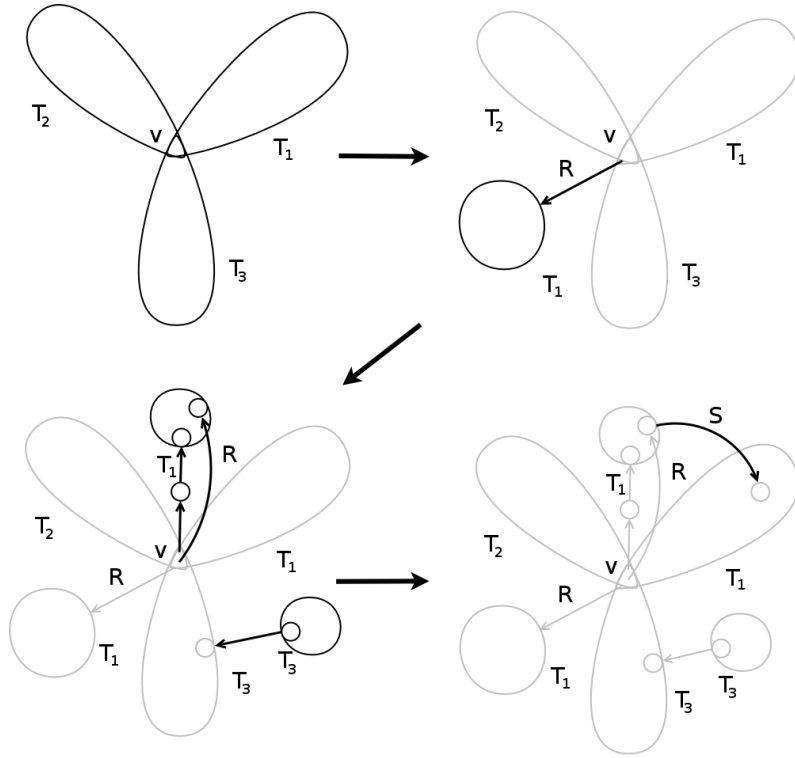
Rysunek 2: Przykładowe przekształcenie ścieżki  $R^+$ -klik – dla czytelności pominięto pozostałe elementy klik oraz połączenia relacjami innymi niż  $R$ .

- (a) Jeśli te punkty są połączone pewną relacją  $R^+$  dla  $R \in T$  i zachodzi  $vR^+w \wedge wR^+v$ , to znaczy, że te elementy są jednej klicie, zatem połączenia między nimi są już ustalone.
- (b) Jeśli te dwa punkty są połączone pewną relacją  $R^+$  dla  $R \in T$ , połączenie nie jest symetryczne (dla ustalenia uwagi założymy, że zachodzi  $vR^+w \wedge \neg wR^+v$ ) i połączenia między tymi punktami nie są jeszcze ustalone, to znajdujemy w  $\mathfrak{M}$  punkt  $w'$  taki, że w  $\mathfrak{M}$  jest ścieżka relacją  $R$  z  $from(v)$  do  $w'$  oraz  $w'$  ma taki sam 1-typ, co  $w$ . Taki punkt istnieje, gdyż 1-typ  $w$  należy do zbioru widocznych 1-typów względem relacji  $R$  zapisanego w rozszerzonym 1-typie  $from(v)$  (patrz obserwacja 4.10). Dodajemy połączenia ze zbioru  $B$  między  $v$  i  $w$  zgodnie z tym, jakie relacje są między  $from(v)$  i  $w'$ .
- (c) Jeśli te dwa punkty nie są połączone żadnym przechodnim domknięciem relacji z  $T$ , to albo są połączone jakąś relacją z  $B$  i wszystkie połączenia między nimi zostały już ustalone, albo w ogóle nie są połączone – wtedy pozostawiamy te punkty niepołączone.

(4) Powtarzamy kroki 1-4 dla nowododanych elementów.

W ten sposób tworzy się przeliczalny model formuły  $\varphi$ . Przykładowy fragment działania powyższego algorytmu można zobaczyć na rysunku 3. Aby się przekonać o poprawności zbudowanego modelu zauważmy najpierw, że formuła  $\rho$  jest spełniona, bo jawnie dodaliśmy do modelu punkt, który ją spełnia. Łatwo również zauważyć, że w trakcie konstrukcji każdemu punktowi zapewniamy wprost wszystkich potrzebnych świadków, więc wszystkie formuły dotyczące świadków są spełnione. Zauważmy również, że:

**Obserwacja 4.10.** *Rozszerzony 1-typ każdego punktu  $v$  tworzonego modelu jest podzbiorem (być może niewłaściwym) rozszerzonego 1-typu  $from(v)$ . W szczególności 1-typ  $v$  jest identyczny z 1-typem punktu  $from(v)$ .*



Rysunek 3: Pojedyncza tura tworzenia modelu rozgałęzionego. Kolejne etapy to: dodanie klik zawierających  $v$  (podpunkt 1), dodanie świadków bezpośrednich (podpunkt 2a), dodanie ścieżek klik (podpunkt 2b) oraz dodanie brakujących połączeń (podpunkt 3b).

Oczywiście wartości funkcji  $from$  w każdym kroku działania algorytmu były ustalane tak, by 1-typ każdego wierzchołka  $v$  był identyczny z 1-typem  $from(v)$ . Mniej oczywistą częścią jest to, że dla każdego wierzchołka  $w$  o 1-typie  $t_w$  i każdej relacji  $R \in T \cup T^{-1}$  takich, że  $\mathfrak{M}' \models vR^+w$ , para  $\langle R, t_w \rangle$  należy do rozszerzonego 1-typu  $from(v)$ . Aby się o tym przekonać, należy przyjrzeć się krokom 1 i 2 budowania modelu i zauważyć, że ilekroć dodajemy jakiś wierzchołek osiągalny z  $v$  relacją  $R$ , to bierzemy pod uwagę albo rozszerzony 1-typ  $from(v)$ , albo rozszerzony typ jakiegoś wierzchołka, który jest osiągalny z  $from(v)$  relacją  $R$ . To znaczy, że 1-typ dodanego wierzchołka jest zapisany w rozszerzonym 1-typie  $from(v)$ .

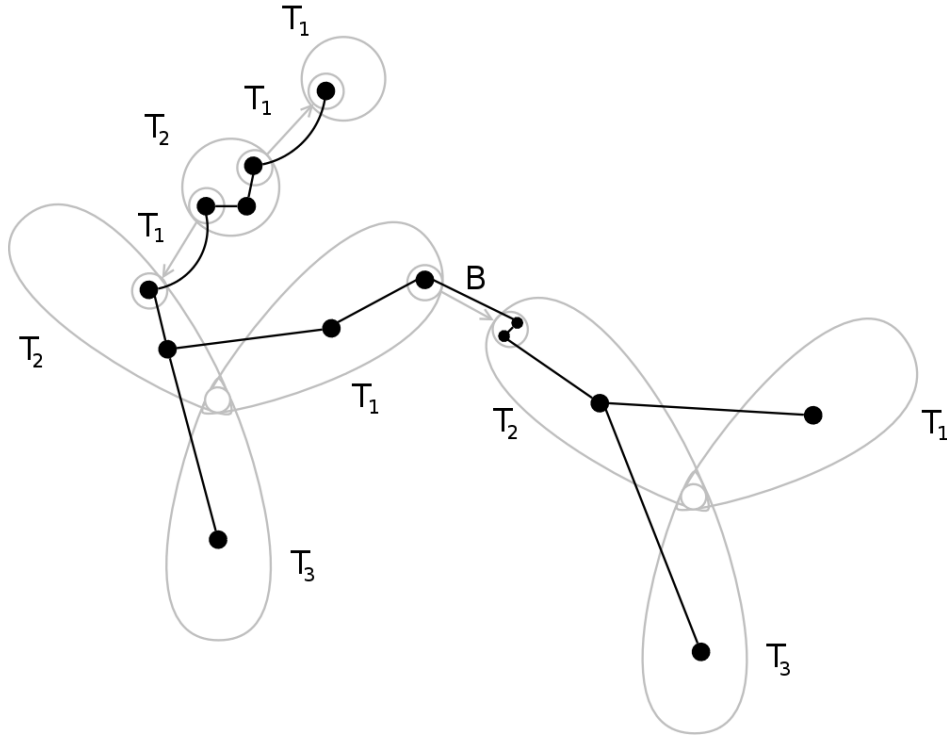
**Obserwacja 4.11.** *Przypuśćmy, że para elementów  $v, w$  modelu  $\mathfrak{M}$  o 2-typie  $t$  spełnia dla pewnego  $i$  formułę  $\delta_i(x, y) = \gamma(x, y) \Rightarrow \epsilon(x, y)$  oraz strażnika  $\gamma(x, y)$ . Jeśli 2-typ  $t'$  pary  $v', w'$  jest podzbiorem  $t$  oraz  $t|_{B \cup U} = t'|_{B \cup U}$ , to  $\mathfrak{M} \models \epsilon(w', v')$ , a zatem również  $\mathfrak{M} \models \delta_i(v', w')$ .*

Prawdziwość tej obserwacji wynika z faktu, że symbole relacyjne z  $T$  nie mogą pojawiać się poza strażnikami. Pozostaje tylko się przekonać, że formuły  $\delta_i(x, y)$  są też spełnione dla dowolnych  $x, y$ .

Weźmy dowolną parę punktów  $v, w$  z modelu  $\mathfrak{M}'$  i formułę  $\delta_i$  :

- Jeśli  $w$  i  $v$  leżą w tej samej klice, to spełniają wszystkie  $\delta_i$  na mocy poprawności konstrukcji z rozdziału 4.1.
- Jeśli nie zachodzi poprzedni punkt oraz  $\delta_i(x, y)$  jest postaci  $xRy \Rightarrow \epsilon(x, y)$  lub  $yRx \Rightarrow \epsilon(x, y)$  dla  $R \in T \cup B$  oraz pewnego  $\epsilon$ , a ponadto  $\mathfrak{M} \models vRw$  (przypadek, gdy  $\mathfrak{M} \models wRv$ , jest symetryczny), to albo połączenie z  $v$  do  $w$  relacją  $R$  zostało ustalone w punkcie 2, albo w punkcie 3. W obu przypadkach 2-typ tej pary jest podzbiorem pewnego 2-typu





Rysunek 4: Przykład drzewa (oznaczonego czarnym kolorem) powstającego na bazie modelu (oznaczonego kolorem szarym). Dla czytelności pominięto niektóre kliki jednoelementowe.

występującego w modelu  $\mathfrak{M}$ , nie zawierającym być może pewnych relacji z  $T$ , więc na mocy obserwacji 4.11 ta para spełnia  $\delta_i$ .

- Jeśli nie zachodzi żaden z wcześniejszych punktów oraz  $\delta_i(x, y)$  jest postaci  $xR^+y \Rightarrow \epsilon(x, y)$  lub  $yR^+x \Rightarrow \epsilon(x, y)$  dla  $R \in T$  i pewnego  $\epsilon$ , oraz zachodzi  $\mathfrak{M} \models vR^+w$  lub  $\mathfrak{M} \models wR^+v$ , to zachodzi jedna z poniższych sytuacji:
  - $v$  i  $w$  są połączone bezpośrednio. Oznacza to, że zostały połączone albo w punkcie 2a, albo w punkcie 2b i są w jednej ścieżce klik. Wtedy połączenie między nimi jest skopiowane z  $\mathfrak{M}$ , być może bez pewnych połączeń z  $T$ , więc na mocy obserwacji 4.11 spełnia  $\delta_i$ .
  - $v$  i  $w$  nie są połączone bezpośrednio i połączenie między nimi powstało w punkcie 2a. Oznacza to, że między tymi punktami ustaliliśmy połączenie w punkcie 3b na podstawie pewnej pary punktów połączonych relacją  $R^+$ , usuwając być może niektóre połączenia relacjami z  $T$ . Na mocy obserwacji 4.11 spełnia  $\delta_i$ .
  - $v$  i  $w$  nie są połączone bezpośrednio i połączenie między nimi powstało w punkcie 2b. To znaczy, że albo te punkty są na jednej ścieżce klik, albo tylko jeden z punktów został dołączony i połączenie między  $v$  i  $w$  zostało ustalone w 3b. W obu przypadkach połączenie jest skopiowane z  $\mathfrak{M}$  na podstawie pewnej pary punktów połączonych relacją  $R^+$ , usuwając być może niektóre połączenia relacjami z  $T$ . Na mocy obserwacji 4.11 spełnia  $\delta_i$ .

Pozostaje zauważyć, że otrzymany model jest rzeczywiście  $|\varphi| \cdot 2^{2^{|\Sigma|+5}}$ -rozgałęziony. Zauważmy, że ten model jest 1-pozytywny, gdyż każdy 2-typ w modelu jest obcięciem pewnego 2-typu z  $\mathfrak{M}$ . Ponadto każda  $R^+$ -klika dla  $R \in T$  została utworzona metodą z rozdziału 4.1, więc ma co najwyżej  $|\varphi| \cdot 2^{2^{|\Sigma|+5}}$  (gdyż liczbę koniunktów można szacować z góry przez długość formuły). Aby się przekonać, że spłaszczenie  $\mathfrak{M}'$  jest drzewem, zauważmy, że dla wszystkich parami

różnych  $a, b, c$  oraz różnych  $R, S \in T$  takich, że  $c \in R\text{-klika}(a)$  oraz  $b \in S\text{-klika}(a)$  2-ty pary  $b, c$  jest 0-pozytywny, gdyż w trakcie konstrukcji w żadnym etapie nie łączymy takich punktów. Ponadto relacje z  $T$  oraz relacje z  $B$  między elementami niepołączonymi żadnym przechodnim domknięciem są w każdym kroku tworzenia algorytmu dodawane tylko między takimi klikami, z których przynajmniej jedna została właśnie dodana, a w dodatku dodawane jest najwyżej jedno połączanie relacją z  $T$  między różnymi klikami. Przykład można zobaczyć na rysunku 4

□

**Wniosek 4.12.** *Każda spełnialna formuła języka  $[\text{GF}^2+^+]$  ma przeliczalny model.*

## 5 Ujęcie algorytmiczne

### 5.1 Opis algorytmu

Na mocy twierdzenia z rozdziału 4.2 każda spełnialna formuła rachunku  $[\text{GF}^2+^+]$  ma model rozgałęziony. Algorytm rozstrzygający może zatem działać następująco: zgadnij element początkowy, wszystkie kliki, w których on się znajduje, oraz ścieżki klik do elementów będących świadkami dla elementu początkowego. Następnie powtarzaj te czynności dla nowododanych elementów tak długo, aż rozszerzone 1-typy wszystkich dodawanych elementów będą się powtarzały. Takie podejście daje jednak algorytm w  $2\text{NEXPTIME}$ , dlatego skupimy się na nieco bardziej skomplikowanej wersji tego algorytmu, która poprawia jego złożoność do  $2\text{EXPTIME}$ .

Aby łatwo kontrolować powstające elementy modelu, będziemy się posługiwać trzema pojęciami:

- 1-typem, zdefiniowanym tak, jak we wcześniejszych rozdziałach
- pełnym 1-typem, który oprócz informacji z 1-typu zawiera informację o 1-typach bezpośrednich następników relacjami, które nie są przechodnio domykane, oraz dla każdej relacji przechodnio domykanej  $R$  informację o 1-typach punktów osiągalnych ścieżką relacją  $R$ , a także informację o 1-typach punktów osiągalnych ścieżką relacją  $R^{-1}$ , przy czym pomijamy punkty, które są w jednej klice z rozważanym punktem
- typem kliki, składającym się z rozmiaru kliki, pełnych 1-typów wszystkich punktów kliki, informacji o połączeniach między tymi elementami oraz funkcji *obietnica*, która dla danego 1-typu  $t$  oraz liczby  $b \in \{-1, 1\}$ , oznaczającej, czy myślimy o następnikach, czy o poprzednikach, zwraca długość ścieżki  $(R^b)^+$ -klik z bieżącej kliki do pewnej kliki zawierającej element typu  $t$

Typ kliki opisuje dokładnie wszystkie lokalne własności kliki, więc dla uproszczenia w tym podrozdziale będziemy go utożsamiać z kliką opisywaną przez ten typ.

Algorytm zaczyna od zgadnięcia typu kliki początkowej  $K$ . Następnie sprawdza, czy lokalnie wszystko się w niej zgadza, to znaczy czy połączenia między punktami w klice nie prowadzą do niespełnienia formuły, czy wszystkich następników zapisanych w pełnych 1-typach punktów można odpowiednio połączyć z punktami bieżącej kliki, czy w klice istnieje punkt, który ma w następnikach typ zapewniany przez funkcję obietnica oraz czy elementy kliki mają zgodne 1-typy.

W kolejnym etapie algorytm zapewnia wszystkim elementom kliki bezpośrednich świadków, być może zgadując nowe typy klik i połączenia z niektórymi punktami zgadniętych klik do  $K$ . Algorytm sprawdza, czy nowe kliki są poprawne. Następnie dla każdej  $R^+$ -kliki  $K'$  i połączonej relacją  $R$  z wcześniejszą kliką sprawdza, czy 1-typy wszystkich elementów z  $K'$  są zapisane w

pełnych 1-typach elementów z  $K$  tak, aby mieć pewność, że da się te elementy połączyć relacjami binarnymi w taki sposób, aby zapewnić spełnienie badanej formuły. Ponadto algorytm dba o to, by zbiory 1-typów osiągalnych relacją  $R$  z  $K'$  były podzbiorami zbiorów osiągalnych  $R$  z punktów  $K$  i odwrotnie. Następnie algorytm sprawdza, czy informacja zapisana w typie  $K$  (w tym funkcja *obietnica*) oraz zgadnięte kliki gwarantują zapewnienie wszystkich świadków dla punktów z kliki.

Następnie algorytm zgaduje typy klik, które są na ścieżce do klik zawierających punkty o 1-typie zapisanym w funkcji *obietnica* z  $K$ . Dla łatwiejszego panowania nad zależnościami algorytm będzie zgadywał tylko pierwszy element tej ścieżki i sprawdzał, czy zgadnięta klika ma odpowiednio mniejszą wartość funkcji *obietnica* lub realizuje dany 1-typ. Oczywiście dla tych klik algorytm również sprawdza, czy są poprawne oraz czy ich pełne 1-typy są zgodne z informacjami z  $K$ .

Jeśli wszystko się zgadza, to rekurencyjnie wywołujemy algorytm dla wszystkich dodanych klik. Gdy wykonamy dostatecznie dużo operacji (podwójnie wykładniczo), to mamy gwarancję, że jakiś typ kliki się powtórzył i algorytm kończy działanie.

## 5.2 Algorytm

Podobnie jak we wcześniejszym rozdziale, weźmy z języka  $[GF^2+^+]$  dowolną formułę w postaci normalnej:  $\varphi = \exists x.\rho(x) \wedge \bigwedge_{i=1}^j \forall xy.\delta_i(x, y) \wedge \bigwedge_{i=1}^k \forall x.\alpha_i(x) \Rightarrow \exists y.\psi_i(x, y)$ . Niech  $U$  będzie zbiorem symboli relacji unarnych występujących w  $\varphi$ ,  $B$  oraz  $T = \{T_1, \dots\}$  będą zbiorami symboli relacji binarnych z  $\varphi$  takimi, że wszystkie relacje przechodnio domykane w  $\varphi$  należą do  $T$ , natomiast pozostałe do  $B$ . Niech  $\Sigma = U \cup B \cup T$ . Dla łatwiejszego szacowania założymy, że  $|\Sigma| \geq 3$ .

Zauważmy, że możliwych 1-typów w modelu jest  $m = 2^{|\Sigma|}$ . Uzupełnimy teraz definicję rozszerzonego 1-typu tak, aby zawierał on wszystkie informacje potrzebne algorytmowi podczas działania:

**Definicja 5.1.** *Pełnym 1-typem* elementu  $v$  modelu  $\mathfrak{M}$  będziemy nazywać zbiór składający się z 1-typu tego elementu oraz następujących krotek:

- $\langle R^+, t \rangle$  takich, że  $R \in T \cup T^{-1}$  oraz dla pewnego  $w \neq v$  o 1-typie  $t$  zachodzi  $\mathfrak{M} \models vR^+w$  oraz nie zachodzi  $\mathfrak{M} \models wR^+v$
- $\langle R, t \rangle$  takich, że  $R \in B \cup T \cup (B \cup T)^{-1}$  oraz dla pewnego  $w$  o 1-typie  $t$  takiego, że  $w$  nie jest z  $v$  w żadnej klice, zachodzi  $\mathfrak{M} \models vRw$

Zauważmy, że aby zapamiętać pełen 1-typ elementu, wystarczy  $2|T| \cdot m + 2(|T| + |B|) \cdot m$  bitów, co można oszacować z góry przez  $4 \cdot |\varphi| \cdot 2^{|\varphi|}$ , czyli wystarcza wykładnicza pamięć. Algorytm będzie pracował na klikach, dlatego zdefiniujemy też typ kliki:

**Definicja 5.2.** Typem  $R^+$ -kliki rozmiaru  $n$  w modelu  $\mathfrak{M}$  nazywamy zbiór składający się z następujących elementów:

- *rozmiar* - rozmiaru kliki
- *relacja* - nazwy relacji  $R$
- *specjalny*  $\in \{0, \dots, n\}$  – wyróżnionego wierzchołka, przy czym jeśli *specjalny* = 0, to żaden wierzchołek nie jest wyróżniony

- funkcji *1typy* :  $\{1, \dots, n\} \rightarrow \{t \mid t \text{ jest pełnym 1-typem}\}$ , zawierającej informacje o pełnych 1-typach wszystkich elementów klikli
- funkcji *2typy* :  $\{1, \dots, n\}^2 \rightarrow \{t \mid t \text{ jest 2-typem}\}$ , zawierającej informacje na temat 2-typów wszystkich par elementów w tej klicie
- funkcji *obietnica* :  $\{-1, 1\} \times \{t \mid t \text{ jest 1-typem}\} \rightarrow \{0, \dots, m + 1\}$  takiej, że jeśli dla pewnych  $b, t$  zachodzi  $obietnica(b, t) > 0$  to istnieje punkt  $w$  nie należący do  $R^+$ -klikli( $v$ ) o 1-typie  $t$  taki, że z  $R^+$ -klikli( $v$ ) istnieje w  $\mathfrak{M}$  ścieżka  $(R^b)^+$ -klikli do  $R^+$ -klikli( $w$ ) o długości co najwyżej  $obietnica(b, t)$

W algorytmie wszystkie funkcje będziemy przechowywać jako zbiory par, zawierające dodatkowo nazwy funkcji. Ponadto, jeśli  $K$  jest typem klikli, to przez  $K.rozmiar$  będziemy oznaczać jej rozmiar,  $K.relacja$  nazwę relacji,  $K.specjalny$  wyróżniony wierzchołek, natomiast przez  $K.1typy$ ,  $K.2typy$  oraz  $K.obietnica$  będziemy oznaczać odpowiednie funkcje z typu  $K$ .

Aby zapamiętać typ klikli rozmiaru co najwyżej  $n$  wystarczy  $\log n + \log |\Sigma| + \log(n + 1) + n \cdot 4 \cdot |\varphi| \cdot 2^{|\varphi|} + n^2 \cdot 2^{|\Sigma|} + (\lceil \log(m + 2) \rceil) \cdot 2m$  bitów pamięci. Dla  $n \leq |\varphi| \cdot 2^{2^{|\Sigma|+5}}$  (a typy tylko tak dużych klikli będziemy budować) daje to wykładniczą ilość potrzebnej pamięci. Wynika z tego, że różnych typów klikli tego rozmiaru może być najwyżej podwójnie wykładniczo, a dokładną liczbę można oszacować z góry przez  $2^{2^{8|\varphi|}}$ .

Nie każdy opis klikli jest sensowny. Powiemy, że typ klikli jest *lokalnie poprawny*, jeśli jest akceptowany przez poniższą funkcję:

**Funkcja *sprawdz – lokalna – poprawność*( $K$ )**

1. Dla każdych  $b, t$  takich, że  $obietnica(b, t) > 0$ , jeśli w klicie pełen 1-typ żadnego elementu nie zawiera pary  $\langle (K.relacja^b)^+, t \rangle$ , to przejdź do stanu odrzucającego *false*.
2. Dla każdych  $R \in T \cup T^{-1}$  oraz 1-typu  $t$  jeśli para  $\langle R, t \rangle$  należy do pełnego 1-typu pewnego elementu, a para  $\langle R^+, t \rangle$  nie należy, to przejdź do stanu *false*.
3. Dla każdych  $b \in \{-1, 1\}$ , 1-typu  $t$  oraz pary elementów  $v, w$ , jeśli para  $\langle (K.relacja^b)^+, t \rangle$  należy do pełnego 1-typu dokładnie jednego z elementów  $w, v$ , to przejdź do stanu *false*.
4. Dla każdego  $i$ , jeżeli któraś para elementów w klicie nie spełnia formuły  $\delta_i$ , to przejdź do stanu *false*.
5. Dla każdego wierzchołka  $v$  o pełnym 1-typie  $t_v$  i każdej pary  $\langle S^+, t_w \rangle$  należącej do jego pełnego 1-typu zgadnij punkt  $w$  o 1-typie  $t_w$  oraz połączenia między  $v$  i  $w$  relacjami z  $B$  tak, aby dla każdego  $i$  zachodziło  $\delta_i(v, w)$  oraz, jeśli formułę  $\delta_i$  można zapisać w postaci  $xS^+y \Rightarrow \gamma(x, y)$ , zachodziło  $\gamma(v, w)$ . Jeśli nie udało się zgadnąć, przejdź do stanu odrzucającego *false*.
6. Dla każdego wierzchołka  $v$  o pełnym 1-typie  $t_v$  i każdej pary  $\langle S, t_w \rangle$  należącej do jego pełnego 1-typu zgadnij punkt  $w$  o typie  $t_w$ , ustal  $vS^b w$  oraz zgadnij połączenia między  $v$  a  $w$  relacjami z  $B$  (z wyjątkiem już ustalonego, jeśli  $S \in B$ ) tak, by dla każdego  $i$  zachodziło  $\delta_i(v, w)$ . Jeśli nie udało się zgadnąć, przejdź do stanu *false*.

Sprawdzenie powyższych własności można wykonać w niedeterministycznym wielomianowym czasie względem rozmiaru klikli.

Instancją klikli typu  $K$  będziemy nazywać taką kliklę, w która jest lokalnie zgodna z typem  $K$ , to znaczy rozmiar, wszystkie o 1-typy i 2-typy w klicie zgadzają się z tym, co jest zapisane w typie klikli. Zauważmy, że każda lokalnie poprawna klikla ma jakąś instancję.

Teraz zdefiniujemy funkcję, która zgaduje lokalnie poprawne typy klik.

**Funkcja *zgadnij – typ – klik*( $R$ )**

1. Zgadnij liczbę  $K.rozmiar \in \{1, \dots, |\varphi| \cdot 2^{2^{|\Sigma|+5}}\}$ .
2. Ustal  $K.relacja = R$ .
3. Zgadnij  $k.specjalny \in \{0, \dots, K.rozmiar\}$ .
4. Dla  $i \in \{1, \dots, K.rozmiar\}$  zgadnij pełen 1-typ  $K.1typy(i)$ .
5. Dla  $i, j \in \{1, \dots, K.rozmiar\}$  zgadnij 2-typ  $K.2typy(i, j)$ .
6. Dla każdych  $b \in \{-1, 1\}$  oraz 1-typu  $t$  zgadnij wartość  $K.obietnica(b, t)$  z zakresu  $0, \dots, 2^{|\Sigma|} + 1$ .
7. Wywołaj *sprawdz – lokalna – poprawnosc*( $K$ ).
8. Zwróć  $K$ .

Powyższa funkcja działa w czasie niedeterministycznym wykładniczym względem rozmiaru sygnatury ze względu na to, że pełne 1-typy mają wykładniczy rozmiar.

Algorytm będzie działał rekurencyjnie. Na początku jednak będzie zawierał procedurę przygotowującą:

**Algorytm A.**

1. Wczytaj formułę  $\varphi = \exists x.\rho(x) \wedge \bigwedge_{i=1}^j \forall xy.\delta_i(x, y) \wedge \bigwedge_{i=1}^k \forall x.\alpha_i(x) \Rightarrow \exists y.\psi_i(x, y)$  języka  $[GF^2+^+]$  w postaci normalnej. Ustal jej sygnaturę  $\Sigma$ , zbiór  $T = \{T_1, \dots\}$  relacji przechodnio domykanych w  $\varphi$  oraz zbiór  $B$  pozostałych relacji binarnych występujących w  $\varphi$ .
2. Niech  $K = \text{zgadnij – typ – klik}(T_1)$ . Jeśli  $K.specjalny \neq 0$ , to przejdź do *false*
3. Zgadnij  $i \in \{1, \dots, K.rozmiar\}$ . Jeśli wierzchołek  $v$  o pełnym 1-typie takim, jak  $K.1typy(i)$  nie spełnia  $\rho(v)$ , przejdź do *false*.
4. Wywołaj *sprawdz – klike*( $K, 1$ ).

Funkcja *sprawdz – klike* zajmuje się realizowaniem zobowiązań zapewnianych przez funkcję *obietnica* oraz wywołaniem funkcji *sprawdz – punkt* dla wszystkich punktów opisywanych w typie klik, z wyjątkiem punktu wyróżnionego.

**Funkcja *sprawdz – klike*( $K, licznik$ )**

1. Jeśli  $licznik > 2^{2^{8|\varphi|}}$  to przejdź do stanu akceptującego *true*.
2. Dla każdych  $S \in T, b \in \{-1, 1\}$  oraz 1-typu  $t$ , dla których  $n = K.obietnica(b, t) > 0$  wykonaj:
  - (a) Zgadnij punkt  $v$  z typu  $K$  taki, że  $\langle (S^b)^+, t \rangle$  należy do pełnego 1-typu  $v$ , a ponadto  $K' = \text{zgadnij – typ – klik}(S)$  oraz  $w \in \{1, \dots, K'.n\}$ .
  - (b) Sprawdź, czy każdy 1-typ  $t'$  należący do  $K'$  jest dopuszczalny dla  $v$ , tzn. czy  $\langle S^+, t' \rangle$  należy do pełnego 1-typu  $v$ . Analogicznie sprawdź, czy dla każdego 1-typu  $t'$  należącego do  $K$  w każdym pełnym 1-typie z  $K'$  znajduje się para  $\langle (S^{-b})^+, t' \rangle$ .

- (c) Sprawdź, czy pełen typ każdego z punktów w  $K'$  po przycięciu do  $(S^b)^+$  jest podzbiorem typu  $v$ . Jeśli nie, przejdź do *false*. Analogicznie sprawdź, czy  $t$  przycięty do  $(S^{-b})^+$  jest podzbiorem każdego z pełnych 1-typów z  $K'$ . Jeśli któraś z tych rzeczy nie zachodzi, przejdź do *false*.
  - (d) Zgadnij połączenia relacjami z  $B$  między  $v$  a  $w$  tak, aby wszystkie formuły  $\delta_i$  były spełnione dla tej pary.
  - (e) Jeśli w  $K'$  nie ma wierzchołka typu  $t$  oraz  $n = 1$  lub  $K'.obietnica(b, t)$  nie jest mniejsza niż  $K.obietnica(b, t)$  to przejdź do *false*.
  - (f) Wywołaj uniwersalnie (tzn. w osobnym wątku) *sprawdz – klike*( $K', licznik + 1$ ).
3. Dla każdego  $v \in \{1, \dots, K.rozmiar\} \setminus \{K.specjalny\}$  wywołaj uniwersalnie *sprawdz – punkt*( $v, K', licznik$ ).

Funkcja *sprawdz – punkt*( $v, K, licznik$ ) będzie budowała otoczenie elementu  $v$  z uwzględnieniem już ustalonej kliki typu  $K$  tak, aby to otoczenie gwarantowało spełnienie wszystkich potrzebnych podformuł  $\varphi$  przez punkt  $v$ .

**Funkcja** *sprawdz – punkt*( $v, K = \{R, \dots\}, licznik$ )

1. Utwórz pusty zbiór *nowe*.
2. Dla każdej  $S \in T \setminus \{R\}$  oblicz  $TYP(S) = zgadnij – typ – klike(S)$  oraz dodaj  $TYP(S)$  do zbioru *nowe*. Jeśli  $TYP(S).1typy(TYP(S).specjalny) \neq K.1typy(v)$  to przejdź do stanu *false*. Ustal również  $TYP(R) = K$ .
3. Dla każdego  $i$  sprawdź, czy formuła  $\alpha_i(v) \Rightarrow \exists y.\psi_i(v, y)$  jest już spełniona. Jeśli nie jest, to zachodzi jeden z przypadków:
  - $\psi_i(x, y)$  można zapisać w postaci  $x = y \wedge \gamma(x, y)$ , przejdź do stanu *false*
  - $\psi_i(x, y)$  można zapisać w postaci  $xSy \wedge \gamma(x, y)$  dla  $S \in B \cup T \cup (B \cup T)^{-1}$ , wtedy zgadnij z pełnego 1-typu  $v$  parę  $\langle S, t \rangle$ ,  $K' = zgadnij – typ – klike(T_1)$  zawierającą punkt  $w$  o 1-typie  $t$  oraz połączenia relacjami z  $B$  między  $v$  a  $w$ . Jeśli nie zachodzi  $vSw \wedge \gamma(v, w)$  to przejdź do stanu *false*. Do zbioru *nowe* dodaj  $K'$ .
  - $\psi_i(x, y)$  można zapisać w postaci  $xS^+y \wedge \gamma(x, y)$  dla  $S \in T \cup T^{-1}$ , wtedy zgadnij 1-typ  $t_w$  punktu  $w$  takiego, że  $\gamma(v, w)$ , i sprawdź, czy  $TYP(S).obietnica(b, t) > 0$ . Jeśli nie, przejdź do *false*.
4. Dla każdego typu  $K' \in nowe$  wywołaj uniwersalnie *sprawdz – klike*( $K', licznik + 1$ ).

### 5.3 Analiza algorytmu

Własność stopu przedstawionego algorytmu jest oczywista ze względu na zmienną *licznik* pilnującą, aby algorytm po ustalonej liczbie tur się zatrzymał. Zauważmy również, że algorytm można tak zaimplementować, by zużywał jedynie wykładniczą ilość pamięci. Opis pojedynczego typu kliki zajmuje jedynie wykładniczo wiele miejsca, łatwo więc zauważyć, że procedura przygotowawcza oraz funkcja *sprawdz – klike* wymagają tylko wykładniczej ilości pamięci. W funkcji *sprawdz – punkt* może się natomiast pojawiać więcej typów klik:  $|T|$  klik zawierających  $v$ , nie więcej niż  $k$  typów klik wynikających z konieczności spełnienia formuł postaci  $xSy \wedge \gamma(x, y)$  oraz najwyżej  $2|\Sigma|m$  typów klik powstających przy realizowaniu obietnic zawartych w funkcji *obietnica*. Na przechowanie wykładniczej liczby informacji wykładniczej długości wystarcza wykładnicza pamięć, zatem algorytm  $A$  jest z klasy AEXSPACE. Pozostaje udowodnić, że algorytm rzeczywiście rozwiązuje badany problem spełnialności.

**Twierdzenie 5.3.** *Algorytm  $A$  akceptuje formułę  $\varphi$  wtedy i tylko wtedy, gdy  $\varphi$  jest spełnialna.*

*Dowód „ $\Leftarrow$ ”.* Jeśli  $\varphi$  jest spełnialna, to z twierdzenia 4.9 wiemy, że ma  $|\varphi| \cdot 2^{2|\Sigma|+5}$ -rozgałęziony model  $\mathfrak{M}$ . Niech  $v$  będzie elementem  $\mathfrak{M}$ , który spełnia  $\rho$  oraz  $\mathfrak{M}'$  będzie spłaszczeniem  $\mathfrak{M}$  o korzeniu  $T_1^+$ -klika( $v$ ). Wystarczy zauważyć, że jeśli algorytm  $A$  będzie zgadywał typy klik zgodnie z tym, jak wyglądają odpowiednie kliki w  $\mathfrak{M}$  w kolejności takiej, że zgadując kolejne typy klik dla pewnego typu kliki  $K$  wybiera te kliki, które są synami instancji  $K$  w  $\mathfrak{M}'$ , to wszystkie warunki sprawdzane przez algorytm będą spełnione, a zatem algorytm nie przejdzie do stanu odrzucającego i w pewnym momencie skończy działanie akceptując formułę.  $\square$

*Dowód „ $\Rightarrow$ ”.* Przypuśćmy, że algorytm zaakceptował formułę  $\varphi$ . Na podstawie przebiegu obliczeń zbudujemy model  $\mathfrak{M}$  formuły  $\varphi$  składający się dokładnie z tych klik, których typy powstawały w czasie działania algorytmu. Zaczniemy od instancji kliki zgadniętej na początku, a następnie będziemy tworzyć rozgałęziony model przez dodawanie kolejnych instancji klik, które tworzy algorytm tak, że w wierzchołki dodane w  $i$ -tym kroku będą miały świadków fragment ścieżki do świadka wśród wierzchołków dodanych w  $i + 1$  kroku. Gdy kliki zaczną się powtarzać, to ustalimy, że model od tego miejsca jest okresowy.

W 1. kroku tworzenia modelu do  $\mathfrak{M}$  dodajemy instancję kliki, której typ jest zgadywany przez algorytm  $A$  na samym początku.

W  $i + 1$ . kroku będziemy zapewniać świadków dla elementów, które zostały dodane w  $i$ -tym kroku. W tym celu weźmy dowolną klikę  $K$  dodaną w  $i$  – tym kroku. Jeśli typ  $K$  tej kliki jest identyczny z jakimś typem dodanym we wcześniejszym kroku, to oznaczamy tę klikę jako okresową i na razie zostawiamy. Jeśli natomiast typ kliki jest widziany pierwszy raz, to dodajemy po  $k$  instancji każdej z klik utworzonych w 2. kroku funkcji *sprawdz* – *klike*( $K, i$ ) i łączymy je z odpowiednim punktem zgodnie z tym, co zgadł algorytm. Następnie dla każdego punktu  $v$  tej kliki, dodajemy po jednej instancji każdej kliki zgadniętej w 2 kroku funkcji *sprawdz* – *punkt*( $v, K, i$ ) w ten sposób, że sklejamy  $v$  z wierzchołkiem tej kliki o tym samym pełnym 1-typie. Na koniec dodajemy po jednej kopii każdej kliki zgadniętej w 3 kroku funkcji *sprawdz* – *punkt*( $v, K, i$ ) i łączmy  $v$  zgodnie z połączeniami zgadniętymi przez algorytm.

Zauważmy, że w ten sposób wykonamy nie więcej niż  $2^{2^{8|\varphi|}}$  kroków, gdyż po  $2^{2^{8|\varphi|}}$  krokach na pewno któryś typ kliki się powtórzy i klika zostanie oznaczona jako okresowa.

*Drzewem klik* będziemy nazywali takie drzewo, w którym etykieta każdego wierzchołka zawiera informację o typie pewnej kliki oraz informację, czy ta klika jest okresowa, natomiast połączenia między klikami są etykietowane informacją, które dwa wierzchołki klik są połączone (albo sklejone) i w jaki sposób.

Kliki dodawane przez nas do modelu we wcześniejszych krokach w naturalny sposób tworzą takie drzewo klik, w którym  $v$  jest synem  $w$ , jeśli klika opisywana przez  $v$  została dodana jako świadek dla  $w$ . W tym drzewie być może niektóre liście odpowiadają klikom okresowym. Dla każdego takiego liścia  $w$  istnieje w drzewie wierzchołek *przodek*( $w$ ), którego etykieta opisuje klikę tego samego typu, ale nie okresową. W kolejnych (być może nieskończenie wielu) krokach w miejsce każdego liścia  $w$  odpowiadającego klice okresowej wstawiamy wierzchołek *przodek*( $w$ ) wraz z całym jego poddrzewem.

Na koniec pozostaje ustalić relacje między tymi punktami  $v, w$ , które leżą w różnych klikach i dla pewnej relacji  $R \in T$  zachodzi  $vR^+w$ . Zauważmy, że w takiej sytuacji  $\langle R^+, v \rangle$  należy do pełnego 1-typu  $w$ , gdyż algorytm dba o to, by w kolejnych klikach nie pojawiały się elementy, które nie należą do pełnych 1-typów elementów z wcześniejszych klik. Ponadto klika zawierająca  $w$  jest lokalnie poprawna, co w szczególności oznacza, że można połączyć  $v$  z  $w$  tak, by te punkty spełniały wszystkie formuły  $\delta_i$ . Z lokalnej poprawności klik wynika również, że wszystkie pary

elementów w klikach spełniają formuły  $\delta_i$ .

Otrzymany model oczywiście spełnia formułę  $\exists x.\rho x$ , gdyż jawnie w 1. kroku dodaliśmy punkt, który ją spełnia. Model spełnia również wszystkie formuły  $\delta_i$  dla każdej pary połączonych (pośrednio lub bezpośrednio) wierzchołków. Ponadto każdemu wierzchołkowi jawnie zapewniliśmy wszystkich potrzebnych świadków, zatem otrzymany model jest modelem formuły  $\varphi$ .  $\square$

Algorytm działa dla formuł w postaci normalnej. Jeśli formuła nie jest w postaci normalnej, to w wykładniczym czasie można ją przekształcić do zbioru formuł w postaci normalnej, a następnie niedeterministycznie zgadnąć jeden element tego zbioru i dla niego przeprowadzić resztę obliczeń. Te dodatkowe czynności wstępne nie zwiększają złożoności pamięciowej ani czasowej algorytmu.

Dolna granica wynika z 2EXPTIME-trudności problemu spełnialności dla  $[GF^2 + TRANS]$ , zaprezentowanego w [14] – można bowiem zastąpić relacje przechodnie przez przechodnie domknięcia pewnych relacji nie występujących w innych miejscach formuły.

**Wniosek 5.4.** *Problem spełnialności dla logiki  $[GF^2+^+]$  jest 2EXPTIME-zupełny.*

## 6 Podsumowanie

Pokazaliśmy, że przełożenie logiki PDL na pewne rozszerzenie logiki ze strażnikami z równością jest trudne – nawet po ograniczeniu do fragmentu strzeżonego dodanie operacji składania relacji w mocno ograniczonym zakresie prowadzi natychmiast do nierozstrzygalności problemu spełnialności. Pozytywnym wynikiem jest rozstrzygalność  $GF^2$  ze przechodnim domknięciem. W bardzo podobny sposób można pokazać, że jeśli będziemy rozpatrywać zwrotne i przechodnie domknięcie, to problem spełnialności pozostanie 2EXPTIME-zupełny.

Otrzymany wynik uzupełnia wiedzę o wpływie przechodniości na rozstrzygalność logiki ze strażnikami. Ciekawym pytaniem otwartym jest co się stanie, jeśli do  $GF^2$  bez równości dodamy możliwość stosowania zarówno przechodniego domknięcia, jak i złożenia, w dodatku pozwalając na dowolne ich kombinacje w strażnikach. Ze względów praktycznych warto również się zastanowić, ile trzeba zabrać z logiki z przechodnio domkniętymi strażnikami, aby problem spełnialności stał się wykładniczy. Otwartym problemem pozostaje również pytanie, czy problem spełnialności  $GF^2$  ze złożeniem jest rozstrzygalny, jeśli pozwolimy na składanie ze sobą jedynie dwóch relacji.

## Literatura

- [1] H. Andreka, I. Nemeti, J. van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217-274, 1998.
- [2] S. Arora, R. Fagin. On winning strategies in Ehrenfeucht–Fraiss’e games. *Theoretical Computer Science*, 174(1-2):97–121, 1997.
- [3] J. van Benthem. Modal Correspondence Theory, *dissertation, Mathematical Institute, University of Amsterdam*, 1976.
- [4] J. van Benthem. ‘Correspondence Theory’. In *D. Gabbay, F. Guenther, Handbook of Philosophical Logic, vol. II*, Reidel, Dordrecht, 1984.
- [5] E. Berger, E. Grädel, Y. Gurevich. The classical decision problem. *Springer-Verlag*, 1996.
- [6] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer. Alternation. *Journal of the ACM* 28, 114-133, 1981.



- [7] M. J. Fischer, R. E. Ladner. Propositional modal logic of programs. In *Proceedings of the Ninth Annual ACM Symposium on theory of Computing (Boulder, Colorado, United States, May 04 - 04, 1977)*. *STOC '77*. ACM, New York, NY, 286-294, 1977.
- [8] H. Ganzinger, C. Meyer, and M. Veanes. The two-variable guarded fragment with transitive relations. In *Proc. LICS'99. IEEE Computer Soc. Press*, 1999.
- [9] E. Grädel, P. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, Mar, 1997.
- [10] E. Grädel, I. Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99*, Trento, 1999.
- [11] E. Grädel, M. Otto, E. Rosen. Undecidability results on two-variable logics. In *Proceedings of 14th Symposium on Theoretical Aspects of Computer Science STACS'97, Lecture Notes in Computer Science Nr. 1200, Springer*, pp. 249–260, 1997.
- [12] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64(4):1719-1742, 1999.
- [13] Y. Kazakov. Saturation-Based Decision Procedures for Extensions of the Guarded Fragment. *PhD thesis*, Universität des Saarlandes, Saarbrücken, Germany, March 2006.
- [14] E. Kieroński, The two-variable guarded fragment with transitive guards is 2EXPTIME-Hard. In *Proc. of 6th International Conference on Foundations of Software Science and Computational Structures (FOSSACS), Lecture Notes in Computer Science*, vol. 2620, pp. 299-312, 2003.
- [15] E. Kieroński, Results on the guarded fragment with equivalence or transitive relations. In *Proc. of 14th Annual Conference, Computer Science Logic (CSL)*, pp. 309-324, 2005.
- [16] V. R. Pratt 1978. A practical decision method for propositional dynamic logic (Preliminary Report). In *Proceedings of the Tenth Annual ACM Symposium on theory of Computing (San Diego, California, United States, May 01 - 03, 1978)*. *STOC '78*. ACM, New York, NY, 326-337, 1978.
- [17] D. Scott. A decision method for validity of sentences in two variables. *J. Symb. Logic* 27, 477, 1962.
- [18] W. Szwaast, L. Tendera. The guarded fragment with transitive guards. *Annals of Pure and Applied Logic*, 128, 227-276, 2004.