

mgr inż. Artur Basiura  
Wydział Elektrotechniki, Automatyki,  
Informatyki i Inżynierii Biomedycznej  
Akademia Górniczo-Hutnicza im. S. Staszica w Krakowie

Szanowna Pani  
prof. dr hab. Ewa Grabska  
Zakład Projektowania i Grafiki Komputerowej  
Wydział Fizyki, Astronomii i Informatyki Stosowanej  
Uniwersytet Jagielloński

## Odpowiedź na recenzję rozprawy doktorskiej

Szanowna Pani Profesor, bardzo dziękuję za wnikliwą ocenę merytoryczną i redakcyjną mojej pracy doktorskiej, pozytywne opinie, jak również zawarte uwagi krytyczne. Stanowią one istotne wskazówki, służące poprawie jakości realizowanych przeze mnie przyszłych prac badawczych. Poniżej zamieściłem odpowiedzi na postawione przez Panią pytania oraz odniosłem się do uwag zamieszczonych w recenzji.

**UWAGA 1.** *W podstawowej definicji grafu bazowego będącego systemem*  
 $BG = (V, E, \Sigma, \Gamma, \text{typ}, \text{atr}, \text{val}, A, AT)$

*1.1 .Dla poszczególnych krawędzi nie są wyznaczone pary wierzchołków*  
*Prawidłowa definicja zbioru krawędzi powinna wyglądać następująco:*  
 *$E$  – jest skończonym niepustym zbiorem krawędzi, gdzie  $E \subset V \times V$*

*1.2 funkcja  $\text{atr}$  przyporządkowuje typom poszczególnych wierzchołków i krawędzi, zbiory*  
*typów atrybutów. W takim razie przeciwdziedzina tej funkcji powinna być rodziną wszystkich*  
*podzbiorów sumy typów atrybutów tzn.  $\mathcal{P}(AT^\Sigma \cup AT^\Gamma)$*

Odpowiedź:

Zgadzam się z uwagami Recenzenta. W pierwszym przypadku powinno nastąpić doprecyzowanie definicji krawędzi :  $E \subset P_2(V)$  (dwuelementowe podzbiory  $V$ ), natomiast w przypadku funkcji  $\text{atr}$  rzeczywiście chodzi o odwzorowanie  $\text{atr}: \Sigma \cup \Gamma \rightarrow \mathcal{P}(AT^\Sigma \cup AT^\Gamma)$

**UWAGA 2.** Niezrozumiałe jest dla mnie stwierdzenie że grafy IAG oraz KAG są rozszerzeniem grafu BG

Odpowiedź:

Niefortunnie zostało użyte stwierdzenie „rozszerzenie”, w tym przypadku chodziło o analogie do programowania obiektowego i polimorfizmu. Grafy IAG i KAG dziedziczą wszystkie właściwości grafu BG, dodatkowo w stosunku do grafu BG definiują nowe typy wierzchołków, krawędzi i tym samym nowe właściwości. Z tego powodu użyto stwierdzenia „rozszerzają”. Jednak z punktu widzenia algorytmicznego, każdy zestaw operacji jakie można przeprowadzić na grafach BG, można przeprowadzić również na grafach IAG i KAG. Grafy IAG i KAG są więc grafami BG.

**UWAGA 3.** W definicji 4 zbiór grafów bazowych jest oznaczony w ten sam sposób jak graf bazowy z definicji 1. Jest to niedopuszczalna kolizja oznaczeń

Odpowiedź:

Zgadzam się z Recenzentem w Definicji 4, BG jest oznaczany jako zbiór grafów bazowych, natomiast Definicja 1 wprowadza oznaczenie BG jako grafu bazowego. Jest to błąd i w Definicji 4 powinien zostać poprawiony.

**UWAGA 4.** W definicji 4 zbiór reguł zmian struktury  $R$  powinien być zdefiniowany przed zbiorem reguł transformacyjnych  $\Pi$

Odpowiedź:

Zgadzam się z uwagą Recenzenta, wcześniejsze wprowadzenie pojęcia dotyczącego zbioru reguł  $R$  w definiensie, niewątpliwie uporządkowuje definicje. Aktualnie jest ona niepełna. Najpierw powinien zostać wprowadzony zbiór reguł zmiany struktury grafów. Dopiero w następnym kroku zdefiniowany zbiór reguł transformacji, który wykorzystuje wcześniej zdefiniowane pojęcia. Przesyłam poprawioną definicję, rzeczywiście jest ona teraz bardziej czytelna:

Definicja 4. Regułami transformacji grafowych  $\Phi$ , nad zbiorem grafów BG nazywamy trójkę postaci:

$$\Phi = (G, \Pi, R),$$

gdzie:

- $G$  – graf początkowy podlegający transformacji należący do zbioru grafów bazowych BG
- $R$  - zbiór reguł zmiany struktury grafów
- $\Pi$  - zbiór reguł transformacyjnych postaci:

$$f: BG^n \times (R \cup \emptyset) \rightarrow BG^k, \text{ gdzie } n, k > 0$$

**UWAGA 5.** Każda definicja składa się z wyrażenia definiowanego (definiendum) oraz wyrażenia definiującego (definiens). W definicji 5 definiendum jest kontekst działania algorytmu  $K_g$ , natomiast czwórka  $(\Phi, F_c, F_w, H)$  to definiens. W def. 5  $K_g$  nie powinno wystąpić również w roli definiens

Odpowiedź:

Zgadzam się z uwagą Recenzenta. W definiendum nie powinno występować odwołanie do kontekstu działania algorytmu  $K_g$ . Prawidłowa definicja to

Definicja 5. Kontekst działania algorytmu  $K_g$  definiujemy jako czwórkę postaci:

$$K_g = (\Phi, F_c, F_w, H)$$

gdzie:

$\Phi$  – reguły transformacji grafowych (zob. Definicja 4)

$F_c$  – Funkcja celu

$F_w$  – Funkcja sprawdzająca poprawność rozwiązania

$H$  – Zbiór heurystyk wykorzystywanych do przeszukiwania zbioru

**UWAGA 6.** Autor powinien rozważyć atrybuowane gramatyki grafowe z diagramem sterującym, w których produkcje są definiowane wraz z predykatami stosowności i porównać efektywność tego formalizmu w zagadnieniach projektowych z zaproponowanym przez siebie formalizmem zawierającym grafy IAG i KAG oraz reguły transformacji

Odpowiedź:

Atrybuowane gramatyki grafowe z diagramami sterującymi są mechanizmem szeroko wykorzystywanym w naszym zespole naukowym, szczególnie w kontekście przetwarzania grafów w środowisku rozproszonym. Można tu zacytować szereg prac począwszy od rozprawy habilitacyjnej prof. Kotulskiego [1], poprzez rozprawę habilitacyjną promotora dr hab. Adama Sędziwego i szereg artykułów innych pracowników Katedry Informatyki Stosowanej AGH [2, 3]. Należy tu też zaznaczyć, że w rozwój tej teorii uczestniczyła dr hab. Barbara Strug z Uniwersytetu Jagiellońskiego [4, 5, 6, 7].

W rozprawie doktorskiej diagramy sterujące zapewniają wysoką efektywność finalnego przetwarzania - chociaż ich wykorzystanie jest ukryte jedynie w komentarzu na stronie 62:

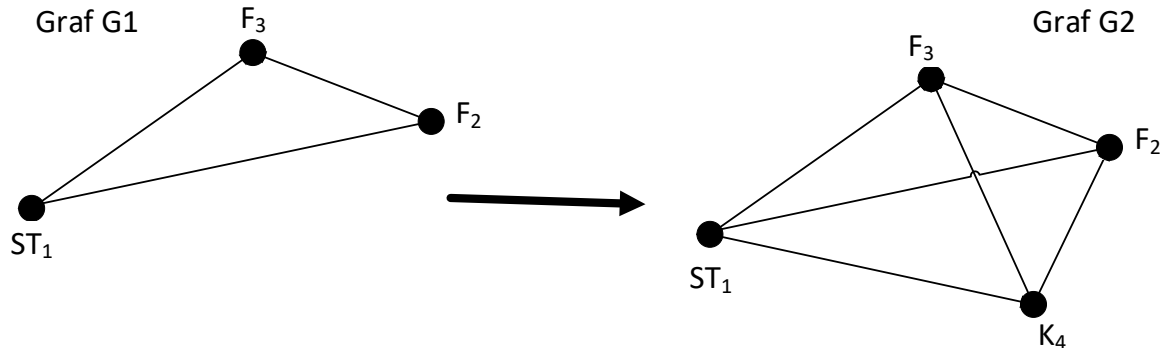
*„Należy zauważyć, że chociaż powyższe transformacje są operacjami stosunkowo złożonymi, jeśli porównać je z „typowymi” produkcjami gramatyk grafowych [35, 53, 54, 55], to mogą być one zrealizowane jako sekwencje takich właśnie, atomowych operacji (produkcji).”*

Nie rozpisywałem jednak złożonych produkcji na sterowanymi diagramem sterującym (zależnymi od kontekstu) sekwencjami produkcji elementarnych, gdyż uznałem, że jest to kwestia implementacyjna, dla której udowodniono już użyteczność notacji.

Diagram sterowania może być uzależniony od omawianego w pracy kontekstu przetwarzania  $K_g$ , czy też analizując same operacje może być składnikiem reguły R.

Poniżej pokazano przykład grafu sterowania, dla nowo tworzonego grafu KAG, dotyczącego możliwej konfiguracji dla złączonych obszarów dwóch segmentów. Graf omawiany poniżej omawiany był w pracy w rozdziale 5.3 i jego fragment zilustrowany jest na Rysunku 28. Początkowo graf przechowuje informacje o możliwej przestrzeni instalacji dla słupa  $ST_1$  (o indeksie 77, czyli  $I(s) = 77$ ). Zakłada możliwość instalacji opraw o typach  $F_1, F_2$  jak zostało to zaprezentowane na Rysunku 1 (zaprezentowanym poniżej) i grafie G1. W wyniku operacji optymalizacji segmentu i punktu świetlnego do którego odnosi się graf KAG, został wyznaczony kąt instalacji  $K_4$  (o indeksie 194, czyli  $I(k) = 194$ ) dla obu opraw ( $F_1, F_2$ ). Konieczne jest wykonanie operacji dodania nowego wierzchołka do grafu G1. Nowy graf został przedstawiony na Rysunku 1 i grafie G2. Produkcja ta jest produkcją

złożoną, a w szczególności trudno jest oszacować jej złożoność, oraz musiałaby być zmodyfikowana gdyby typów opraw było więcej niż dwa. W oparciu o diagramy sterujące można rozwiązać oba te problemy.



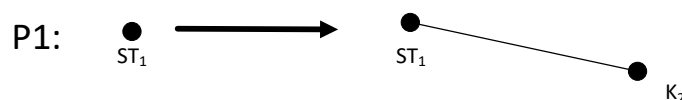
Rysunek 1. Wzbogacenie grafu G1 o nową konfigurację i utworzenie grafu G2

Możliwe jest zastosowanie na grafie G1, serii produkcji sterowanych diagramem sterowania DCG<sub>1</sub>, mających na celu utworzenie grafu G2.

Tabela 1. Graf sterowania DCG<sub>1</sub> – predykat i tranzycja

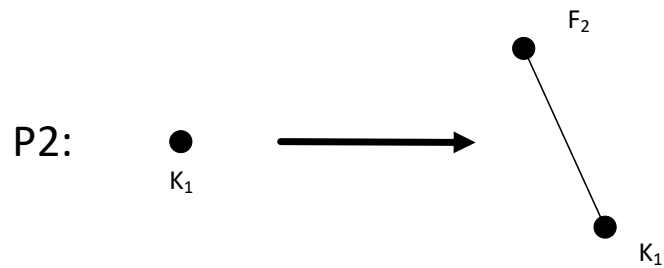
Stan	Wait	Predykat [ $\pi$ ]	Tranzycja [T]
N1	$\exists Vi \in RS$	$\Pi1: \text{command}(Vi) = \text{addNewAngle};$ $\exists s \in V: \text{typ}(s) = ST, I(s) = 77$	$T1=(N1, N2, P1(s), RS-\{Vi\})$
N2	true	$\Pi2: \exists s, k \in V: I(s)=77, I(k)=194: (\exists f \in V:$ $(s, f) \in E \text{ and } (f, k) \notin E)$	$T2=(N2, N2, P2(s, k), RS)$
N2	true	not $\Pi2$ :	$T3=(N2, N1, \emptyset, RS)$

W diagramie DCG<sub>1</sub> użyte są dwie produkcje P1 i P2. P1 jest produkcją zaprezentowaną na Rysunku 2. Produkcja ta wyzwalana jest przez predykat:  $\exists s \in V: \text{typ}(s) = ST, I(s) = 77$ . Czyli dla wierzchołka typu słup (ST) o indeksie 77. Celem produkcji jest utworzenie krawędzi łączącej wierzchołek typu ST (słup), z nowo utworzonym wierzchołkiem typu K (Kąt montażu).



Rysunek 2. Produkcja P1

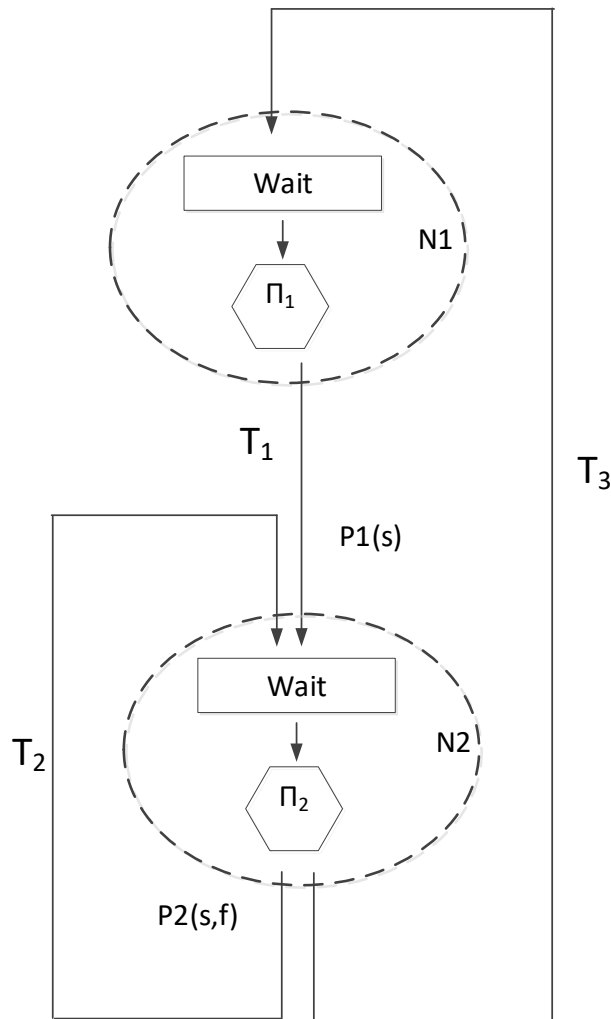
Druga produkcja jest zaprezentowana na Rysunku 3.



Rysunek 3. Produkcja P2

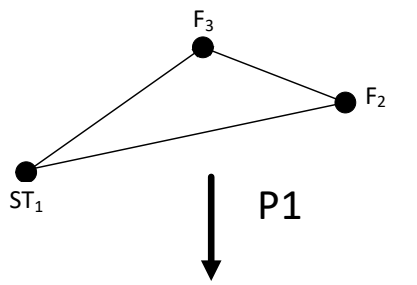
Produkcja ta wyzwalana jest przez predykat  $\Pi_2: \exists s, k \in V: l(s)=77, l(k)=194 : (\exists f \in V: (s, f) \in E \text{ and } (f, k) \notin E)$ .  
 W wyniku działania produkcji tworzona jest krawędź pomiędzy wierzchołkiem definiującym kąt montażu i oprawę.

Pełny Graf sterowania został zaprezentowany na Rysunku 4.

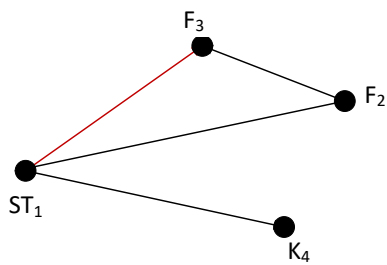


Rysunek 4. Graf sterowania  $DCG_1$

Przykładowo operacje zaprezentowaną na Rysunku 1, możemy wyrazić w postaci kilku kroków pokazanych na Rysunku 5.

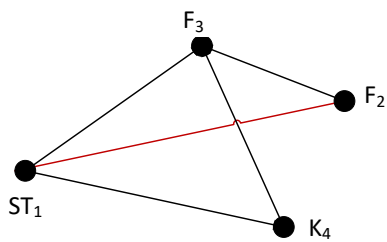


↓ P1

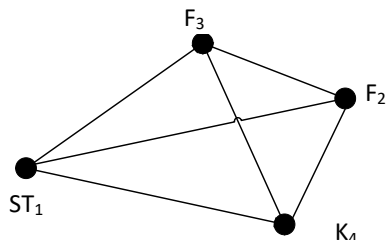


Kolorem czerwonym zaznaczono istnienie szukanej krawędzi (określonej w predykcji  $\Pi_2$ )

↓ P2(s,k)



↓ P2(s,k)



Rysunek 5. Przykład utworzenia grafu  $G_2$  jako sekwencja produkcji diagramu sterowania  $DCG_1$  na grafie  $G_1$

Jeżeli w grafie istniałoby więcej wierzchołków reprezentujących typy lamp, to automatycznie zostałyby utworzone krawędzie modelujące relacje pomiędzy nimi, a nowym wierzchołkiem definiujący kąt montażu.

***UWAGA 7.*** Edycja rozprawy doktorskiej nie jest staranna np. Autor często odwołuje się do nieistniejącego rozdziału 0

Odpowiedź:

Zgadzam się z Recenzentem, w pracy istotnie pojawiły się błędy redakcyjne czego przykładem są wspomniane odwołania do rozdziału 0, które w rzeczywistości dotyczą rozdziału 10.

Pozdrawiam i łączę wyrazy szacunku,

Artur Basiura

**Bibliografia:**

- [1] Kotulski L., "Rozproszone transformacje grafowe: teoria i zastosowania", Wyd. Naukowe AGH, Kraków 2013, ISBN 987-83-7464-630-7, 2013
- [2] Kotulski L., Sędziwy A., "GRADIS - The multiagent environment supported by graph transformations. Simulation Modelling Practice and Theory", Vol.18, Issue 10, pp. 1515-1525, 2010
- [3] Kotulski L., Sędziwy A., Strug B., "Conditional Synchronization in Multi-agent Graph-based Knowledge System", ICCS 2015, 1043-1051, 2015
- [4] Kotulski L., Strug B., "Multi-agent System for Distributed Adaptive Design", Key Engineering Materials, vol. 486, pp. 217-220, 2011
- [5] Kotulski L., Strug B., "Supporting communication and cooperation in distributed representation for adaptive design", Advanced Engineering Informatics 27, pp. 220-229, 2013
- [6] Kotulski L., Strug B., "Using Graph Transformations in Distributed Adaptive Design System," LNCS vol. 5337, pp. 477-486, 2008
- [7] Kotulski L., Strug B., "Distributed Adaptive Design with Hierarchical Autonomous Graph Transformation Systems", LNCS vol. 4488 pp. 870-877, 2007
- [8] Kotulski L., Sędziwy A., "Derivation Control Environment As A Tool For An Efficient Distributed Graph Transformations Coordination", Conference: Parallel and Distributed Computing, IEEE Xplore, 2008