



# Obwody elektryczne: Kategorie hipergraphów i operady

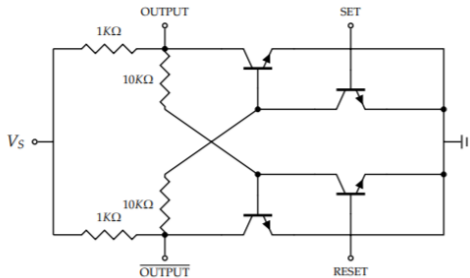
## Wstęp

Radosław Ćwikowski  
Marcin Caputa

7.02.2020

Obwody elektryczne, sieci reakcji chemicznych, automaty skończone, procesy Markowa: są to wszystkie powszechnie opisywane modele systemów fizycznych lub obliczeniowych za pomocą diagramów sieciowych. Tutaj na przykład rysujemy schemat, który modeluje flip-flop, obwód elektryczny - ważny w pamięci komputera - który może przechowywać trochę informacji.

# Przykładowy obraz



Diagramy sieciowe mają sprawdzone narzędzie. W tym rozdziale jesteśmy zainteresowani zrozumieniem wspólnej struktury matematycznej, którą dzielą się na, do celów tłumaczenie między nimi i ich zjednoczenie; na przykład niektóre rodzaje procesów Markowa mogą być symulowane, a zatem rozwiązywane za pomocą obwodu rezystorów. Kiedy zrozumiemy podstawowe struktury, które są wspólne dla języków diagramów sieciowych, możemy wykonać łatwe porównywanie odpowiednich modeli matematycznych. Na pierwszy rzut oka schematy sieciowe wyglądają zupełnie inaczej niż schematy okablowania, które do tej pory widzieliśmy. Na przykład druty nie są przekierowywane w powyższym przypadku, podczas gdy w kategorii - w tym kategoriach monoidalnych widocznych w teoriach zasobów lub współprojektowaniu - każdy morfizm ma domenę i kodomenę, co daje mu poczucie kierunku.

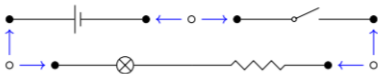
# Przykładowe elementy obwodów elektrycznych



# Przykładowe elementy obwodów elektrycznych



# Przykładowe elementy obwodów elektrycznych





Konstrukcje uniwersalne są kluczowe dla teorii kategorii. Pozwalają nam definiować obiekty, przynajmniej do izomorfizmu, opisując ich związek z innymi przedmiotami. W tej sekcji naszym głównym zadaniem jest konkretne zrozumienie kolimitów w kategorii  $\mathbf{Fin}$  Zestaw zbiorów skończonych i funkcji. Chodzi o to, aby wziąć kilka zestawów - powiedzmy dwa, piętnaście lub zero - użyć funkcji między nimi, aby wyznaczyć te elementy jeden zestaw „należy traktować tak samo” jak elementy w innym zestawie, a następnie scalić odpowiednio się zestawia.

Wypychacze to sposób łączenia zestawów. Podobnie jak połączenie podzbiorów, wypychanie może się łączyć dwa zestawy w nierozłączny sposób: elementy jednego zestawu można identyfikować z elementami innymi. Jednak konstrukcja wypychająca jest znacznie bardziej ogólna: pozwala (i wymaga) użytkownikowi dokładnie określić, które elementy zostaną zidentyfikowane.

# Pushouts - Wypychacze

**Definition 6.19.** Let  $\mathcal{C}$  be a category and let  $f: A \rightarrow X$  and  $g: A \rightarrow Y$  be morphisms in  $\mathcal{C}$  that have a common domain. The *pushout*  $X +_A Y$  is the colimit of the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & & \downarrow \\ Y & & \end{array}$$

In more detail, a pushout consists of (i) an object  $X +_A Y$  and (ii) morphisms  $\iota_X: X \rightarrow X +_A Y$  and  $\iota_Y: Y \rightarrow X +_A Y$  satisfying (a) and (b) below.

(a) The diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & \tau & \downarrow \iota_X \\ Y & \xrightarrow{\iota_Y} & X +_A Y \end{array} \quad (6.20)$$

commutes. (We will explain the " $\tau$ " symbol below.)

(b) For all objects  $T$  and morphisms  $x: X \rightarrow T$ ,  $y: Y \rightarrow T$ , if the diagram

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & & \downarrow x \\ Y & \xrightarrow{g} & T \end{array}$$

commutes, then there exists a unique morphism  $t: X +_A Y \rightarrow T$  such that

$$\begin{array}{ccc} A & \xrightarrow{f} & X \\ g \downarrow & & \downarrow \iota_X \\ Y & \xrightarrow{\iota_Y} & X +_A Y \end{array} \begin{array}{c} \xrightarrow{x} \\ \downarrow \\ \xrightarrow{t} \end{array} \begin{array}{c} X \\ X +_A Y \\ T \end{array} \quad (6.21)$$

commutes.

Mówimy, że kategoria  $C$  ma skończone limity, jeśli istnieje colimit,  $\text{colim}JD$ . ilekroć  $J$  jest kategorią skończoną, a  $D: J \rightarrow C$  jest diagramem

## Przykład 6.31.

Obiekt początkowy w kategorii  $C$ , jeśli istnieje, jest kolimerem funktora  $! : 0 \rightarrow C$ , gdzie  $0$  to kategoria bez obiektów i morfizmów,  $!$  jest wyjątkowy taki funktor. Rzeczywiście, kokon się skończył! jest tylko przedmiotem  $C$ , więc początkowy kokon się skończył! jest tylko początkowym przedmiotem  $C$ . Zauważ, że  $0$  ma skończenie wiele obiektów (brak); dlatego początkowe obiekty są skończonymi kolimitami.

## Przykład 6.31.

Często chcemy wiedzieć, że kategoria  $C$  ma wszystkie skończone granice (w takim przypadku często porzucają „wszystko” i po prostu mówią „ $C$  ma skończone granice”). Aby sprawdzić, czy  $C$  ma (wszystkie) skończone colimits, wystarczy sprawdzić, czy ma kilka prostszych form colimit, które generują wszystko reszta.

Twierdzenie 6.32. Niech  $C$  będzie kategorią. Następujące są równoważne: 1.  $C$  ma wszystkie skończone limity. 2.  $C$  ma obiekt początkowy i wszystkie wypychania. 3.  $C$  ma wszystkie korektory i wszystkie skończone koprodukty. Dowód. Nie podamy tutaj dokładnych szczegółów, ale kluczowy pomysł jest indukcyjny: jeden Potrafi budować dowolne skończone diagramy przy użyciu podstawowych bloków konstrukcyjnych.

Gdy kategoria  $C$  ma skończone limity, niezwykle przydatnym sposobem na ich spakowanie jest biorąc pod uwagę kategorię cospans w  $C$ .

## Przykład 6.43.

Niech  $C$  będzie kategorią. Cospan w  $C$  to tylko para morfizmów do a wspólny obiekt  $A \rightarrow N \leftarrow B$ . Wspólny obiekt  $N$  nazywany jest wierzchołkiem cospanu a pozostałe dwa obiekty  $A$  i  $B$  nazywane są jego stopami



Jeśli chcemy powiedzieć, że cospans tworzą kategorię, powinniśmy zacząć od tego, jak to zrobić kompozycja działałaby. Załóżmy, że

If we want to say that cospans form a category, we should begin by saying how composition would work. So suppose we have two cospans in  $\mathcal{C}$

$$A \xrightarrow{f} N \xleftarrow{g} B \quad \text{and} \quad B \xrightarrow{h} P \xleftarrow{k} C$$

Since the right foot of the first is equal to the left foot of the second, we might stick them together into a diagram like this:

$$A \xrightarrow{f} N \xleftarrow{g} B \xrightarrow{h} P \xleftarrow{k} C$$

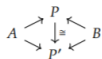
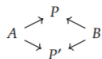
Then, if a pushout of  $N \xleftarrow{g} B \xrightarrow{h} P$  exists in  $\mathcal{C}$ , as shown on the left, we can extract a new cospan in  $\mathcal{C}$ , as shown on the right:

$$\begin{array}{ccc}
 & N & P \\
 & \swarrow \scriptstyle{g} & \nwarrow \scriptstyle{h} \\
 A & \xrightarrow{f} & N \xrightarrow{in} N +_B P \xleftarrow{out} P \\
 & \swarrow \scriptstyle{g} & \nwarrow \scriptstyle{h} \\
 & B & P \\
 & \xrightarrow{h} & \\
 & & C
 \end{array}
 \quad \rightsquigarrow \quad
 \begin{array}{ccc}
 & N +_B P & \\
 & \swarrow \scriptstyle{f \circ in} & \nwarrow \scriptstyle{out} \\
 A & \xrightarrow{f \circ in} & N +_B P \xleftarrow{out} P \\
 & \swarrow \scriptstyle{g} & \nwarrow \scriptstyle{h} \\
 & B & P \\
 & \xrightarrow{h} & \\
 & & C
 \end{array}
 \tag{6.44}$$

mamy dwóch cospansów w  $\mathcal{C}$ .

# Using Directed Acyclic Word Graphs

Może się wydawać, że osiągnęliśmy cel, ale brakuje nam kilku rzeczy. Najpierw, potrzebujesz tożsamości na każdym obiekcie  $C \in \text{Ob } C$ ; ale to nie jest trudne: użyj  $C \rightarrow C \leftarrow C$  gdzie obie mapy są tożsamościami w  $C$ . Co ważniejsze, nie wiemy, że  $C$  ma wszystkie wypychania, więc nie wiemy, że co dwa sekwencyjne morfizmy  $A \rightarrow B \rightarrow C$  można skomponować. Poza tym istnieje stan techniczny, że kiedy tworzymy wypychacze, my tylko uzyskać odpowiedź „aż do izomorfizmu”: wszystko, co jest izomorficzne w przypadku wypychania, liczy się jako wypychanie (sprawdź definicję, aby zobaczyć dlaczego). Chcemy, aby te wszystkie wybory się liczyły tak samo, więc definiujemy dwa cospansy jako równoważne, jeśli występuje izomorfizm między ich odpowiednimi wierzchołkami. Oznacza to, że cospan  $A \rightarrow P \leftarrow B$  i  $A \rightarrow P' \leftarrow B$  na schemacie pokazanym po lewej poniżej są równoważne iff występuje izomorfizm  $P = P'$ , co powoduje, że schemat jest po prawej stronie

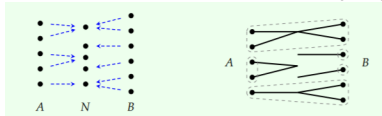


Teraz gdzieś idziemy. Tak długo, jak nasza kategoria  $C$  ma wypychacze:  $\text{Cospan}C$  utworzy kategorię. Ale tak naprawdę jesteśmy bardzo blisko zdobycia więcej. Jeśli domagamy się również, aby  $C$  również miał obiekt początkowy - możemy zaktualizować  $\text{Cospan}C$  do symetrycznej kategorii monoidalnej

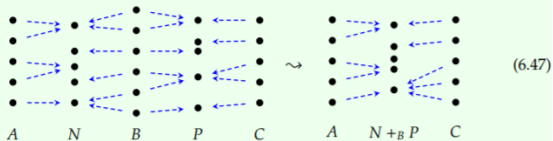
Niech  $C$  będzie kategorią ze skończonymi kolimitami. Następnie istnieje kategoria  $\text{Cospan}C$  z tymi samymi obiektami co  $C$ , tj.  $\text{Ob}(\text{Cospan}C) = \text{Ob}(C)$ , gdzie morfizmy  $A \rightarrow B$  to (klasy równoważności) cospans od  $A$  do  $B$ , i podano skład przez powyższą konstrukcję wypychającą. Ta kategoria ma symetryczną strukturę monoidalną, oznaczoną  $(\text{Cospan}C, \otimes, +)$ . Jednostką monoidalną jest obiekt początkowy  $- \in C$ , a iloczyn monoidalny podaje koprodukt

# Finding maximal cliques in a graph

Kategoria  $\mathbf{FinSet}$  ma skończone limity. Możemy więc zdefiniować symetryczną monoidalną kategorię  $\mathbf{CospanFinSet}$ . Jak to wygląda? Wygląda bardzo podobnie przewody łączące porty. Obiekty  $\mathbf{CospanFinSet}$  są zbiorami skończonymi; narysujmy je jako kolekcje  $\bullet$ 's. Morfizmy są cospanami funkcji. Niech  $A$  i  $N$  będą pięcioelementowymi zestawami, a  $B$  będzie a zestaw sześciu elementów. Poniżej znajdują się dwa obrazy cospanu  $A \rightarrow N \leftarrow B$



Na ilustracji po lewej stronie przedstawiamy po prostu funkcje  $f$  i  $1$  za pomocą rysunku strzałki od każdego  $a \in A$  do  $f(a)$  i każdego  $b \in B$  do  $1(b)$ . Na zdjęciu po prawej stronie my spraw, aby ten obraz bardziej przypominał druty, po prostu narysuj drut tam, gdzie przed nami miał strzałkę i usuń niepotrzebne punkty środkowe. Rysujemy również linię kropkowaną wokół punktów, które są połączone, aby podkreślić ważną perspektywę, że cospans ustalić, że niektóre porty są połączone, tj. należą do tej samej klasy równoważności. Monoidalna kategoria  $\text{CospanFinSet}$  zapewnia następnie dwie operacje łączenia cospans: skład i produkt monoidalny. Skład podaje się, biorąc wypychanie map pochodzących ze wspólnej stopy, jak opisano w definicji 6.45. Oto przykład kompozycji cospan, na której przedstawione są wszystkie funkcje

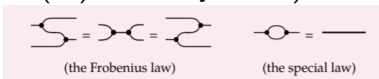


The monoidal product is given simply by the disjoint union of two cospans; in pictures it is simply combining two cospans by stacking one above another.

Kategoria hipergraph jest rodzajem symetrycznej kategorii monoidalnej, której schematy połączeń są sieciami. Wkrótce przekonamy się, że obwody elektryczne można zorganizować w kategorię hypergraph; właśnie nad tym budowaliśmy. Ale aby zdefiniować hipergraph w kategoriach, warto najpierw wprowadzić monoidy Frobenius



Niech  $X$  będzie obiektem w symetrycznej kategorii monoidalnej  $(\mathcal{C}, \otimes, I)$ . Struktura Frobeniusa na  $X$  składa się z 4-krotnej  $(\mu, \eta, \nu, \epsilon)$  takiej, że  $(X, \mu, \eta)$  jest przemianą monoid i  $(X, \nu, \epsilon)$  to komonoid współsprawczy, który spełnia sześć równań powyżej ((ko)asocjatywności, (ko)jedności, (ko)komutatywności), a także



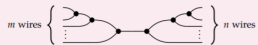
The diagram shows two equations. The first, labeled '(the Frobenius law)', shows three equivalent ways to connect two strands: a braid-like structure, a structure with a dot on the top strand, and a structure with a dot on the bottom strand. The second, labeled '(the special law)', shows a circle on a strand being equal to a straight strand.

następujące trzy równania:

$X$  wymieniony w strukturze Frobeniusa nazywamy specjalną komutatywną Frobenius monoid, lub po prostu monoid Frobenius w skrócie

Obiekt

**Definition 6.54.** Let  $(X, \mu, \eta, \delta, \epsilon)$  be a Frobenius monoid in a monoidal category  $(\mathcal{C}, I, \otimes)$ . Let  $m, n \in \mathbb{N}$ . Define  $s_{m,n}: X^{\otimes m} \rightarrow X^{\otimes n}$  to be the following morphism



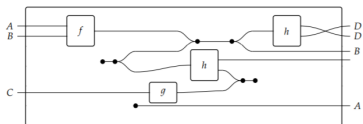
It can be written formally as  $(m-1)$   $\mu$ 's followed by  $(n-1)$   $\delta$ 's, with special cases when  $m=0$  or  $n=0$ .

We call  $s_{m,n}$  the *spider of type  $(m, n)$* , and can draw it more simply as the icon



# Schematy połączeń dla kategorii hypergraph

Wprowadzamy kategorie hypergraph poprzez ich schematy połączeń. Tak jak dla kategorie monoidalne, formalna definicja jest tylko strukturą wymaganą do jednoznacznej interpretacji tych diagramów. Rzeczywiście, nasze zainteresowanie kategoriami hipergrafów najlepiej widać na ich schematach okablowania. Kluczową ideą jest to, że diagramy okablowania dla kategorii hypergraph są diagramami sieci. Oznacza to, oprócz rysowania etykietowanych pól z danymi wejściowymi i wyjściowymi, jak możemy dla kategorii monoidalnych i oprócz gięcia tych drutów, jak możemy Kompaktowe zamknięte kategorie, możemy dzielić, łączyć, kończyć i inicjalizować przewody. Oto przykład schematu okablowania, który przedstawia kompozyt morfizmów w kategorii hypergraph



# Definicja kategorii hypergraph

Kategoria hypergraph jest symetryczną kategorią monoidalną  $(C, I, \otimes)$  w który każdy obiekt  $X$  jest wyposażony w strukturę Frobeniusa  $(X, \mu_X, \eta_X, \epsilon_X)$  tak, że dla wszystkich obiektów  $X, Y$  i taki, że  $I \text{ idl } I$ . Rekwizyt hipergraficzny to kategoria hipergraficzna, która jest również rekwizytem, np.  $\text{Ob}(C) \cong N$  itp.



Celem tego rozdziału jest pokazanie, w jaki sposób możemy skonstruować kategorię hypergraph których morfizmami są obwody elektryczne. Aby to zrobić, musimy najpierw wprowadzić pojęcie zachowującej strukturę mapy dla symetrycznych kategorii monoidalnych, uogólnienie monoidalnych monotonicznych zwanych symetrycznymi funktorami monoidalnymi. Następnie przedstawiamy ogólna metoda - dekorowanych cospansów - do wytwarzania kategorii hipergraficznych. W ten sposób powstanie wiele luźnych końców: kolimity, cospansy, obwody i hipergraph.

**Rough Definition 6.68.** Let  $(\mathcal{C}, I_{\mathcal{C}}, \otimes_{\mathcal{C}})$  and  $(\mathcal{D}, I_{\mathcal{D}}, \otimes_{\mathcal{D}})$  be symmetric monoidal categories. To specify a *symmetric monoidal functor*  $(F, \varphi)$  between them,

- (i) one specifies a functor  $F: \mathcal{C} \rightarrow \mathcal{D}$ ;
- (ii) one specifies a morphism  $\varphi_I: I_{\mathcal{D}} \rightarrow F(I_{\mathcal{C}})$ .
- (iii) for each  $c_1, c_2 \in \text{Ob}(\mathcal{C})$ , one specifies a morphism

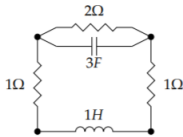
$$\varphi_{c_1, c_2}: F(c_1) \otimes_{\mathcal{D}} F(c_2) \rightarrow F(c_1 \otimes_{\mathcal{C}} c_2),$$

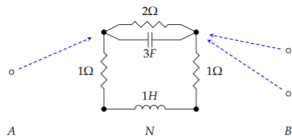
natural in  $c_1$  and  $c_2$ .

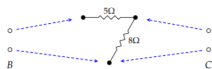
We call the various maps  $\varphi$  *coherence maps*. We require the coherence maps to obey bookkeeping axioms that ensure they are well behaved with respect to the symmetric monoidal structures on  $\mathcal{C}$  and  $\mathcal{D}$ . If  $\varphi_I$  and  $\varphi_{c_1, c_2}$  are isomorphisms for all  $c_1, c_2$ , we say that  $(F, \varphi)$  is *strong*.

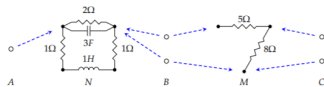


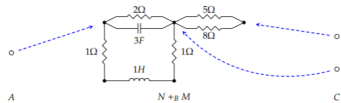
# Przykładowe zastosowanie grafu dla obwodu elektrycznego











Prezentacje Przygotowali:  
Radosław Ćwikowski  
Marcin Caputa