Materiały do wykładu z logiki liniowej

Paweł Urzyczyn
14 czerwca 2006

Logika liniowa, pochodząca od J.-Y. Girarda i rozwijana od wczesnych lat dziewięćdziesiątych, łączy konstruktywizm logiki intuicjonistycznej z symetrią logiki klasycznej. Jest to ciekawy model teoretyczny rozmaitych zjawisk obliczeniowych, związanych np. z gospodarką zasobami czy problemem optymalnych redukcji. Wykład będzie omawiał podstawowe zagadnienia teorii dowodu, semantyki, oraz interpretacje obliczeniowe logiki liniowej, a także zagadnienia tzw. "geometrii interakcji".

1 Powtórzenie z logiki klasycznej

Jednym z systemów dowodzenia dla klasycznego rachunku zdań jest rachunek sekwentów Gerharda Gentzena. Przypomnijmy podstawowe definicje.

Definicja 1.1 *Sekwentem* nazywamy napis postaci „$\alpha_1, \ldots, \alpha_n \vdash \beta_1, \ldots, \beta_m$", gdzie α_i i β_j są formułami. Zarówno m jak n może być zerem. Konwencja: duże litery greckie oznaczają ciągi formuł. Piszemy Γ, α lub α, Γ na oznaczenie ciągu powstałego z Γ przez dopisanie α z tyłu lub z przodu. Za *aksjomaty* rachunku sekwentów przyjmujemy sekwenty postaci: $\alpha \vdash \alpha$, gdzie α jest dowolną formułą.

Wyprowadzeniem sekwentu $\Gamma \vdash \Sigma$ (lub *dowodem formalnym w sensie Gentzena*) nazywamy drzewo etykietowane sekwentami, które ma następujące własności:

- Korzeń ma etykietę $\Gamma \vdash \Sigma$;
- Etykiety liści są aksjomatami;
- Etykiety pozostałych wierzchołków są otrzymane z etykiet ich następników (synów) za pomocą reguł wnioskowania jak na Obrazku 1.

Jeśli sekwent $\vdash \varphi$ ma dowód, to formułę φ nazywamy *twierdzeniem*.

Reguły rachunku sekwentów dzielimy na trzy grupy: reguły *strukturalne* (wymiana, skracanie i osłabianie), reguły *logiczne* (związane ze spójnikami zdaniowymi) i regułę cięcia. Wszystkie reguły oprócz cięcia dzielimy ponadto na prawe i lewe. Dla każdego spójnika mamy jedną lub dwie lewe i jedną lub dwie prawe reguły logiczne, określające sposób wprowadzenia danego spójnika po lewej (prawej) stronie sekwentu. Wyjątkiem są stałe logiczne **ff** (fałsz) i **tt** (prawda), dla których mamy tylko po jednej regule.

Wymiana:	$\dfrac{\Gamma, \varphi, \psi, \Delta \vdash \Sigma}{\Gamma, \psi, \varphi, \Delta \vdash \Sigma}(\mathrm{LW})$	$\dfrac{\Gamma \vdash \Delta, \varphi, \psi, \Sigma}{\Gamma \vdash \Delta, \psi, \varphi, \Sigma}(\mathrm{PW})$
Osłabianie:	$\dfrac{\Gamma \vdash \Sigma}{\Gamma, \varphi \vdash \Sigma}(\mathrm{LO})$	$\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \varphi, \Pi}(\mathrm{PO})$
Skracanie:	$\dfrac{\Gamma, \varphi, \varphi \vdash \Sigma}{\Gamma, \varphi \vdash \Sigma}(\mathrm{LS})$	$\dfrac{\Gamma \vdash \varphi, \varphi, \Sigma}{\Gamma \vdash \varphi, \Sigma}(\mathrm{PS})$
Koniunkcja:	$\dfrac{\Gamma, \varphi \vdash \Sigma}{\Gamma, \varphi \wedge \psi \vdash \Sigma}(\mathrm{LK}) \; \dfrac{\Gamma, \psi \vdash \Sigma}{\Gamma, \varphi \wedge \psi \vdash \Sigma}$	$\dfrac{\Gamma \vdash \varphi, \Pi \quad \Gamma \vdash \psi, \Pi}{\Gamma \vdash \varphi \wedge \psi, \Pi}(\mathrm{PK})$
Alternatywa:	$\dfrac{\Gamma, \varphi \vdash \Sigma \quad \Gamma, \psi \vdash \Sigma}{\Gamma, \varphi \vee \psi \vdash \Sigma}(\mathrm{LA})$	$\dfrac{\Gamma \vdash \varphi, \Pi}{\Gamma \vdash \varphi \vee \psi, \Pi}(\mathrm{PA}) \dfrac{\Gamma \vdash \psi, \Pi}{\Gamma \vdash \varphi \vee \psi, \Pi}$
Implikacja:	$\dfrac{\Gamma \vdash \varphi, \Pi \quad \Delta, \psi \vdash \Pi}{\Gamma, \Delta, \varphi \to \psi \vdash \Sigma, \Pi}(\mathrm{LI})$	$\dfrac{\Gamma, \varphi \vdash \psi, \Pi}{\Gamma \vdash \varphi \to \psi, \Pi}(\mathrm{PI})$
Negacja:	$\dfrac{\Gamma \vdash \varphi, \Pi}{\Gamma, \neg\varphi \vdash \Pi}(\mathrm{LN})$	$\dfrac{\Gamma, \varphi \vdash \Pi}{\Gamma \vdash \neg\varphi, \Pi}(\mathrm{PN})$
Prawda i fałsz:	$\Gamma, \mathbf{ff} \vdash \quad (\mathrm{LF})$	$\Gamma \vdash \mathbf{tt}, \Pi \quad (\mathrm{PP})$
Cięcie:	$\dfrac{\Gamma \vdash \varphi, \Pi \quad \Delta, \varphi \vdash \Sigma}{\Gamma, \Delta \vdash \Pi, \Sigma}(\mathrm{cut})$	

Obrazek 1: Rachunek sekwentów dla klasycznej logiki zdaniowej

Dla rachunku sekwentów mamy takie twierdzenie o pełności:

Twierdzenie 1.2 *Następujące warunki są równoważne:*

1. *Sekwent* $\alpha_1, \ldots, \alpha_n \vdash \beta_1, \ldots, \beta_m$ *ma dowód;*
2. *Formuła* $\alpha_1 \wedge \cdots \wedge \alpha_n \to \beta_1 \vee \cdots \vee \beta_m$ *jest klasyczną tautologią.*

Prawdziwe jest też następujące twierdzenie („*Hauptsatz*”) Gentzena:

Twierdzenie 1.3 (o eliminacji cięcia) *Każdy sekwent, który ma dowód w rachunku sekwentów, ma też taki dowód, który nie używa reguły cięcia.*

Główna zaleta dowodów bez cięcia wynika z następującej *własności podformuł:* wszystkie formuły występujące w przesłance dowolnej reguły (poza cięciem) są podformułami formuł występujących w konkluzji. Dlatego np. w dowodzie sekwentu $\vdash \alpha$ mamy do czynienia tylko z podformułami formuły α, a w dowodzie formuły $(p \to \neg p) \to (\neg p \to p) \to \neg p$ występuje tylko implikacja i negacja. Dowodzi to *konserwatywności* logiki klasycznej nad jej fragmentami, otrzymanymi przez ograniczenie zbioru spójników.

Reguły wymiany pozwalają na dowolne uporządkowanie formuł po obu stronach sekwentu. Można więc sekwent utożsamiać z parą multizbiorów. Reguły skracania pozwalają na następny krok: można zapomnieć o powtórzeniach formuł i uważać sekwent za parę zbiorów. Osłabianie ma zaś taki sens: dodanie „niepotrzebnych" założeń lub konkluzji nie psuje poprawności dowodu. Reguły strukturalne mówią więc, że na poprawność wnioskowania nie wpływa:

- kolejnośc użycia założeń (wymiana);
- wielokrotne używanie założeń (skracanie);
- rezygnacja z użycia założenia (osłabianie).

W obecności reguł strukturalnych, reguły dla koniunkcji i alternatywy znajdujące się na Obrazku 1 (tzw. reguły *addytywne* lub *kontekstowe*) są równoważne regułom *multiplikatywnym* lub *bezkontekstowym*, jak na Obrazku 2. Stałe logiczne, rozumiane jako elementy neutralne operacji $\wedge$ i $\vee$, też mogą być definiowane innymi regułami. Zauważmy, że reguły addytywne z Obrazka 1 wyrażają „porządkowy" sens koniunkcji i alternatywy jako kresu dolnego i górnego, podczas gdy reguły multiplikatywne z Obrazka 2 identyfikują koniunkcję i alternatywę z... przecinkiem (odpowiednio po lewej i prawej stronie sekwentu).

Koniunkcja:	$\dfrac{\Gamma, \varphi, \psi \vdash \Sigma}{\Gamma, \varphi \wedge \psi \vdash \Sigma}$ (L$\wedge$)	$\dfrac{\Gamma \vdash \varphi, \Sigma \quad \Delta \vdash \psi, \Pi}{\Gamma, \Delta \vdash \varphi \wedge \psi, \Sigma, \Pi}$ (P$\wedge$)
Alternatywa:	$\dfrac{\Gamma, \varphi \vdash \Sigma \quad \Delta, \psi \vdash \Pi}{\Gamma, \Delta, \varphi \vee \psi \vdash \Sigma, \Pi}$ (L$\vee$)	$\dfrac{\Gamma \vdash \varphi, \psi, \Sigma}{\Gamma \vdash \varphi \vee \psi, \Sigma}$ (P$\vee$)
Prawda:	$\dfrac{\Gamma \vdash \Sigma}{\Gamma, \mathbf{tt} \vdash \Sigma}$ (L$\mathbf{tt}$)	$\vdash \mathbf{tt}$ (P$\mathbf{tt}$)
Fałsz:	$\mathbf{ff} \vdash$ (L$\mathbf{ff}$)	$\dfrac{\Gamma \vdash \Pi}{\Gamma \vdash \mathbf{ff}, \Pi}$ (P$\mathbf{ff}$)

Obrazek 2: Reguły multiplikatywne (bezkontekstowe)

Rachunek sekwentów dla intuicjonistycznej logiki zdaniowej różni się od klasycznego systemu z Obrazka 1 tym, że po prawej stronie sekwentu może występować co najwyżej jedna formuła. Inaczej mówiąc, wariant intuicjonistyczny dopuszcza tylko sekwenty postaci $\Gamma \vdash \varphi$ i $\Gamma \vdash \varnothing$, gdzie $\varnothing$ oznacza ciąg pusty. Reguły (PW) i (PS) tracą sens. W pozostałych regułach należy przyjąć, że ciąg Σ ma co najwyżej jeden element, a ciąg Π jest pusty. Własności intuicjonistycznego rachunku sekwentów są podobne do własności rachunku klasycznego, w szczególności zachodzi twierdzenie o eliminacji cięcia.

2 Spójniki multiplikatywne i addytywne

Jeśli odrzucimy reguły skracania i osłabiania, to zniknie równoważnośc reguł multiplikatywnych i addytywnych dla koniunkcji i alternatywy. Koniunkcja addytywna z Obrazka 1 okazuje się w istocie innym spójnikiem logicznym, niż koniunkcja multiplikatywna z Obrazka 2. Podobnie jest dla alternatywy. *Klasyczny liniowy rachunek zdań* ma więc cztery spójniki binarne (Obrazek 3) zamiast dwóch. Są też cztery stałe logiczne — elementy neutralne tych czterech działań.

Wymiana:
$$\frac{\Gamma,\varphi,\psi,\Delta\vdash\Sigma}{\Gamma,\psi,\varphi,\Delta\vdash\Sigma}\,(\mathrm{LX}) \qquad \frac{\Gamma\vdash\Delta,\varphi,\psi,\Sigma}{\Gamma\vdash\Delta,\psi,\varphi,\Sigma}\,(\mathrm{RX})$$

Wraz &:
$$\frac{\Gamma,\varphi\vdash\Sigma}{\Gamma,\varphi\mathbin{\&}\psi\vdash\Sigma}\,(\mathrm{L}\&)\ \frac{\Gamma,\psi\vdash\Sigma}{\Gamma,\varphi\mathbin{\&}\psi\vdash\Sigma} \qquad \frac{\Gamma\vdash\varphi,\Sigma\quad\Gamma\vdash\psi,\Sigma}{\Gamma\vdash\varphi\mathbin{\&}\psi,\Sigma}\,(\mathrm{R}\&)$$

Tensor ⊗:
$$\frac{\Gamma,\varphi,\psi\vdash\Sigma}{\Gamma,\varphi\otimes\psi\vdash\Sigma}\,(\mathrm{L}\otimes) \qquad \frac{\Gamma\vdash\varphi,\Sigma\quad\Delta\vdash\psi,\Pi}{\Gamma,\Delta\vdash\varphi\otimes\psi,\Sigma,\Pi}\,(\mathrm{R}\otimes)$$

Plus ⊕:
$$\frac{\Gamma,\varphi\vdash\Sigma\quad\Gamma,\psi\vdash\Sigma}{\Gamma,\varphi\oplus\psi\vdash\Sigma}\,(\mathrm{L}\oplus) \qquad \frac{\Gamma\vdash\varphi,\Sigma}{\Gamma\vdash\varphi\oplus\psi,\Sigma}\,(\mathrm{R}\oplus)\ \frac{\Gamma\vdash\psi,\Sigma}{\Gamma\vdash\varphi\oplus\psi,\Sigma}$$

Par ⅋:
$$\frac{\Gamma,\varphi\vdash\Sigma\quad\Delta,\psi\vdash\Pi}{\Gamma,\Delta,\varphi\parr\psi\vdash\Sigma,\Pi}\,(\mathrm{L}\parr) \qquad \frac{\Gamma\vdash\varphi,\psi,\Sigma}{\Gamma\vdash\varphi\parr\psi,\Sigma}\,(\mathrm{R}\parr)$$

Implikacja:
$$\frac{\Gamma\vdash\varphi,\Sigma\quad\Delta,\psi\vdash\Pi}{\Gamma,\Delta,\varphi\multimap\psi\vdash\Sigma,\Pi}\,(\mathrm{L}\multimap) \qquad \frac{\Gamma,\varphi\vdash\psi,\Sigma}{\Gamma\vdash\varphi\multimap\psi,\Sigma}\,(\mathrm{R}\multimap)$$

Negacja:
$$\frac{\Gamma\vdash\varphi,\Sigma}{\Gamma,\varphi^{\perp}\vdash\Sigma}\,(\mathrm{L}^{\perp}) \qquad \frac{\Gamma,\varphi\vdash\Sigma}{\Gamma\vdash\varphi^{\perp},\Sigma}\,(\mathrm{R}^{\perp})$$

Jedynka:
$$\frac{\Gamma\vdash\Sigma}{\Gamma,1\vdash\Sigma}\,(\mathrm{L}1) \qquad \vdash 1\ (\mathrm{R}1)$$

Dno:
$$\perp\ \vdash\quad(\mathrm{L}\perp) \qquad \frac{\Gamma\vdash\Sigma}{\Gamma\vdash\Sigma,\perp}\,(\mathrm{R}\perp)$$

Zero i Top:
$$\Gamma,0\vdash\Sigma\ (\mathrm{L}0) \qquad \Gamma\vdash\top,\Sigma\ (\mathrm{R}\top)$$

Cięcie:
$$\frac{\Gamma\vdash\varphi,\Sigma\quad\Delta,\varphi\vdash\Pi}{\Gamma,\Delta\vdash\Sigma,\Pi}\,(\mathrm{Cut})$$

Obrazek 3: Multiplikatywno-addytywna logika liniowa

Liniowy rachunek sekwentów zbudowany jest na tych samych zasadach co zwykły. Stosujemy też tę samą terminologię i notację. Aksjomatami nadal są sekwenty postaci $\varphi\vdash\varphi$, nadal są prawe i lewe reguły wprowadzania spójników. Z reguł strukturalnych pozostała tylko wymiana. Będziemy o niej zazwyczaj zapominać, utożsamiając sekwenty z parami multizbiorów.

Cztery spójniki i cztery stałe naszej nowej logiki umieścimy w tabelce-przypominałce:

klasyczne	$\wedge$	$\vee$	**tt**	**ff**
multiplikatywne	$\otimes$	$\parr$	1	$\perp$
addytywne	$\&$	$\oplus$	$\top$	0

Pierwsza intuicja związana z doborem reguł dla tych spójników jest taka: Operacje addytywne to kresy, operacje multiplikatywne to przecinki. Stałe $\top$ i 0 to elementy skrajne (zauważmy, że $\alpha \vdash \top$ i $0 \vdash \alpha$, dla dowolnego α) a stałe 1 i $\perp$ to puste ciągi (odpowiednio po lewej i po prawej: na przykład $1, \Gamma \vdash \Delta$ wtedy i tylko wtedy, gdy $\Gamma \vdash \Delta$). Najważniejsza interpretacja formuł logiki liniowej jest jednak związana z odpowiedniością Curry'ego-Howarda. O formułach myślimy jak o typach, a wnioskowanie postaci $\alpha_1, \ldots, \alpha_n \vdash \beta$ (na razie tylko z jedną konkluzją po prawej stronie) rozumiemy jako opis procesu przekształcenia założeń (danych) typów $\alpha_1, \ldots, \alpha_n$ w konkluzję typu β. Interpretacja spójników $\otimes$, $\&$, $\oplus$ i $\multimap$ może być sformułowana tak:

- Obiekt typu $\alpha \otimes \beta$ to para obiektów: pierwszy typu α, drugi typu β. Utworzenie takiej pary wymaga odrębnych zasobów dla każdego członu. Także skonsumowanie takiej pary polega na zużyciu obu członów.

- Obiekt typu $\alpha \oplus \beta$ to para złożona z obiektu typu α lub obiektu typu β oraz wskaźnika pokazującego o który typ faktycznie chodzi. Utworzenie obiektu typu $\alpha \oplus \beta$ polega na utworzeniu *jednego* obiektu (typu α bądź typu β) i zaopatrzeniu go w odpowiednią flagę. Skonsumowanie obiektu typu $\alpha \oplus \beta$, polega na jego otwarciu i spożyciu zawartości.

- Obiekt typu $\alpha \,\&\, \beta$ to „wirtualna" para obiektów (pierwszy typu α, drugi typu β) z których każdy może potencjalnie zostać utworzony (z tych samych zasobów). Inaczej mówiąc jest to „prawo wyboru" pomiędzy obiektem typu α i obiektem typu β. Utworzenie takiej „wirtualnej" pary jest możliwe wtedy, gdy dysponujemy zasobami wystarczającymi dokładnie na jeden z takich obiektów, ale nie na oba naraz.

- Implikacja liniowa $\alpha \multimap \beta$ reprezentuje typ procesu, w którym przesłanka α jest przetworzona w konkluzję β, przy czym zostaje całkowicie „zużyta", czy też „skonsumowana".

Zauważmy, że mamy nowy symbol dla implikacji („lizak") choć reguły są te same. Pod nieobecność reguł strukturalnych sens implikacji zmienił się jednak istotnie. Mamy też nowy symbol dla negacji, którą też teraz będziemy nieco inaczej interpretować.

- Negacja $\alpha^\perp$ to typ dualny do α. Utworzenie danej typu α to to samo, co skonsumowanie danej typu $\alpha^\perp$ i odwrotnie.

Najtrudniejsza do uchwycenia jest intuicja związana z operacją $\parr$.

- Typ $\alpha \parr \beta$ wyraża pełną i zamkniętą komunikację, lub też „uwikłanie" pomiędzy α i β.

Logika liniowa różni się znacznie od logiki klasycznej a nawet intuicjonistycznej. Aby zrozumieć o co w niej chodzi, należy dokładnie zapomnieć o klasycznym rozumieniu "zdania" jako wyrażenia, któremu "można przypisać wartość prawdy lub fałszu". Ta interpretacja jest nietrafna już dla logiki intuicjonistycznej, gdzie zamiast o wartość logiczną zdania pytamy o możliwość uzasadnienia go poprzez *konstrukcję*.

W przypadku logiki liniowej, taka interpretacja to jeszcze za mało. Poprawne wnioskowanie liniowe polega na *skonstruowaniu* konkluzji z założeń w bardziej dosłownym sensie. Założenia, traktowane w sposób "materialny", są zużywane w trakcie dowodu, nie można też ich bezkarnie porzucić. Można to sformułować tak: konstrukcja intuicjonistyczna to *funkcja*, a konstrukcja liniowa to *akcja* uwzględniająca efekty uboczne w postaci operacji na zasobach.

Typowo francuskim sposobem objaśniania logiki liniowej jest przykład menu, złożonego z przystawki, dania głównego i deseru. W polskim wariancie niech to będzie zupa, drugie i kompot:

$$\textit{10 zł} \multimap (\textit{pomidorowa} \,\&\, \textit{krupnik}) \otimes (\textit{kotlet} \,\&\, \textit{ryba}) \otimes (\textit{kompot jabłkowy} \oplus \textit{kompot śliwkowy})$$

W naszym przykładzie za 10 złotych dostaniemy w stołówce zupę (do wyboru pomidorową lub krupnik), drugie danie (do wyboru kotlet lub rybę), i kompot (taki jaki będzie tego dnia w garnku). W tym ostatnim przypadku, wybór nie należy do nas, ale do kucharza.[1]

3 Tautologie liniowe

Zacznijmy od kilku podstawowych własności naszego systemu. Na początek taki dowód:

$$\dfrac{\Gamma_2 \vdash \alpha \qquad \dfrac{\Gamma_1 \vdash \alpha \multimap \beta \qquad \dfrac{\alpha \vdash \alpha \quad \beta \vdash \beta}{\alpha, \alpha \multimap \beta \vdash \beta}\,(\mathrm{L}\multimap)}{\Gamma_1, \alpha \vdash \beta}\,(\mathrm{cut})}{\Gamma_1, \Gamma_2 \vdash \beta}\,(\mathrm{cut})$$

Udowodniliśmy, że reguła *modus ponens* w postaci

$$\frac{\Gamma_1 \vdash \alpha \multimap \beta \quad \Gamma_2 \vdash \alpha}{\Gamma_1,\, \Gamma_2 \vdash \beta}$$

jest dopuszczalna (zauważmy, że założenia się sumują). Można ją uogólnić tak:

$$\frac{\Gamma_1 \vdash \alpha \multimap \beta, \Delta_1 \quad \Gamma_2 \vdash \alpha,\, \Delta_2}{\Gamma_1,\, \Gamma_2 \vdash \beta,\, \Delta_1,\, \Delta_2}$$

Mamy też regułę (P$\multimap$), czyli twierdzenie o dedukcji: *Jeśli* $\Gamma, \alpha \vdash \beta,\, \Delta$ *to* $\Gamma \vdash \alpha \multimap \beta,\, \Delta$.

Implikacja zachowuje się jak quasiporządek. Mamy bowiem $\alpha \vdash \alpha$, czyli $\vdash \alpha \multimap \alpha$, oraz $\alpha \multimap \beta,\ \beta \multimap \gamma \vdash \alpha \multimap \gamma$. W tym quasiporządku elementem największym jest $\top$ a najmniejszym 0, mamy bowiem $\alpha \vdash \top$ oraz $0 \vdash \alpha$ dla dowolnego α.

Ale uwaga: niebawem okaże się, że $\alpha, \beta \nvdash \alpha$, a nawet $\alpha, \alpha \nvdash \alpha$. Po lewej mamy dwie jednostki zasobu α, po prawej jedną. Ogólniej, z tego, że $\Gamma \vdash \alpha$ *nie wynika*, że $\Gamma, \beta \vdash \alpha$, bo β nie może przepaść bez śladu (nic w przyrodzie nie ginie). Z drugiej strony, mamy na przykład $\alpha,\ \alpha \multimap \alpha \multimap \beta \nvdash \beta$, bo do uzyskania β trzeba założenie $\alpha \multimap \alpha \multimap \beta$ dwukrotnie aplikować do α, więc jedna sztuka α nie wystarczy. Podobnie, $\alpha, \alpha, \alpha \to \beta, \beta \to \beta \to \gamma \nvdash \gamma$. Tu wprawdzie alf mamy dostatek, ale implikacja $\alpha \to \beta$ też się zużywa, a potrzebna jest dwa razy. Z podobnych powodów, liniowe odpowiedniki aksjomatów logiki implikacyjnej

$$\alpha \multimap \beta \multimap \alpha \qquad\qquad (\alpha \multimap \beta \multimap \gamma) \multimap (\alpha \multimap \beta) \multimap \alpha \multimap \gamma$$

nie są twierdzeniami liniowymi (co ściśle udowodnimy później).

Równoważność $\alpha \multimapboth \beta$ definiujemy jako $(\alpha \multimap \beta) \,\&\, (\beta \multimap \alpha)$. Dla dowolnego Γ mamy teraz $\Gamma \vdash \alpha \multimapboth \beta$ wtedy i tylko wtedy, gdy $\Gamma \vdash \alpha \multimap \beta$ oraz $\Gamma \vdash \beta \multimap \alpha$. Następujące równoważności (gdzie $\Box$ jest dowolnym spójnikiem spośród $\otimes$, $\&$, $\parr$, $\oplus$) są twierdzeniami logiki liniowej:

[1] Aha, jeszcze jedno: musimy wszystko zjeść, bo kucharz się obrazi.

1. $\alpha \mathbin{\Box} \beta \multimapboth \beta \mathbin{\Box} \alpha$;
2. $\alpha \mathbin{\Box} (\beta \mathbin{\Box} \gamma) \multimapboth (\alpha \mathbin{\Box} \beta) \mathbin{\Box} \gamma$;
3. $(\alpha \otimes \beta \multimap \gamma) \multimapboth (\alpha \multimap \beta \multimap \gamma)$;
4. $(\alpha \multimap \beta) \mathbin{\&} (\alpha \multimap \gamma) \multimapboth (\alpha \multimap \beta \mathbin{\&} \gamma)$;
5. $(\alpha \multimap \gamma) \mathbin{\&} (\beta \multimap \gamma) \multimapboth (\alpha \oplus \beta \multimap \gamma)$;
6. $\alpha \otimes (\beta \oplus \gamma) \multimapboth (\alpha \otimes \beta) \oplus (\alpha \otimes \gamma)$;
7. $\alpha \mathbin{\&} (\beta \parr \gamma) \multimapboth (\alpha \mathbin{\&} \beta) \parr (\alpha \mathbin{\&} \gamma)$;
8. $(\alpha \multimap \beta) \multimapboth \alpha^\perp \parr \beta$.

Zauważmy, że zamieniając spójniki $\otimes$ i $\&$ na $\wedge$, spójniki $\oplus$ i $\parr$ na $\vee$, implikację $\multimap$ na zwykłą strzałkę, a $^\perp$ na negację, otrzymamy z każdego twierdzenia liniowego klasyczną tautologię. Czasami (nie zawsze) uda się też operacja odwrotna, ale zwykle musimy wybrać koniunkcję (alternatywę) odpowiednią do sytuacji. Na przykład, nie uda nam się udowodnić formuły (3), jeśli zamiast $\otimes$ wstawimy do niej $\&$.

Na osobne podkreślenie załuguje równoważność (8) wyrażająca liniową implikację za pomocą $\parr$ i negacji, podobnie jak jest w logice klasycznej. W logice liniowej, jak w klasycznej, mamy też pary wzajemnie dualnych spójników, powiązanych prawami de Morgana:

9. $(\alpha \otimes \beta)^\perp \multimapboth \alpha^\perp \parr \beta^\perp$ oraz $(\alpha \parr \beta)^\perp \multimapboth \alpha^\perp \otimes \beta^\perp$;
10. $(\alpha \mathbin{\&} \beta)^\perp \multimapboth \alpha^\perp \oplus \beta^\perp$ oraz $(\alpha \oplus \beta)^\perp \multimapboth \alpha^\perp \mathbin{\&} \beta^\perp$;
11. $\perp^\perp \multimapboth 1$ oraz $1^\perp \multimapboth \perp$;
12. $0^\perp \multimapboth \top$ oraz $\top^\perp \multimapboth 0$;
13. $\alpha^{\perp\perp} \multimapboth \alpha$.

Ostatni punkt mówi, że negacja jest dualna sama do siebie. O danej typu $\alpha^\perp$ myślimy jako o danej "odwrotnej" do danej typu α. Przesłanie do kogoś α to to samo co otrzymanie od niego $\alpha^\perp$. Dobrym skojarzeniem jest wyobrażenie prądu elektrycznego jako

- przepływu ładunku dodatniego od bieguna $\oplus$ do bieguna $\ominus$, lub
- przepływu ładunku ujemnego od bieguna $\ominus$ do bieguna $\oplus$, lub
- obu tych przepływów występujących jednocześnie i stanowiących nierozerwalnie związane dualne aspekty tego samego zjawiska.

Przy tej interpretacji, dualność pomiędzy $\&$ i $\oplus$, wyrażona przez równoważności (10), ma czytelny sens intuicyjny. Wytworzenie (przesłanie) danej typu $\alpha \mathbin{\&} \beta$ to to samo co skonsumowanie (pobranie) danej typu $\alpha^\perp \oplus \beta^\perp$. Trudno tu uciec od analogii z grą, w której gracz wykonuje ruch polegający na wyborze między α i β. Wtedy konsumuje on $\alpha \mathbin{\&} \beta$ i przekazuje przeciwnikowi $\alpha^\perp \oplus \beta^\perp$.

Na koniec, jeszcze jedna seria przykładów. Uwaga: te implikacje nie dają się odwrócić.

14. $(\alpha \multimap \beta) \oplus (\alpha \multimap \gamma) \multimap (\alpha \multimap \beta \oplus \gamma)$;
15. $(\alpha \mathbin{\&} \beta) \oplus (\alpha \mathbin{\&} \gamma) \multimap \alpha \mathbin{\&} (\beta \oplus \gamma)$;
16. $\alpha \oplus (\beta \mathbin{\&} \gamma) \multimap (\alpha \oplus \beta) \mathbin{\&} (\alpha \oplus \gamma)$;
17. $(1 \mathbin{\&} \alpha) \otimes (1 \mathbin{\&} \alpha) \multimap 1 \mathbin{\&} (\alpha \otimes \alpha)$;
18. $\alpha \otimes (\beta \mathbin{\&} \gamma) \multimap (\alpha \otimes \beta) \mathbin{\&} (\alpha \otimes \gamma)$.

4 Modalności

Logika multiplikatywno-addytywna jest ciekawa, ale przez ścisły zakaz używania reguł strukturalnych jej siła wyrazu jest ograniczona.[2] Dużo ciekawiej będzie, jeśli dopuścimy możliwość *kontrolowanego* stosowania reguł strukturalnych. Inaczej mówiąc, możemy pozwolić na swobodę w użyciu tych założeń (danych), które są w jawny sposób zadeklarowane jako założenia (dane) wielokrotnego użytku.

Wyrażamy to za pomocą nowych, jednoargumentowych spójników „!" (oczywiście) i „?" (być może). Spójniki te są nazywane *modalnościami* (chociaż ta nazwa jest myląca) lub operatorami *wykładniczymi*. Możemy je interpretować tak:

- Obiekt typu $!\alpha$ to gotowość do utworzenia *dowolnej żądanej* liczby obiektów typu α.
- Obiekt typu $?\alpha$ to gotowość do skonsumowania *pewnej nieznanej* liczby obiektów typu α.

Reguły wnioskowania dla spójników wykładniczych widać na Obrazku 4.

Oczywiście:	$\dfrac{\Gamma \vdash \Sigma}{\Gamma, !\varphi \vdash \Sigma}$ (O !)	$\dfrac{\Gamma, \varphi \vdash \Sigma}{\Gamma, !\varphi \vdash \Sigma}$ (L !)	$\dfrac{!\Gamma \vdash \varphi, ?\,\Sigma}{!\Gamma \vdash !\varphi, ?\,\Sigma}$ (R !)	$\dfrac{\Gamma, !\varphi, !\varphi \vdash \Sigma}{\Gamma, !\varphi \vdash \Sigma}$ (S !)
Być może:	$\dfrac{\Gamma \vdash \Sigma}{\Gamma \vdash ?\,\varphi, \Sigma}$ (O ?)	$\dfrac{\Gamma \vdash \varphi, \Sigma}{\Gamma \vdash ?\,\varphi, \Sigma}$ (R ?)	$\dfrac{!\Gamma, \varphi \vdash ?\,\Sigma}{!\Gamma, ?\,\varphi \vdash ?\,\Sigma}$ (L ?)	$\dfrac{\Gamma \vdash ?\,\varphi, ?\,\varphi, \Sigma}{\Gamma \vdash ?\,\varphi, \Sigma}$ (S ?)

Obrazek 4: Modalności (operacje wykładnicze)

Oto kilka przykładów twierdzeń z modalnościami. Równoważność (22) wskazuje pochodzenie nazwy "wykładniczy". Implikacji (24–26) nie da się odwrócić.

19. $(!\alpha)^\perp \multimapboth ?\alpha^\perp$ oraz $(?\alpha)^\perp \multimapboth !\alpha^\perp$;
20. $!\alpha \multimapboth !!\alpha$;

[2] Ale znowu nie tak bardzo, bo ta logika jest PSPACE-zupełna.

21. $!\alpha \multimap\!\circ\ 1 \mathbin{\&} \alpha \mathbin{\&} (!\alpha \otimes !\alpha)$;
22. $!(\alpha \mathbin{\&} \beta) \multimap\!\circ\ !\alpha \otimes !\beta$;
23. $!(\alpha \mathbin{\&} \beta) \multimap\!\circ\ !(!\alpha \mathbin{\&} !\beta)$;
24. $!(\alpha \mathbin{\&} \beta) \multimap !\alpha \mathbin{\&} !\beta$;
25. $!(!\alpha \multimap \beta) \multimap !\alpha \multimap !\beta$;
26. $!(\alpha \multimap \beta \mathbin{\&} \gamma) \multimap !(\alpha \multimap \beta) \mathbin{\&} !(\alpha \multimap \gamma)$.

5 Eliminacja cięcia

Ze względu na dualizm występujący pomiędzy spójnikami klasycznej logiki liniowej i symetrię zadaną przez operację $^\perp$, często przyjmuje się następujące uproszczenie składni.

- Zakładamy, że w zbiorze zmiennych zdaniowych określona jest bijektywna inwolucja, oznaczana przez $^\perp$. Inaczej mówiąc, przyjmujemy, że dla każdej zmiennej p mamy dokładnie jedną zmienną q spełniającą warunki $p^\perp = q$ i $q^\perp = p$. Wtedy oczywiście zachodzi też $p^{\perp\perp} = p$ i $q^{\perp\perp} = q$.
- Jeśli formuła α nie jest zmienną to $\alpha^\perp$ definiujemy przez prawa de Morgana (9–12, 19) ze stron 7–8. Na przykład $(\alpha \otimes \beta)^\perp$ to „skrót” wyrażenia $\alpha^\perp \parr \beta^\perp$.
- Implikację liniową definiujemy równaniem $(\alpha \multimap \beta) = \alpha^\perp \parr \beta$.

W tym ujęciu spójniki $^\perp$ i $\multimap$ przestają być formalnie częścią składni i stają się meta-językowymi skrótami, podobnie jak $\multimap\!\circ$. Korzystając ze wszechobecnego dualizmu. można też znacznie uprościć reguły wnioskowania. Reguły ($\mathrm{L}^\perp$) i ($\mathrm{R}^\perp$) stwierdzają, że dowolna formuła występująca po lewej stronie sekwentu, to to samo co formuła dualna umieszczona po stronie prawej. Sekwent $\Gamma \vdash \Delta$ jest więc równoważny sekwentowi $\vdash \Gamma^\perp, \Delta$ i można się ograniczyć do rachunku sekwentów prawostronnych. Zmniejsza to liczbę reguł od razu o połowę, bo przecież lewe reguły są lustrzanymi odbiciami prawych (Obrazek 5).

Dowód eliminacji cięcia dla klasycznej logiki liniowej przeprowadzimy dla takiego prawostronnego rachunku. Analogiczny dowód można wykonać dla rachunku dwustronnego, mamy jednak wtedy więcej przypadków.

Główna idea dowodu jest taka, żeby każdorazowo eliminować cięcie dotyczące możliwie najdłuższej formuły, zastępując je jednym lub kilkoma cięciami, w których formuła „wycinana” jest krótsza. Jeśli to się od razu nie udaje, to przesuwamy cięcie w górę. Dlatego zawsze wybieramy cięcie „maksymalne”, tj. takie, że wszystkie cięcia znajdujące się powyżej dotyczą krótszych formuł.

Ta metoda działa doskonale w większości przypadków, ale „zacina się” w kontakcie z regułą skracania. W przypadku logiki liniowej problem dotyczy reguły (S?). Dlatego odwołamy się do sposobu pochodzącego jeszcze od Gentzena i rozszerzymy nasz system o dodatkową regułę zwaną *multicięciem* lub *cięciem wielokrotnym*:

$$\frac{\vdash \Gamma,\, !\varphi^\perp \qquad \vdash \Delta,\, (?\varphi)^n}{\vdash \Gamma, \Delta}\,(\mathrm{mix})$$

Aksjomat: $\vdash \varphi, \varphi^{\perp}$.

Wymiana: $$\frac{\vdash \Gamma, \varphi, \psi, \Delta}{\vdash \Gamma, \psi, \varphi, \Delta}$$

Wraz &: $$\frac{\vdash \Gamma, \varphi \quad \vdash \Gamma, \psi}{\vdash \Gamma, \varphi \mathbin{\&} \psi}$$

Tensor ⊗: $$\frac{\vdash \Gamma, \varphi \quad \vdash \Delta, \psi}{\vdash \Gamma, \Delta, \varphi \otimes \psi}$$

Plus ⊕: $$\frac{\vdash \Gamma, \varphi}{\vdash \Gamma, \varphi \oplus \psi} \qquad \frac{\vdash \Gamma, \psi}{\vdash \Gamma, \varphi \oplus \psi}$$

Par ⅋: $$\frac{\vdash \Gamma, \varphi, \psi}{\vdash \Gamma, \varphi \invamp \psi}$$

Jedynka: $\vdash 1$

Dno: $$\frac{\vdash \Gamma}{\vdash \Gamma, \perp}$$

Top: $\vdash \Gamma, \top$ (nie ma reguły dla zera)

Oczywiście: $$\frac{\vdash \varphi, ?\,\Sigma}{\vdash !\varphi, ?\,\Sigma}\ (\,!\,)$$

Być może: $$\frac{\vdash \Gamma}{\vdash \Gamma, ?\,\varphi}\ (\mathrm{O?}) \qquad \frac{\vdash \Gamma, \varphi}{\vdash \Gamma, ?\,\varphi}\ (\,?\,) \qquad \frac{\vdash \Gamma, ?\,\varphi, ?\,\varphi}{\vdash \Gamma, ?\,\varphi}\ (\mathrm{S?})$$

Cięcie: $$\frac{\vdash \Gamma, \varphi \quad \vdash \Delta, \varphi^{\perp}}{\vdash \Gamma, \Delta}$$

Obrazek 5: Prawostronny rachunek sekwentów

Napis $(?\varphi)^n$ oznacza n-krotne wystąpienie formuły $?\varphi$, przy czym n jest dowolną liczbą naturalną, być może zerem. Nietrudo się przekonać, że reguła (mix) jest dopuszczaczalna, więc jej dodanie do systemu jest nieszkodliwe. W szczególności, dla $n = 1$ reguła (mix) to zwykłe cięcie. Odtąd, gdy mowa jest o „cięciu" to chodzi zarówno o cięcie zwykłe jak multicięcie, chyba że wyraźnie zaznaczono inaczej.

Definicja 5.1 *Stopniem cięcia* nazwiemy długość formuły wycinanej, tj. formuły φ przy zwykłym cięciu jak na Obrazku 5, lub formuły $!\varphi$ w przypadku reguły (mix) jak wyżej. *Stopień dowodu* to największy stopień cięcia w tym dowodzie. (Zatem dowód stopnia zero to dowód bez cięcia.) *Wagą cięcia* w dowodzie nazywamy rozmiar części dowodu powyżej tego cięcia (mierzony liczbą wierzchołków drzewa). Cięcie jest *maksymalne* jeśli powyżej niego występują tylko cięcia mniejszych stopni.

Lemat 5.2 *Przypuśćmy, że pewien sekwent ma dowód kończący się maksymalnym cięciem stopnia k (tj. że wszystkie pozostałe cięcia w tym dowodzie są stopni mniejszych niż k). Wtedy istnieje dowód tego sekwentu o stopniu mniejszym niż k.*

Dowód: Dowód jest przez indukcję ze względu na dwa parametry:

- Stopień $k > 0$ ostatniego cięcia;
- Wagę tego cięcia.

Rozważamy różne przypadki, w zależności od tego w jaki sposób otrzymano przesłanki ostatniego cięcia. W każdym z tych przypadków eliminujemy to ostatnie cięcie, wprowadzając być może jakieś nowe cięcia. Ale te nowe cięcia są albo mniejszego stopnia, albo są maksymalne, tego samego stopnia i mniejszej wagi. Dla uproszczenia będziemy zakładać, że zamiast ciągów formuł mamy multizbiory (tj. reguła wymiany jest używana domyślnie).

Przypadek 1: Jedna z przesłanek jest aksjomatem. Jeśli cięcie jest zwykłe, to wygląda to tak:

$$\dfrac{\dfrac{\vdots}{\vdash \Gamma, \varphi} \qquad \vdash \varphi^\perp, \varphi}{\vdash \Gamma, \varphi}$$

i możemy po prostu cięcie usunąć, pozostawiając dowód z lewej strony. W przypadku multicięcia (gdzie $n \neq 1$) musi to wyglądać tak:

$$\dfrac{\dfrac{\vdots}{\vdash \Gamma,\ (?\varphi)^n} \qquad \vdash\ !\varphi^\perp,\ ?\varphi}{\vdash \Gamma,\ ?\varphi}$$

i dowód przerabiamy tak:

$$\dfrac{\dfrac{\vdots}{\vdash \Gamma,\ (?\varphi)^n}}{\vdash \Gamma,\ ?\varphi}$$

Podwójna linia oznacza serię kontrakcji (dla $n > 1$) lub pojedyncze osłabienie (dla $n = 0$). A więc w tym przypadku cięcie zostaje po prostu usunięte.

Przypadek 2: Obie przesłanki są otrzymane za pomocą reguł wprowadzających formułę wycinaną i formułę dualną. Ten przypadek to główny krok indukcyjny: redukujemy tu stopień cięcia. Na początek niech dowód kończy się regułą wprowadzającą $\varphi = \alpha \mathbin{\&} \beta$.

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\, \alpha\end{matrix} \qquad \begin{matrix}(2)\\ \vdots\\ \vdash \Gamma,\, \beta\end{matrix}}{\vdash \Gamma,\, \alpha \mathbin{\&} \beta} \qquad \dfrac{\begin{matrix}(3)\\ \vdots\\ \vdash \Pi,\, \alpha^\perp\end{matrix}}{\vdash \Pi,\, \alpha^\perp \oplus \beta^\perp}}{\vdash \Gamma, \Pi}$$

Zamiast do formuły $\alpha \mathbin{\&} \beta$ można zastosować cięcie do α i otrzymać dowód mniejszego stopnia:

$$\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\, \alpha\end{matrix} \qquad \begin{matrix}(3)\\ \vdots\\ \vdash \Pi, \alpha^\perp\end{matrix}}{\vdash \Gamma, \Pi}$$

Jeśli $\varphi = \alpha \otimes \beta$ to dowód

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\, \alpha\end{matrix} \qquad \begin{matrix}(2)\\ \vdots\\ \vdash \Delta,\, \beta\end{matrix}}{\vdash \Gamma,\, \alpha \otimes \beta} \qquad \dfrac{\begin{matrix}(3)\\ \vdots\\ \vdash \Pi,\, \alpha^\perp,\, \beta^\perp\end{matrix}}{\vdash \Pi,\, \alpha^\perp \parr \beta^\perp}}{\vdash \Gamma, \Delta, \Pi}$$

zastępujemy przez dowód o dwóch cięciach mniejszego stopnia:

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\, \alpha\end{matrix} \qquad \begin{matrix}(3)\\ \vdots\\ \vdash \Pi,\, \alpha^\perp,\, \beta^\perp\end{matrix}}{\vdash \Gamma, \Pi,\, \beta^\perp} \qquad \begin{matrix}(2)\\ \vdots\\ \vdash \Delta,\, \beta\end{matrix}}{\vdash \Gamma, \Delta, \Pi}$$

Z powodu symetrii nie musimy osobno rozważać przypadku $\varphi = \alpha \parr \beta$ ani przypadku $\varphi = \alpha \oplus \beta$. Możemy jednak mieć cięcie dla stałej 1:

$$\dfrac{\vdash 1 \qquad \dfrac{\begin{matrix}\vdots\\ \vdash \Gamma\end{matrix}}{\vdash \Gamma,\, \bot}}{\vdash \Gamma}$$

które można łatwo usunąć. Skoro nie ma reguły wprowadzającej zero, nie ma też takiego cięcia w przypadku 2. Niech więc $\varphi = {!\alpha}$. Mamy tu do czynienia z multicięciem. Z lewej strony mamy regułę (!), z prawej (O?), (S?) lub (?). Możliwość pierwsza wygląda tak:

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix}}{\vdash ?\Gamma,\ !\varphi^\perp} \qquad \dfrac{\begin{matrix}(2)\\ \vdots\\ \vdash \Sigma,\ (?\varphi)^n\end{matrix}}{\vdash \Sigma,\ (?\varphi)^n,\ ?\varphi}\,(\mathrm{O}?)}{\vdash ?\Gamma, \Sigma}$$

Stosując cięcie o jeden krok wcześniej, dostajemy dowód tego samego stopnia, ale mniejszy. Nowe cięcie jest maksymalne, więc teza wynika z założenia indukcyjnego.

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix}}{\vdash ?\Gamma,\ !\varphi^\perp} \quad \begin{matrix}(2)\\ \vdots\\ \vdash \Sigma,\ (?\varphi)^n\end{matrix}}{\vdash ?\Gamma, \Sigma}$$

Jeśli z prawej mamy regułę (S?) to dowód jest taki:

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix}}{\vdash ?\Gamma,\ !\varphi^\perp} \qquad \dfrac{\begin{matrix}(2)\\ \vdots\\ \vdash \Sigma,\ (?\varphi)^{n+1}\end{matrix}}{\vdash \Sigma,\ (?\varphi)^n}\,(\mathrm{S}?)}{\vdash ?\Gamma, \Sigma}$$

Tutaj też możemy wykonać cięcie o jeden krok wcześniej i zmniejszyć wagę cięcia. Zauważmy jednak, że dla $n = 1$ zwykłe cięcie musi zostać zastąpione multicięciem. To jest właśnie to miejsce w dowodzie eliminacji cięcia, gdzie potrzebna jest reguła (mix).

Pozostaje do rozpatrzenia sytuacja gdy z prawej jest (?):

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix}}{\vdash ?\Gamma,\ !\varphi^\perp} \qquad \dfrac{\begin{matrix}(2)\\ \vdots\\ \vdash \Sigma,\ (?\varphi)^n,\ \varphi\end{matrix}}{\vdash \Sigma,\ (?\varphi)^n,\ ?\varphi}\,(\,?\,)}{\vdash ?\Gamma, \Sigma}$$

Teraz, zgodnie z duchem przypadku 2, obniżymy stopień cięcia. Ale nie wszędzie.

$$\begin{array}{c}\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix} \qquad \dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash ?\Gamma,\ \varphi^\perp\end{matrix}}{\vdash ?\Gamma,\ !\varphi^\perp} \quad \begin{matrix}(2)\\ \vdots\\ \vdash \Sigma,\ (?\varphi)^n,\ \varphi\end{matrix}}{\vdash ?\Gamma,\ \Sigma,\ \varphi}}{\vdash ?\Gamma,\ ?\Gamma,\ \Sigma}\\ \hline\hline \vdash ?\Gamma,\ \Sigma\end{array}$$

Podwójna linia oznacza serię skracań. Mamy teraz dwa cięcia: jedno mniejszego stopnia, drugie maksymalne stopnia k, mniejszej wagi.

Przypadek 3: Przynajmniej jedna z przesłanek reguły cięcia nie wprowadza formuły wycinanej. W tym przypadku „przesuwamy” cięcie w górę zmniejszając jego wagę. Przypuśćmy najpierw, że formuła wycinana zwykłym cięciem nie jest wprowadzana z lewej, i że mamy tam regułę (&).

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\,\alpha,\,\varphi\end{matrix}\qquad\begin{matrix}(2)\\ \vdots\\ \vdash \Gamma,\,\beta,\,\varphi\end{matrix}}{\vdash \Gamma,\,\alpha \mathbin{\&} \beta,\,\varphi}\qquad\begin{matrix}(3)\\ \vdots\\ \vdash \Sigma,\,\varphi^{\perp}\end{matrix}}{\vdash \Gamma,\Sigma,\,\alpha \mathbin{\&} \beta}$$

Wtedy stosujemy cięcie do obu przesłanek reguły (&), otrzymując dwa maksymalne cięcia stopnia k, ale o mniejszej wadze.

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma,\,\alpha,\,\varphi\end{matrix}\qquad\begin{matrix}(3)\\ \vdots\\ \vdash \Sigma,\,\varphi^{\perp}\end{matrix}}{\vdash \Gamma,\Sigma,\,\alpha}\qquad\dfrac{\begin{matrix}(2)\\ \vdots\\ \vdash \Gamma,\,\beta,\,\varphi\end{matrix}\qquad\begin{matrix}(3)\\ \vdots\\ \vdash \Sigma,\,\varphi^{\perp}\end{matrix}}{\vdash \Gamma,\Sigma,\,\beta}}{\vdash \Gamma,\Sigma,\,\alpha \mathbin{\&} \beta}$$

Jeśli po lewej stronie jest reguła bezkontekstowa o dwóch przesłankach, tj. ($\otimes$) lub (mix) to dowód wygląda mniej więcej tak:

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma_1,\,\varphi\end{matrix}\qquad\begin{matrix}(2)\\ \vdots\\ \vdash \Gamma_2\end{matrix}}{\vdash \Gamma,\varphi}\,(\ast)\qquad\begin{matrix}(3)\\ \vdots\\ \vdash \Sigma,\,\varphi^{\perp}\end{matrix}}{\vdash \Gamma,\Sigma}\,(\mathrm{cut})$$

Stosujemy wtedy cięcie do tej przesłanki, która zawiera formułę φ i nadal można poprawnie zastosować regułę $(\ast)$:

$$\dfrac{\dfrac{\begin{matrix}(1)\\ \vdots\\ \vdash \Gamma_1,\,\varphi\end{matrix}\qquad\begin{matrix}(3)\\ \vdots\\ \vdash \Sigma,\,\varphi^{\perp}\end{matrix}}{\vdash \Gamma_1,\Sigma}\,(\mathrm{cut})\qquad\begin{matrix}(2)\\ \vdots\\ \vdash \Gamma_2\end{matrix}}{\vdash \Gamma,\Sigma}\,(\ast)$$

Po lewej stronie może też być reguła z jedną przesłanką, tj. jedna z reguł ($\oplus$), (⅋), ($\perp$), ($!$), (O?), ($?$), (S?), lub reguła ($\top$) bez przesłanek. Te przypadki są jeszcze prostsze.

Jeśli cięcie jest wielokrotne i wycinana formuła $!\varphi$ nie jest wprowadzana z lewej, to możliwe przypadki są podobne do rozpatrzonych wyżej. Trochę inaczej jest, gdy $!\varphi$ jest wprowadzone z lewej strony regułą ($!$), a z prawej strony mamy regułę inną niż (O?), (S?) i ($?$).

Przypuśćmy, że jest to reguła bezkontekstowa o dwóch przesłankach, tj. (⊗) lub (mix):

$$
\begin{prooftree}
\AxiomC{(1)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma,\ \varphi^{\perp}$}
\UnaryInfC{$\vdash ?\Gamma,\ !\varphi^{\perp}$}
\AxiomC{(2)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \Sigma_1,\ (?\varphi)^k$}
\AxiomC{(3)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \Sigma_2,\ (?\varphi)^l$}
\RightLabel{$(\,*\,)$}
\BinaryInfC{$\vdash \Sigma,\ (?\varphi)^{k+l}$}
\RightLabel{(mix)}
\BinaryInfC{$\vdash ?\Gamma, \Sigma$}
\end{prooftree}
$$

Musimy teraz zastosować cięcie do obu przesłanek reguły $(\,*\,)$, każdorazowo otrzymując maksymalne cięcie o mniejszej wadze. Na końcu konieczna jest seria kontrakcji.

$$
\begin{prooftree}
\AxiomC{(1)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma,\ \varphi^{\perp}$}
\UnaryInfC{$\vdash ?\Gamma,\ !\varphi^{\perp}$}
\AxiomC{(2)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \Sigma_1,\ (?\varphi)^k$}
\RightLabel{(mix)}
\BinaryInfC{$\vdash ?\Gamma, \Sigma_1$}
\AxiomC{(1)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash ?\Gamma,\ \varphi^{\perp}$}
\UnaryInfC{$\vdash ?\Gamma,\ !\varphi^{\perp}$}
\AxiomC{(3)}
\noLine
\UnaryInfC{$\vdots$}
\noLine
\UnaryInfC{$\vdash \Sigma_2,\ (?\varphi)^l$}
\RightLabel{(mix)}
\BinaryInfC{$\vdash ?\Gamma, \Sigma_2$}
\RightLabel{$(\,*\,)$}
\BinaryInfC{$\vdash ?\Gamma, ?\Gamma, \Sigma$}
\doubleLine
\UnaryInfC{$\vdash ?\Gamma, \Sigma$}
\end{prooftree}
$$

Jeśli prawa reguła jest kontekstowa lub nie ma dwóch przesłanek, to operacja eliminacji cięcia jest jeszcze prostsza. Zostawiamy ją dla własnej przyjemności czytelnika.[3] ∎

Twierdzenie 5.3 (o eliminacji cięcia) *Jeśli sekwent ma dowód to ma dowód bez cięcia.*

Dowód: Dowód jest przez indukcję ze względu na dwa parametry:

- Największy stopień cięcia w dowodzie;
- Liczba cięć tego stopnia.

Wybieramy maksymalne cięcie największego stopnia k i do części dowodu kończącej się tym cięciem stosujemy Lemat 5.2. Jeśli było to jedyne cięcie stopnia k, to maleje pierwszy parametr, w przeciwnym razie maleje drugi parametr. ∎

Jak już mówiliśmy, twierdzenie o eliminacji cięcia zachodzi też dla wariantu dwustronnego. (Zauważmy na przykład, że reguła (L⊸) to w istocie szczególny przypadek reguły (R⊗).)

Z eliminacji cięcia wynikają same dobre rzeczy, przede wszystkim *własność podformuł:*

W dowodzie bez cięcia dla sekwentu $\vdash \Gamma$ *występują wyłącznie podformuły formuł z* Γ.

Oznacza to, że klasyczna logika liniowa jest *konserwatywna* nad swoimi fragmentami wyznaczonymi przez wybór niektórych spójników, na przykład każde twierdzenie liniowe, w którym

[3] Jeśli ktoś ma ochotę zakodować wszystkie przypadki w LaTeXu, to proszę bardzo :)

występują tylko operacje $\oplus$, $\otimes$, $\parr$ i $\&$ można udowodnić używając tylko reguł *multiplikatywno-addytywnej logiki liniowej*, tj. nie odwołując się do reguł dla modalności. Z Twierdzenia 5.3 wynikają też różne rezultaty negatywne. Na przykład teraz dopiero umiemy ściśle uzasadnić, że sekwent $p, p \vdash p$ nie ma dowodu. Po prostu nie ma reguły, która mogłaby być ostatnia w takim wyprowadzeniu. Podobnie, można teraz już udowodnić, że implikacje odwrotne do (14–18, 24–26) na stronach 8 i 9 nie są twierdzeniami liniowymi. Także następujące dwa metatwierdzenia są prostymi konsekwencjami eliminacji cięcia.

Wniosek 5.4

1. *Jeśli* $\vdash \alpha \otimes \beta$ *to* $\vdash \alpha$ *i* $\vdash \beta$.
2. *Jeśli* $\vdash \alpha \oplus \beta$ *to* $\vdash \alpha$ *lub* $\vdash \beta$.

Z części (1) Wniosku 5.4 wynika, że warunki $\vdash \alpha \otimes \beta$ i $\vdash \alpha \& \beta$ są równoważne. Ale $\Gamma \vdash \alpha \otimes \beta$ zwykle znaczy zupełnie co innego niż $\Gamma \vdash \alpha \& \beta$.

6 Intuicjonistyczna logika liniowa

System przedstawiony na Obrazkach 3 i 4 nazywamy *klasyczną* logiką liniową. Mamy też *intuicjonistyczną* logikę liniową (ILL), otrzymaną tak, jak zwykłą logikę intuicjonistyczną można otrzymać ze zwykłej logiki klasycznej: przez ograniczenie liczby formuł po prawej stronie sekwentu do co najwyżej jednej. (Na Obrazku 6, symbol σ oznacza jedną formułę lub ciąg pusty.) Wtedy jednak tracą sens nie tylko pewne reguły, ale też niektóre spójniki: $\parr$, ?, $^\perp$ i $\perp$.

Czasami przydaje się $\perp$, ale wtedy trzeba przyjąć takie reguły:

$$\perp\ \vdash \qquad (\mathrm{L}\perp) \qquad\qquad\qquad \frac{\Gamma \vdash}{\Gamma \vdash \perp}\ (\mathrm{R}\perp)$$

Twierdzenie o eliminacji cięcia jest prawdziwe dla intuicjonistycznej logiki liniowej (także ze stałą $\perp$). Dowód wymaga w istocie powtórzenia całego dowodu dla klasycznej logiki liniowej (z podwójną przyjemnością, bo tu reguły nie mogą być jednostronne).

(Nie)konserwatywność

Wiadomo, że zwykła logika klasyczna jest istotnym rozszerzeniem logiki intuicjonistycznej, bo na przykład prawo Peirce'a $((p \to q) \to p) \to p$ albo zasada wyłączonego środka $\neg p \vee p$ są tautologiami klasycznymi ale nie intuicjonistycznymi. Ale formuła $((p \multimap q) \multimap p) \multimap p$ nie ma wcale liniowego dowodu, a liniowym odpowiednikiem zasady wyłączonego środka jest formuła $p^\perp \parr p$, nie należąca do intuicjonistycznej składni. Najprostszym klasycznym twierdzeniem liniowym, które nie ma dowodu intuicjonistycznego, jest więc zapewne formuła

$$((((0 \multimap p) \multimap q) \multimap r) \multimap 0) \multimap q.$$

Wymiana:
$$\dfrac{\Gamma, \varphi, \psi, \Delta \vdash \sigma}{\Gamma, \psi, \varphi, \Delta \vdash \sigma}\,(\mathrm{LX})$$

Wraz &:
$$\dfrac{\Gamma, \varphi \vdash \sigma}{\Gamma, \varphi \mathbin{\&} \psi \vdash \sigma}\ (\mathrm{L\&})\ \dfrac{\Gamma, \psi \vdash \sigma}{\Gamma, \varphi \mathbin{\&} \psi \vdash \sigma} \qquad \dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \mathbin{\&} \psi}\ (\mathrm{R\&})$$

Tensor $\otimes$:
$$\dfrac{\Gamma, \varphi, \psi \vdash \sigma}{\Gamma, \varphi \otimes \psi \vdash \sigma}\ (\mathrm{L}\otimes) \qquad \dfrac{\Gamma \vdash \varphi \quad \Delta \vdash \psi}{\Gamma, \Delta \vdash \varphi \otimes \psi}\ (\mathrm{R}\otimes)$$

Plus $\oplus$:
$$\dfrac{\Gamma, \varphi \vdash \sigma \quad \Gamma, \psi \vdash \sigma}{\Gamma, \varphi \oplus \psi \vdash \sigma}\ (\mathrm{L}\oplus) \qquad \dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \oplus \psi}\ (\mathrm{R}\oplus)\ \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \oplus \psi}$$

Implikacja:
$$\dfrac{\Gamma \vdash \varphi \quad \Delta, \psi \vdash \sigma}{\Gamma, \Delta, \varphi \multimap \psi \vdash \sigma}\ (\mathrm{L}\multimap) \qquad \dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \multimap \psi}\ (\mathrm{R}\multimap)$$

Jedynka:
$$\dfrac{\Gamma \vdash \sigma}{\Gamma, 1 \vdash \sigma}\ (\mathrm{L1}) \qquad \vdash 1\ (\mathrm{R1})$$

Zero i Top:
$$\Gamma, 0 \vdash \sigma\ (\mathrm{L0}) \qquad \Gamma \vdash \top\ (\mathrm{R}\top)$$

Oczywiście:
$$\dfrac{\Gamma \vdash \sigma}{\Gamma, !\varphi \vdash \sigma}\ (\mathrm{O}\ !) \qquad \dfrac{\Gamma, \varphi \vdash \sigma}{\Gamma, !\varphi \vdash \sigma}\ (\mathrm{L}\ !) \qquad \dfrac{!\Gamma \vdash \varphi}{!\Gamma \vdash !\varphi}\ (\mathrm{R}\ !) \qquad \dfrac{\Gamma, !\varphi, !\varphi \vdash \sigma}{\Gamma, !\varphi \vdash \sigma}\ (\mathrm{S}\ !)$$

Cięcie:
$$\dfrac{\Gamma \vdash \varphi \quad \Delta, \varphi \vdash \sigma}{\Gamma, \Delta \vdash \sigma}\ (\mathrm{Cut})$$

Obrazek 6: Intuicjonistyczna logika liniowa

Najpierw zobaczmy dowód klasyczny dla $((((0 \multimap p) \multimap q) \multimap r) \multimap 0) \multimap q$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{0 \vdash p,\ r}{\vdash 0 \multimap p,\ r} \quad q \vdash q}{(0 \multimap p) \multimap q \vdash r,\ q}}{\vdash ((0 \multimap p) \multimap q) \multimap r,\ q} \quad 0 \vdash}{(((0 \multimap p) \multimap q) \multimap r) \multimap 0 \vdash q}}{\vdash ((((0 \multimap p) \multimap q) \multimap r) \multimap 0) \multimap q}$$

Próbując odtworzyć dowód intuicjonistyczny widzimy, że końcówka musi wyglądać tak:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\\ (0 \multimap p) \multimap q \vdash r\end{matrix}}{\vdash ((0 \multimap p) \multimap q) \multimap r} \quad 0 \vdash q}{(((0 \multimap p) \multimap q) \multimap r) \multimap 0 \vdash q}}{\vdash ((((0 \multimap p) \multimap q) \multimap r) \multimap 0) \multimap q}$$

Próba dowodu sekwentu $(0 \multimap p) \multimap q \vdash r$ prowadzi zaś do konieczności wyprowadzenia sekwentów $\vdash 0 \multimap p$ i $q \vdash r$. To się oczywiście nie uda. Jeszcze prostszy przykład można podać, jeśli dopuścimy w logice intuicjonistycznej stałą $\perp$. Wtedy klasycznie można udowodnić

$$p, \perp \multimap \perp \vdash p$$

ale nie da się tego zrobić intuicjonistycznie. Mamy jednak też dobrą wiadomość.

Fakt 6.1 *Jeśli sekwent $\Gamma \vdash \alpha$ w języku intuicjonistycznej logiki liniowej ma dowód klasyczny i nie zawiera wystąpień przynajmniej jednego z symboli $\multimap$ i 0, to ma dowód intuicjonistyczny.*

Dowód: Rozpatrzmy klasyczny dowód bez cięcia dla $\Gamma \vdash \alpha$. Jeśli to nie jest dowód intuicjonistyczny to istnieje taki sekwent w tym dowodzie, w którym po prawej stronie są przynajmniej dwie formuły. Wybierzmy taki sekwent położony możliwie najniżej. Musi on być przesłanką reguły (L$\multimap$). Istotnie, w przypadku pozostałych reguł, które mogą wystąpić w naszym dowodzie (reguły dla $^\perp$ i $\parr$ są wykluczone z powodu własności podformuł) jeśli konkluzja ma tylko jedną formułę po prawej to i przesłanka ma jedną. A zatem nasz „zły" sekwent $\Delta \vdash \delta, \gamma$ występuje w kontekście

$$\frac{\Delta \vdash \delta, \gamma \qquad \Sigma, \beta \vdash}{\Delta, \Sigma, \delta \multimap \beta \vdash \gamma}$$

w którym druga przesłanka (L$\multimap$) ma pustą prawą stronę. Pozostaje zauważyć, że jedyną regułą, która pozwala na uzyskanie sekwentu o pustej prawej stronie jest reguła (L0). ∎

Translacja Girarda

Zobaczymy teraz, jak zwykłą logikę intuicjonistyczną można zinterpretować w intuicjonistycznej i klasycznej logice liniowej. Poniższa *translacja Girarda* oparta jest na interpretacji implikacji $\alpha \to \beta$ jako złożenia operacji zwielokrotnienia ! i liniowej implikacji $\multimap$.

Definicja 6.2 Dla dowolnej formuły φ zwykłego rachunku zdań określamy liniową formułę φ^*:

- $p^* = p$, $\mathbf{tt}^* = 1$, $\mathbf{ff}^* = 0$;
- $(\alpha \wedge \beta)^* = \alpha^* \,\&\, \beta^*$;
- $(\alpha \vee \beta)^* = !\alpha^* \oplus !\beta^*$;
- $(\alpha \to \beta)^* = !\alpha^* \multimap \beta^*$.

W definicji powyżej pominęliśmy negację, którą uważamy za szczególny przypadek implikacji ($\neg\alpha = \alpha \to \mathbf{ff}$). Zatem $(\neg\alpha)^* = !\alpha^* \multimap 0$. Jak zwykle przyjmujemy, że $\Gamma^* = \{\gamma^* \mid \gamma \in \Gamma\}$.

Dla zwięźlejszego sformułowania następnego twierdzenia przyjmijmy taką konwencję, że notacja $\Gamma \vdash \varphi$ może także oznaczać sekwent z pustą prawą stroną. W tym przypadku napis $!\Gamma^* \vdash \varphi^*$ oznacza sekwent $!\Gamma^* \vdash$.

Twierdzenie 6.3 *Następujące warunki są równoważne:*

1) $\Gamma \vdash \varphi$ *w zwykłej logice intuicjonistycznej;*

2) $!\Gamma^* \vdash \varphi^*$ *w intuicjonistycznej logice liniowej;*

3) $!\Gamma^* \vdash \varphi^*$ *w klasycznej logice liniowej.*

Dowód: (1 $\Rightarrow$ 2) Indukcja ze względu na rozmiary dowodu dla $\Gamma \vdash \varphi$ w systemie, w którym nie występują sekwenty z pustą prawą stroną, a reguły dla koniunkcji i alternatywy są addytywne (jak na Obrazku 7). Pomijamy nietrudny dowód równoważności tego systemu z innymi systemami intuicjonistycznego rachunku zdań.

Strukturalne:
$$\frac{\Gamma, \varphi, \psi, \Delta \vdash \sigma}{\Gamma, \psi, \varphi, \Delta \vdash \sigma}\ (\mathrm{LW}) \qquad \frac{\Gamma, \varphi, \varphi \vdash \sigma}{\Gamma, \varphi \vdash \sigma}\ (\mathrm{LS}) \qquad \frac{\Gamma \vdash \sigma}{\Gamma, \varphi \vdash \sigma}\ (\mathrm{LO})$$

Koniunkcja:
$$\frac{\Gamma, \varphi \vdash \sigma}{\Gamma, \varphi \wedge \psi \vdash \sigma}\ (\mathrm{LK})\ \frac{\Gamma, \psi \vdash \sigma}{\Gamma, \varphi \wedge \psi \vdash \sigma} \qquad \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi}\ (\mathrm{PK})$$

Alternatywa:
$$\frac{\Gamma, \varphi \vdash \sigma \quad \Gamma, \psi \vdash \sigma}{\Gamma, \varphi \vee \psi \vdash \sigma}\ (\mathrm{LA}) \qquad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi}\ (\mathrm{PA})\ \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi}$$

Implikacja:
$$\frac{\Gamma \vdash \varphi \quad \Delta, \psi \vdash \sigma}{\Gamma, \Delta, \varphi \to \psi \vdash \sigma}\ (\mathrm{LI}) \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}\ (\mathrm{PI})$$

Prawda i fałsz:
$$\Gamma, \mathbf{ff} \vdash \sigma \quad (\mathrm{LF}) \qquad \Gamma \vdash \mathbf{tt} \quad (\mathrm{PP})$$

Cięcie:
$$\frac{\Gamma \vdash \varphi, \quad \Delta, \varphi \vdash \sigma}{\Gamma, \Delta \vdash \sigma}\ (\mathrm{cut})$$

Obrazek 7: Rachunek sekwentów dla intuicjonistycznej logiki zdaniowej

Mamy szereg przypadków, zależnie od ostatniej reguły użytej w tym dowodzie. Aksjomat $\varphi \vdash \varphi$ tłumaczy się na:

$$\frac{!\varphi \vdash !\varphi}{!\varphi \vdash \varphi}\ (\mathrm{R}\,!)$$

Podobnie aksjomat (LF) tłumaczy się na (L0). Ponieważ po lewej stronie mamy zawsze wykrzykniki, więc przypadek (PP) i przypadki reguł strukturalnych są łatwe. Jeśli dowód sekwentu $\Gamma, \alpha \wedge \beta \vdash \sigma$ kończy się regułą (LK) zastosowaną do $\Gamma, \alpha \vdash \sigma$, to z założenia indukcyjnego mamy $!\Gamma^*, !\alpha^* \vdash \sigma^*$. Ponieważ $!(\alpha^* \mathbin{\&} \beta^*) \vdash !\alpha^*$, więc stosując cięcie dostajemy $!\Gamma^*, !(\alpha^* \mathbin{\&} \beta^*) \vdash \sigma^*$. Inne przypadki są podobne, najbardziej złożony jest ten, gdy dowód kończy się regułą (LI), tj. gdy $\Gamma, \Delta, \alpha \to \beta \vdash \sigma$ otrzymano z $\Gamma \vdash \alpha$ i $\Delta, \beta \vdash \sigma$. Założenie indukcyjne pozwala nam skonstruować taki dowód:

$$\dfrac{\dfrac{\dfrac{\dfrac{!\alpha^* \vdash !\alpha^* \quad \beta^* \vdash \beta^*}{!\alpha^* \multimap \beta^*, !\alpha^* \vdash \beta^*}}{!(!\alpha^* \multimap \beta^*), !\alpha^* \vdash \beta^*}}{!(!\alpha^* \multimap \beta^*) \vdash !\alpha^* \multimap !\beta^*} \qquad \dfrac{\dfrac{!\Gamma^* \vdash \alpha^*}{!\Gamma^* \vdash !\alpha^*} \quad !\Delta^*, !\beta^* \vdash \sigma^*}{!\Gamma^*, !\Delta^*, !\alpha^* \multimap !\beta^* \vdash \sigma^*}}{!\Gamma^*, !\Delta^*, !(!\alpha^* \multimap \beta^*) \vdash \sigma^*}$$

Pozostałe przypadki pomijamy.

($2 \Rightarrow 1$) Rozpatrzmy dowód sekwentu $!\Gamma^* \vdash \varphi^*$ w ILL i zamieńmy w nim wszystkie wystąpienia $\oplus$, $\&$, $\otimes$, $\multimap$, 1, $\top$, 0 odpowiednio na $\vee$, $\wedge$, $\wedge$, $\to$, $\mathbf{tt}$, $\mathbf{tt}$, $\mathbf{ff}$. Usuńmy też wszystkie wykrzykniki. Otrzymujemy coś, co nie musi być dowodem sekwentu $\Gamma \vdash \varphi$, ale z czego taki dowód można łatwo otrzymać. Jedyne różnice są takie: tam gdzie użyto reguł (L1) i (L0) otrzymujemy łatwe do wyprowadzenia

$$\frac{\Gamma \vdash \sigma}{\Gamma, \mathbf{tt} \vdash \sigma} \qquad \text{i} \qquad \Gamma, \mathbf{ff} \vdash \sigma,$$

a tam gdzie zastosowano (R !) i (L !) konkluzja staje się taka sama jak przesłanka. Zatem po niewielkich poprawkach uzyskamy intuicjonistyczny dowód w systemie z Obrazka 1.

Implikacja ($2 \Rightarrow 3$) jest oczywista, natomiast trudniejsza jest implikacja ($3 \Rightarrow 2$), która wymagać będzie kilku faktów pomocniczych. Przez *wysokość* dowodu rozumiemy tu wysokość drzewa (długość najdłuższej gałęzi od liścia do korzenia), nie licząc kroków dowodu stosujących reguły wymiany. Przez prostą indukcję ze względu na budowę dowodów bez cięcia pokazujemy, że w klasycznej logice liniowej:

A) *Jeśli* $\Gamma \vdash \Delta, \alpha \multimap \beta$ *ma dowód bez cięcia, to ma też dowód bez cięcia tej samej wysokości kończący się zastosowaniem reguły* (R$\multimap$)*, aksjomatu lub reguły bez przesłanek.*

B) *Jeśli* $\Gamma \vdash \Delta, \alpha \,\&\, \beta$ *ma dowód bez cięcia, to ma też dowód bez cięcia tej samej wysokości, kończący się zastosowaniem reguły* (R$\&$)*, aksjomatu lub reguły bez przesłanek.*

Dowód polega na pokazaniu, że reguły (R$\multimap$) i (R$\&$) można zawsze „permutować” z innymi regułami. A więc (R$\multimap$) i (R$\&$) mogą być zawsze stosowane jako ostatnie, chyba że użyto aksjomatu lub jednej z reguł (L0), (R$\top$). Z powyższych faktów, przez indukcję ze względu na wysokość dowodów bez cięcia, otrzymujemy taki wniosek:

C) *Jeśli dany sekwent ma dowód, to ma też dowód bez cięcia, w którym każdy sekwent postaci* $\Gamma \vdash \alpha \,\&\, \beta$ *jest konkluzją reguły* (R$\&$)*, a każdy sekwent postaci* $\Gamma \vdash \alpha \multimap \beta$ *jest konkluzją reguły* (R$\multimap$).

Na potrzeby dalszego dowodu, formułą *prostą* nazwiemy 0, każdą zmienną i każdą formułę postaci $\alpha \oplus \beta$. Sekwent jest *dobry* jeśli jego prawa strona ma jedną z form:

(a) $!\Gamma, \varphi$, gdzie φ jest prosta (b) $!\Gamma$ (c) α^*

oraz wszystkie formuły w tym sekwencie są postaci α^* lub $!\alpha^*$. Dalej powiemy, że dowód jest *dobry*, gdy składa się wyłącznie z dobrych sekwentów oraz:

- reguły (R$\multimap$) i (R$\&$) są stosowane tylko wtedy, gdy po prawej stronie przesłanki jest tylko jedna formuła.

Pokażemy teraz, że

D) *Jeśli dobry sekwent ma dowód, to ma dobry dowód.*

Postępujemy przez indukcję ze względu na rozmiar dowodu bez cięcia, spełniającego warunki faktu (C) powyżej. Przypuśćmy najpierw, że ostatnia reguła w dowodzie jest prawa. Jeśli nasz sekwent jest typu (a) to formułą wprowadzoną przez tę regułę musi być φ, bo inaczej musiałaby to być reguła (R!), a formuła φ nie jest postaci $?\varphi'$. Zatem ostatnio użyta reguła to (R1) lub (R$\oplus$), bo nie ma prawej reguły dla zera. W pierwszym przypadku dowód jest oczywiście dobry. W drugim $\varphi = !\alpha^* \oplus !\beta^*$ i poprzedni sekwent dowodu jest dobrym sekwentem typu (b). Teza wynika z założenia indukcyjnego.

Jeśli ostatni sekwent jest typu (b) to jedyna możliwa prawa reguła to (!) i wtedy nasz sekwent jest postaci $\Gamma \vdash !\alpha^*$. Poprzedni sekwent jest więc dobry typu (c). Jeśli ostatni sekwent jest typu (c) oraz $\alpha^* = \beta^* \mathbin{\&} \gamma^*$ lub $\alpha^* = \beta^* \multimap \gamma^*$ to poprzedni sekwent jest też dobry typu (c). Zauważmy, że reguła (R$\multimap$) lub (R&) została prawidłowo zastosowana, teza więc znowu wynika z założenia indukcyjnego. Dla $\alpha^* = !\beta^* \oplus !\gamma^*$ mamy poprzednik typu (b), dla $\alpha^* = 1$ poprzednika nie ma i teza jest trywialnie spełniona.

Ani $\parr$ ani $^\perp$ nie mogą wystąpić w dowodzie bez cięcia, bo dobry sekwent nie zawiera tych spójników. Zatem jedyną lewą regułą, dla której prawa strona przesłanki może być inna niż prawa strona konkluzji, jest (L$\multimap$). Dla pozostałych reguł teza jest oczywistą konsekwencją założenia indukcyjnego. Przypuśćmy więc, że ostatnią regułą w dowodzie jest (L$\multimap$) i że uzyskany sekwent jest typu (a). Ponieważ wprowadzona implikacja musi być postaci α^* lub postaci $!\alpha^*$, więc mamy tu dwie możliwości:

$$\frac{\Gamma \vdash !\beta, !\Sigma, \varphi \qquad \Delta, \gamma \vdash !\Pi}{\Gamma, \Delta, !\beta \multimap \gamma \vdash !\Sigma, !\Pi, \varphi} \qquad \text{lub} \qquad \frac{\Gamma \vdash !\beta, !\Sigma \qquad \Delta, \gamma \vdash !\Pi, \varphi}{\Gamma, \Delta, !\beta \multimap \gamma \vdash !\Sigma, !\Pi, \varphi}$$

Przesłanki są wtedy dobrymi sekwentami typów (a) i (b) lub odwrotnie. Jeśli sekwent końcowy jest typu (b) to obie przesłanki są typu (b), niech więc ostatni sekwent będzie typu (c). Teraz znowu są dwie możliwości, zależnie od tego, z której przesłanki pochodzi formuła α^*:

$$\frac{\Gamma \vdash !\beta \qquad \Delta, \gamma \vdash \alpha^*}{\Gamma, \Delta, !\beta \multimap \gamma \vdash \alpha^*} \qquad \text{lub} \qquad \frac{\Gamma \vdash !\beta, \alpha^* \qquad \Delta, \gamma \vdash}{\Gamma, \Delta, !\beta \multimap \gamma \vdash \alpha^*}$$

W pierwszym przypadku przesłanki są dobrymi sekwentami typów (b) i (c) i postępujemy jak zwykle. Podobnie w drugim przypadku jeśli α^* jest prosta, to przesłanki są dobrymi sekwentami typów (a) i (b). Jeśli α^* nie jest prosta, to albo $\alpha^* = \beta^* \mathbin{\&} \gamma^*$ albo $\alpha^* = \beta^* \multimap \gamma^*$. Ale ten przypadek jest niemożliwy, bo wtedy ostatnia reguła powinna być prawa. To kończy dowód faktu (D).

Przypuśćmy teraz, że mamy dobry dowód sekwentu postaci $!\Gamma^* \vdash \varphi^*$ w klasycznej logice liniowej. Jak w dowodzie implikacji ($2 \Rightarrow 1$) zamieniamy w tym dowodzie spójniki liniowe na „zwykłe" i otrzymujemy klasyczny dowód, w którym reguła (PI) jest stosowana tylko do sekwentów „intuicjonistycznych". Pozostaje pokazać ostatni fakt pomocniczy:

E) *Jeśli* $\Gamma \vdash \varphi_1, \ldots, \varphi_n$ *klasycznie, przy czym reguła (PI) jest stosowana tylko do sekwentów z co najwyżej jedną formułą po prawej stronie, to* $\Gamma \vdash \varphi_1 \vee \cdots \vee \varphi_n$ *intuicjonistycznie.*

Dowód (E) przebiega przez łatwą indukcję ze względu na dowód. ∎

Translacja Girarda w naturalny sposób wykorzystuje modalność !, ale nie jest to konieczne. Istnieje translacja z logiki intuicjonistycznej do fragmentu multiplikatywno-addytywnego logiki liniowej.[4] Translacja nie jest wielomianowa, ale pośrednio można z niej uzyskać dowód PSPACE-zupełności logiki multiplikatywno-addytywnej.

7 Nierozstrzygalność

Pokażemy teraz, że zdaniowa logika liniowa, zarówno klasyczna jak intuicjonistyczna jest nierozstrzygalna. Do tego potrzebne będą pewne przygotowania.

Teorie liniowe

Czasem wygodnie jest rozszerzyć system logiki liniowej o dodatkowe specyficzne aksjomaty postaci $\Gamma \vdash \alpha$. Na przykład dodanie aksjomatów $p, p, p, p \vdash q$ i $q \vdash p \otimes p \otimes p \otimes p$ pozwala używać jednego q zamiast czterech egzemplarzy p. Nam potrzebne będą tylko aksjomaty postaci $p_1, \ldots, p_n \vdash \alpha$, gdzie $p_1, \ldots, p_1$ są zmiennymi zdaniowymi. Nazwiemy je *łatwymi* aksjomatami. Twierdzenie o eliminacji cięcia dla takiej *teorii* liniowej przyjmuje następującą postać.

Twierdzenie 7.4 *Przypuśćmy, że sekwent* $\Delta \vdash \Pi$ *ma dowód w teorii otrzymanej przez dodanie pewnej liczby łatwych aksjomatów do systemu logiki liniowej. Wtedy istnieje dowód dla* $\Delta \vdash \Pi$*, w którym wszystkie cięcia są rezolucjami postaci:*

$$\frac{\Gamma \vdash \alpha \qquad \alpha, \Sigma \vdash \Xi}{\Gamma, \Sigma \vdash \Xi},$$

gdzie $\Gamma \vdash \alpha$ *jest łatwym aksjomatem.*

Dowód: Postępujemy podobnie jak przy zwykłej eliminacji cięcia, traktując rezolucje jak zwykłe reguły i eliminując pozostałe cięcia. ∎

Automaty z licznikami

Definicja 7.5 *Dwulicznikowym automatem alternującym* nazwiemy układ $\mathcal{A} = \langle Q, q_0, q_f, \mathcal{I} \rangle$, gdzie Q jest skończonym zbiorem *stanów*, z wyróżnionym stanem *początkowym* q_0 i stanem *końcowym* q_f, a funkcja $\mathcal{I}$ przypisuje każdemu stanowi q (być może za wyjątkiem stanu q_f) *instrukcję* $\mathcal{I}_q$ jednej z postaci:

1) $(q : c := c + 1;\ \texttt{goto}\ p)$;

2) $(q : c := c - 1;\ \texttt{goto}\ p)$;

3) $(q : d := d + 1;\ \texttt{goto}\ p)$;

4) $(q : d := d - 1;\ \texttt{goto}\ p)$;

[4] Lincoln, P., Scedrov, A., Shankar, N., Linearizing intuitionistic implication, *Annals of Pure and Applied Logic* 60 (1993) 151–177.

5) $(q : \texttt{goto}\ p\ \texttt{or}\ r)$;

6) $(q : \texttt{goto}\ p\ \texttt{and}\ r)$,

gdzie $p, r \in Q$. *Konfiguracją* automatu nazwiemy trójkę (q, m, n), gdzie q jest stanem, a liczby $m, n \in \mathbb{N}$ interpretujemy jako wartości liczników c i d. Konfigurację $(q_0, 0, 0)$ nazywamy *początkową*, a konfigurację $(q_f, 0, 0)$ *końcową*. Jeśli instrukcja $\mathcal{I}_q$ jest typu (6) to konfiguracja (q, m, n) jest *uniwersalna*, inne konfiguracje są *egzystencjalne*.

Mówimy, że konfiguracja (q, m, n) jest *akceptowana*, gdy jest to konfiguracja końcowa, lub gdy instrukcja $\mathcal{I}_q$ jest postaci

(1) oraz konfiguracja $(p, m + 1, n)$ jest akceptowana.

(2) oraz $m > 0$ i konfiguracja $(p, m - 1, n)$ jest akceptowana.

(3) oraz konfiguracja $(p, m, n + 1)$ jest akceptowana.

(4) oraz $n > 0$ i konfiguracja $(p, m, n - 1)$ jest akceptowana.

(5) oraz przynajmniej jedna z konfiguracji (p, m, n), (r, m, n) jest akceptowana.

(6) oraz obie konfiguracje (p, m, n), (r, m, n) są akceptowane.

Problem stopu dla dwulicznikowych automatów alternujących formułujemy tak: *Czy konfiguracja początkowa danego automatu jest akceptowana?*

Lemat 7.6 *Problem stopu dla dwulicznikowych automatów alternujących jest nierozstrzygalny.*

Dowód: Jak wiadomo, problem stopu dla deterministycznych automatów z dwoma licznikami jest nierozstrzygalny. Takie automaty mają jednak instrukcje postaci

7) $(q : \texttt{if}\ c = 0\ \texttt{then goto}\ p\ \texttt{else goto}\ r)$;

8) $(q : \texttt{if}\ d = 0\ \texttt{then goto}\ p\ \texttt{else goto}\ r)$,

którymi nie dysponujemy. Test na zero można jednak symulować z pomocą alternacji. Na przykład instrukcję (7) zastępujemy przez taki ciąg[5] instrukcji ($p_1, \ldots, p_5$ są nowymi stanami):

$$\begin{aligned}
q\ &: \texttt{goto}\ p_1\ \texttt{or}\ p_3;\\
p_1 &: c := c - 1; \texttt{goto}\ p_2;\\
p_2 &: c := c + 1; \texttt{goto}\ r;\\
p_3 &: \texttt{goto}\ p_4\ \texttt{and}\ p;\\
p_4 &: \texttt{goto}\ p_5\ \texttt{or}\ q_f;\\
p_5 &: d := d - 1; \texttt{goto}\ p_4;
\end{aligned}$$

Przypuśćmy, że konfiguracja (q, m, n) jest akceptowana przez „nowy” automat alternujący. Jeśli akceptowana jest konfiguracja (p_1, m, n) i dalej (r, m, n), to jest to możliwe tylko dla

[5] Może ktoś ma ochotę zrobić obrazek?

$m \neq 0$. Druga możliwość jest taka, że akceptowana jest konfiguracja (p_3, m, n) i dalej (p, m, n) i (p_4, m, n) Ale to ostatnie może się zdarzyć tylko dla $m = 0$. A zatem w obu przypadkach (q, m, n) jest akceptowana przez „stary" automat deterministyczny. Podobnie argumentując stwierdzimy, że zachodzi też implikacja odwrotna, a więc nasze automaty są równoważne z punktu widzenia akceptowania konfiguracji początkowej. (Ścisły dowód należałoby przeprowadzić przez indukcję ze względu na definicję akceptowania.) ∎

Ustalmy teraz pewien dwulicznikowy automat alternujący $\mathcal{A} = \langle Q, q_0, q_f, \mathcal{I}\rangle$, dla którego zdefiniujemy teorię liniową o aksjomatach wyznaczonych przez instrukcje automatu. Dla uproszczenia przyjmijmy, że wszystkie stany automatu mogą występować w roli zmiennych zdaniowych. Użyjemy też zmiennych zdaniowych c i d reprezentujących liczniki. Dla dowolnej instrukcji automatu $\mathcal{I}_q$ mamy jeden lub dwa aksjomaty jak następuje:

(1) $q \vdash p \otimes c$;

(2) $q, c \vdash p$;

(3) $q \vdash p \otimes d$;

(4) $q, d \vdash p$;

(5) $q \vdash p$ oraz $q \vdash r$;

(6) $q \vdash p \oplus r$.

Lemat 7.7 *Konfiguracja (q, m, n) jest akceptowana wtedy i tylko wtedy, gdy $q, c^m, d^n \vdash q_f$.*

Dowód: Zaczniemy od łatwiejszej implikacji z lewej do prawej. Stosujemy indukcję ze względu na definicję akceptowania. Skoro $q_f \vdash q_f$, to teza zachodzi dla konfiguracji końcowej. Jeśli $\mathcal{I}_q$ jest typu (1) to z założenia indukcyjnego mamy $p, c^{m+1}, d \vdash q_f$ i stosujemy rezolucję:

$$\frac{q \vdash p \otimes c \qquad \dfrac{p, c^{m+1}, d^n \vdash q_f}{p \otimes c,\ c^m, d^n \vdash q_f}}{q, c^m, d^n \vdash q_f}$$

Jeśli $\mathcal{I}_q$ jest typu (6) to z założenia indukcyjnego wynika $p, c^m, d^n \vdash q_f$ i $r, c^m, d^n \vdash q_f$. A więc:

$$\frac{q \vdash p \oplus r \qquad \dfrac{p,\ c^m, d^n \vdash q_f \qquad r,\ c^m, d^n \vdash q_f}{p \oplus r,\ c^m, d^n \vdash q_f}}{q, c^m, d^n \vdash q_f}$$

Pozostałe przypadki pomijamy i przechodzimy do dowodu implikacji odwrotnej. Załóżmy, że sekwent $q, c^m, d^n \vdash q_f$ ma dowód. Ponieważ występują tu tylko zmienne, więc można zakładać, że ostatnim krokiem dowodu jest cięcie takie jak w Twierdzeniu 7.4. Po lewej stronie każdego aksjomatu występuje stan maszyny, który musi pozostać w konkluzji, a więc użyty aksjomat odpowiada instrukcji $\mathcal{I}_q$. Jeśli jest to instrukcja typu (1), to nasz dowód kończy się tak

$$\frac{q \vdash p \otimes c \qquad p \otimes c,\, c^m, d^n \vdash q_f}{q, c^m, d^n \vdash q_f}$$

Ostatnią regułą w dowodzie sekwentu $p \otimes c,\ c^m,\ d^n \vdash q_f$ nie może być rezolucja, bo po lewej stronie nie ma zmiennej reprezentującej stan. A więc sekwent ten został otrzymany z $p,\ c^{m+1}, d^n \vdash q_f$ za pomocą reguły (L$\otimes$). Na mocy założenia indukcyjnego konfiguracja $(p, m+1, n)$ jest akceptowana, więc także (q, m, n) jest akceptowana.

Rozpatrzmy jeszcze przypaadek, gdy $\mathcal{I}_q$ jest typu (6) i dowód kończy się tak:

$$\frac{q \vdash p \oplus r \qquad p \oplus r,\, c^m, d^n \vdash q_f}{q, c^m, d^n \vdash q_f}$$

Jak w poprzednim przypadku zauważamy, że sekwent $p \oplus r,\ c^m, d^n \vdash q_f$ otrzymano z pomocą reguły (L$\oplus$) i stosujemy założenie indukcyjne do sekwentów $p, c^m, d^n \vdash q_f$ i $r, c^m, d^n \vdash q_f$. Resztę dowodu opuszczamy. ∎

Twierdzenie 7.8 *Zdaniowa logika liniowa jest nierozstrzygalna.*

Dowód: Niech $\mathcal{A}$ będzie dwulicznikowym automatem alternującym. Przypuśćmy, że aksjomaty wyznaczone przez ten automat są postaci $p_{j1}, \ldots, p_{jk_j} \vdash \alpha_j$ dla $j = 1, \ldots, n$. Z Lematu 7.7 wynika, że konfiguracja początkowa automatu $\mathcal{A}$ jest akceptowana wtedy i tylko wtedy, gdy formuła $q_0 \vdash q_f$ ma dowód używający tych aksjomatów. Jest to równoważne temu, że formuła

$$!(p_{11} \multimap \ldots \multimap p_{1k_1} \multimap \alpha_1) \multimap \ldots \multimap\, !(p_{n1} \multimap \ldots \multimap p_{nk_n} \multimap \alpha_n) \multimap q_0 \multimap q_f$$

jest twierdzeniem liniowym. ∎

Uwaga: W dowodzie nierozstrzygalności użyto tylko spójników $\otimes$, $\oplus$, $\multimap$, $!$. Ponieważ w tym fragmencie klasyczna logika liniowa jest konserwatywna nad intuicjonistyczną (Fakt 6.1), więc Wniosek 7.8 zachodzi też dla tej ostatniej.

Niektóre fragmenty logiki liniowej są rozstrzygalne, na przykład wiadomo, że fragment multiplikatywny (spójniki $\otimes$, $⅋$, 1, $\bot$) jest NP-zupełny, a fragment multiplikatywno-addytywny (bez modalności) jest PSPACE-zupełny.[6][7] (Łatwo zauważyć, że wysokość multiplikatywno-addytywnego dowodu bez cięcia jest liniowa ze względu na rozmiar sekwentu wynikowego — zatem problem jest oczywiście w PSPACE. Trudność wynika np. z przypisu 4, lub z bezpośredniego kodowania problemu QBF.)

Otwartym problemem pozostaje rozstrzygalność fragmentu multiplikatywno-wykładniczego, oraz fragmentu *tensor-bang-lollipop*: $\otimes$, $!$, $\multimap$. Zauważmy, że bez $\oplus$ konstrukcja z Lematu 7.7 wystarczy do reprezentowania niedeterministycznych automatów bez testu na zero. Takie automaty (bez stanów) to w istocie *sieci Petriego*.

[6] Lincoln, P., Mitchell, J., Scedrov, A., Shankar, N., Decision problems for propositional linear logic, *Annals of Pure and Applied Logic* 56 (1992) 239–311.

[7] Lincoln, P., Deciding provability of linear logic formulas, in: *Advances in Linear Logic*, (Girard, J.-Y., Lafont, Y., Regnier, L., eds.) Cambridge University Press, 1995, pp. 109–122.

Definicja 7.9 Krotkę postaci $\mathcal{N} = \langle S, T, F \rangle$, gdzie S jest skończonym zbiorem *miejsc*, T jest skończonym zbiorem *przejść* (przy czym $S \cap T = \emptyset$), a $F : (S \times T) \cup (T \times S) \to \mathbb{N}$ nazywamy *siecią Petriego*. Każda funkcja $C : S \to \mathbb{N}$ nazywana jest *konfiguracją* sieci. Konfiguracja C' jest *osiągalna w jednym kroku* z konfiguracji C, gdy istnieje takie przejście t, że dla dowolnego miejsca $s \in S$ zachodzą warunki $C(s) - F(s,t) \geq 0$ oraz $C'(s) = C(s) - F(s,t) + F(t,s)$. Natomiast C' jest po prostu *osiągalna* z C gdy istnieje taki ciąg $C = C_0, C_1, \ldots, C_n = C'$, że każda C_{i+1} jest osiągalna w jednym kroku z C_i. Możemy to zapisać tak: $C \twoheadrightarrow C'$.

Grafowa interpretacja Definicji 7.9 jest taka: Wartość $F(s,t)$ to liczba krawędzi od miejsca s do przejścia t, a wartość $F(t,s)$ to liczba krawędzi w przeciwną stronę. Konfiguracja sieci określa liczbę pionków znajdujących się w każdym miejscu sieci.[8] Zmiana konfiguracji (poprzez przejście t) składa się z dwóch faz: w fazie pierwszej zabieramy $F(s,t)$ pionków z każdego miejsca s, w fazie drugiej umieszczamy $F(t,s)$ nowych pionków w każdym s. Jeśli pionków jest gdzieś za mało, to taka zmiana jest niewykonalna nawet gdy $-F(s,t) + F(t,s) > 0$.

Problem osiągalności dla sieci Petriego polega na rozstrzygnięciu, czy w danej sieci jedna konfiguracja jest osiągalna z innej. Mamy takie sławne twierdzenie z trudnym dowodem:

Twierdzenie 7.10 *Problem osiągalności dla sieci Petriego jest rozstrzygalny.*

Bardziej abstrakcyjnym modelem, podobnym do sieci Petriego, jest *system dodawania wektorów* (VAS). Taki system to po prostu skończony zbiór wektorów $v_1, \ldots, v_n \in \mathbb{Z}^k$. Stany systemu to wektory liczb naturalnych, a wektor u jest *osiągalny* z wektora w w jednym kroku, gdy $w + v_i = u$ dla pewnego i. Osiągalność w wielu krokach definiuje się w oczywisty sposób. Nietrudo zauważyć, że problem osiągalności dla VAS jest tak samo trudny jak dla sieci Petriego.

Twierdzenie 7.11 *Problem osiągalności dla sieci Petriego (systemów dodawania wektorów) redukuje się w pamięci logarytmicznej do logiki tensor-bang-lollipop*

Dowód: Postepujemy podobnie jak w dowodzie Twierdzenia 7.8, tj. definiujemy teorię liniową zawierającą dla każdego przejścia t aksjomat

$$s_1^{F(s_1,t)}, \ldots, s_n^{F(s_n,t)} \vdash s_1^{F(t,s_1)} \otimes \cdots \otimes s_n^{F(t,s_n)},$$

gdzie $s_1, \ldots, s_n$ to wszystkie stany sieci. Ze względu na brak stanów dowód jest nieco trudniejszy. Aby go ułatwić, zauważmy najpierw, że

$$\textit{jeśli } C_1 \twoheadrightarrow C_1' \textit{ i } C_2 \twoheadrightarrow C_2' \textit{ to } C_1 + C_2 \twoheadrightarrow C_1' + C_2', \tag{$*$}$$

gdzie suma konfiguracji jest „po współrzędnych", tj. $C_1 + C_2(s) = C_1(s) + C_2(s)$. Powiedzmy teraz, że formuła $s_{i_1}^{r_1} \otimes \cdots \otimes s_{i_m}^{r_m}$ *reprezentuje* konfigurację C, gdy dla dowolnego $s_i \in S$, wartość $C(s_i)$ jest sumą liczb r_j dla $i_j = i$. Jeśli formuła $\alpha_1 \otimes \cdots \otimes \alpha_d$ reprezentuje C, to powiemy też, że ciąg formuł $\alpha_1, \ldots, \alpha_d$ *reprezentuje* C.

Lemat, który należy udowodnić, jest taki: *Przypuśćmy, że ciąg* Γ *reprezentuje konfigurację* C*, natomiast formuła* α *reprezentuje konfigurację* C'*. Konfiguracja* C' *jest osiągalna z* C *wtedy i tylko wtedy, gdy w teorii jak wyżej można wyprowadzić sekwent* $\Gamma \vdash \alpha$.

[8] Obrazek mile widziany :)

Dowód implikacji z lewej do prawej jest łatwy (por. Lemat 7.7). Implikacja odwrotna wymaga badania dowodu z rezolucjami. Tym razem taki dowód może się kończyć krokiem typu (R$\otimes$) i wtedy korzystamy z obserwacji ($*$). Jeśli dowód kończy się zastosowaniem rezolucji, to korzystamy z tego, że nasza definicja reprezentowania ignoruje rozmieszczenie tensorów po lewej stronie sekwentu. Szczegóły pozostawiamy czytelnikowi. ∎

Twierdzenie 7.11 ustala jedynie *redukcję* w jedną stronę. Można jednak sformułować problem dotyczący automatów na drzewach, którego rozstrzygalność jest *równoważna* rozstrzygalności fragmentu *tensor-bang-lollipop* logiki liniowej.[9]

8 Naturalna dedukcja i termy liniowe

Reguły naturalnej dedukcji dla intuicjonistycznej logiki liniowej przedstawione są na Obrazku 8. Dla każdego spójnika mamy regułę (lub reguły) wprowadzania tego spójnika i regułę (lub reguły) eliminacji tego spójnika. Reguły wprowadzania służą do dowodzenia formuł zbudowanych z odpowiednich spójników, reguły eliminacji mówią jak można wykorzystać formułę danej postaci do wyprowadzenia innej formuły. Wszystko to jest zrelatywizowane do multizbioru założeń: napis $\Gamma \vdash \varphi$ czytamy „formuła φ wynika z multizbioru założeń Γ.”

$$\varphi \vdash \varphi \ \text{(aksjomat)}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \,\&\, \psi}\ (+\&) \qquad \frac{\Gamma \vdash \varphi \,\&\, \psi}{\Gamma \vdash \varphi}(-\&)\frac{\Gamma \vdash \varphi \,\&\, \psi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \varphi \quad \Delta \vdash \psi}{\Gamma, \Delta \vdash \varphi \otimes \psi}\ (+\otimes) \qquad \frac{\Gamma, \varphi, \psi \vdash \sigma \quad \Delta \vdash \varphi \otimes \psi}{\Gamma, \Delta \vdash \sigma}\ (-\otimes)$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \oplus \psi}\ (+\oplus)\ \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \oplus \psi} \qquad \frac{\Gamma, \varphi \vdash \sigma \quad \Gamma, \psi \vdash \sigma \quad \Delta \vdash \varphi \oplus \psi}{\Gamma, \Delta \vdash \sigma}\ (-\oplus)$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \multimap \psi}\ (+\multimap) \qquad \frac{\Gamma \vdash \varphi \multimap \psi \quad \Delta \vdash \varphi}{\Gamma, \Delta \vdash \psi}(-\multimap)$$

$$\vdash 1\ (+1) \qquad \frac{\Gamma \vdash \sigma \quad \Delta \vdash 1}{\Gamma, \Delta \vdash \sigma}\ (-1) \qquad \Gamma \vdash \top\ (+\top) \qquad \frac{\Delta \vdash 0}{\Gamma, \Delta \vdash \varphi}\ (-0)$$

$$\frac{\Gamma \vdash \sigma \quad \Delta \vdash !\varphi}{\Gamma, \Delta \vdash \sigma}\ (-\mathrm{O}!) \qquad \frac{\Gamma, !\varphi, !\varphi \vdash \sigma \quad \Delta \vdash !\varphi}{\Gamma, \Delta \vdash \sigma}\ (-\mathrm{S}!)$$

$$\frac{\Delta_1 \vdash !\gamma_1 \cdots \Delta_n \vdash !\gamma_n \quad !\Gamma \vdash \varphi}{\Delta_1, \ldots, \Delta_n \vdash !\varphi}\ (+!) \qquad \frac{\Gamma \vdash !\varphi}{\Gamma \vdash \varphi}\ (-\mathrm{D}!)$$

Obrazek 8: Naturalna dedukcja

Reguły naturalnej dedukcji odzwierciedlają konstruktywny sens spójników liniowych. Na przy-

[9] de Groote, Ph., Guillaume, B., Salvati, S., Vector addition tree automata, in: *Proc. 19th Annual Symp. on Logic in Computer Science* (LICS 2004), pp. 64–73.

kład uzasadnienie formuły $\varphi \,\&\, \psi$ polega na uzasadnieniu dwóch alternatywnych konstrukcji: dla φ i dla ψ, wykorzystujących potencjalnie te same zasoby. Dla $\otimes$, zasoby muszą być osobne, a reguła $(-\otimes)$ mówi, że wykorzystanie tensora jest możliwe wtedy, gdy umiemy wykorzystać obie składowe. W przypadku modalności ! mamy trzy sposoby wykorzystania konstrukcji $!\varphi$ wyrażone przez reguły $(-\mathrm{O}!)$, $(-\mathrm{S}!)$ i $(-\mathrm{D}!)$. Nieco dziwna jest reguła wprowadzania wykrzyknika $(+!)$, naturalniejsze na pierwszy rzut oka jest coś takiego:

$$\frac{!\Gamma \vdash \varphi}{!\Gamma \vdash !\varphi}\ (\text{plus}\,!)$$

Ta reguła okaże się jednak za chwilę technicznie niewygodna.

Jak wiadomo, dowody w intuicjonistycznym rachunku zdań można z powodzeniem reprezentować za pomocą termów rozszerzonego rachunku lambda. Także dla logiki liniowej możemy zdefiniować odpowiedni rachunek lambda. Reguły przypisania typów dla tego rachunku to odpowiednio „udekorowane” reguły naturalnej dedukcji (Obrazek 9).

$$x : \varphi \vdash x : \varphi \ (\text{aksjomat})$$

$$\frac{\Gamma \vdash M : \varphi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N\rangle : \varphi \,\&\, \psi}\ (+\&) \qquad \frac{\Gamma \vdash M : \varphi \,\&\, \psi}{\Gamma \vdash \pi_1(M) : \varphi}(-\&)\frac{\Gamma \vdash M : \varphi \,\&\, \psi}{\Gamma \vdash \pi_2(M) : \psi}$$

$$\frac{\Gamma \vdash M : \varphi \quad \Delta \vdash N : \psi}{\Gamma, \Delta \vdash M \otimes N : \varphi \otimes \psi}\ (+\otimes) \qquad \frac{\Gamma, x : \varphi, y : \psi \vdash M : \sigma \quad \Delta \vdash N : \varphi \otimes \psi}{\Gamma, \Delta \vdash \mathbf{let}\ x \otimes y = N\ \mathbf{in}\ M : \sigma}\ (-\otimes)$$

$$\frac{\Gamma \vdash M : \varphi}{\Gamma \vdash \mathbf{inl}_{\varphi\oplus\psi}(M) : \varphi \oplus \psi}\ (+\oplus) \ \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathbf{inr}_{\varphi\oplus\psi}(M) : \varphi \oplus \psi}$$

$$\frac{\Gamma, x : \varphi \vdash M : \sigma \quad \Gamma, y : \psi \vdash N : \sigma \quad \Delta \vdash P : \varphi \oplus \psi}{\Gamma, \Delta \vdash \mathbf{case}\ P\ \mathbf{of}\ [x]M, [y]N : \sigma}\ (-\oplus)$$

$$\frac{\Gamma, x : \varphi \vdash M : \psi}{\Gamma \vdash \lambda x.\, M : \varphi \multimap \psi}\ (+\multimap) \qquad \frac{\Gamma \vdash M : \varphi \multimap \psi \quad \Delta \vdash N : \varphi}{\Gamma, \Delta \vdash MN : \psi}(-\multimap)$$

$$\vdash \bullet : 1 \ (+1) \qquad \frac{\Gamma \vdash M : \sigma \quad \Delta \vdash P : 1}{\Gamma, \Delta \vdash \mathbf{let}\ \bullet = P\ \mathbf{in}\ M : \sigma}\ (-1)$$

$$\vec{x} : \Gamma \vdash \mathit{Top}(\vec{x}) : \top \ (+\top) \qquad \frac{\vec{y} : \Delta \vdash M : 0}{\vec{x} : \Gamma, \vec{y} : \Delta \vdash \varepsilon_\varphi(\vec{x}, M) : \varphi}\ (-0)$$

$$\frac{\Gamma \vdash M : \sigma \quad \Delta \vdash N : !\varphi}{\Gamma, \Delta \vdash \mathbf{discard}\ N\ \mathbf{before}\ M : \sigma}\ (-\mathrm{O}!) \quad \frac{\Gamma, x : !\varphi, y : !\varphi \vdash M : \sigma \quad \Delta \vdash N : !\varphi}{\Gamma, \Delta \vdash \mathbf{share}\ N\ \mathbf{as}\ x, y\ \mathbf{in}\ M : \sigma}\ (-\mathrm{S}!)$$

$$\frac{\Delta_1 \vdash P_1 : !\gamma_1 \cdots \Delta_n \vdash P_n : !\gamma_n \quad \vec{x} : !\Gamma \vdash M : \varphi}{\Delta_1, \ldots, \Delta_n \vdash \mathbf{promote}\ P_1, \ldots, P_n\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ M : !\varphi}\ (+!) \quad \frac{\Gamma \vdash N : !\varphi}{\Gamma \vdash \mathbf{consume}(N) : \varphi}\ (-\mathrm{L}!)$$

Obrazek 9: Liniowy rachunek lambda

„Kanoniczny” obiekt typu $\varphi \,\&\, \psi$ to para (rekord) złożona z dwóch „metod”, z których tylko jedną można (trzeba) wykorzystać. Operacja wyboru współrzędnej i to nic innego tylko rzutowanie $\pi_i(M)$. Obiekt postaci $M \otimes N$ to też para, ale taka, której oba człony muszą być wykorzystane (reprezentuje to wyrażenie **let**...). Typ $\varphi \oplus \psi$ to suma prosta typów φ i ψ, z którym związane są włożenia (lewe i prawe) oraz instrukcja **case**. Trzy sposoby wykorzystania powtarzalnej danej typu $!\varphi$ wyrazone są instrukcjami **discard**, **share**, i **consume**. Instrukcja **promote** realizuje „zwielokrotnienie" obiektu M w kontekście, w którym wszystkim jego zmiennym wolnym $\vec{x}$ można przypisać „znaczenia" $\vec{P}$ odpowiednich typów.

W termach liniowych występują zmienne wolne i związane. W termie **let** $x \otimes y = N$ **in** M zmienne x i y są związane w M (ale nie w N). W termie **case** P **of** $[x]M, [y]N$, zakresem wiązania x jest M, a zakresem wiązania y jest N. Podobnie, M jest zakresem wiązania x, y w termie **share** N **as** x, y **in** M i zakresem wiązania $\vec{x}$ w termie **promote** $P_1, \ldots, P_n$ **for** $\vec{x}$ **in** M.

Normalizacja i redukcja

Dowód, w którym bezpośrednio po wprowadzeniu jakiegoś spójnika następuje jego eliminacja, można zwykle „uprościć". Na ogół polega to na zastąpieniu pewnych założeń dowodami odpowiednich formuł. Taka *normalizacja* dowodów podobna jest w naturze do eliminacji cięcia. Dowód *normalny* nie zawierający fragmentów dających się „uprościć" ma zwykle bardzo prostą budowę, i jest łatwiejszy do analizy.

Za pomocą notacji termowej łatwo jest sformalizować zjawisko normalizacji z pomocą reguł beta-redukcji. Redukcje liniowego rachunku lambda mamy na Obrazku 10.

$$(\lambda x.\, M)N \quad \to \quad M[N/x]; \qquad\qquad \pi_i(\langle M_1, M_2\rangle) \quad \to \quad M_i;$$

$$\mathbf{let}\ \bullet = \bullet\ \mathbf{in}\ M \quad \to \quad M; \qquad\qquad \mathbf{let}\ x \otimes y = P \otimes Q\ \mathbf{in}\ M \quad \to \quad M[P/x, Q/y];$$

$$\mathbf{case}\ \mathbf{inl}(P)\ \mathbf{of}\ [x]M, [y]N \quad \to \quad M[P/x]; \qquad \mathbf{case}\ \mathbf{inr}(Q)\ \mathbf{of}\ [x]M, [y]N \quad \to \quad N[Q/y];$$

$$\mathbf{discard}\ (\mathbf{promote}\ \vec{P}\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ N)\ \mathbf{before}\ M \quad \to \quad \mathbf{discard}\ \vec{P}\ \mathbf{before}\ M;$$

$$\mathbf{share}\ (\mathbf{promote}\ \vec{P}\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ M)\ \mathbf{as}\ y, z\ \mathbf{in}\ N \quad \to$$
$$\qquad \mathbf{share}\ \vec{P}\ \mathbf{as}\ \vec{x}_1, \vec{x}_2\ \mathbf{in}\ N[(\mathbf{promote}\ \vec{x}_1\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ M)/y, (\mathbf{promote}\ \vec{x}_2\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ M)/z];$$

$$\mathbf{consume}(\mathbf{promote}\ \vec{P}\ \mathbf{for}\ \vec{x}\ \mathbf{in}\ M) \quad \to \quad M[\vec{P}/\vec{x}];$$

Obrazek 10: Reguły redukcji

Poniższe fakty podamy bez dowodu:

Twierdzenie 8.12 *Liniowy rachunek lambda ma następujące własności:*

1. *Typy zachowują się przy redukcji: jeśli* $\Gamma \vdash M : \varphi$ *oraz* $M \twoheadrightarrow M'$ *to* $\Gamma \vdash M' : \varphi$.
2. *Własność Churcha-Rossera: jeśli* $M \twoheadrightarrow M_1$ *i* $M \twoheadrightarrow M_2$ *to* $M_1 \twoheadrightarrow M_3$ *i* $M_2 \twoheadrightarrow M_3$*, dla pewnego* M_3.
3. *Silna normalizacja: jeśli* $\Gamma \vdash M : \varphi$*, to każdy ciąg redukcji* $M = M_0 \to M_1 \to M_2 \to \cdots$ *jest skończony.*

Silną normalizację dla liniowego rachunku lambda udowodnił Nick Benton[10] sprowadzając zadanie do silnej normalizacji polimorficznego rachunku lambda.

Zauważmy jeszcze, że gdybyśmy zamiast reguły (+!) przyjęli coś takiego:

$$\frac{!\Gamma \vdash M : \varphi}{!\Gamma \vdash \mathbf{lift}(M) : !\varphi}\ (\text{plus}\,!)$$

to przy naturalnych regułach redukcji:

$$\begin{aligned}\mathbf{discard\ lift}(M)\ \mathbf{before}\ N \quad &\to \quad N;\\ \mathbf{share\ lift}(M)\ \mathbf{as}\ x, y\ \mathbf{in}\ N \to \quad &N[x := M, y := M],\end{aligned}$$

Twierdzenie 8.12 nie byłoby prawdziwe. Przyczyną jest brak zamkniętości ze względu na podstawienie. Mamy bowiem na przykład

$$y : q \multimap !p,\ z : q \vdash yz : !p, \qquad \text{oraz} \qquad x : !p \vdash \mathbf{lift}(\mathbf{consume}(x)) : !p,$$

ale $$y : q \multimap !p,\ z : q \nvdash \mathbf{lift}(\mathbf{consume}(yz)) : !p,$$

a stąd wyniknie, że term $(\lambda x.\,\mathbf{lift}(\mathbf{consume}(x)))(yz)$ „straci" typ $!p$ na skutek redukcji.

Jeśli M nie ma zmiennych wolnych, to oczywiście można zdefiniować

$$\mathbf{lift}(M) := \mathbf{promote\ for\ in}\ M.$$

Programowanie z liniowymi typami

Dodawanie liczb naturalnych możemy z pomocą następnika i poprzednika zdefiniować tak:

$$\mathit{Add}(x, y) := \mathbf{if}\ \mathit{zero}(x)\ \mathbf{then}\ y\ \mathbf{else}\ \mathit{Add}(\mathit{pred}(x), \mathit{succ}(y))$$

Jeśli teraz *zero* jest typu $\mathbf{int} \to \mathbf{bool}$ a funkcje *pred* i *succ* są typu $\mathbf{int} \to \mathbf{int}$ to tak zdefiniowana funkcja *Add* jest typu $\tau = \mathbf{int} \times \mathbf{int} \to \mathbf{int}$. Ściśle rzecz biorąc, w naszej definicji uczestniczy jednak jeszcze jedna funkcja: jest to operator punktu stałego **Fix**, który jest w tym przypadku typu $(\tau \to \tau) \to \tau$. Nasza definicja może więc być zapisana tak:

$$\mathit{Add} = \mathbf{Fix}(\lambda A \lambda X.\,\mathbf{if}\ \mathit{zero}(\pi_1(X))\ \mathbf{then}\ \pi_2(X)\ \mathbf{else}\ A(\mathit{pred}(\pi_1(X)), \mathit{succ}(\pi_2(X)))).$$

Status zmiennych x i y w tej definicji jest inny: drugi argument jest używany raz, pierwszy raz albo dwa razy. Ale oba argumenty są używane. Zatem w liniowym języku funkcyjnym, operacja *Add* powinna mieć typ $\sigma = !\mathbf{int} \otimes \mathbf{int} \multimap \mathbf{int}$. Będziemy jednak mieli problem z jej zdefiniowaniem: funkcja, do której stosujemy operator punktu stałego może nie używać argumentu A. Typem tego argumentu powinno być więc $!\sigma$, co zresztą nie jest od rzeczy, jeśli pomyślimy o tym, że dodawanie jest operacją wielokrotnego użytku.

Przypuśćmy, że $\mathit{pred} : \mathbf{int} \multimap \mathbf{int}$ i niech $\mathit{Pred}(x) = \mathbf{promote}\ x\ \mathbf{for}\ x'\ \mathbf{in}\ \mathit{pred}(\mathbf{consume}(x'))$. Dla $x : !\mathbf{int}$, term $\mathit{Pred}(x)$ ma też typ $!\mathbf{int}$. Jeśli $A : !\sigma$, $x : !\mathbf{int}$, $y : \mathbf{int}$, to term

$$\mathbf{share}(x)\ \mathbf{as}\ x_1, x_2\ \mathbf{in\ if}\ \mathit{zero}(\mathbf{consume}(x_1))\ \mathbf{then\ discard}\ x_2\ \mathbf{before}\ y\ \mathbf{else}\ A(\mathit{Pred}(x_2), y)$$

[10] Benton, N., Strong normalisation for the linear term calculus, *J. Functional Programming* 5 (1995), 65–80.

ma typ **int**. A więc dla $X : !\mathbf{int} \otimes \mathbf{int}$ także następujące wyrażenie $M(A, X)$ jest typu **int**.

$$\begin{array}{l}\mathbf{let}\ x \otimes y = X\ \mathbf{in}\ \mathbf{share}(x)\ \mathbf{as}\ x_1, x_2\ \mathbf{in} \\ \qquad\qquad\qquad \mathbf{if}\ \mathit{zero}(\mathbf{consume}(x_1))\ \mathbf{then}\ \mathbf{discard}\ x_2\ \mathbf{before}\ y\ \mathbf{else}\ A(\mathit{Pred}(x_2), y).\end{array}$$

Zatem funkcja $\lambda A \lambda X. M(A, X)$ ma typ $!\sigma \multimap \sigma$ i możemy zdefiniować dodawanie $\mathit{Add}' : \sigma$ jeśli mamy stałą $\mathbf{Fix} : (!\sigma \multimap \sigma) \multimap \sigma$. Ponieważ jednak zarówno **Fix** jak $\lambda A \lambda X. M(A, X)$ nie mają zmiennych wolnych, więc możemy ostatecznie przyjąć,[11] że $\mathit{Add} = \mathbf{lift}(\mathit{Add}')$.

9 Semantyka fazowa

Niech $\langle M, \cdot, \varepsilon \rangle$ będzie półgrupą przemienną z jednością i niech $\perp \subsetneq M$ będzie wyróżnionym podzbiorem M. Taką półgrupę nazywamy *przestrzenią fazową* i na podzbiorach $X \subseteq M$ określamy dwie operacje:

$$\begin{array}{lcl} X^{\perp} & = & \{z \in M \mid \forall x{\in}M (x \in X \Rightarrow xz \in \perp)\}; \\ C(X) & = & X^{\perp\perp}. \end{array}$$

Lemat 9.1 *Działanie C jest operacją domknięcia, tj. dla dowolnych $X, Y \subseteq M$:*

1. $X \subseteq C(X)$;
2. *jeśli* $X \subseteq Y$ *to* $C(X) \subseteq C(Y)$;
3. $C(C(X)) = C(X)$.

Dowód: Pierwsza własność jest oczywista. Własność (2) wynika od razu z tego, że $X \subseteq Y$ implikuje $Y^{\perp} \subseteq X^{\perp}$. Aby pokazać (3), zauważmy, że z (1) wynika zarówno $C(X)^{\perp} \subseteq X^{\perp}$ jak też $X^{\perp} \subseteq C(X^{\perp})$. ∎

Zbiór $X \subseteq M$ jest *faktem*, gdy jest *domknięty*, tj. gdy $C(X) = X$. (Zauważmy, że każdy zbiór postaci $X^{\perp}$ jest domknięty.) W rodzinie $\mathcal{C}$ wszystkich zbiorów domkniętych można zdefiniować operacje odpowiadające spójnikom logiki liniowej. Zaczniemy od definicji stałych:

$$0 = M^{\perp} \qquad\qquad 1 = \perp^{\perp} \qquad\qquad \top = M \qquad\qquad \perp = \perp$$

Niech $I = 1 \cap \{x \in M \mid xx = x\}$ oraz $X \cdot Y = \{xy \mid x \in X \text{ oraz } y \in Y\}$. Określamy dalej:

$$\begin{array}{lcl} X \oplus Y & = & C(X \cup Y); \\ X \otimes Y & = & C(X \cdot Y); \\ !X & = & C(I \cap X). \end{array}$$

Pozostałe operacje definiujemy idąc po linii najmniejszego oporu:

$$\begin{array}{lcl} X \,\&\, Y & = & (X^{\perp} \oplus Y^{\perp})^{\perp} \\ X \parr Y & = & (X^{\perp} \otimes Y^{\perp})^{\perp} \\ X \multimap Y & = & X^{\perp} \parr Y \\ ?X & = & (!X^{\perp})^{\perp}. \end{array}$$

[11] To jest pewne uproszczenie. Implementacja operatora **Fix** może wymagać, aby jego argument też był wielokrotnego użytku.

Jeśli teraz wartościowanie v przypisuje zmiennym zdaniowym zbiory domknięte w przestrzeni fazowej M, to w oczywisty sposób możemy zdefiniować *znaczenie* $[\![\varphi]\!]_v$ dla dowolnej formuły φ, które zawsze jest zbiorem domkniętym.

Lemat 9.2 *Jeśli $v(p) \in \mathcal{C}$ dla wszystkich zmiennych p formuły φ, to także $[\![\varphi]\!]_v \in \mathcal{C}$.*

Dowód: Dowód sprowadza się do łatwego sprawdzenia, że stałe są domknięte (zauważmy np. że $\perp = \{\varepsilon\}^\perp$) i że operacje nie wyprowadzają poza $\mathcal{C}$. ∎

Formuła φ jest *spełniona* przez wartościowanie v, gdy $\varepsilon \in [\![\varphi]\!]_v$. Mamy następujące twierdzenie o pełności, którego dowód pomijamy.[12]

Twierdzenie 9.3 *Formuła φ jest twierdzeniem liniowym wtedy i tylko wtedy, gdy jest spełniona przez każde wartościowanie w każdej przestrzeni fazowej.*

Ćwiczenia

1. Udowodnić, że $X \multimap Y = (X \cdot Y^\perp)^\perp = \{z \in M \mid \forall x \in M(x \in X \Rightarrow xz \in Y)\}$, dla faktów X, Y.
2. Udowodnić, że $A^\perp \cap B^\perp = (A \cup B)^\perp$.
3. Udowodnić, że $X \mathbin{\&} Y = X \cap Y$, dla dowolnych faktów X i Y. (A zatem iloczyn zbiorów domkniętych jest domknięty.)
4. Udowodnić, że $X \subseteq Y$ wtedy i tylko wtedy, gdy $\varepsilon \in X \multimap Y$.
5. Udowodnić poprawność. Wskazówka: Jeśli $\gamma_1, \ldots, \gamma_n \vdash \varphi$ to $[\![\gamma_1]\!]_v \otimes \cdots \otimes [\![\gamma_n]\!]_v \subseteq [\![\varphi]\!]_v$.

10 Przestrzenie zgodności

Kategoria przestrzeni zgodności nie definiuje semantyki logiki liniowej w takim sensie do jakiego przywykliśmy. Jest to raczej semantyka *dowodów*, czy też programów liniowych, a nie formuł. Inaczej mówiąc, chodzi bardziej o model dla typów danych reprezentujących spójniki liniowe, niż o model dla wnioskowania liniowego. Ale tę konstrukcję można udoskonalić i otrzymać kategorię dla której zachodzi twierdzenie o pełności w tradycyjnym sensie.[13]

Zaczynamy od *pajęczyny*, czyli zbioru A ze zwrotną i symetryczną *relacją zgodności* $\frown$. W takiej pajęczynie *kliką* nazywamy dowolny podzbiór $a \subseteq A$, w którym każde dwa elementy są zgodne. Zbiór wszystkich klik w A to *przestrzeń zgodności* $Coh(A)$. Alternatywnie, przestrzenią zgodności możemy nazwać rodzinę $\mathcal{A}$ podzbiorów zbioru A, spełniającą warunki:

1. Jeśli $a \in \mathcal{A}$ i $a' \subseteq a$ to $a' \in \mathcal{A}$.
2. Jeśli $\mathcal{B} \subseteq \mathcal{A}$ oraz dla dowolnych $a, b \in \mathcal{B}$ zachodzi $a \cup b \in \mathcal{A}$ to $\bigcup B \in \mathcal{A}$.

[12] Patrz Girard, J.-Y., Linear logic: its syntax and semantics, in : *Advances in Linear Logic*, (Girard, J.-Y., Lafont, Y., eds.) Cambridge University Press, 1995, pp. 1–42.
[13] Girard, J.-Y, On denotational completeness, *Theoretical Computer Science*, 227 (1999), 249–273.

Fakt 10.1 *Definicje powyżej są równoważne: Po pierwsze, $Coh(A)$ ma własności (1) i (2). Po drugie, jeśli $\mathcal{A}$ jest rodziną o własnościach (1) i (2) oraz $|\mathcal{A}| = \{x \mid \{x\} \in \mathcal{A}\}$, to zbiór $|\mathcal{A}|$ z relacją $x \mathrel{\overset{\frown}{\smile}} y \Leftrightarrow \{x, y\} \in \mathcal{A}$ jest pajęczyną i do tego $\mathcal{A} = Coh(|\mathcal{A}|)$.*

Dowód: Część pierwsza jest łatwa. W części drugiej, aby z $a \in Coh(|\mathcal{A}|)$ wywnioskować, że $a \in \mathcal{A}$, należy zastosować (2) do rodziny wszystkich singletonów elementów a. ∎

Fakt 10.2 *Przestrzeń zgodności jest zupełnym porządkiem częściowym.*

Dowód: Najmniejszym elementem jest pusta klika, a kresem górnym zbioru skierowanego jest jego suma. ∎

Najprostszym przykładem przestrzeni zgodności jest przestrzeń z relacją identyczności. Jest ona izomorficzna z „płaskim” cpo, w którym wszystkie elementy, oprócz najmniejszego, są nieporównywalne. Tak należy sobie wyobrażać interpretację atomowych typów danych, jak **bool** i **int**.

Stosujemy następujące oznaczenia:

- Piszemy $x \frown y$, gdy $x \mathrel{\overset{\frown}{\smile}} y$ i $x \neq y$.
- Piszemy $x \asymp y$, gdy $x \frown y$ nie zachodzi.
- Piszemy $x \smile y$, gdy $x \mathrel{\overset{\frown}{\smile}} y$ nie zachodzi.

Zatem na przykład $x \asymp x$ ale nie $x \smile x$. Relacja $\asymp$ jest relacją zgodności *dualną* do $\mathrel{\overset{\frown}{\smile}}$. Jeśli $\mathcal{A}$ jest przestrzenią zgodności o pajęczynie $\langle A, \mathrel{\overset{\frown}{\smile}}\rangle$ to przez $\mathcal{A}^\perp$ oznaczymy $Coh(\langle A, \asymp\rangle)$. Oczywiście $\mathcal{A}^{\perp\perp}$ to to samo co $\mathcal{A}$. Inne spójniki liniowe także odpowiadają pewnym operacjom na przestrzeniach. Niech $\mathcal{A}$ i $\mathcal{B}$ będą przestrzeniami zgodności. Wtedy $\mathcal{A} \otimes \mathcal{B}$ i $\mathcal{A} \mathbin{\rotatebox[origin=c]{180}{\&}} \mathcal{B}$ mają pajęczyny o dziedzinie $|\mathcal{A}| \times |\mathcal{B}|$ z relacjami odpowiednio $\mathrel{\overset{\frown}{\smile}}_\otimes$ i $\mathrel{\overset{\frown}{\smile}}_{\rotatebox[origin=c]{180}{\&}}$ określonymi tak:

$$\begin{array}{lcl} (x,y) \mathrel{\overset{\frown}{\smile}}_\otimes (x',y') & \Leftrightarrow & x \mathrel{\overset{\frown}{\smile}}_A x' \ \text{ oraz } \ y \mathrel{\overset{\frown}{\smile}}_B y' \\ (x,y) \frown_{\rotatebox[origin=c]{180}{\&}} (x',y') & \Leftrightarrow & x \frown_A x' \ \text{ lub } \ y \frown_B y' \end{array}$$

Zauważmy, że (zgodnie z oczekiwaniami) zachodzi tożsamość $(\mathcal{A} \mathbin{\rotatebox[origin=c]{180}{\&}} \mathcal{B})^\perp = \mathcal{A}^\perp \otimes \mathcal{B}^\perp$. Oczywiście przyjmujemy, że $\mathcal{A} \multimap \mathcal{B} = \mathcal{A}^\perp \mathbin{\rotatebox[origin=c]{180}{\&}} \mathcal{B}$. Spójniki addytywne odpowiadają konstrukcjom, w których pajęczyna jest określona na sumie prostej $|\mathcal{A}| + |\mathcal{B}|$ pajęczyn $|\mathcal{A}|$ i $|\mathcal{B}|$ (co ostatecznie wyjaśnia tajemnicę terminologii „adddytywny/multiplikatywny”) przy czym relacje $\mathrel{\overset{\frown}{\smile}}_\oplus$ i $\mathrel{\overset{\frown}{\smile}}_\&$ zawierają w sobie obie relacje $\mathrel{\overset{\frown}{\smile}}_A$ i $\mathrel{\overset{\frown}{\smile}}_B$. Dodatkowo, dla $x \in |\mathcal{A}|$, $y \in |\mathcal{B}|$ definiujemy:

$$x \frown_\& y \quad \text{oraz} \quad x \smile_\oplus y$$

Zauważmy, że kliki w przestrzeni $\mathcal{A} \mathbin{\&} \mathcal{B}$ można utożsamiać z parami uporządkowanymi $\langle a, b\rangle$, gdzie $a \in \mathcal{A}$, $b \in \mathcal{B}$. Przestrzeń $\mathcal{A} \mathbin{\&} \mathcal{B}$ jest w pewnym sensie produktem kartezjańskim przestrzeni $\mathcal{A}$ i $\mathcal{B}$ (Ćwiczenie 3).

Tożsamości znane z logiki liniowej pozostają prawdziwe dla przestrzeni zgodności. Na przykład $(\mathcal{A} \mathbin{\&} \mathcal{B})^\perp = \mathcal{A}^\perp \oplus \mathcal{B}^\perp$, a przestrzenie $\mathcal{A} \otimes (\mathcal{B} \oplus \mathcal{C})$ i $(\mathcal{A} \otimes \mathcal{B}) \oplus (\mathcal{A} \otimes \mathcal{C})$ są izomorficzne. Niestety, „sklejają się” dualne stałe (ale, jak powiada Girard,[12] „nobody is perfect”). Elementem neutralnym obu operacji $\otimes$ i $\rotatebox[origin=c]{180}{\&}$ jest przestrzeń o jednoelementowej pajęczynie. Jest tylko jedna taka przestrzeń, można ją oznaczać w zależności od nastroju przez 1 lub przez $\perp$. Podobnie, przestrzeń z pustą pajęczyną odgrywa rolę obu stałych 0 i $\top$.

Przekształcenia stabilne i morfizmy liniowe

Najwyższy czas zdefiniować odpowiednie dla przestrzeni zgodności pojęcie przekształcenia (morfizmu w kategorii takich przestrzeni).

Definicja 10.3 Funkcja $F : \mathcal{A} \to \mathcal{B}$ jest *stabilna*, gdy jest ciągłym przekształceniem porządku $\langle \mathcal{A}, \subseteq \rangle$ w porządek $\langle \mathcal{B}, \subseteq \rangle$ (tj. zachowuje kresy niepustych zbiorów skierowanych) i dodatkowo dla dowolnych $a, a' \in \mathcal{A}$ spełniających warunek $a \cup a' \in \mathcal{A}$ zachodzi $F(a \cap a') = F(a) \cap F(a')$.

Jako ciągła, funkcja stabilna jest monotoniczna, zatem w równości $F(a \cap a') = F(a) \cap F(a')$ istotnym żądaniem jest tylko inkluzja z prawej do lewej. Zastrzeżenie $a \cup a' \in \mathcal{A}$ jest potrzebne aby można było np. określić funkcję o własności $F(\{0\}) = F(\{1\}) = \{0\}$ i $F(\emptyset) = \emptyset$.

Przykładem funkcji niestabilnej jest „równoległa alternatywa” (*parallel or*), czyli funkcja ciągła $F : \mathbf{bool} \mathbin{\&} \mathbf{bool} \to \mathbf{bool}$, taka że $F(\{\mathbf{tt}\}, \emptyset) = F(\emptyset, \{\mathbf{tt}\}) = \{\mathbf{tt}\}$ oraz $F(\{\mathbf{ff}\}, \{\mathbf{ff}\}) = \{\mathbf{ff}\}$. Mamy bowiem z ciągłości $F(\emptyset, \emptyset) = \emptyset$, tymczasem stabilność wymaga tutaj $F(\emptyset, \emptyset) = \{\mathbf{tt}\}$.

Sens stabilności można wyrazić tak: za każdy element w zbiorze $F(a)$ „odpowiedzialny” jest jeden, ściśle określony podzbiór zbioru a.

Lemat 10.4 *Niech $F : \mathcal{A} \to \mathcal{B}$ będzie stabilne. Dla dowolnego $a \in \mathcal{A}$ i dowolnego $y \in F(a)$ istnieje najmniejszy podzbiór $a_0 \subseteq a$ taki, że $y \in F(a_0)$. Na dodatek, zbiór a_0 jest skończony.*

Dowód: Zbiór a jest skierowaną sumą swoich skończonych podzbiorów. Na mocy ciągłości, mamy więc $y \in F(a')$ dla pewnego skończonego $a' \subseteq a$. Jeśli a_0 jest minimalnym takim zbiorem, oraz dla jakiegoś $a_1 \subseteq a$ zachodzi $y \in f(a_1)$, to ze stabilności wynika teraz, że $y \in f(a_0) \cap f(a_1) = f(a_0 \cap a_1)$. Skoro a_0 jest minimalny, to $a_0 = a_0 \cap a_1$, czyli $a_0 \subseteq a_1$. ∎

Definicja 10.5 *Śladem* przekształcenia stabilnego $F : \mathcal{A} \to \mathcal{B}$ nazywamy zbiór par

$$Tr(F) = \{\langle a, y\rangle \mid a \text{ jest minimalnym zbiorem o własności } y \in F(a)\}.$$

Fakt 10.6 *Jeśli F i G są stabilne, oraz $Tr(F) = Tr(G)$ to $F = G$.*

Dowód: Zauważmy, że $F(a) = \{y \mid \exists a'(a' \subseteq a \wedge \langle a', y\rangle \in Tr(F))\}$. ∎

Niech $\mathcal{A}_{\mathit{fin}}$ oznacza rodzinę wszystkich klik skończonych w $\mathcal{A}$. Możemy teraz zdefiniować przestrzeń śladów $\mathcal{A} \Rightarrow \mathcal{B}$ przyjmując jako pajęczynę zbiór $\mathcal{A}_{\mathit{fin}} \times |\mathcal{B}|$ z relacją $\langle a, y\rangle \frown \langle a', y'\rangle$ zachodzącą w trzech przypadkach:

- gdy $a \cup a' \notin \mathcal{A}$;
- gdy $a = a'$ oraz $y \frown_B y'$;
- gdy $a \neq a'$, $a \cup a' \in \mathcal{A}$ oraz $y \smile_B y'$.

Fakt 10.7 *Elementy przestrzeni $\mathcal{A} \Rightarrow \mathcal{B}$ to dokładnie ślady przekształceń stabilnych z $\mathcal{A}$ do $\mathcal{B}$.*

Pora zrobić zasadnicze odkrycie. Konstrukcja przestrzeni śladów $\mathcal{A} \Rightarrow \mathcal{B}$ składa się z dwóch prostszych operacji. Najpierw definiujemy pajęczynę zbiorów skończonych, a potem tworzymy pewną nową przestrzeń na produkcie pajęczyn. Nazwijmy to po imieniu. Jeśli $\mathcal{A}$ jest dowolną przestrzenią, to przez $!\mathcal{A}$ oznaczymy przestrzeń z pajęczyną $\mathcal{A}_{\mathit{fin}}$ i relacją $a \frown a'$ zachodzącą gdy $a \cup a' \in \mathcal{A}$. Mamy teraz taką konkluzję:

Fakt 10.8 *Przestrzeń $\mathcal{A} \Rightarrow \mathcal{B}$ jest izomorficzna z $!\mathcal{A} \multimap \mathcal{B}$.*

Fakt ten wynika w naturalny sposób z następujących obserwacji:

Definicja 10.9 Przekształcenie stabilne $F : \mathcal{A} \to \mathcal{B}$ jest *liniowe* wtedy i tylko wtedy, gdy $F(\emptyset) = \emptyset$ oraz $F(a \cup a') = F(a) \cup F(a')$, gdy tylko $a \cup a' \in \mathcal{A}$. Łatwo widzieć, że ślad przekształcenia liniowego składa się z par postaci $\langle \{x\}, y\rangle$ (funkcja liniowa jest zdeterminowana przez swoje zachowanie na singletonach). Dla takich funkcji przyjmujemy więc po prostu, że $Tr(F) = \{\langle x, y\rangle \mid y \in F(\{x\})\}$.

Fakt 10.10 *Elementy przestrzeni $\mathcal{A} \multimap \mathcal{B}$ to dokładnie ślady przekształceń liniowych z $\mathcal{A}$ do $\mathcal{B}$.*

Interpretacja dowodów liniowych

Ze względu na równoważność pomiędzy $\alpha \otimes \beta \multimap \gamma$ i $\alpha \multimap \beta^\perp \parr \gamma$, każdy sekwent

$$\gamma_1, \ldots, \gamma_n \vdash \varphi_1, \ldots, \varphi_m$$

można reprezentować przez dowolną formułę postaci

$$\gamma_1 \otimes \cdots \otimes \gamma_n \otimes \varphi_1^\perp \otimes \cdots \otimes \varphi_{i-1}^\perp \otimes \varphi_{i+1}^\perp \otimes \cdots \otimes \varphi_m^\perp \multimap \varphi_i$$

lub postaci

$$\gamma_1 \otimes \cdots \otimes \gamma_{i-1} \otimes \gamma_{i+1} \otimes \cdots \otimes \gamma_n \otimes \varphi_1^\perp \otimes \cdots \otimes \varphi_m^\perp \multimap \gamma_i^\perp$$

Przypuśćmy, że każdej zmiennej zdaniowej przyporządkowano pewną przestrzeń zgodności. Możemy teraz utożsamiać formuły złożone z przestrzeniami otrzymanymi przez zastosowanie odpowiednich operacji. To utożsamienie przenosi się dalej na dowolne sekwenty. Dowodowi w logice liniowej można więc przypisywać element przestrzeni reprezentującej sekwent końcowy tego dowodu. Dla uproszczenia ograniczamy się do sekwentów z jedną formułą po każdej stronie (patrz wyżej) i rozważamy przestrzenie z dokładnością do izomorfizmów pomiędzy $\mathcal{A} \otimes \mathcal{B} \multimap \mathcal{C}$ i $\mathcal{A} \multimap \mathcal{B}^\perp \parr \mathcal{C}$.

- Aksjomat $\vdash \mathcal{A}^\perp, \mathcal{A}$ utożsamiamy z sekwentem $\mathcal{A} \vdash \mathcal{A}$ i reprezentujemy przez funkcję identycznościową z $\mathcal{A}$ do $\mathcal{A}$ o śladzie $\{\langle z, z\rangle \mid z \in |\mathcal{A}|\}$.
- Dowód kończący się cięciem postaci $\dfrac{\mathcal{A} \vdash \mathcal{B} \quad \mathcal{B} \vdash \mathcal{C}}{\mathcal{A} \vdash \mathcal{C}}$ reprezentujemy przez złożenie funkcji reprezentujących dowody przesłanek.
- Jeśli $F : \mathcal{A} \to \mathcal{B}$ reprezentuje dowód $\mathcal{A} \vdash \mathcal{B}$ oraz $G : \mathcal{C} \to \mathcal{D}$ reprezentuje dowód $\mathcal{C} \vdash \mathcal{D}$ to dowód $\mathcal{A} \otimes \mathcal{C} \vdash \mathcal{B} \otimes \mathcal{D}$, kończący się wprowadzeniem $\otimes$ z prawej, reprezentujemy przez funkcję o śladzie $\{\langle\langle x, y\rangle, \langle z, u\rangle\rangle \mid \langle x, z\rangle \in Tr(F) \wedge \langle y, u\rangle \in Tr(G)\}$.

- Wprowadzenie $\invamp$ z prawej jest przy naszych uproszczeniach operacją nie zmieniającą interpretacji sekwentu.
- Jeśli $F : \mathcal{A} \to \mathcal{B}$ reprezentuje dowód $\mathcal{A} \vdash \mathcal{B}$ oraz $G : \mathcal{A} \to \mathcal{C}$ reprezentuje dowód $\mathcal{A} \vdash \mathcal{C}$ to dowód $\mathcal{A} \vdash \mathcal{B} \,\&\, \mathcal{C}$, kończący się wprowadzeniem $\&$ z prawej, reprezentujemy przez funkcję o śladzie, który jest sumą śladów $Tr(F) \cup Tr(G)$ (zakładamy tu cichcem, że te ślady są rozłączne).
- Jeśli $F : \mathcal{A} \to \mathcal{B}$ reprezentuje dowód $\mathcal{A} \vdash \mathcal{B}$, to dowód $\mathcal{A} \vdash \mathcal{B} \oplus \mathcal{C}$, kończący się wprowadzeniem $\oplus$ z prawej, jest reprezentowany przez funkcję $G = \mathrm{in}_1 \circ F$, gdzie in_1 jest włożeniem $\mathcal{A}$ w $\mathcal{A} \oplus \mathcal{B}$.
- Jeśli $F : !\mathcal{A} \to \mathcal{B}$ reprezentuje dowód $!\mathcal{A} \vdash \mathcal{B}$, to dowód $!\mathcal{A} \vdash !\mathcal{B}$, kończący się wprowadzeniem $!$, jest reprezentowany przez funkcję o śladzie
$$\{\langle b_1 \cup \cdots \cup b_n, \{x_1, \ldots, x_n\}\rangle \mid \langle b_i, x_i\rangle \in Tr(F),\ \text{dla } i = 1, \ldots n\}.$$
- Jeśli $F : 1 \to \mathcal{A}$ reprezentuje dowód $\vdash \mathcal{A}$, to dowód $!\mathcal{B} \vdash \mathcal{A}$ reprezentuje funkcja stale równa $F(\iota)$, gdzie ι jest jedynym elementem pajęczyny $|1|$.
- Jeśli $F : !\mathcal{A} \otimes !\mathcal{A} \multimap \mathcal{C}$ reprezentuje dowód $!\mathcal{A} \otimes !\mathcal{A} \vdash \mathcal{C}$, to dowód $!\mathcal{A} \vdash \mathcal{C}$ jest reprezentowany przez funkcję o śladzie $\{\langle a \cup b, x\rangle \mid \langle\langle a, b\rangle, x\rangle \in Tr(F)\}$.
- Wreszcie jeśli $F : \mathcal{A} \multimap \mathcal{B}$ reprezentuje dowód sekwentu $\mathcal{A} \vdash \mathcal{B}$, to funkcja o śladzie $\{\langle\{x\}, y\rangle \mid \langle x, y\rangle \in Tr(F)\}$ reprezentuje dowód $!\mathcal{A} \vdash \mathcal{B}$.

Twierdzenie 10.11 *Funkcja reprezentująca dowód nie zmienia się przy eliminacji cięcia.*

Dowód: Zostawiamy czytelnikowi. ∎

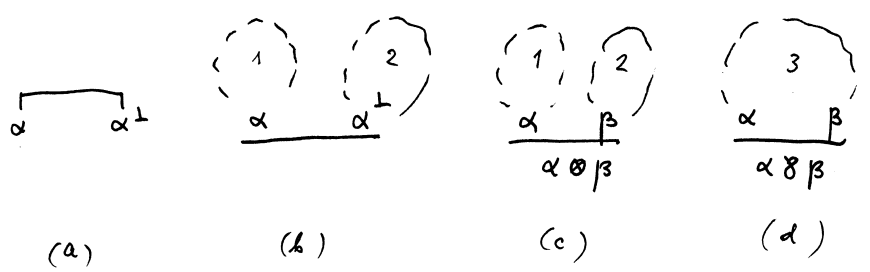
Ćwiczenia

1. Pokazać, że równoważności 1–13 z rozdziału 3 tłumaczą się na izomorfizmy odpowiednich przestrzeni zgodności.
2. Jeśli funkcja ciągła $F : \mathcal{A} \to \mathcal{B}$ jest stabilna, to z tego, że $a \cup a' \in \mathcal{A}$ i $a \cap a' = \emptyset$ wynika inkluzja $F(a) \cap F(a') \subseteq F(\emptyset)$. Czy każda funkcja o tej własności jest stabilna?
3. Udowodnić, że przestrzeń $\mathcal{A}\&\mathcal{B}$ jest produktem kartezjańskim w kategorii przestrzeni zgodności, w której morfizmami są przekształcenia stabilne. (Należy zdefiniować stabilne „rzutowania” $\pi_1 : \mathcal{A} \,\&\, \mathcal{B} \to \mathcal{A}$ i $\pi_2 : \mathcal{A} \,\&\, \mathcal{B} \to \mathcal{A}$ i udowodnić, że dla dowolnej przestrzeni $\mathcal{C}$ i dowolnych stabilnych $f_1 : \mathcal{C} \to \mathcal{A}$, $f_2 : \mathcal{C} \to \mathcal{B}$ istnieje dokładnie jedna funkcja stabilna $f_1 \,\&\, f_2 : \mathcal{C} \to \mathcal{A} \,\&\, \mathcal{B}$ spełniająca równości $\pi_i \circ (f_1 \,\&\, f_2) = f_i$ dla $i = 1, 2$.)
4. Udowodnić, że przestrzenie $(\mathcal{A} \,\&\, \mathcal{B}) \Rightarrow \mathcal{C}$ i $\mathcal{A} \Rightarrow (\mathcal{B} \Rightarrow \mathcal{C})$ są izomorficzne.
5. Udowodnić, że przestrzenie $\mathcal{A} \multimap \mathcal{B}$ i $\mathcal{B}^\perp \multimap \mathcal{A}^\perp$ są izomorficzne.
6. Co jest koproduktem[14] przestrzeni zgodności ze względu na przekształcenia stabilne? A ze względu na morfizmy liniowe?
7. Niech $F : \mathcal{A} \to \mathcal{B}$ i $G : \mathcal{B} \to \mathcal{C}$ będą przekształceniami liniowymi. Udowodnić, że $Tr(G \circ F)$ to złożenie relacji $Tr(F)$ i $Tr(G)$.

[14] Przestrzeń $\mathcal{C}$ nazwiemy *koproduktem* przestrzeni $\mathcal{A}$ i $\mathcal{B}$ ze względu na przekształcenia stabilne, wtedy i tylko wtedy, gdy mamy dwa przekształcenia stabilne $l : \mathcal{A} \to \mathcal{C}$ i $r : \mathcal{B} \to \mathcal{C}$ o takiej własności: Dla dowolnej przestrzeni $\mathcal{D}$ i dowolnych stabilnych $\alpha : \mathcal{A} \to \mathcal{D}$, $\beta : \mathcal{B} \to \mathcal{D}$ istnieje dokładnie jedno przekształcenie stabilne $\alpha + \beta : \mathcal{C} \to \mathcal{D}$ spełniające warunki $(\alpha + \beta) \circ l = \alpha$, $(\alpha + \beta) \circ r = \beta$. Analogiczna definicja koproduktu stosuje się do przekształceń liniowych.

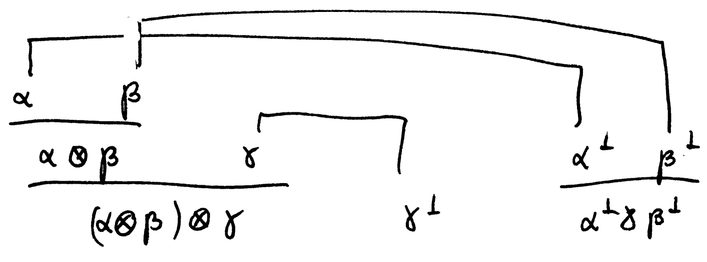
11 Sieci multiplikatywne

Sieci dowodowe to pewien sposób zapisywania dowodów w prawostronnym rachunku sekwentow. Najładniejsze własności mają sieci multiplikatywne, więc od nich zaczniemy. Taka sieć to graf niezorientowany, którego wierzchołki są etykietowane formułami. Niektóre z tych wierzchołków uważamy za *wyjścia* (konkluzje) sieci, przy czym wyjście typu α możemy także nazywać *wejściem* (założeniem) typu $\alpha^{\perp}$. Definicja indukcyjna sieci jest objaśniona na Obrazku 11. Najprostsza sieć (a) ma dwa wyjścia α i $\alpha^{\perp}$ i odpowiada aksjomatowi $\vdash \alpha, \alpha^{\perp}$. Sieć ($b$) odpowiada regule cięcia i realizuje komunikację pomiędzy podsieciami (1) i (2). Jej wyjściami są wszystkie wyjścia podsieci (1) i (2) oprócz tych połączonych ze sobą cięciem. Sieć (c) łączy ze sobą wyjścia α i β rozłącznych podsieci składowych w jedno wyjście typu $\alpha \otimes \beta$, a sieć (d) łączy ze sobą wyjścia tej samej sieci w jedno „podwójne" wyjście typu $\alpha \parr \beta$.



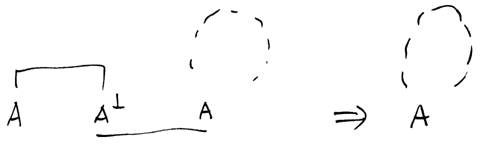
Obrazek 11: Sieci multiplikatywne

Na Obrazku 12 mamy przykład sieci o trzech wyjściach $(\alpha \otimes \beta) \otimes \gamma$, $\gamma^{\perp}$, $\alpha^{\perp} \parr \beta^{\perp}$. Zauważmy, że te sama sieć odpowiada dwóm różnym dowodom zapisanym w tradycyjny sposób, bo wprowadzenie $\otimes$ i $\parr$ może odbywać się w różnej kolejności. W przypadku sieci tego problemu nie ma. Dlatego też prostsza jest eliminacja cięcia.



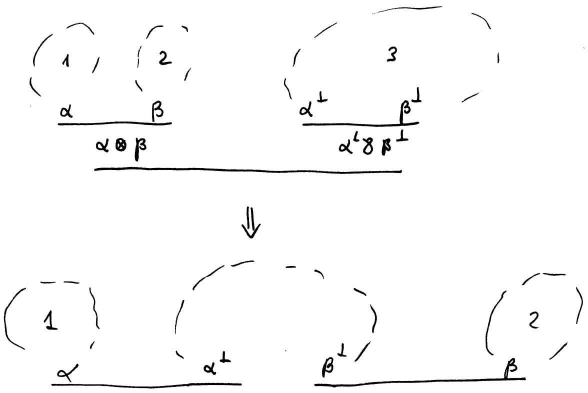
Obrazek 12: Przykład sieci

Mamy tylko dwie reguły eliminacji cięcia, przedstawione na Obrazkach 13 i 14. Niepotrzebne są czynności polegające na permutowaniu kroków dowodu.



Obrazek 13: Eliminacja cięcia z aksjomatem

Reguły z Obrazków 13 i 14 można stosować do dowolnej sieci, zawierającej redeks, tj. prawidłowo zbudowaną *podsieć* postaci takiej jak po lewej stronie. Pozostałe połączenia redeksu z resztą sieci pozostają bez zmian. Redukcję z Obrazka 13 można sobie wyobrażać jako rozdzielenie dwużyłowego przewodu łączącego „podwójne" wyjścia $\alpha \otimes \beta$ i $\alpha^\perp \parr \beta^\perp$ i zastąpienie go dwoma pojedynczymi połączeniami typu α i typu β. Intuicja podpowiada, że komunikacja realizowana przez całą sieć pozostaje przy tej operacji bez zmian.



Obrazek 14: Eliminacja cięcia multiplikatywnego

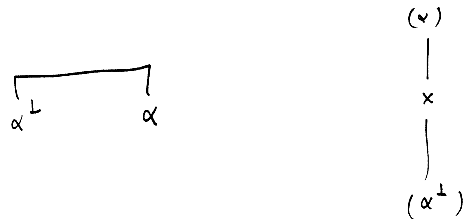
Dla sieci dowodowych zachodzi twierdzenie o eliminacji cięcia: Każdą sieć można przekształcić w sieć bez cięcia o tych samych konkluzjach.

Termy liniowe jako sieci

Za pomocą multiplikatywnych sieci dowodowych można reprezentować lambda-termy *liniowe*, tj. takie, w których każda zmienna jest używana dokładnie raz. Ściślej, jeśli M jest termem liniowym, oraz $\Gamma \vdash M : \tau$, gdzie $\Gamma = \{x_1 : \gamma_1, \ldots, x_n : \gamma_n\}$, to określamy sieć dowodową $\underline{M}$

o konkluzjach $\tau, \gamma_1^\perp, \ldots, \gamma_n^\perp$. Przy tym strzałkę $\to$ w typach $\tau, \gamma_1, \ldots, \gamma_n$ utożsamiamy z lizakiem $\multimap$. Na rysunkach poniżej, po lewej stronie mamy sieci, a po prawej „zwykłe" grafowe reprezentacje odpowiednich termów. Ewentualne podobieństwo jest całkowicie zamierzone.

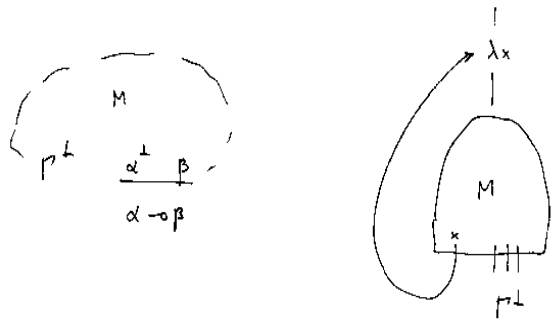
Sieć reprezentująca zmienną, a dokładniej asercję typową $x : \alpha \vdash x : \alpha$, to po prostu aksjomat odpowiedniego typu. (Obrazek 15).



Obrazek 15: Zmienna to aksjomat

Abstrakcja na Obrazku 16 jest typu $\alpha \multimap \beta$, czyli $\alpha^\perp \parr \beta$. Mamy tu wyprowadzenie kończące się zastosowaniem reguły

$$\frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x\, M : \alpha \to \beta}$$

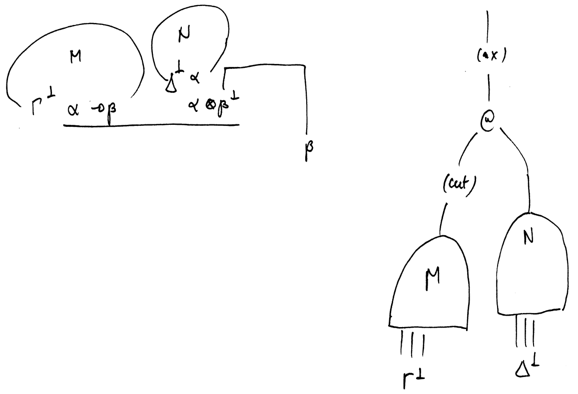


Obrazek 16: Abstrakcja

A na Obrazku 17 mamy wyprowadzenie kończące się zastosowaniem reguły

$$\frac{\Gamma \vdash M : \alpha \to \beta \quad \Delta \vdash N : \alpha}{\Gamma, \Delta \vdash MN : \beta}$$

Beta-redukcja termów liniowych jest poprawnie reprezentowana przez eliminację cięcia, ale jeden krok beta-redukcji wymaga trzech kroków redukcji w sieciach. Zauważmy, że beta-

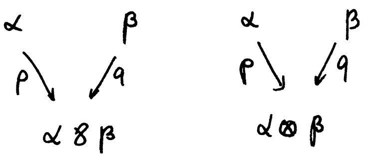


Obrazek 17: Aplikacja

redukcja reprezentowana jako operacja grafowa (Obrazek 20) dokładnie odpowiada eliminacji cięcia pomiędzy $\alpha \multimap \beta$ i $\alpha \otimes \beta^\perp$. Co więcej, można ją uważać za operację całkowicie lokalną, nie wymagającą manipulacji całym grafem a jedynie „przestawienia połączeń" (Obrazek 21).

Geometria interakcji

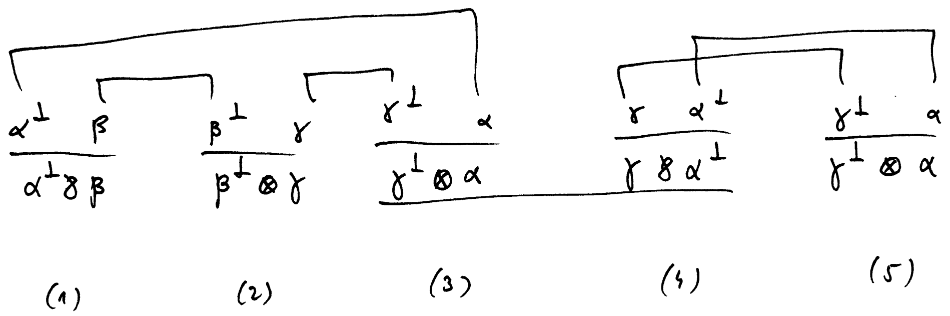
W multiplikatywnej sieci dowodowej orientujemy i oznaczamy krawędzie jak na Obrazku 18.



Obrazek 18: Krawędzie zorientowane z etykietami

Inne krawędzie pozostają bez etykiet (lub z „neutralną" etykietą 1) a ich orientacja jest nieistotna. Przez *ścieżkę* w tak udekorowanym grafie rozumiemy taki ciąg krawędzi, w którym nie dochodzi do „cofania" tj. przejścia przez jakąś krawędź w jedną stronę i natychmiast w stronę odwrotną. Każdej takiej ścieżce przypisujemy *wagę*, czyli słowo nad alfabetem $\{p, q, p^*, q^*\}$ otrzymane przez zapisanie etykiet kolejnych krawędzi na tej ścieżce. Przy tym, jesli ścieżka przechodzi przez daną krawędź zgodnie z jej orientacją to zapisujemy jej etykietę (np. p) „normalnie", a jeśli ścieżka przechodzi przez krawędź „pod prąd" to zapisujemy etykietę z gwiazdką, np. p^*. Żeby było trudniej, wszystko zapisujemy od prawej, np. w sieci na Obrazku 19 mamy ścieżkę od konkluzji (1) do (5) o wadze pp^*qp^*.

To jeszcze nie koniec, bo niektóre wagi będziemy ze sobą utożsamiać. W istocie definiujemy pewną algebrę wolną o stałych $p, q, 0, 1$ i operacjach $\cdot, +, ^*$. Kropka to oczywiście składanie,



Obrazek 19: Przykład sieci

gwiazdka odpowiada odwróceniu „kierunku ruchu", a plus pozwala na sumowanie „przepływu" odbywającego się dwoma różnymi drogami. Równania definiujące naszą *algebrę dynamiczną* są takie: Mnożenie jest łączne, dodawanie jest łączne i przemienne, mamy też zwykłe prawo rozdzielności jak w półpierścieniu. Jedynka jest elementem neutralnym mnożenia, zero elementem neutralnym dodawania. Ponadto zakładamy, że $0 \cdot x = x \cdot 0 = 0$ oraz:

$$p^*p = q^*q = 1 \qquad\qquad p^*q = q^*p = 0$$

$$(xy)^* = y^*x^* \qquad (x+y)^* = x^* + y^* \qquad x^{**} = x$$

Pierwsze dwa równania mówią, że przejście przez $\xrightarrow{p}$ (odp. $\xrightarrow{q}$) a następnie przez $\xleftarrow{p}$ (odp. $\xleftarrow{q}$) ma wagę neutralną. Następne dwa równania stwierdzają, że po przejściu przez $\xrightarrow{p}$ (odp. $\xrightarrow{q}$) *nie wolno* przechodzić bezpośrednio przez $\xleftarrow{q}$ (odp. $\xleftarrow{p}$). Zero interpretujemy bowiem jako wagę ścieżki niedozwolonej. Taka sytuacja występuje wtedy, gdy ścieżka wchodzi do cięcia przez lewą składową wycinanej formuły a wychodzi przez prawą, lub odwrotnie. Tak właśnie jest w przypadku wspomnianej wyżej ścieżki o wadze $pp^*qp^* = 0$.

Zauważmy, że ścieżka ma wagę niezerową wtedy i tylko wtedy, gdy zamieniając każde wystąpienie $x \in \{p, q\}$ na instrukcję $push(x)$, a każde wystąpienie x^* na $pop(x)$, otrzymamy możliwy do wykonania ciąg operacji na stosie (o odpowiedniej zawartości początkowej).

Przykład 11.1 Rozpatrzmy sieć z Obrazka 19 i zapomnijmy na chwilę, że występuje w niej cięcie, tj. potraktujmy wierzchołki (3) i (4) jak „wirtualne wyjścia". Komunikacja w tej sieci wyrażona być może macierzą rozmiaru 5x5:

$$M = \begin{bmatrix} 0 & qp^* & pq^* & 0 & 0 \\ pq^* & 0 & qp^* & 0 & 0 \\ qp^* & pq^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & qq^* + pp^* \\ 0 & 0 & 0 & qq^* + pp^* & 0 \end{bmatrix}$$

Pozycja na przecięciu i-tej kolumny i j-tego wiersza przedstawia komunikację od (wirtualnego) wyjścia (i) do (wirtualnego) wyjścia (j), możliwą bez wykorzystania cięcia. Cięcie możemy też przedstawić macierzą rozmiaru 5x5:

$$\sigma = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Komunikacja wykorzystująca cięcie (dowolną ilość razy) wyraża się więc sumą

$$\sum_{n=0}^{\infty} M(\sigma M)^n.$$

Ponieważ ostatecznie interesują nas tylko prawdziwe wyjścia sieci, mnożymy powyższe wyrażenie z obu stron przez macierz

$$1 - \sigma^2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

i dostajemy wyrażenie

$$\mathrm{Ex}(M, \sigma) = (1 - \sigma^2)\Big[\sum_{n=0}^{\infty} M(\sigma M)^n\Big](1 - \sigma^2),$$

które dla postrachu nazywamy *formułą egzekucyjną*, a dla lepszego efektu zapisujemy tak:

$$\mathrm{Ex}(M, \sigma) = (1 - \sigma^2)M(1 - \sigma M)^{-1}(1 - \sigma^2),$$

co uzasadniamy „formalnymi" rachunkami:

$$(1 - \sigma M)[\sum_{n=0}^{\infty} M(\sigma M)^n] = \sum_{n=0}^{\infty} M(\sigma M)^n - \sum_{n=0}^{\infty} M(\sigma M)^{n+1} = (\sigma M)^0 = 1.$$

Przykład jw. można uogólnić i udowodnić następujące własności:

1) Formuła egzekucyjna jest niezmiennikiem eliminacji cięcia.

2) Suma $\sum_{n=0}^{\infty} M(\sigma M)^n$ jest w istocie skończona, tj. prawie wszystkie składniki są zerami.

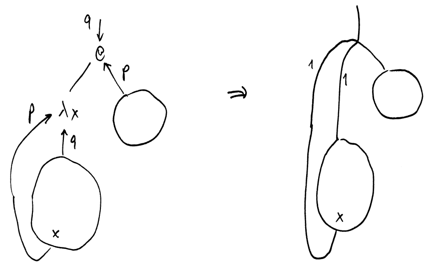
Własność (1) powyżej wynika stąd, że waga każdej ścieżki, która nie jest „rozerwana" przez eliminację cięcia, pozostaje przy tej operacji niezmieniona. Oznacza to w szczególności, że

3) Ścieżki o niezerowej wadze to dokładnie ścieżki *uporczywe*, tj. takie, które nie znikają przy żadnej redukcji i pozostają obecne w sieci normalnej.

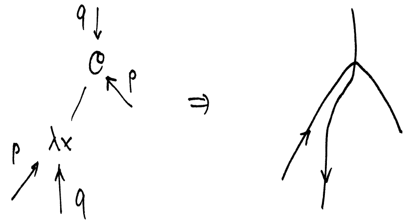
4) Nie ma niezerowych pętli.

Własność (2) wynika z twierdzenia o eliminacji cięcia: żadna ścieżka nie może przechodzić przez cięcia więcej razy niż liczba kroków potrzebnych do eliminacji wszystkich cięć.[15]

Na koniec zauważmy jeszcze, że etykietowanie sieci indukuje etykietowanie lambda-termów (Obrazki 20–21) przy którym beta-redukcja zachowuje wagi ścieżek, mamy więc także tu własności (3) i (4). Ścieżki uporczywe pozostają obecne w postaci normalnej termu, zatem badanie ścieżek niezerowych pozwala na uzyskanie informacji o kształcie postaci normalnej.



Obrazek 20: Beta-redukcja w grafie



Obrazek 21: Beta-redukcja jako operacja lokalna

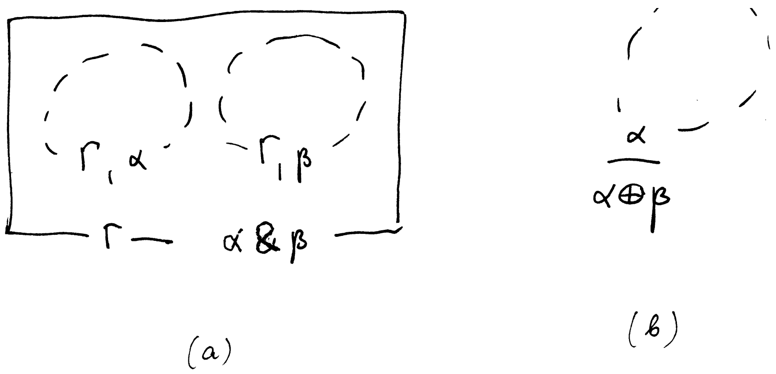
Ćwiczenia

1. Udowodnić twierdzenie o eliminacji cięcia dla sieci multiplikatywnych.
2. Udowodnić, że beta-redukcja termów liniowych jest poprawnie zaimplementowana przez eliminację cięcia.
3. Udowodnić własności (1)–(4).

[15] Cięcie na ścieżce w każdym kroku albo znika, albo zmienia sie na cięcie mniejszej rangi.

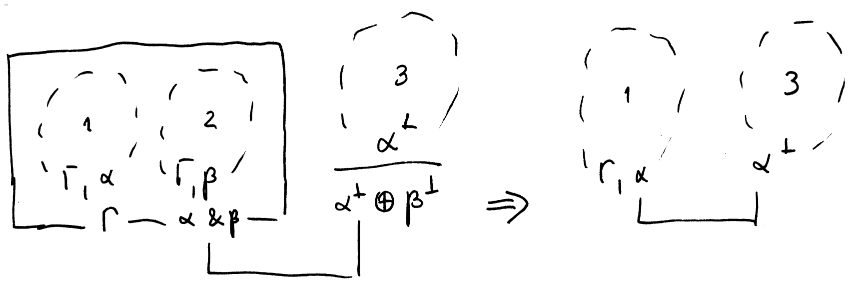
12 Sieci zapudłowane

Sieci dowodowe dla logiki z operatorami addytywnymi i modalnymi reprezentujemy za pomocą *pudełek*. O sieci znajdującej się w pudełku można myśleć jak o procesie, oczekującym na możliwość komunikacji ze światem zewnętrznym. Pudełko reprezentuje tu synchronizację, która jest realizowana poprzez eliminację cięcia. Na Obrazku 22 widzimy jak można rozszerzyć definicję sieci na przypadek operatorów addytywnych.



Obrazek 22: Sieci dla spójników addytywnych

Sieci (*a*) i (*b*) odpowiadają regułom wprowadzania & i ⊕. Pierwsza z nich powstaje poprzez umieszczenie dwóch podsieci w pudełku.[16] Jedna z tych podsieci zniknie, a druga, na skutek eliminacji cięcia, zostanie połączona z „odbiornikiem" (Obrazek 23).



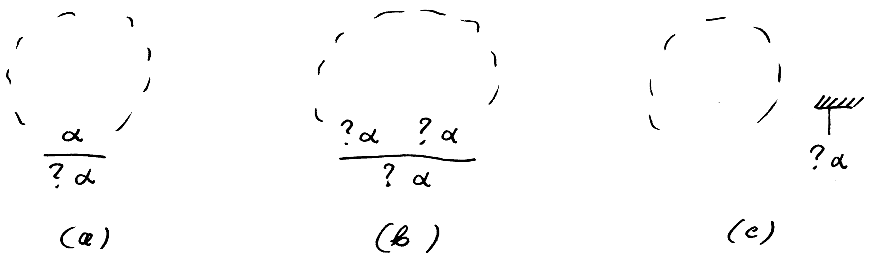
Obrazek 23: Addytywna eliminacja cięcia

Kolej na modalności. Dla operacji „być może" mamy trzy sposoby tworzenia sieci, odpowiadające trzem regułom wprowadzania pytajnika: derelikcji (*a*), skracaniu (*b*) i osłabianiu (*c*). W tym ostatnim przypadku wyjście $?\alpha$ jest połączone z uziemieniem (czy też z `/dev/null`) i nie ma związku z resztą sieci.

Wprowadzanie wykrzyknika, podobnie jak wprowadzanie &, odbywa się za pośrednictwem pudełka. „Klasyczne" rozwiązanie odpowiada zwykłym regułom

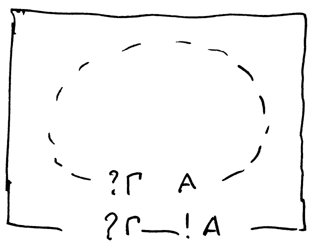
$$\frac{\vdash ?\Gamma, \alpha}{\vdash ?\Gamma, !\alpha} \quad \text{czyli} \quad \frac{!\Delta \vdash \alpha}{!\Delta \vdash !\alpha}$$

[16] A tak naprawdę to w cylindrze Lafonta.



Obrazek 24: Wprowadzanie znaku zapytania

i jest przedstawione na Obrazku 25. W tym ujęciu, sieci dowodowe ściśle odpowiadają dowodom w prawostronnym rachunku sekwentów z Obrazka 5.



Obrazek 25: Pudło ortodoksyjne

Ale czasem wygodniej jest skorzystać z nieco innych reguł dla wykrzyknika:

$$\frac{\vdash \Gamma, \alpha}{\vdash ?\Gamma, !\alpha} \qquad \text{czyli} \qquad \frac{\Delta \vdash \alpha}{!\Delta \vdash !\alpha}$$

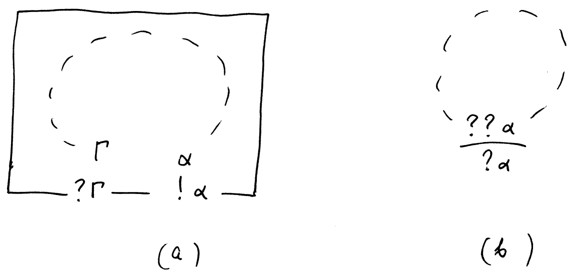
Te reguły są oczywiście poprawne, ale dla pełności systemu musimy dodać jeszcze jeden sposób wprowadzania pytajnika, który nazwiemy regułą delta:

$$\frac{\vdash \Gamma, ??\alpha}{\vdash \Gamma, ?\alpha}\ (\delta)$$

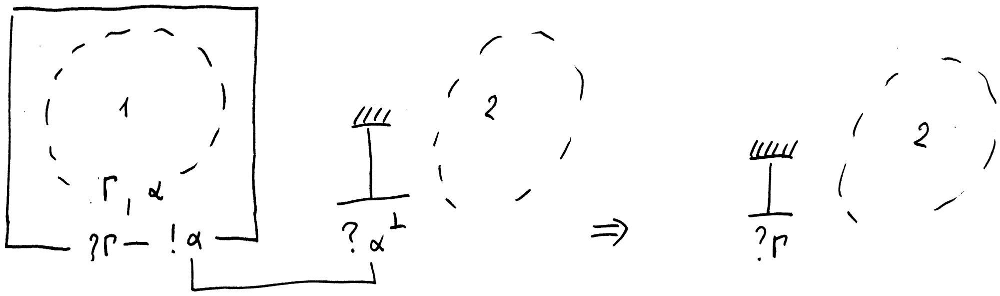
Nieortodoksyjne pudło mamy na Obrazku 26(a), a obok jeszcze jeden sposób stworzenia sieci o konkluzji $?\alpha$.

Zarówno w przypadku pudła ortodoksyjnego jak i nieortodoksyjnego, wyjście wprowadzające wykrzyknik nazywamy *głównymi drzwiami* pudełka. Pozostałe wyjścia to *drzwi boczne*.

Eliminację cięcia dla tak zdefiniowanych sieci przedstawiają Obrazki 27–31. Komunikacja z osłabianiem jest realizowana poprzez wyrzucenie pudełka wraz z zawartością, co odpowiada obliczeniu, w którym dana typu $!\alpha$ nie jest wykorzystana wcale (Obrazek 27).

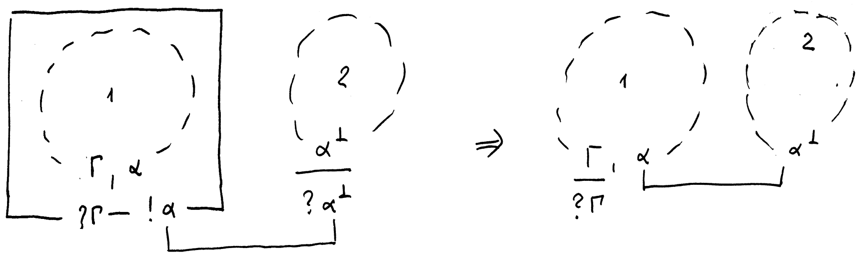


Obrazek 26: Nieortodoksyjne sieci wykładnicze



Obrazek 27: Przesyłanie danych na `/dev/null`

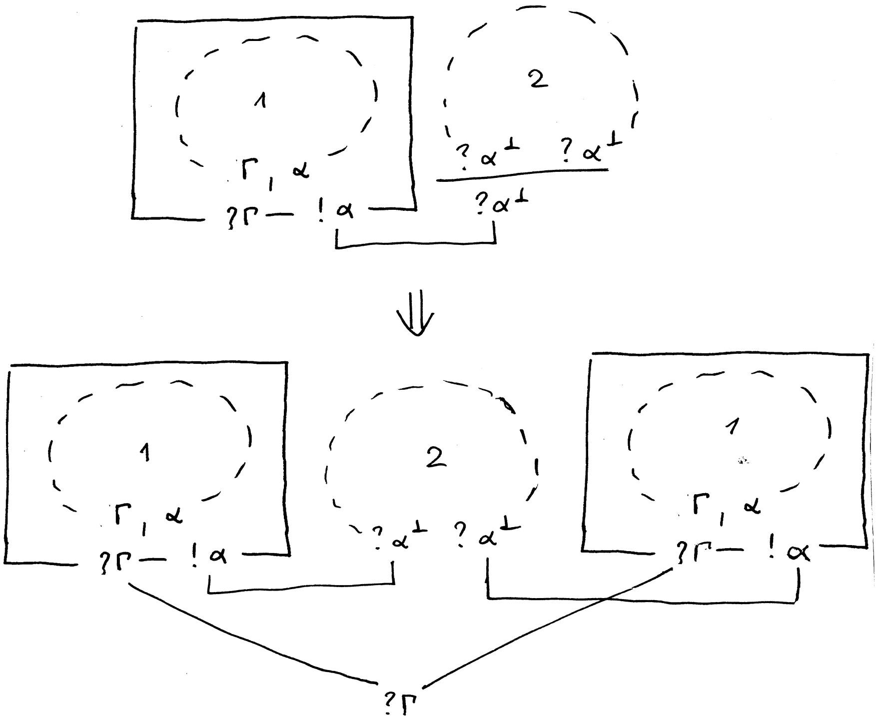
Jednorazowe użycie danej to cięcie z derelikcją (Obrazek 28). Jego realizacja polega na porzuceniu pudełka po wyjęciu z niego zawartości (do jednorazowego spożycia).



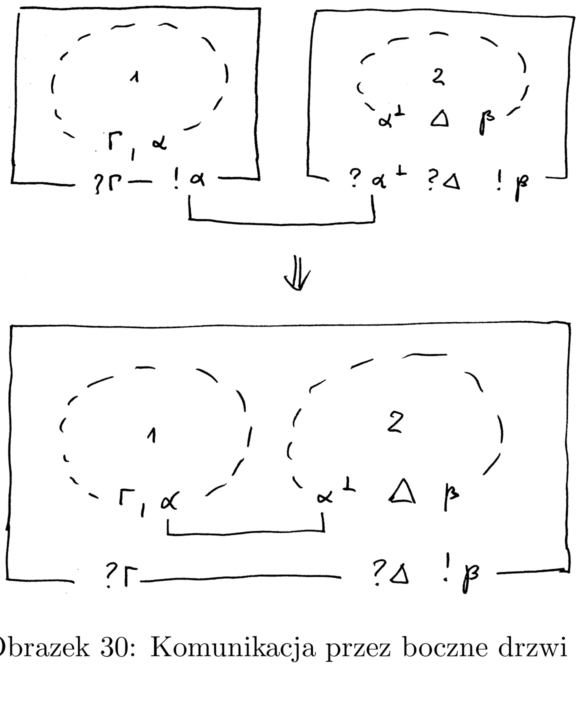
Obrazek 28: Jednorazowe zużycie zawartości pudełka

Cięcie ze skracaniem odpowiada sytuacji, w której dana typu $!\alpha$ jest rozdzielana na dwie takie same i przekazywana do dwóch różnych odbiorników. Wymaga to podwojenia pudła (Obrazek 29).

Następny obrazek przedstawia eliminację cięcia pomiędzy głównymi drzwiami jednego pudełka i bocznymi drzwiami drugiego. Taka komunikacja jest jak szeroka rura łącząca jedno pudło z drugim. Topologicznie to to samo, co jedno pudło (Obrazek 30).

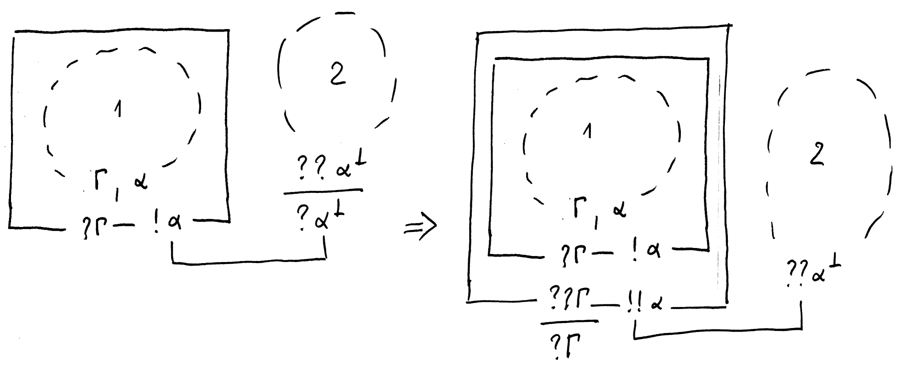


Obrazek 29: Rozmnażanie pudełek



Obrazek 30: Komunikacja przez boczne drzwi

Wreszcie może się zdarzyć, że głowne drzwi pudełka są połączone cięciem z regułą delta. Aby umożliwić komunikację, tj. przesłać sygnał typu $!\alpha$ do odbiornika typu $!!\alpha$ (czyli nadajnika typu $??\alpha^\perp$) musimy „podnieść" rangę wysyłanego sygnału o jeszcze jeden wykrzyknik. A więc musimy nasze pudełko... włożyć do drugiego pudełka (Obrazek 31).

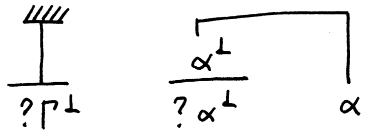


Obrazek 31: Matrioszka

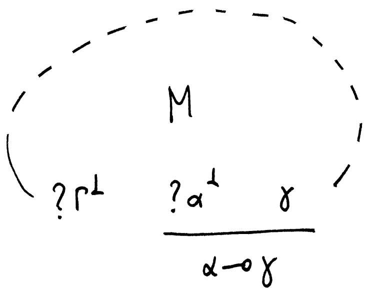
Twierdzenie o eliminacji cięcia dla nieortodoksyjnych sieci nie jest natychmiastową konsekwencją Twierdzenia 5.3, bo przecież nasze sieci są trochę inne. Co gorsza, na Obrazku 31 mamy redukcję podnoszącą stopień cięcia. Ale szczęśliwie tylko na chwilę. Formuła $??\alpha^\perp$ jest otrzymana w sieci (2) na jeden z czterech sposobów: jako boczne drzwi pudełka, przez derelikcję, aksjomat, lub jeszcze jedno wystąpienie reguły delta. Postępując przez indukcję ze względu na liczbę wystąpień delty w sieci (2), wyeliminujemy wreszcie wszystkie cięcia wyższych stopni.

Rachunek lambda

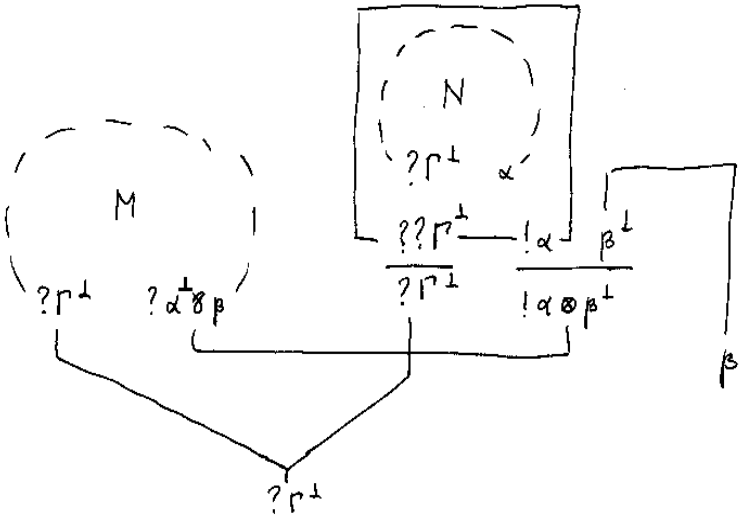
Term M, który w otoczeniu $\Gamma = \{x_1 : \gamma_1, \ldots, x_n : \gamma_n\}$ ma typ τ, można przedstawić jako sieć o konkluzjach $?\gamma_1^\perp, \ldots, ?\gamma_n^\perp, \tau$. Na Obrazkach 32–34 mamy kolejno termy postaci $x : \alpha$, $\lambda x^\alpha.M^\gamma : \alpha \to \gamma$ oraz $M^{\alpha\to\beta}N^\alpha : \beta$. Niestety, beta-redukcja jest implementowana przez eliminację cięcia tylko „z dokładnością do reguł strukturalnych": rezultat może się różnić od oczekiwanego sposobem „obsługi" pytajników.



Obrazek 32: Zmienna



Obrazek 33: Abstrakcja



Obrazek 34: Aplikacja

Geometria interakcji

Orientacja i etykiety krawędzi w sieciach multiplikatywno-wykładniczych są przedstawione na Obrazku 35.

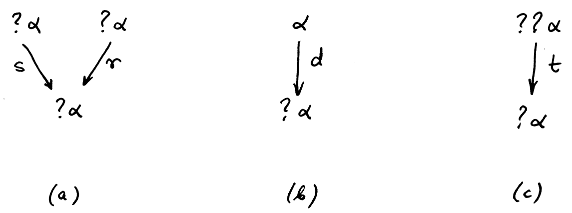
Algebra ścieżek ma więc dodatkowe stałe r, s, d, t. Ścieżka może przechodzić przez pudełko, którego drzwi utożsamia się wtedy z odpowiednią konkluzją „wewnętrznej" podsieci. Ale każdy odcinek przechodzący przez pudełko oznaczany jest wykrzyknikiem. Na przykład na Obrazku 36 można wskazać ścieżkę od A do $?(B^\perp \otimes A)$, której przypiszemy wagę $t \cdot !(dq) \cdot d$. Uprościmy tę wagę do $tddq$, gdy uzupełnimy aksjomaty algebry dynamicznej o dodatkowe równania:

$$r^*r = s^*s = d^*d = t^*t = 1 \qquad r^*s = s^*r = 0;$$

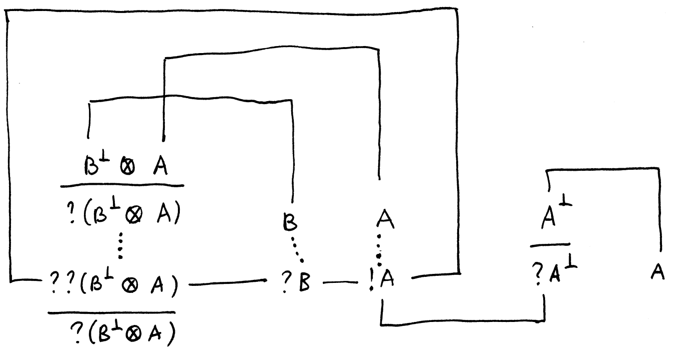
$$(!x)^* = !x^* \qquad !x!y = !(xy) \qquad !(x+y) = !x + !y;$$

$$!xr = r!x \qquad !xs = s!x \qquad !xd = dx \qquad !xt = t!!x.$$

Objaśnienie równości $t \cdot !(dq) \cdot d = tddq$ ułatwia Obrazek 36. Nasza ścieżka wchodzi do pudełka przez cięcie pomiędzy derelikcją (etykieta d) i głównymi drzwiami. Eliminacja tego cięcia (Obrazek 28) spowoduje zniknięcie pudełka i pojawienie się dodatkowej krawędzi oznaczonej



Obrazek 35: Znakowane przejścia



Obrazek 36: Przykład sieci

przez d. W rezultacie wagą naszej ścieżki po deformacji będzie właśnie $tddq$. To zjawisko wyraża aksjomat $!xd = dx$. Pozostawiamy czytelnikowi przyjemność uzupełnienia kolorowym ołówkiem:

- Obrazka 29, aby objaśnić aksjomaty $!xr = r!x$ oraz $!xs = s!x$;
- Obrazka 30, aby objaśnić aksjomat $!x!y = !(xy)$;
- Obrazka 31, aby objaśnić aksjomat $!xt = t!!x$.

Niestety, waga ścieżek nie jest tym razem niezmiennikiem eliminacji cięcia (np. w przypadku komunikacji pomiędzy bocznymi drzwiami tego samego pudełka). Ścieżka o niezerowej wadze może nawet po prostu zniknąć. Na szczęście dzieje się tak tylko wtedy, gdy nasza ścieżka jest osłabiona przez uziemienie. Dla pozostałych ścieżek zachodzi nadal równoważność: ścieżka jest uporczywa wtedy i tylko wtedy, gdy jej waga nie jest zerem.

Ćwiczenia

1. Udowodnić twierdzenie o eliminacji cięcia dla sieci multiplikatywno-wykładniczych.
2. Zaproponować reguły eliminacji cięcia dla sieci z ortodoksyjnymi pudełkami.
3. Pokazać, że beta-redukcja nie jest poprawnie zaimplementowana przez eliminację cięcia.
4. Pokazać przykład niezerowej i uporczywej ścieżki, która zmienia wagę podczas eliminacji cięcia.

13 Optymalne redukcje

Problem *optymalnych redukcji* polega na tym aby ewaluację lambda-wyrażenia, czy też wyrażenia w funkcyjnym języku programowania, realizować bez powtarzania tych samych czynności. Oznacza to, że nie można dopuszczać do duplikowania redeksów, a nawet takich wyrażeń, które mogą stać się redeksami „w przyszłości”. Naturalne rozwiązanie tego zadania polega na takiej grafowej reprezentacji termów, w której kopie tego samego podwyrażenia są reprezentowane przez ten sam podgraf, a rozdzielenia ich dokonuje się tylko wtedy gdy jest to konieczne. Beta-redukcja realizowana jest jako lokalna transformacja grafu jak na Obrazku 20. Taka grafowa implementacja redukcji nie jest zadaniem łatwym, ponieważ to samo wyrażenie może być podczas ewaluacji kopiowane wielokrotnie. Konieczna jest wtedy pełna kontrola nad strukturą współdzielenia poszczególnych fragmentów ewaluowanego wyrażenia. *Algorytm Lampinga* rozwiązuje to zadanie za pomocą klasyfikacji poszczególnych wierzchołków grafu ze względu na „głębokość” ich występowania w argumentach aplikacji, oraz wprowadzenie dwóch dodatkowych typów wierzchołków *croissant* i *bracket*, których rolą jest kontrolowanie interakcji pomiędzy wierzchołkami o różnej głębokości.

Na następnej stronie mamy kopie kilku ilustracji z książki

- Asperti, A., Guerrini, S., *The Optimal Implementation of Functional Programming Languages*, Cambridge University Press, 1998.

Fig. 3.13–3.14 definiują początkową reprezentację lambda-termu jako grafu. Zasada jest taka, że każdej zmiennej występującej w termie odpowiada dokładnie jedno „wejście” u dołu. Dlatego w przypadku aplikacji łączymy wejścia do obu składowych za pomocą „rozgałęźnika” (*fan node*). Argument aplikacji ma „głębokość” większą o 1 od operatora. Wierzchołki typu *bracket* stanowią tu przejście pomiędzy poziomem n i $n+1$.

Fig. 3.10 przedstawia komplet wierzchołków używanych w grafach Lampinga, a dwie następne ilustracje przedstawiają reguły redukcji grafów. Na ilustracji o numerze 3.11 indeksy głębokości komunikujących się wierzchołków są takie same, a na ilustracji 3.12 mamy $i < j$.

Bez wchodzenia w dalsze szczegóły zauważmy teraz uderzającą analogię pomiędzy konstrukcją Lampinga i eliminacją cięcia w sieciach dowodowych. Jeśli wrócimy do Obrazków 32–34 i trochę je poprawimy, to otrzymamy w istocie Fig. 3.13. Należy w tym celu tylko zamienić derelikcję na rogala a regułę delta na kwadratową klamrę. Teraz policzmy pudełka otaczające dowolny wierzchołek grafu i zobaczymy, że otrzymamy dokładnie jego głębokość.

Oczywiście etykietowanie z Obrazków 18 i 35 można przenieść na grafy Lampinga. Etykiety r i s postawimy przy wejściach do *fan node*, etykiety p i q przy lambdach i małpach, d przy derelikcji czyli rogalu, a t przy klamerce. Zobaczmy teraz, że redukcje z rysunków 3.11 i 3.12 zachowują wagi ścieżek, przechodzących przez wierzchołki w nich uczestniczące. Na przykład prawa górna redukcja z Fig. 3.11 odpowiada równaniu $t^*t = 1$, a środkowa redukcja z Fig. 3.12 równaniu $!x \cdot d = d!x$. A więc algorytm Lampinga to w istocie pewien proces „lokalnej eliminacji cięcia”. Zadziwiające, że w istocie ta sama technika pojawiła się tu całkowicie niezależnie od prac Girarda i innych nad geometrią interakcji. Tu niespodziewanie kończą się te notatki, które może zachęcą czytelnika do samodzielnego zgłebienia oryginalnej literatury.

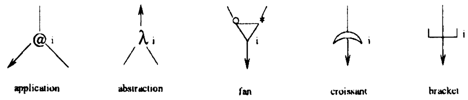


Fig. 3.10. Sharing graph operators.

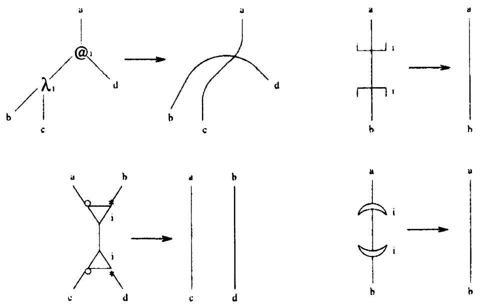


Fig. 3.11. Interaction rules.

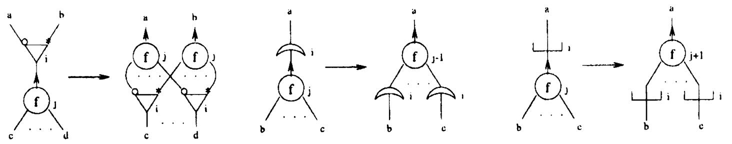


Fig. 3.12. Action rules.

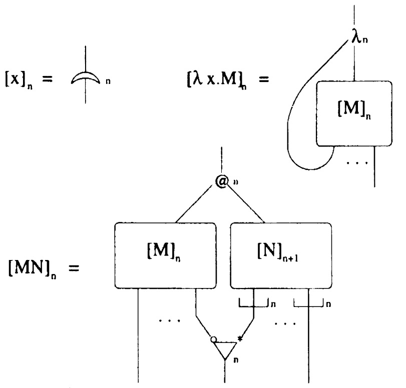


Fig. 3.13. Initial translation.

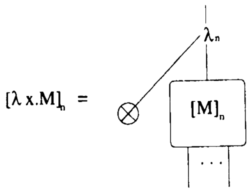


Fig. 3.14. The translation of $\lambda x.M$ when $x \notin FV(M)$.

14 Literatura

Zaczęło się od takiej pracy (zajmującej cały zeszyt czasopisma):

- J.-Y. Girard, Linear logic, *Theoretical Computer Science*, 50(1):1–101, 1987.

Może ona nadal służyć jako wprowadzenie do logiki liniowej, ale od tego czasu ukazało się wiele innych publikacji o takim charakterze. Na przykład:

- Gallier, J., Constructive Logics. Part II: Linear Logic and Proof Nets, Raport DEC, Paryż 1991.
- J.-Y. Girard, Linear logic: A survey. In P. de Groote, editor, *The Curry-Howard Isomorphism* (P. de Groote, ed.), volume 8 of *Cahiers du centre de logique*, Universite catholique de Louvain, 1995, pp. 193–255.
- J.-Y. Girard, Linear logic: its syntax and semantics, in [1], pp. 1–42.
- Danos, V., Di Cosmo, R., *The Linear Logic Primer*, `http://www.pps.jussieu.fr/~dicosmo/CourseNotes/LinLog`.
- P.-L. Curien. Introduction to linear logic and ludics, Part I. *Advances in Mathematics (China)*, 34(1-3):513–544, 2005.
- A.S. Troelstra. *Lectures on Linear Logic*, volume 29 of *CLSI Lecture Notes*. Center for the Study of Language and Information, Stanford, 1992.

Ostatnia pozycja jest książką całkowicie poświęconą logice liniowej i napisaną w bardzo przystępny sposób, niestety, we własnej notacji autora, co znakomicie uatrakcyjnia lekturę. Następne dwie pozycje książkowe to zbiory artykułów dotyczących logiki liniowej i różnych zagadnień pokrewnych.

- [1] J.-Y. Girard, Y. Lafont, and L. Regnier, editors. *Advances in Linear Logic*, vol. 222 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1995.
- [2] T. Ehrhard, J.-Y. Girard, P. Ruet, and P. Scott, editors. *Linear Logic in Computer Science*, volume 316 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 2004.

Ukazały się też dwa zeszyty specjalne *Theoretical Computer Science* poświęcone logice liniowej: numer 227(1-2), 1999 i numer 294(2), 2003.

Teraz prace związane z obliczeniowymi interpretacjami logiki liniowej.

- S. Abramsky. Computational interpretations of linear logic. *Theoretical Computer Science*, 111(1-2):3–57, 1993.
- Benton, N., Bierman, G., de Paiva, V., Term assignment for intuitionistic linear logic, Cambridge University, 1992. `http://research.microsoft.com/~nick/tr262.ps`
- Benton, N., Strong normalisation for the linear term calculus, *J. Functional Programming* 5 (1995), 65–80.

Nierozstrzygalność i złożoność:

- Lincoln, P., Deciding provability of linear logic formulas, in [1], pp. 109–122.
- Lincoln, P., Scedrov, A., Shankar, N., Linearizing intuitionistic implication, *Annals of Pure and Applied Logic* 60 (1993) 151–177.

- Lincoln, P., Mitchell, J., Scedrov, A., Shankar, N., Decision problems for propositional linear logic, *Annals of Pure and Applied Logic* 56 (1992) 239–311.
- de Groote, Ph., Guillaume, B., Salvati, S., Vector addition tree automata, in: *Proc. 19th Annual Symp. on Logic in Computer Science* (LICS 2004), pp. 64–73.
- Lafont, Y., The undecidability of second order linear logic without exponentials, *Journal of Symbolic Logic*, 61(2):541–548, 1996.

Semantyka, zwłaszcza przestrzenie zgodności:

- J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.
- R.M. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1998.
- Girard, J.-Y, On denotational completeness, *Theoretical Computer Science*, 227 (1999), 249–273.
- Blute, R., Scott, Ph., Category theory for linear logicians, in [2], pp. 1–64.

Sieci dowodowe:

- Guerrini, S., Correctness of multiplicative proof nets is linear, *Proc. 14th Annual Symp. on Logic in Computer Science* (LICS 1999), pp. 454–463.
- Guerrini, S., Proof nets and the λ-calculus, in [2], pp. 65–118.

Oryginalne prace Girarda o geometrii interakcji:

- J.-Y. Girard, Towards a geometry of interaction, *Contemporary Mathematics*, 92, 1989, 69–108.

  J.-Y. Girard, Geometry of interaction I: interpretation of system F, in: *Logic Colloquium '88*, Forro, Bonotto, Valentini and Zanardo eds., Elsevier, 1989, pp. 221–260.

  J.-Y. Girard, Geometry of interaction 2: deadlock-free algorithms, in: *Proc. COLOG'88*, P. Martin-Löf and G. Mints, eds. LNCS 417, Springer-Verlag, 1988, pp. 76–93.

  J.-Y. Girard, Geometry of interaction III: accomodating the additives, in [1], pp. 329–389.

Prace o geometrii interakcji bardziej nadające się dla zwykłych śmiertelników (zob. też Danos i Di Cosmo powyżej).

- Asperti, A., Laneve, C., Danos, V., Regnier, L., Paths in the lambda-calculus: three years of communications without understanding, *Proc. 9th Annual Symp. on Logic in Computer Science* (LICS 1994), pp. 26–436.
- V. Danos and L. Regnier, Proof-nets and the Hilbert space, in [1], pp. 307–328.

Optymalne redukcje i okolice

- Lévy, J.-J., Optimal reductions in the lambda-calculus, in: To H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, J.P. Seldin and J.R. Hindley, editors, Academic Press, 1980, pp. 159–191.
- Lamping, J., An algorithm for optimal lambda calculus reduction, *Proc. Symp. Principles of Programming Languages*, 1990, pp. 16–30.

- A. Asperti and S. Guerrini. *Optimal Implementation of Functional Programming Languages*, volume 45 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1998.
- H.G. Mairson, From Hilbert space to Dilbert space: Context semantics made simple, in M. Agrawal and A. Seth, editors, *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science*, volume 2556 of *Lecture Notes in Computer Science*, pages 2–17. Springer-Verlag, 2002.
- Neergaard, P.M., Mairson, H.G., Types, Potency, and Idempotency: Why Nonlinearity and Amnesia Make a Type System Work. In *Proc. 9th Int'l Conf. Functional Programming* , ACM, 2004, pp. 138–149.

Jak prosta ścieżkologia może się do czegoś przydać.

- Z. Spławski, P. Urzyczyn, Type fixpoints: iteration vs. recursion, *Proc. 4th Int'l Conf. Functional Programming*, ACM, 1999, pp. 102–113.
- T. Knapik, D. Niwiński, P. Urzyczyn, Deciding monadic theories of hyperalgebraic trees, *Proc. TLCA'01*, (S. Abramsky, Ed.), LNCS 2044, Springer-Verlag, Berlin, 2001, pp. 253–267.

Najmodniejsze od kilku sezonów: ludics.

- J.-Y. Girard, Locus Solum: From the rules of logic to the logic of rules. *Mathematical Structures in Computer Science*, 11(3):301–506, 2001.
- J.-Y. Girard, From foundations to ludics. *Bulletin of Symbolic Logic*, 9(2):131–168, 2003.
- P.-L. Curien, Introduction to linear logic and ludics, Part II. `http://www.pps.jussieu.fr/~curien/LL-ludintroII.pdf`

Nowość: wykłady Girarda, Rzym 2004 (wersja angielska). Uwaga: około 500 stron

- J.-Y. Girard, *The Blind Spot. Lectures on proof-theory, Roma Tre, October-December 2004.* `http://iml.univ-mrs.fr/~girard/coursang/coursang.html`

Podziękowania

Za komentarze i poprawki dziękuję Panom: Sławomirowi Kolasińskiemu, Łukaszowi Lwowi, Pawłowi Parysowi, Adamowi Warskiemu i Piotrowi Wilkinowi.