EDUARDO NAHUM OCHS

CATEGORIAS, FILTROS E INFINITESIMAIS NATURAIS

DISSERTAÇÃO DE MESTRADO

Departamento de Matemática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, abril de 1999

EDUARDO NAHUM OCHS

CATEGORIAS, FILTROS E INFINITESIMAIS NATURAIS

DISSERTAÇÃO DE MESTRADO

Dissertação apresentada ao Departamento de Matemática da PUC/RJ como parte dos requisitos para obtenção do título de Mestre em Matemática Aplicada.

Orientador: Nicolau Corção Saldanha.

Departamento de Matemática

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, abril de 1999

Oh, don’t you know that anyone can join
And they come and they call and they fall on the floor
Don’t you see when you’re looking at me
That I’ll never end?
Transcend, transcend
Ain’t it strange
(...)
Do you go to the temple tonight?
Oh no, I don’t think so
Do you not go to the palace of answers with me Marie?
Oh no, I don’t think so, no
See when they offer me book of gold,
I know soon still that platinum is coming
And when I look inside of your temple
It looks just like the inside of the brain of anyone man,
And when he beckons his finger to me,
Well, I *MOVE* in another direction,
I *MOVE* in another dimension
I spin, I spiral, and I splatter
Hand of God, I feel the finger,
Hand of God, and I start to whirl
And I whirl, and I whirl,
Don’t get dizzy, do not fall now,
Turn, God, God — strange
Go, go on, Go,
Like a dervish,
Turn, God, — strange — make a move
Turn, Lord, — strange — I don’t get nervous
Oh I just move in another dimension
Come, move in another dimension
Strange, strange
Ain’t it strange

Patti Smith/Ivan Kral: *“Ain’t it strange”*
de *Radio Ethiopia*, 1976

a:

Iggy Pop
Angel Vianna
F. M. Alexander
Vange Leonel
J. Y. Girard
Richard Stallman

Agradecimentos:

...a Carlos Tomei, que me tirou do buraco e me trouxe para a Matemática;
a Nicolau Saldanha, pela paciência e senso de realidade;
a Luiz Carlos Pereira, Edward Hermann Haeusler e Paulo Henrique Viana de Barros, professores e membros da banca;
ao CNPq, pelo apoio financeiro;
ao Departamento de Matemática da PUC/RJ, pelo ambiente empolgante e pela sua fantástica capacidade de sobreviver em tempos difíceis;
a Renato Vidal e a outras pessoas que se dispuseram a discutir comigo idéias desta tese durante a sua elaboração;
a todas as pessoas que me ajudaram ou me influenciaram (elas saberão quem são).

Resumo:

A análise não-standard permite falar de infinitesimais e assim admite para alguns teoremas demonstrações mais simples do que as da análise classica. Sempre é possivel traduzir uma demonstração não-standard para uma standard, talvez bem mais longa e complicada que a original.

Uma certa classe de demonstrações admite uma tradução simples e que preserva a estrutura da prova original; essa tradução é feita trocando o ultrafiltro por um filtro adequado e os infinitesimais por constantes com certas propriedades universais.

Quando trocamos o quociente por um ultrafiltro por um quociente por um filtro a lógica que o sistema obedece fica mais fraca; ela continua sendo booleana, mas passa a ter mais de dois valores de verdade. O modelo montado com o filtro é um *topos booleano*, e para entendê-lo melhor introduzimos λ-cálculo, dedução natural, categorias e topoi em geral.

Abstract:

Some theorems of classical analysis admit shorter proofs in non-standard analysis, as in NSA we can formalize easily arguments involving infinitesimals. It is always possible to find a standard counterpart of a non-standard proof, but it may be much longer and much more complex. Hovewer, there's a class of proofs in NSA that admit a simple translation that preserves proof structure, done by replacing the ultrafilter by a filter and infinitesimals by certain constants with universal properties.

When we take the quotient with a filter we get a system with a weaker logic, still boolean but with more than two truth-values. This system is a *boolean topos*, and to understand how it works we introduce λ-calculus, natural deduction, categories and topoi in general.

Sumário:

Chapter 1

Introdução

Em geral uma prova de um teorema não é só uma seqüência formal de símbolos; uma prova é também a expressão da idéia que temos de como mostrar que aquele teorema é verdade, e nesse sentido podemos falar de provas que estão mais próximas da nossa idéia da prova, e provas que estão mais distantes. Claro que não dá pra dizer que cada prova tem *uma idéia* por trás, e que essa uma idéia precisa ser deformada até se tornar aceitável matematicamente e depois "desdeformada", no sentido de que uma vez que a idéia tenha sido formalizada podemos construir uma ponte entre a versão formal e a intuitiva, e outros matemáticos podem ter acesso a exatamente a mesma idéia intuitiva que nós. Não só a intuição de cada matemático é diferente, como no processo de formalização a própria idéia original vai se transformando, possivelmente para se tornar mais compatível com as outras idéias que já temos e com a nossa linguagem.

Se a nossa linguagem for restritiva demais e forçarmos todas as idéias a se encaixarem nela muito rápido muitas idéias vão ser formalizadas às pressas, vão perder a ligação com a idéia intuitiva e vão se tornar áridas; outras vão ser simplesmente descartadas, para abrir espaço para outras idéias que possam ser formalizadas mais facilmente. Num caso extremo disso acabaríamos fazendo só "matemática morta" — algo como o estilo Bourbaki, mas certamente sem tanto brilho.

Uma área em que é especialmente difícil encontrar o meio-termo certo entre as idéias intuitivas e a formalidade é a Análise, e ela tem um problema especialmente interessante: muitos dos argumentos mais intuitivos são em termos de "infinitesimais", ou pelo menos de "quantidades suficientemente pequenas", que podem ser formalizados tanto em termos de 'ϵ's, 'δ's e limites — o modo clássico, que todo mundo aceita, mas às vezes é preciso transformar bastante uma idéia com infinitesimais para pô-la nesta forma — quanto usando análise não-standard, que permite infinitesimais mas que é pouco conhecida e que não está bem no caminho entre as idéias intuitivas e a análise clássica: ela é bastante difícil, cheia de idiossincrasias próprias, e não há uma boa tradução de provas não-standard para provas clássicas.

O grande tema deste trabalho é uma linguagem que se pretende a ser um meio-termo entre certos tipos de argumentos intuitivos, a análise clássica e a análise não-standard. Essa linguagem é inspirada nos rabiscos que a gente faz num canto de papel enquanto está tentando entender certos tipos de problemas de Cálculo e Análise; os rabiscos são como o esqueleto da solução, e é possível preencher aos poucos os detalhes que faltam até chegarmos a uma solução suficientemente formal. Na nossa linguagem intermediária é possível dar uma boa interpretação para infinitesimais, e construções feitas nela podem ser traduzidas facilmente para a análise clássica, ao contrário de demonstrações arbitrárias em análise não-standard.

Essa linguagem intermediária é mais fraca que a linguagem das "análises" em dois sentidos: 1) ela é capaz de expressar menos coisas, e 2) ela tem mais *modelos*, isto é, pode ser interpretada de mais formas. Algumas dessas formas nos levam a bons modos de expôr as próprias fundações lógicas dessa linguagem: como estamos mais interessados em *demonstrações* do que só em *verdades* é mais natural pensar em termos de teoria da prova, o que nos remete ao *intuicionismo*, que é um sistema lógico com uma teoria da prova muito mais bem comportada que a da lógica clássica; um dos melhores modos de entender o intuicionismo é começar pelo *isomorfismo de Curry-Howard*, que estabelece uma correspondência entre *construções* (vistas como termos de λ-cálculo) e dedução natural intuicionista. O isomorfismo de Curry-Howard vai ser tão onipresente ao longo do texto que vamos ter muito poucas oportunidades de nos referir a ele pelo nome.

Depois que sabemos um pouco de intuicionismo e λ-cálculo um passo natural é ver o que são *topoi*, que dão um modo de interpretar a análise e a teoria dos conjuntos com uma lógica mais fraca, intuicionista ao invés de clássica. O nosso modelo intermediário entre as duas análises, com infinitesimais, pode ser visto como um topos. A lógica que ele obedece está entre a clássica e a intuicionista, é booleana mas tem mais de dois valores de verdade.

O modelo intermediário para as "análises" é muito simples e pode ser entendido sem grandes pré-requisitos; ele está descrito na seção 7.5. Os dois últimos capítulos do texto são capítulos de notas: o penúltimo é uma introdução rápida a topoi (que só aparecem explicitamente lá), e o último é um amontoado de esboços sobre aplicações desse modelo e de outras técnicas relacionadas. Nós vamos nos concentrar em dar boas introduções aos tópicos mais básicos e vamos fingir que topoi, o modelo intermediário e as suas aplicações são só curiosidades secundárias, ou objetivos que não foram atingidos.

1.1 Um primeiro problema

Vamos tomar como primeiro exemplo o problema básico de cálculo de variações, só porque ele gera um diagrama suficientemente interessante (fig. 1). Considere uma função suave do intervalo $[a, b]$ (o espaço dos 'x') em $\mathbb{R}$ (o espaço dos 'y'); essa função ao invés de ganhar um nome especial vai ser referida só pelo seu "tipo", $x \to y$. A partir dessa função $x \to y$ podemos construir uma outra função de $[a, b]$ em $\mathbb{R}$, a sua derivada, $x \to y'$; juntando a função identidade $x \to x$ e as funções $x \to y$ e $x \to y'$ podemos formar uma função de $[a, b]$ em $\mathbb{R}^3$, $x \to (x, y, y')$. Usando j como abreviatura para (x, y, y') esse processo transforma uma função $x \to y$ suave em uma função $x \to j$, e podemos considerá-lo como uma função indo do espaço dos '$x \to y$'s, que vamos definir como sendo o conjunto das funções suaves de $[a, b]$ em $\mathbb{R}$, no espaço das '$x \to j$'; o "tipo" da função correspondente a esse processo seria $(x \to y) \to (x \to j)$, e, de novo, vamos usar o nome do tipo como nome para a função. Não é claro como o espaço das '$x \to j$' deva ser definido: podemos aceitar como pontos dele todas as funções suaves de $[a, b]$ em $\mathbb{R}^3$ ou impôr condições nas $x \to j$ para que elas sejam a imagem por $(x \to y) \to (x \to j)$ de alguma função $x \to y$, mas felizmente isso não vai nos importar: para completar a formulação do problema só precisamos agora de uma função $j \to L$ que dê a "densidade de energia" para a curva $x \to y$; obtendo o $x \to j$ correspondente ao $x \to y$ e compondo-o com a $j \to L$ obtemos uma função $x \to L$, que deve poder ser integrada no intervalo $[a, b]$ para dar um valor que vamos chamar de I e que diz a energia total da curva $x \to y$. O problema básico de cálculo de variações é analisar quais curvas

$x \to y$ são mínimos locais de I, para certas $j \to L$ e para certas definições de quem é o espaço das $x \to y$ válidas; por exemplo, podemos só estar interessados nas $x \to y$ com comprimento de arco igual a uma constante, ou seja, as com $\int_a^b \sqrt{1 + y'(x)^2}\, dx = A$.

O diagrama para esse problema é o que aparece abaixo. A variável s e a seta $s \to (x \to y)$ são introduzidas para permitir que deformemos a curva $x \to y$ variando o parâmetro s, e em algum momento (na verdade só nos capítulos de notas) vamos aprender como derivar formalmente I com relação a s, e examinando a derivação formal vamos encontrar condições que fazem com que essa derivação formal valha de verdade.

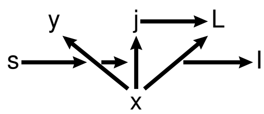


Figura 1.

Esse diagrama deixa bem visível como certos objetos podem ser construídos, ou *deduzidos*, a partir de outros: por exemplo, a partir de x e $x \to y$ obtemos um y, ou "se eu sei x e $x \to y$ então sei y"; só que ele está numa forma econômica demais, em que não há setas ou letras repetidas: a seta entre x e y só aparece uma vez, ao invés de aparecer uma no lado direito de $s \to (x \to y)$, outra por si só e outra no lado esquerdo de de $(x \to y) \to (x \to j)$. Inicialmente a forma abaixo ("em árvore") vai ser bem mais útil.

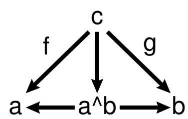
$$\dfrac{\dfrac{\dfrac{\dfrac{s \quad s \to (x \to y)}{x \to y} \qquad (x \to y) \to (x \to j)}{x \to j} \qquad j \to L}{x \to L} \qquad (x \to L) \to I}{I}$$

Figura 2.

Veja que essa árvore pode ser lida de dois modos: numa interpretação *funcional* as barras dizem coisas como "se eu tenho um ponto α e uma função $\alpha \to \beta$ então eu tenho um ponto β" e "se eu tenho uma função $\alpha \to \beta$ e uma $\beta \to \gamma$ eu posso formar a composta $\alpha \to \beta$"; numa interpretação *lógica* interpretamos cada tipo como uma proposição e lemos as barras como "se α é verdade e α implica β então β é verdade" e "se α implica β e β implica γ então, por transitividade, α implica γ".

Vamos esquecer por um momento que conhecemos a árvore da figura 2 e pensar que só temos as funções $s \to (x \to y)$, $(x \to y) \to (x \to j)$, $j \to L$ e $(x \to L) \to I$. Se nós tivermos técnicas que mostrem facilmente que existe *exatamente uma* construção natural para a função $s \to I$ — talvez passando para a interpretação lógica e mostrando que "existe uma demonstração, essencialmente única, de $s \to I$ a partir das hipóteses" — então podemos falar da função $s \to I$ *sem dar explicitamente a sua construção, e sem risco de ambiguidade*. Isso pode nos levar a um cálculo diferencial formal bem poderoso, em que seria permitido usar derivadas como I_s sem precisarmos definir explicitamente a função $s \to I$, e sem precisarmos expandir imediatamente I_s numa expressão enorme escrita em termos de derivadas mais familiares.

1.2 Setas naturais



Digamos que $\mathbf{E}_{a\wedge b}$ seja o espaço dos pares ordenados formados por um a e um b, ou seja, $\mathbf{E}_{a\wedge b} = \mathbf{E}_a \times \mathbf{E}_b$. Então existe uma função natural indo de $\mathbf{E}_{a\wedge b}$ em $\mathbf{E}_a$ (ou: "uma função levando cada $a \wedge b$ num a"), que é a projeção na primeira coordenada, e uma indo de $\mathbf{E}_{a\wedge b}$ em $\mathbf{E}_b$, a projeção na segunda coordenada.

Digamos que temos uma função f que leva cada c num a (i.e., esse a é tomado como sendo $f(c)$) e uma outra função, g, que leva cada c num b ($b := g(c)$). Temos um jeito natural de construir uma função $c \to a \wedge b$ a partir delas: basta tomar para cada c o par ordenado formado pelo a e pelo b correspondentes.

A coisa curiosa é que o diagrama montado com f, g e essas funções naturais *comuta*, i.e., os dois jeitos naturais de obter uma função $c \to a$ a partir do diagrama — tomar a própria f ou tomar a composta $c \to a \wedge b \to a$ — dão o mesmo resultado, e idem para $c \to b$. Claro que não podemos esperar que todo diagrama comute desde que algumas das suas setas tenham sido escolhidas do modo natural, mas em um número surpreendentemente grande de casos práticos basta checar que um diagrama obedece umas poucas condições — fáceis de verificar — para ver que ele comuta. Essas condições vão ser abordadas no capítulo 4, e essa idéia de *objetos naturais*, junto com uma notação inspirada em considerar **Set** como a categoria principal, vai nos permitir introduzir vários conceitos básicos de categorias rapidamente.

Além disso muitos desses diagramas vão poder ser interpretados tanto funcionalmente quanto logicamente, como as árvores da seção anterior. Repare que construímos uma função $c \to a \wedge b$ a partir de uma $c \to a$ e de uma $c \to b$, mas poderíamos ter feito o contrário: a partir de uma $c \to a \wedge b$ é possível obter a $c \to a$ e a $c \to b$ correspondentes compondo a $c \to a \wedge b$ com as projeções. Essas operações são a inversa uma da outra, e dão uma bijeção (também chamada de "natural") entre as '$c \to a \wedge b$'s e os pares formados por uma $c \to a$ e uma $c \to b$. Na interpretação lógica podemos considerar que uma seta $\alpha \to \beta$ existe se e só se α implica β em algum sentido, por exemplo, se β é demonstrável a partir de α; então as setas $a \wedge b \to a$ e $a \wedge b \to b$ sempre existem (se temos uma noção decente de demonstrabilidade, é claro), e a seta $c \to a \wedge b$ existe se e só se tanto a $c \to a$ quanto a $c \to b$ existirem, e temos de novo uma bijeção.

1.3 Tipos

Vamos usar a letra $\mathcal{A}$ para denotar o conjunto dos *tipos atômicos* de um sistema, junto com os seus espaços associados. Formalmente, $\mathcal{A} = (A, E)$, onde A é o conjunto dos *nomes de tipos atômicos* e E é uma função que leva cada membro de A no seu espaço associado; para $a \in A$ costumamos escrever $\mathbf{E}_a$ ao invés de $E(a)$, como fizemos na seção sobre o problema variacional.

Em geral vamos usar as letras minúsculas romanas $a, b, \ldots, z, \mathbb{n}, \mathbb{r}$ e $\mathbb{z}$ como nomes de tipos atômicos, e vamos convencionar que $\mathbf{E}_{\mathbb{n}} = \mathbb{N}$, $\mathbf{E}_{\mathbb{r}} = \mathbb{R}$ e $\mathbf{E}_{\mathbb{z}} = \mathbb{Z}$. Como essas letras são usadas como *nomes* de tipos, não como *variáveis* de tipos, duas letras diferentes denotam tipos diferentes; consideramos x e y tipos diferentes mesmo que $\mathbf{E}_x = \mathbf{E}_y$.

Definição: um *ponto do tipo* α é um ponto de $\mathbf{E}_\alpha$. Às vezes vamos chamar um ponto de tipo α simplesmente de "um α".

Quando falarmos só de "um ponto", ou "um ponto de $\mathbf{Set}_{\mathcal{A}}$", onde $\mathbf{Set}_{\mathcal{A}}$ é o "universo dos pontos tipados", que vamos definir daqui a pouco, em geral vamos querer que esse ponto carregue a informação de qual é o seu tipo; nesse sentido se $\mathbf{E}_x = \mathbf{E}_y = \mathbb{R}$ então o 2 pertencente ao espaço dos $x$zes é diferente do 2 pertencente ao espaço dos ys, e para indicar isso vamos escrevê-los como 2^x e 2^y ("constantes tipadas"). 2^x, 2^y, $2^{\mathtt{r}}$ e $2^{\mathtt{n}}$ são pontos diferentes, já que são constantes tipadas e têm tipos diferentes; o fato de que temos uma função do espaço de $x$zes no espaço de ys que é essencialmente a identidade costuma ser deixado de lado. Pode parecer estranho trabalhar desse jeito, mas estamos procurando propriedades sintáticas.

Alguns exemplos de constantes tipadas que vão ser usadas no texto:

$0^{\mathtt{n}}, 1^{\mathtt{n}}, 2^{\mathtt{n}}, \ldots,$

$\ldots, -1^{\mathtt{r}}, 0^{\mathtt{r}}, 1^{\mathtt{r}}, 2^{\mathtt{r}}, \ldots,$

$\pi^{\mathtt{r}}, \sqrt{2}^{\mathtt{r}}, \ldots,$

$(\sqrt{\ })^{\mathtt{n}\to\mathtt{r}}, (\sqrt[3]{\ })^{\mathtt{r}\to\mathtt{r}}, \ldots$

$(+)^{\mathtt{n}\to(\mathtt{n}\to\mathtt{n})}, (+)^{\mathtt{r}\to(\mathtt{r}\to\mathtt{r})}, (-)^{\mathtt{z}\to(\mathtt{z}\to\mathtt{z})}, \ldots$

Os exemplos das três primeiras linhas são óbvios. Na quarta linha, $(\sqrt{\ })^{\mathtt{n}\to\mathtt{r}}$ é a função que leva cada natural na sua raiz quadrada (real), e $(\sqrt[3]{\ })^{\mathtt{r}\to\mathtt{r}}$ leva cada real na sua raiz cúbica. Na quinta linha o primeiro $(+)$ é a soma habitual, mas ao invés de a considermos como uma função de $\mathbb{N}\times\mathbb{N}$ em $\mathbb{N}$ nós a consideramos como uma função que quando recebe um natural retorna uma *função*, e essa outra função quando receber um outro natural retorna o resultado da soma dos dois. Pra tornar mais claro: $(+)(2)$ é uma função (de tipo $\mathtt{n} \to \mathtt{n}$), e $((+)(2))(3) = 5$. O $(-)$ é a mesma coisa, mas com a subtração, e sobre os inteiros; $((-)(6))(2) = 4$. Repare que nós estamos omitindo a anotação de tipo sempre que ela é clara a partir do contexto. Vamos fazer isso sempre que possível.

Regras de formação de tipos

Vamos começar com o caso que nos interessa mais e depois formalizar o caso geral. Temos três regras binárias de formação de tipos, $\to$, $\wedge$ e $\vee$, e uma 0-ária, $\bot$; definimos $\mathcal{A}^{\to,\wedge,\vee,\bot}$ como sendo o conjunto dos tipos gerados por $\mathcal{A}$ e por essas regras, ou seja, o menor conjunto de tipos que contém os tipos de $\mathcal{A}$, é fechado por essas regras (i.e., $\bot$ é um tipo de $\mathcal{A}^{\to,\wedge,\vee,\bot}$, e se α e β são tipos de $\mathcal{A}^{\to,\wedge,\vee,\bot}$ então $\alpha \to \beta$, $\alpha \wedge \beta$ e $\alpha \vee \beta$ são tipos de $\mathcal{A}^{\to,\wedge,\vee,\bot}$) e que é *livre*, isto é, não obedece nenhuma igualdade desnecessária; ou seja, cada tipo de $\mathcal{A}^{\to,\wedge,\vee,\bot}$ ou é atômico ou pode ser expresso como $R(\alpha_1, \ldots, \alpha_n)$ de exatamente uma forma, onde R é uma regra n-ária e α_1, ..., α_n são tipos de $\mathcal{A}^{\to,\wedge,\vee,\perp}$. Além disso cada tipo pode ser expresso de exatamente um modo em termos só de regras de formação e de tipos atômicos, e essa expressão (que dizemos que é o *nome por extenso* daquele tipo) tem comprimento finito. Por exemplo, o que chamamos de $a \to b \wedge \perp$ é na verdade uma notação mais curta para $\to(a, \wedge(b, \perp()))$ (que é o modo "oficial" de escrever $a \to b \wedge \perp$ em termos de regras de formação e tipos atômicos).

Para podermos usar essa notação mais curta temos que estabelecer qual é a precedência dos conectivos $\to$, $\vee$ e $\wedge$, e vamos convencionar que ela é a seguinte: o $\wedge$ "gruda mais rápido" nos seus vizinhos que o $\vee$, e o $\vee$ gruda mais rápido que o $\to$. Por exemplo, $a \to b \wedge c \vee d$ é pra ser interpretado como $a \to (b \wedge c) \vee d$, que é pra ser interpretado como $a \to ((b \wedge c) \vee d)$.

Ainda não definimos quem é o espaço associado a um tipo não atômico. Queremos que cada tipo de $\mathcal{A}^{\to,\wedge,\vee,\perp}$ esteja associado a um espaço, como em $\mathcal{A}$, e pra isso vamos estipular o seguinte: para α e β tipos de $\mathcal{A}^{\to,\wedge,\vee,\perp}$, com os espaços associados $\mathbf{E}_\alpha$ e $\mathbf{E}_\beta$,

$\mathbf{E}_{\alpha\to\beta}$ é o conjunto de *todas* as funções indo do conjunto $\mathbf{E}_\alpha$ no conjunto $\mathbf{E}_\beta$.

$\mathbf{E}_{\alpha\wedge\beta}$ é $\mathbf{E}_\alpha \times \mathbf{E}_\beta$, i.e., $\mathbf{E}_{\alpha\wedge\beta}$ é o conjunto dos pares ordenados em que o primeiro membro é um ponto de $\mathbf{E}_\alpha$ e o segundo é um ponto de $\mathbf{E}_\beta$.

$\mathbf{E}_{\alpha\vee\beta}$ é a união disjunta $\mathbf{E}_\alpha \amalg \mathbf{E}_\beta$, onde para nós a operação $\amalg$ vai ter um significado muito preciso: temos dois símbolos especiais, fixos, $\mathcal{L}$ e $\mathcal{R}$ (de "left" e "right"), e $A \amalg B$ vai ser definida como sendo $\{\, (\mathcal{L}, a) \mid a \in A \,\} \cup \{\, (\mathcal{R}, b) \mid b \in A \,\}$; ou seja, cada ponto de $A \amalg B$ indica claramente se ele veio do conjunto à esquerda do $\amalg$ ou do conjunto à direita.

Além disso, $\mathbf{E}_\perp$ é definido como sendo o conjunto vazio.

Note que não estamos dizendo como interpretar $a \wedge b \wedge c$, $a \vee b \vee c$ ou $a \to b \to c$. Pelas definições que estamos usando — e que não vão ser estendidas para cuidar desses casos — os conectivos binários não são nem comutativos nem associativos, e isso até se reflete nos espaços associados: um ponto de $\mathbf{E}_{a\wedge(b\wedge c)}$ é da forma $(a, (b, c))$, enquanto um ponto de $\mathbf{E}_{(a\wedge b)\wedge c}$ é da forma $((a, b), c)$. Nos capítulos 2 e 3 vamos ver que, por exemplo, $a \wedge b \to b \wedge a$ não é um axioma do nosso sistema lógico, e sim um *teorema*; relendo a demonstração desse teorema usando a interpretação funcional encontramos exatamente a função indo de $\mathbf{E}_{a\wedge b}$ para

$\mathbf{E}_{b\wedge a}$ que recebe um par ordenado e retorna o par com os membros do par original na ordem trocada.

Agora o caso geral. Digamos que $R_1, \ldots, R_n$ sejam regras de formação de tipos, onde cada R_i é uma regra k_i-ária e está associada a um procedimento que gera o espaço $\mathbf{E}_{R_i(\alpha_1,\ldots,\alpha_{k_i})}$ a partir de $\mathbf{E}_{\alpha_1}, \ldots, \mathbf{E}_{\alpha_{k_i}}$. Então definimos $\mathcal{A}^{R_1,\ldots,R_n}$ de modo correspondente a como definimos $\mathcal{A}^{\to,\wedge,\vee,\bot}$.

Quando for claro a partir do contexto quais são as regras de formação de tipos nós vamos escrever $\mathcal{A}^*$ ao invés de $\mathcal{A}^{R_1,\ldots,R_n}$.

Em geral não vamos nem dizer explicitamente nem quem é o $\mathcal{A}$ nem quais são as regras de formação de tipos, mas as regras em princípio vão sempre incluir as regras $\to$, $\wedge$, $\vee$ e $\bot$ já mencionadas e mais uma, $\top$, 0-ária, com $\mathbf{E}_\top$ definido como sendo um conjunto com um ponto só; esse ponto vai ser chamado de $\top$ ou $\top^\top$. Um tipo formado por uma regra que nunca foi mencionada não vai se comportar de forma muito diferente de um tipo atômico que tenha um nome estranho, e por isso podemos imaginar que $\mathcal{A}^*$ já vem com todas as regras de formação de tipos que vão nos interessar. Nós só vamos estar interessados em umas poucas regras específicas, não em regras arbitrárias, e não vamos nunca precisar de definições como "$\mathcal{A}^*$ é fechado pela aplicação de todas as regras de formação de tipos do conjunto tal, que tem enumeráveis regras de cada aridade finita".

Vamos passar a nos referir a $\mathcal{A}^*$ como $\mathbf{Set}_{\mathcal{A}}$. A notação "$\mathbf{Set}_{\mathcal{A}}$" é para enfatizar que estamos numa "categoria de conjuntos tipados", mas isso só vai importar a partir do capítulo 4.

1.4 Árvores

A definição tradicional diz que uma árvore é um grafo acíclico conexo, mas não é essa que queremos usar. Para nós o diagrama com barras horizontais na seção 1.1 é o que queremos chamar de uma "árvore de dedução", e o diagrama abaixo (que aparece em duas versões, uma com as regras explícitas e outra com as regras omitidas) é a "árvore de formação" para o tipo $a \to b \wedge \bot$.

$$\dfrac{a \quad \dfrac{b \quad \dfrac{}{\bot}\,\bot}{b\wedge\bot}\,\wedge}{a\to(b\wedge\bot)}\,\to \qquad\qquad \dfrac{a \quad \dfrac{b \quad \dfrac{}{\bot}}{b\wedge\bot}}{a\to(b\wedge\bot)}$$

Como essa noção de árvore costuma causar uma certa confusão nós vamos dar uma definição formal para ela.

Definicão: uma *lista* é um par $(n, (a_1, \dots, a_n))$, onde $n \in \mathbb{N}$ e $(a_1, \dots, a_n)$ é uma n-upla. Usamos a notação $[a_1, \dots, a_n]$ para denotar uma lista (no caso com n elementos).

Uma *árvore* é uma tripla em que o primeiro membro é o conjunto (finito) das *posições*, ou *nós*, da árvore, o segundo é uma função que associa a cada posição o seu conteúdo, e o terceiro é um conjunto em que cada elemento é uma descrição de uma das barras da árvore; cada descrição dessas é ou um par da forma (*nó abaixo da barra*, *lista dos nós acima da barra*) ou uma tripla da forma (*nó abaixo*, *lista dos nós acima*, *anotação*); essa anotação costuma ser escrita à direita da barra na representação gráfica, e em geral é usada para indicar o nome da regra de dedução ou de formação a que a barra corresponde. Nós vamos impôr as seguintes condições sobre o conjunto das descrições de barras:

1) Todo nó da árvore está acima de exatamente uma barra, exceto por um nó que não está acima de barra nenhuma, e que é chamado de *raiz* da árvore.

2) A árvore não tem ciclos, i.e., não há uma seqüência $a_1, a_2, \dots, a_n$ $(n \geq 2)$ de nós tais que cada nó a_i esteja acima do nó a_{i+1} e o nó a_n esteja acima do a_1.

3) Nenhuma barra tem nós repetidos acima dela.

Uma conseqüência bem conhecida dessas condições é que para cada nó de uma árvore vai existir exatamente um caminho desse nó para a raiz e descendo uma barra em cada passo.

Os nós que não têm ninguém acima deles são chamados de *folhas* da árvore.

Um exemplo:

$$\dfrac{\dfrac{a \quad \dfrac{a}{b}\,R}{c} \qquad b \quad \dfrac{}{a}\,R'}{c} \qquad\qquad \dfrac{\dfrac{n_2 \quad \dfrac{n_1}{n_3}}{n_4} \qquad n_5 \quad \dfrac{}{n_6}}{n_7}$$

a árvore à esquerda na figura acima tem a seguinte representação formal, se os nós são nomeados como na àrvore à direita, acima:

$(\{n_1, n_2, n_3, n_4, n_5, n_6, n_7\},$
$\{(n_1, a), (n_2, a), (n_3, b), (n_4, c), (n_5, b), (n_6, a), (n_7, c)\},$
$\{(n_3, [n_1], R),$
$(n_4, [n_2, n_3]),$

$(n_6, [\,], R'),$
$(n_7, [n_4, n_5, n_6])\}$.

Vamos definir a *altura* de um nó de uma árvore indutivamente da seguinte forma: se ele é uma folha ou se ele vem logo abaixo de uma barra que não tem ninguém acima, a altura dele é 1; senão a altura dele é o máximo da altura dos nós acima dele, mais 1. A altura de uma árvore é considerada como sendo a altura da sua raiz. A árvore da figura acima tem altura 4.

A altura de um tipo é definida como sendo a altura da árvore de formação daquele tipo, e também vai ser conveniente definir indutivamente o *comprimento* de um tipo, do seguinte modo: tipos atômicos e tipos formados pela aplicação de uma regra 0-ária têm comprimento 1; um tipo da forma $R(\alpha_1, \ldots, \alpha_n)$, onde $n \geq 1$, tem comprimento $\mathrm{compr}(\alpha_1) + \ldots + \mathrm{compr}(\alpha_n) + 1$. Em termos de árvores o comprimento de um tipo é o número de nós na árvore de formação daquele tipo. Por exemplo, $a \wedge (a \to \bot)$ tem comprimento 5. Repare que os parênteses não são contados.

Observações:

1) Mais tarde o conceito de árvore vai sofrer algumas extensões simples: os nós vão poder ter certos tipos de anotações e algumas barras vão poder ser barras duplas. Nós não vamos definir como modificar a representação formal para acomodar essas modificações; aliás, nós nunca mais vamos descer a tantos detalhes.

2) Nesta seção nós usamos "acima" e "abaixo" no sentido de "imediatamente acima" e "imediatamente abaixo".

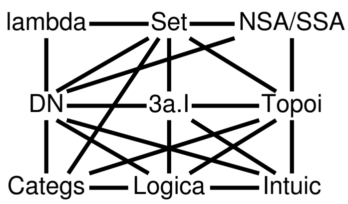
3) A representação formal de uma árvore não vai mais nos importar, e duas árvores que são iguais módulo mudança de nomes de nós vão ser consideradas iguais. Se duas árvores só diferirem no conteúdo dos seus nós e nas anotações nas barras nós vamos dizer que elas têm "a mesma estrutura", ou que são "paralelas".

4) O significado de uma barra vai ser sempre que o objeto de baixo é obtido a partir dos de cima; vamos dizer que o objeto abaixo da barra vem "depois" dos objetos de cima.

Um mapa dos assuntos

Nós vamos passar pelos nove assuntos do diagrama abaixo à esquerda. As linhas indicam relações entre eles, e cada uma das relações vai ser explicada daqui a pouco. Considere que os nove assuntos estão numerados como os números de um telefone de teclas, i.e., como na matriz abaixo à direita; um parágrafo começado com, por exemplo, (45), vai estar dizendo a relação entre os assuntos 4 e 5 — DN (dedução natural) e "3a.I" (a "terceira interpretação").

A ordem dos que assuntos que aparecem explicitamente (e têm capítulos dedicados a eles) é λ-cálculo — dedução natural — categorias — análise não-standard e análise semi-standard — topoi; ou seja, 1 4 7 3 6. **Set** (na verdade $\mathbf{Set}_{\mathcal{A}}$) aparece como o modelo mais natural para λ-cálculo tipado; a interpretação lógica é a interpretação mais natural (tradicionalmente) para dedução natural; a terceira interpretação para dedução natural junta as duas outras interpretações, a funcional ($\mathbf{Set}_{\mathcal{A}}$) e a lógica; NSA e SSA são "versões não convencionais" para **Set**; intuicionismo é uma "versão não convencional" para lógica, em que as deduções do sistema DN ainda valem; topoi são versões não-convencionais de teoria dos conjuntos em que a lógica é pelo menos intuicionista.



$$\begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

(12) **Set** é um modelo para λ-cálculo tipado;

(14) árvores em DN podem ser interpretadas como λ-termos tipados e vice-versa (o chamado "isomorfismo de Curry-Howard");

(23) NSA é um "modelo não-standard" para **Set**;

(24) uma árvore em DN pode ser interpretada como uma construção de um ponto a partir de outros em **Set**;

(25), (45), (58): a "terceira interpretação", descrita em **Set** no capítulo 3, é um modo de juntar as duas interpretações mais naturais de DN, a funcional e a lógica, numa só;

(26) **Set** é um topos;

(27) **Set** inspira todas as nossas construções em categorias (e a notação para elas), e **Set** é uma categoria cartesiana (bi)fechada;

(34), (36) os modelos $\mathbf{Set}^{\mathcal{U}}$ para NSA e $\mathbf{Set}^{\mathcal{F}}$ para SSA são topoi bastante bem comportados: os $\mathbf{Set}^{\mathcal{F}}$ são booleanos e subextensionais e os $\mathbf{Set}^{\mathcal{U}}$ são, além disso, topoi com lógica 2-valuada. No capítulo de notas indicamos como usar uma espécie de dedução natural para fazer cálculo diferencial com infinitesimais, e a tradução de demonstrações feitas dessa forma para demonstrações clássicas passa por uma interpretação em topoi, usando subpontos para "desfazer" o quociente $\mathbf{Set}^{\mathcal{F}} \to \mathbf{Set}^{\mathcal{U}}$.

(47) o capítulo sobre o teorema da dedução mostra como transformar árvores do sistema DN para o sistema CCC (um sistema dedutivo em que as deduções correspondem às construções em categorias cartesinas fechadas) e vice-versa;

(48), (49) a primeira interpretação natural para DN é a em que os tipos correspondem a sentenças lógicas, e a lógica intuicionista é a que considera como teoremas as sentenças dedutíveis em DN;

(56), (59), (69) a terceira interpretação é a que motiva topoi, e usando ela é que mostramos como interpretar sentenças de primeira ordem num topos e mostramos que a lógica de qualquer topos é pelo menos intuicionista;

(67) a definição de topos é categórica: um topos é uma categoria que tem todos os limites finitos, tem todos os colimites finitos, tem exponenciais e tem um objeto classificador.

(78) a definição mais fácil (mas não a mais a clara) para álgebras de valores de verdade é categórica; veja a seção sobre álgebras de Heyting no capítulo 6.

(89) a interpretação lógica aceita modelos (i.e., álgebras de valores de verdade) que não são nem sequer booleanos, são só intuicionistas. Veja a seção 6.4.

Chapter 2

Um pouco de λ-cálculo

Quando o λ-cálculo começou a ser desenvolvido, nas décadas de 1920 e 1930, ainda havia dois conceitos de função: em um, que hoje em dia se tornou dominante, uma função é um conjunto de pares ordenados, e no outro, que foi praticamente esmagado, uma função é alguma espécie de *operação* sobre os seus argumentos que produz um resultado. O segundo é o que vai nos interessar mais, e o λ-cálculo é uma tentativa, bastante bem-sucedida, de formalizá-lo.

Em λ-cálculo a operação que recebe uma função f e um valor x e retorna $f(f(x))$ pode ser expressa como um *termo* como $\lambda f.\lambda x.f(f\,x)$; esse termo está numa forma que é completamente independente do domínio e do contradomínio de f e do conjunto onde o x mora. Isso parece ótimo, parece que nos dá generalidade à beça, mas também traz algumas complicações: é possível expressar como termos algumas operações que são tremendamente difíceis de interpretar. Por exemplo, $\lambda x.xx$ representa a operação que toma um x (uma função?) e retorna o resultado de aplicá-lo a si mesmo. Nenhuma função, daquelas que são consideradas como conjuntos de pares ordenados, pode ser aplicada a si mesma[1]; se vamos tentar interpretar termos como funções no sentido de conjuntos de pares é possível que tenhamos ou que considerá-los como funções parciais ou que restringir cuidadosamente os seus domínios...

Há uma outra alternativa, que vai funcionar magicamente bem. Se nos restringirmos a termos *tipados* não só os termos vão ser fáceis de interpretar (em $\mathbf{Set}_{\mathcal{A}}$) como também vamos ter um algoritmo que diz, para quaisquer dois

[1]por outro lado a operação "identidade", que recebe um argumento e o retorna, pode ser aplicada a si mesma, e o resultado é a própria identidade...

termos dados, se eles são equivalentes ou não. O λ-cálculo restrito a termos tipados ("λ-cálculo tipado") é sob muitos aspectos muito mais bem-comportado que o λ-cálculo não-tipado, e vamos nos concentrar na versão tipada, com umas poucas escapadelas para comentar como as coisas funcionam diferente na versão não-tipada.

Agora a idéia mais importante. Os termos são entidadas sintáticas *finitas* descrevendo operações, e as transformações que levam termos em outros termos equivalentes (e portanto descrições de operações em descrições de outras operações equivalentes) são transformações sintáticas finitas entre expressões finitas; uma *demonstração* de que duas operações são equivalentes pode ser dada por uma seqüência dessas transformações (explicitamente!), da mesma forma que uma *demonstração* de que uma sentença é verdadeira em ZFC pode ser dada por uma seqüência finita de outras sentenças de ZFC que constitua uma "prova" da sentença original. Cada sentença da seqüência é finita e os critérios que dizem se uma sentença é conseqüência lógica imediata das anteriores podem ser expressos de forma puramente sintática; é uma situação bastante parecida com a dos termos equivalentes, só que ZFC é muito mais complicado que λ-cálculo...

O que vamos fazer é usar o λ-cálculo como a fundação para uma outra teoria dos conjuntos, da qual ZFC é um modelo. Essa outra teoria é baseada em uma outra lógica, o intuicionismo, que é mais fraca que a lógica clássica, e o melhor modo de introduzir o intuicionismo é através de uma certa conexão entre ele e o λ-cálculo: o isomorfismo de Curry-Howard, que diz como interpretar termos de λ-cálculo tipado como demonstrações em cálculo proposicional intuicionista e vice-versa. Até o fim do capítulo 7 vamos só entender como ver o nosso conhecido $\mathbf{Set}_{\mathcal{A}}$ sob a ótica dessas novas fundações; depois na seção (6.4) vamos ver rapidamente uma álgebra de valores de verdade que é "estritamente intuicionista" (ou, mais corretamente, "não-booleana"), no sentido de não obedecer $\neg\neg a \to a$. No capítulo 8 surgem modelos para essas fundações nos quais podemos interpretar infinitesimais; esses modelos são bastante simples e podem ser entendidos sem precisar de muito material dos capítulos anteriores. Nos capítulos de notas vamos introduzir os topoi, que são modelos muito mais gerais para as fundações novas; $\mathbf{Set}_{\mathcal{A}}$ e os modelos com infinitesimais, $\mathbf{Set}^{\mathbb{I}}$, $\mathbf{Set}^{\mathcal{F}}$ e $\mathbf{Set}^{\mathcal{U}}$ são topoi, e há outros topoi mais complicados, por exemplo os

$\mathbf{Set}^{\mathbf{P}}$, em que a lógica é não-booleana.

O nosso primeiro modelo para λ-cálculo tipado se chama $\Lambda_{\mathcal{A}}$ e é essencialmente $\mathbf{Set}_{\mathcal{A}}$ com um pouco de estrutura a mais. Na verdade não temos um só $\Lambda_{\mathcal{A}}$ e sim vários: um é o modelo *canônico*, em que as constantes de cada tipo vão corresponder exatamente aos pontos daquele tipo em $\mathbf{Set}_{\mathcal{A}}$, e em que o conjunto das constantes é disjunto do conjunto das variáveis; os outros são modelos *não-canônicos*, e neles cada constante de tipo α vai estar associada a um ponto do $\mathbf{E}_\alpha$, mas a função que leva as constantes nos pontos associados a elas não precisa ser nem injetiva nem sobrejetiva; além disso um modelo não-canônico pode ter símbolos que representam ao mesmo tempo constantes e variáveis.

Um modelo $\Lambda_{\mathcal{A}}$ é formado por um *conjunto de termos*, uma *função de interpretação* (que na verdade é uma função parcial), e uma relação de equivalência entre os termos, chamada de *λβη-equivalência*, que é induzida por uma relação de *redução* entre termos. A interpretação vai ser definida primeiro nos termos atômicos, depois nos termos fechados (que vamos definir na seção 2.3) e depois nos outros termos. Quanto aos termos, eles vão ser formados a partir dos termos atômicos (que são as constantes e as variáveis) por duas operações, a *aplicação*, que vai corresponder à aplicação de uma função a um valor em $\mathbf{Set}_{\mathcal{A}}$ (por exemplo, $(\sqrt{\ })^{\mathrm{n}\to\mathrm{r}}4^{\mathrm{n}}$ vai ser equivalente a 2^{r} no modelo), e a *abstração*, que é uma espécie de inversa da aplicação: não só $\lambda x.(fx)$ vai ser equivalente a f, como se φ é uma expressão (i.e., um termo) então $(\lambda x.\varphi)a$ vai ser equivalente à expressão obtida substituindo-se cada ocorrência de x em φ por a; por exemplo, $(\lambda x.((+)(x))(\sqrt{\ }(x)))(4)$, $((+)(4))(\sqrt{\ }(4))$, $((+)(4))(2)$ e 6 são termos equivalentes no modelo. Estamos enfatizando que algumas dessas equivalências são "no modelo" porque temos também a outra relação de equivalência, a $\lambda\beta\eta$-equivalência, que é mais fraca que a equivalência no modelo e é puramente sintática, i.e., não depende dos valores atribuídos às constantes.

2.1 Termos

O conjunto de termos é gerado a partir do conjunto de termos atômicos e das duas regras de formação de um modo bem parecido com como geramos os tipos de $\mathcal{A}^*$ a partir dos tipos atômicos, mas os termos de $\Lambda_{\mathcal{A}}$ vão ser todos tipados: cada termo de $\Lambda_{\mathcal{A}}$ tem exatamente um *tipo*, que é um tipo de $\mathcal{A}^*$; vamos definir

T_α como sendo o conjunto dos termos de tipo α, e o conjunto dos termos de $\Lambda_{\mathcal{A}}$ vai ser $\bigcup_{\alpha \in \mathcal{A}^*} T_\alpha$.

Termos atômicos. Para cada tipo α vamos ter um conjunto V_α infinito enumerável de *variáveis de tipo* α, e um conjunto C_α, possivelmente vazio, e não necessariamente disjunto de V_α, de *constantes de tipo* α. A união desses dois conjuntos, $A_\alpha := C_\alpha \cup V_\alpha$, é o *conjunto dos termos atômicos de tipo* α. Durante este capítulo vamos chamar termos atômicos de *símbolos*. Símbolos que são só variáveis são chamados de *variáveis puras*, e símbolos que são só constantes são chamados de *constantes puras*.

Toda vez que um símbolo aparecer com um superscrito, como em $f^{\alpha \to \beta}$, vamos estar querendo dizer que ele é do tipo indicado pelo superscrito. Esse superscrito é chamado de *anotação de tipo*, e muitas vezes vai ser omitido. Vamos usar anotações de tipo também em termos.

Regras de formação de termos. As regras de formação são as seguintes: se α^α é um termo de tipo α e $\varphi^{\alpha \to \beta}$ é um termo de tipo $\alpha \to \beta$, então $\varphi^{\alpha \to \beta}\alpha^\alpha$ é um termo de tipo β; e se ψ^β é um termo de tipo β e x^α é uma variável (não um termo!) de tipo α, então $\lambda x^\alpha.\psi^\beta$ é um termo de tipo $\alpha \to \beta$.

A regra que vamos usar para omitir parênteses é que a aplicação sempre começa a partir da esquerda e não precisa de parênteses quando só há dois termos envolvidos, i.e., $a\,b\,c\,d$ quer dizer $((a\,b)c)d$ e escrevemos fx ao invés de $f(x)$; além disso a abstração começa a partir da direita e "acontece depois das aplicações": $\lambda x.\lambda y.\lambda z.abcd$ é $\lambda x.(\lambda y.(\lambda z.((ab)c)d))$. Para evitar maiores complicações, quando um termo obtido por abstração aparecer à direita numa aplicação ele sempre vai aparecer entre parênteses; por exemplo, escrevemos $a(\lambda x.w)$ ao invés de $a\,\lambda x.w$.

Vamos definir a árvore de formação de um termo da mesma forma que definimos a árvore de formação de um tipo, e, como fizemos no caso das árvores de formação de tipos, vamos pular a definição formal e dar só um exemplo. Se x tem tipo a, f tem tipo $a \to b$ e g tem tipo $b \to c$, a árvore de formação de $\lambda x.g(f(x))$ é a da esquerda na figura abaixo; nós vamos sempre pôr os tipos numa árvore separada (no caso, a da direita) ao invés de grudar as anotações de tipos nos termos (poderíamos ter escrito, por exemplo, x^a, $f^{a\to b}$, $g^{b \to c}$, $(\lambda x.g(f(x)))^{a \to c}$). Ao longo do texto as árvores de tipos vão se tornar cada vez mais importantes que as de termos, e é por isso que escolhemos pôr os parâmetros à esquerda das funções nas barras de aplicação: porque a árvore de tipos correspondente fica com uma estrutura melhor.

$$\dfrac{\dfrac{\dfrac{x \quad f}{f(x)} \quad g}{g(f(x))}}{\lambda x.g(f(x))} \qquad\qquad \dfrac{\dfrac{\dfrac{a \quad a \to b}{b} \quad b \to c}{c}}{a \to c}$$

Os termos que aparecem nos nós da árvore de formação de um termo φ — e que portanto correspondem a trechos de φ — são chamados de *subtermos* de φ. Um subtermo α de φ junto com o nó em que ele aparece (e portanto o trecho a que ele corresponde) é chamado de uma *ocorrência* de α em φ. Cada ocorrência de um símbolo x em φ está ou *presa* (e se estiver presa está presa a um certo λx) ou está *livre*, e o processo para decidir isso é o seguinte: comece no nó em que esse x ocorre, e vá descendo uma barra por vez. Se você atravessar uma barra de abstração cuja variável no λ seja exatamente o x, pare; aquela ocorrência do x está presa a esse λ. Se isso não acontecer até você chegar na raiz, então aquela ocorrência do x está livre. Um exemplo: em $(\lambda x.fx(\lambda x.x)x)x$, a primeira e a terceira ocorrências do x estão presas ao primeiro λ; a segunda está presa ao segundo λ e a quarta (e última, já que os $x$zes nos λx não são considerados ocorrências do x!) está livre.

É evidente que constantes puras não podem estar presas.

2.2 Redução

Termos limpos. Dizemos que φ é um *termo limpo* se cada variável x abstraída em φ só é abstraída uma vez, e se todas as ocorrências de x em φ estão presas. Por exemplo, nem $(\lambda x.fx(\lambda x.x)x)x$ nem $f(\lambda x.a)(\lambda x.b)$ e nem $f(\lambda x.a)x$ são limpos, mas $(\lambda y.fy(\lambda z.z)y)x$, $f(\lambda x.a)(\lambda y.b)$ e $f(\lambda y.a)x$ são.

α-conversão. Se φ e ψ são dois termos, dizemos que φ *se α-converte a ψ em um passo* se φ e ψ têm a mesma árvore de tipos e a seguinte coisa acontece: existem variáveis x e y tais que λx ocorre em φ e nem y nem λy ocorrem em φ, e ψ é obtido a partir de ψ substituindo uma ocorrência de λx por λy e cada ocorrência de x que estava presa por aquele λx por y. Por exemplo: $(\lambda x.fx(\lambda x.x)x)x$ se α-converte a $(\lambda y.fy(\lambda x.x)y)x$ em um passo; $a(\lambda x.b)$ se α-converte a $a(\lambda y.b)$ em um passo.

É fácil ver que qualquer termo pode ser convertido num termo limpo através de uma seqüência de α-conversões; além disso, qualquer subtermo de um termo limpo também é limpo.

Uma relação de redução (ou de conversão) em um passo induz uma relação de redução e uma de equivalência dos jeitos óbvios: um termo se reduz a outro em n passos se existe uma seqüência de n reduções em um passo indo do primeiro termo ao segundo; um termo se reduz a outro se se reduz em n passos, para algum $n \in \mathbb{N}$ (onde para nós $\mathbb{N}$ inclui 0); um termo é equivalente a outro se existe uma seqüência de termos indo de um ao outro em que entre cada dois termos consecutivos da seqüência ou um se reduz ao outro ou o outro se reduz ao um.

β-redução. É muito trabalhoso definir a β-redução em termos não-limpos (consulte [HS], p.7, para os detalhes), mas a definição que só se aplica a termos limpos vai ser suficiente para nós. Se φ é um termo limpo e $(\lambda x.\psi)\chi$ é um subtermo de φ, então dizemos que *φ se β-reduz a φ' em um passo* se φ' é obtido a partir de φ substituindo uma ocorrência de $(\lambda x.\psi)\chi$ por $\psi[x\backslash\chi]$, onde $\psi[x\backslash\chi]$ é simplesmente uma notação formal para o termo obtido substituindo-se cada ocorrência de x em ψ por χ; por exemplo, $((+)y\,y)[y\backslash(\sqrt{}4)] = (+)(\sqrt{}4)(\sqrt{}4)$, e $(\lambda y.(+)y\,y)(\sqrt{}4)$ se β-reduz em um passo a $(+)(\sqrt{}4)(\sqrt{}4)$.

η-redução. Se φ é um termo limpo, $\lambda x.\psi x$ é um subtermo de x e x não ocorre em ψ, dizemos que *φ se η-reduz a φ'* se φ' é o termo obtido substituindo-se uma ocorrência de $\lambda x.\psi x$ em φ por ψ.

$\lambda\beta\eta$-redução e $\lambda\beta\eta$-equivalência. Dizemos que um termo φ se $\lambda\beta\eta$-reduz a ψ em um passo se φ se α-converte a ψ em um passo, ou se β-reduz em um passo, ou se η-reduz em um passo a ψ. A partir da definição de $\lambda\beta\eta$-redução em um passo definimos do jeito natural o que é um termo se $\lambda\beta\eta$-reduzir a outro e o que é dois termos serem $\lambda\beta\eta$-equivalentes. Como essas são as relações que nos interessam mais vamos ter notações especiais para elas: $\varphi \twoheadrightarrow \psi$ se φ se $\lambda\beta\eta$-reduz a ψ, e $\varphi \equiv \psi$ se φ é $\lambda\beta\eta$-equivalente a ψ.

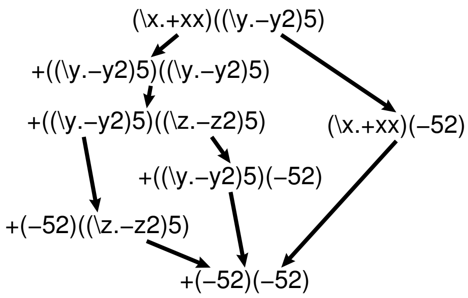
2.3 Termos irredutíveis

Vamos dizer que um termo é *irredutível* se ele é limpo e não aceita nenhum passo de β-redução e nenhum passo de η-redução; ou seja, um termo é irredutível se é limpo e não tem subtermos nem da forma $(\lambda x.\varphi)\psi$ nem da forma $\lambda x.\xi\,x$, onde o x

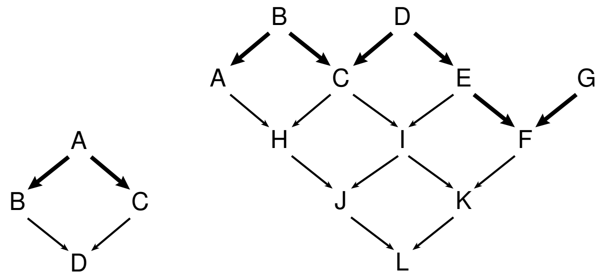
não ocorre em ξ. Por exemplo: 2, x, $(+)((-)5\,2)((-)5\,2)$ e $\lambda f.\lambda g.\lambda h.f(\lambda x.g(hx))$ são irredutíveis.

Será que qualquer termo pode ser $\lambda\beta\eta$-reduzido a um termo irredutível? Será que esse termo irredutível é único (módulo α-conversão)? Para λ-cálculo tipado a resposta para essas duas perguntas é "sim"; vamos dar só uma idéia da prova, e o leitor interessado em mais detalhes pode consultar os capítulos 4 e 6 de [GLT] e o apêndice 2 de [HS]. Em primeiro lugar, vamos definir uma *sequência de reduções* começando num termo φ_0 como uma seqüência de termos $\varphi_0, \varphi_1, \ldots$, finita ou infinita, em que na passagem de cada termo para o seguinte ou damos um passo de β-redução, ou damos um passo de η-redução, ou passamos de um termo não-limpo para um termo limpo via uma α-conversão. É possível ver que para qualquer termo φ_0 sempre existe uma seqüência de reduções finita indo dele para um termo irredutível ([GLT], cap. 4), e que também não existem seqüências infinitas de reduções começando de um termo ([GLT], cap. 6); exceto pela escolha dos nomes das variáveis nos passos de α-conversão há apenas um número finito de termos que podem seguir um certo termo de uma seqüência de reduções, e portanto há, módulo α-equivalência, apenas um número finito de termos acessíveis a partir de um termo φ_0 por seqüências de reduções; idem para seqüências genéricas de $\lambda\beta\eta$-reduções.

Exemplo: $(\lambda.(+)x\,x)(\lambda y.(-)y\,2)5$. Cada seta do diagrama abaixo é um passo de α-conversão ou de β-redução. Note que como $(+)$ e $(-)$ são constantes ainda não temos regras que permitam reduzir $(-)5\,2$ a 3 e $(+)3\,3$ a 6; as regras desse tipo vão vir da interpretação no modelo.



Fato: a $\lambda\beta\eta$-redução obedece o que chamamos de *propriedade do diamante* (ver [HS], apêndice 2): se um termo A se $\lambda\beta\eta$-reduz a B e a C, então existe um termo D tal que $B \twoheadrightarrow D$ e $C \twoheadrightarrow D$; veja a figura abaixo à esquerda. Isso é suficiente para garantir que se dois termos φ e ψ são $\lambda\beta\eta$-equivalentes então existe um termo χ tal que $\varphi \twoheadrightarrow \chi$ e $\psi \twoheadrightarrow \chi$: por exemplo, na figura abaixo à direita temos um caso em que $A \equiv G$ porque $A \twoheadleftarrow B \twoheadrightarrow C \twoheadleftarrow D \twoheadrightarrow E \twoheadrightarrow F \twoheadleftarrow G$, e para conseguirmos um termo L tal que $A \twoheadrightarrow L$ e $G \twoheadrightarrow L$ nós “completamos os diamantes” do diagrama obtendo os termos intermediários H, I, J e K, e, por fim, L.



Daí é imediato ver que para cada termo φ existe exatamente um termo irredutível (módulo α-equivalência), ψ, ao qual φ se reduz: se $\psi \twoheadrightarrow \psi_1$ e $\psi \twoheadrightarrow \psi_2$, então existe ψ_3 com $\psi_1 \twoheadrightarrow \psi_3$ e $\psi_2 \twoheadrightarrow \psi_3$, e essas duas últimas reduções são α-conversões; daí ψ_1 e φ_2 são α-equivalentes.

Com isso já temos um algoritmo que diz se dois termos são $\lambda\beta\eta$-equivalentes: encontre um termo irredutível associado a cada um deles, por exemplo montando seqüências de reduções começando em cada um deles e estendendo-as até chegar a termos irredutíveis; se os dois termos irredutíveis que obtivemos forem α-equivalentes então $\varphi \equiv \psi$, senão não. Uma conseqüência trivial disso é que o sistema é *consistente*: se φ e ψ são montados só com constantes e aplicações, sem nenhuma abstração, então eles já são irredutíveis e são $\lambda\beta\eta$-equivalentes se e só se forem iguais como termos; em particular, se c_1 e c_2 são duas constantes temos $c_1 \equiv c_2$ se e só se tivermos $c_1 = c_2$.

Se $\varphi \twoheadrightarrow \psi$ e ψ é irredutível então dizemos que ψ é a *forma normal* de φ; a forma normal de um termo é única módulo α-equivalência. O processo de encontrar a forma normal é chamado de *normalização*.

Observação: os resultados correspondentes em λ-cálculo não-tipado são bem mais fracos. é possível provar um lema do diamante para λ-cálculo não-tipado e com isso a forma normal, se existir, vai ser única módulo α-equivalência, mas às vezes ela não existe: por exemplo, $(\lambda x.xx)(\lambda y.yy) \twoheadrightarrow (\lambda y.yy)(\lambda y.yy)$ por β-redução. Também podemos ter termos φ e ψ não α-equivalentes tais que

$\varphi \twoheadrightarrow \psi$ e $\psi \twoheadrightarrow \varphi$ e termos com seqüências de reduções em que aparecem termos de comprimento arbitrariamente grande.

Já estamos quase em condições de entender a função de interpretação. Precisamos de duas definições auxiliares:

Termos fechados e combinadores. Se todos os símbolos livres de um termo são constantes dizemos que ele é um termo *fechado*; por exemplo, se f e g são constantes (mesmo que sejam variáveis também) então $\lambda x.g(fx)$ é um termo fechado. Um *combinador* é um termo que não tem símbolos livres, ou seja, em que todos os símbolos são variáveis e estão presos por algum λ. Por exemplo, $\lambda f.\lambda g.\lambda x.fx(gx)$ é um combinador, e $\lambda x.g(fx)$ não. A idéia é que em $\lambda f.\lambda g.\lambda x.fx(gx)$ o x, o f e o g estão fazendo papel de variáveis, não de constantes; vamos ver isso com detalhes em breve.

2.4 Combinadores ao invés de λ

Digamos que para quaisquer tipos α, β e γ nós tenhamos constantes $\mathbf{I}^{\alpha\to\alpha}$, $\mathbf{K}^{\alpha\to(\beta\to\alpha)}$ e $\mathbf{S}^{(\alpha\to(\beta\to\gamma))\to((\alpha\to\beta)\to(\alpha\to\gamma))}$. Essas constantes vão representar certos combinadores; vamos definir uma operação '$*$' sobre termos que substitui cada ocorrência delas pelo combinador correspondente. Quando passamos de φ para φ^*,

cada	$\mathbf{I}^{\alpha\to\alpha}$	vira	$\lambda x^\alpha.x$,
cada	$\mathbf{K}^{\alpha\to(\beta\to\alpha)}$	vira	$\lambda x^\alpha.\lambda y^\beta.x$ e
cada	$\mathbf{S}^{(\alpha\to(\beta\to\gamma))\to((\alpha\to\beta)\to(\alpha\to\gamma))}$	vira	$\lambda f^{\alpha\to(\beta\to\gamma)}.\lambda g^{\alpha\to\beta}.\lambda x^\alpha.fx(gx)$.

Dizemos que um termo é um *λ-polinômio* quando ele é formado a partir de símbolos atômicos só por aplicação, sem abstrações. Se φ é um λ-polinômio, dizemos que ele é um **SKI**-polinômio sobre $a_1, \ldots, a_n$ quando todos os símbolos atômicos que aparecem nele são '**S**'s, '**K**'s ou '**I**'s ou estão na lista $a_1, \ldots, a_n$.

Vamos esquecer temporariamente a distinção entre as constantes **S**, **K** e **I** e os combinadores que elas representam. Vamos dizer que dois termos φ e ψ são *congruentes* (notação: $\varphi \cong \psi$) quando tivermos $\varphi^* \equiv \psi^*$; até o fim desta seção a relação que vai nos importar é a congruência, não a equivalência, porque queremos entender isto aqui:

Teorema: qualquer termo é congruente a um **SKI**-polinômio sobre os seus símbolos livres.

Vamos descrever um algoritmo que encontra esse **SKI**-polinômio. Primeiro é bom entender um pouco melhor como esses '**S**'s, '**K**'s e '**I**'s funcionam. Repare que eles poderiam ter sido definidos pelas equações abaixo:

$$\begin{array}{rcll} \mathbf{I}x & \cong & x & (\text{“identidade”}), \\ \mathbf{K}xy & \cong & x & (\text{“constante”}), \\ \mathbf{S}fgx & \cong & fx(gx) & (\text{“split”}). \end{array}$$

Vamos direto para o passo mais difícil do algoritmo. Digamos que temos um termo da forma $\lambda x.\varphi$, onde φ é um λ-polinômio. Queremos uma operação '$'$' sobre termos que "expulse" as ocorrências de x em φ: φ vai ser congruente a $\varphi' x$, e aí vamos ter $\lambda x.\varphi \cong \lambda x.\varphi' x \cong \varphi'$. Um exemplo:

$$\begin{aligned} (xa)(bx) &\cong ((xa)(bx))'x \\ &\cong \mathbf{S}(xa)'(bx)'x \\ &\cong \mathbf{S}(\mathbf{S}x'a')(\mathbf{S}b'x')x \\ &\cong \mathbf{S}(\mathbf{SI}(\mathbf{K}a))(\mathbf{S}(\mathbf{K}b)\mathbf{I})x \end{aligned}$$

$$\begin{aligned} \lambda x.(xa)(bx) &\cong ((xa)(bx))' \\ &\cong \mathbf{S}(\mathbf{SI}(\mathbf{K}a))(\mathbf{S}(\mathbf{K}b)\mathbf{I}) \end{aligned}$$

A operação '$'$' pode ser definida de vários modos. Um deles é este: se $\varphi = x$, então $\varphi' = \mathbf{I}$; se φ não contém x, então $\varphi' = \mathbf{K}\varphi$; se φ não cai em nenhum desses dois casos então φ é da forma $\varphi_1\varphi_2$, e aí definimos φ' como $\mathbf{S}\varphi_1'\varphi_2'$.

Pra passar para o caso geral definimos para cada variável v a operação $[\cdot]_v$ que expulsa os vs de termos que sejam da forma $\lambda v.\varphi$, onde φ é um λ-polinômio: $[\lambda x.\varphi]_x$ é definida como φ', e as outras são similares. Note que $[\lambda x.\varphi]_x \cong \lambda x.\varphi$. Usando uma operação dessas em cada abstração de um termo obtemos o **SKI**-polinômio que queríamos. Por exemplo,

$$\begin{aligned} \lambda f.\lambda x.f(fx) &\cong [\lambda f.[\lambda x.f(fx)]_x]_f \\ &= [\lambda f.\mathbf{S}(\mathbf{K}f)(\mathbf{S}(\mathbf{K}f)\mathbf{I})]_f \\ &= \mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I}))(\mathbf{S}(\mathbf{S}(\mathbf{KS})(\mathbf{S}(\mathbf{KK})\mathbf{I})(\mathbf{KI}))\ . \end{aligned}$$

O **SKI**-polinômio obtido por esse algoritmo em geral é horrível, mas repare o seguinte: em $\mathbf{Set}_{\mathcal{A}}$ temos, para quaisquer tipos α, β e γ, pontos de $\mathbf{E}_{\alpha\to\alpha}$, $\mathbf{E}_{\alpha\to(\beta\to\alpha)}$ e $\mathbf{E}_{(\alpha\to(\beta\to\gamma))\to((\alpha\to\beta)\to(\alpha\to\gamma))}$ que são bons candidatos a serem os valores de $\mathbf{I}^{\alpha\to\alpha}$, $\mathbf{K}^{\alpha\to(\beta\to\alpha)}$ e $\mathbf{S}^{(\alpha\to(\beta\to\gamma))\to((\alpha\to\beta)\to(\alpha\to\gamma))}$: para o $\mathbf{I}$, a função que espera um x e retorna o mesmo x (a identidade); para $\mathbf{K}$, a função que recebe um x e retorna a função constante que espera um y e retorna o x que tinha sido dado antes; e, para o $\mathbf{S}$, a função que espera um f, depois um g e depois um x, e retorna o resultado de $fx(gx)$. Se um termo φ é fechado e cada uma das suas constantes livres corresponde a um ponto de $\mathbf{Set}_{\mathcal{A}}$ (com o tipo adequado) então temos um ponto de $\mathbf{Set}_{\mathcal{A}}$ candidato a ser o valor de φ: obtenha, por esse algoritmo, um **SKI**-polinômio congruente a φ; calcule o "valor" dele interpretando cada aplicação entre termos nele como a aplicação natural em $\mathbf{Set}_{\mathcal{A}}$. Não é nada óbvio que esse modo de atribuir valores a termos seja bem-comportado — seria preciso ver, por exemplo, que dois termos equivalentes são associados ao mesmo valor — mas esse argumento já dá uma indicação, ainda que vaga, de que *todo termo fechado deve ser interpretável*. Na próxima seção vamos ver um modo muito melhor de chegar à função de interpretação.

2.5 A interpretação dos termos no modelo

Seja $[\cdot]'$ uma função qualquer que atribua valores com os tipos certos às constantes de $\Lambda_{\mathcal{A}}$, ou seja, uma função com domínio $\bigcup_{\alpha\in\mathcal{A}^*} C_\alpha$ que leve cada constante $c^\alpha \in C_\alpha$ em um ponto $[c^\alpha]' \in \mathbf{E}_\alpha$. Queremos estender essa $[\cdot]'$ a uma função parcial $[\cdot]$ que obedeça as seguintes condições:

Condição sobre os termos atômicos: $[\cdot]$ coincide com $[\cdot]'$ sobre os termos atômicos, i.e., $[c_\alpha] = [c_\alpha]'$ para toda constante c^α e $[v_\alpha]$ não está definida se v^α é uma variável que não é uma constante;

Condição sobre a aplicação: se $[\alpha^\alpha]$ e $[\varphi^{\alpha\to\beta}]$ estão definidas então $[\varphi^{\alpha\to\beta}\alpha^\alpha]$ está definida e vale $[\varphi^{\alpha\to\beta}]([\alpha^\alpha])$;

Condição sobre a abstração: essa fica mais fácil de definir se x^α for ao mesmo tempo uma variável e uma constante e se o conjunto das constantes puras de tipo α contiver uma constante para cada ponto de $\mathbf{E}_\alpha$; vamos supor que cada constante dessas seja representada pelo próprio ponto correspondente de $\mathbf{E}_\alpha$. Nesse caso a condição fica sendo a seguinte: se φ^β é um termo limpo e $[\varphi^\beta]$ está definido, então $[\lambda x^\alpha.\varphi^\beta]$ é a função f de $\mathbf{E}_{\alpha\to\beta}$ que leva cada $\alpha^\alpha \in \mathbf{E}_\alpha$

(que, repare, também é uma constante de C_α) em $[\varphi^\beta[x^\alpha \backslash \alpha^\alpha]]$.

Condição sobre a $\lambda\beta\eta$-equivalência: se φ e ψ forem dois termos $\lambda\beta\eta$-equivalentes então $[\varphi] = [\psi]$, isto é, ou $[\varphi]$ e $[\psi]$ são ambas definidas e têm o mesmo valor ou são ambas indefinidas.

Minimalidade: dentre todas as funções parciais $[\cdot]$ que obedeçam as quatro condições acima vamos tomar a que tenha o menor domínio possível.

O fato é que para qualquer atribuição de valores para as constantes, $[\cdot]$, a extensão $[\cdot]$ existe e é única; além disso os termos interpretáveis por ela (i.e., os em que a função parcial $[\cdot]$ está definida) são exatamente os termos fechados. Não vamos demonstrar nada disso, só dar alguns exemplos esclarecedores. Seja $\varphi = (\lambda f^{\mathtt{n}\to\mathtt{n}}.\lambda x^{\mathtt{n}}.f(f x))(\lambda y^{\mathtt{n}}.(+)\, y\, 2)$. Podemos usar a condição sobre a $\lambda\beta\eta$-equivalência e normalizar φ; com isso $[\varphi] = [\lambda x^{\mathtt{n}}.(+)\,((+)\, x\, 2)\, 2]$. Substituindo $x^{\mathtt{n}}$ por cada natural em $(+)\,((+)\, x\, 2)\, 2$ e calculando $[(+)\,((+)\, 0\, 2)\, 2]$, $[(+)\,((+)\, 1\, 2)\, 2]$, etc (usando a condição sobre a aplicação) vemos que $[\varphi]$ é a função que leva cada natural na soma dele com 4. Outro modo de obter $[\varphi]$ teria sido ver que $[\lambda y^{\mathtt{n}}.(+)\, y\, 2]$ é a função $\mathtt{n} \to \mathtt{n}$ dada por $y \mapsto y+2$, e aplicando essa função duas vezes a cada valor de x em $[\lambda x^{\mathtt{n}}.f(f x)]$ teríamos obtido de novo que $[\varphi]$ é $x \mapsto x+4$. Por outro lado, se tivéssemos tentado interpretar $\varphi' = (\lambda f^{\mathtt{n}\to\mathtt{n}}.\lambda x^{\mathtt{n}}.f(f x))(\lambda y^{\mathtt{n}}.(+)\, y\, v^{\mathtt{n}})$, onde $v^{\mathtt{n}}$ é uma variável pura, teríamos sempre esbarrado em alguma situação em que precisaríamos ter um valor para $[v^{\mathtt{n}}]$ para obter o valor de $[\varphi']$; usando a condição de minimalidade é possível ver que $[\varphi']$ tem que ser indefinida.

2.6 As reduções e a árvore dos tipos

Vamos retomar um exemplo que apareceu na seção 2.1: a árvore de formação do termo $\lambda x.g(f\,x)$ e a sua árvore de tipos.

$$\dfrac{\dfrac{\dfrac{x \quad f}{f(x)} \quad g}{g(f(x))}}{\lambda x.g(f(x))} \qquad\qquad \dfrac{\dfrac{\dfrac{a \quad a \to b}{b} \quad b \to c}{c}}{a \to c}$$

Digamos que x, f e g sejam constantes; o x tem que ser também uma variável, mas o nosso sistema permite que símbolos sejam ao mesmo tempo variáveis e constantes. O valor de $f\,x$ é obtido a partir do valor de x e do de f

por aplicação, e daí, pelo menos a princípio, o valor de $f\,x$ *depende* dos valores de x e de f. Da mesma forma, o valor de $g(f\,x)$ é obtido a partir dos de g e de $f\,x$, e depende desses valores; como o de $f\,x$ é obtido a partir dos valores de f e de x, podemos considerar que o valor de $g(f\,x)$ depende dos de x, f e g, e, aliás, só depende desses valores, já que a operação de aplicação está fixa. Assim, podemos interpretar a árvore de formação de $g(f\,x)$ como descrevendo um procedimento que a partir dos valores de x, f e g obtém o valor de $g(f\,x)$; munidos de valores para x, f e g nós seguimos as instruções dadas pela árvore, obtemos alguns valores intermediários e no final obtemos o valor que queríamos, aliás, o valor que a árvore diz ser o valor de $g(f\,x)$.

As barras que correspondem a aplicações são bem fáceis de interpretar, mas as que correspondem a introduções de 'λ's são bem piores. A barra acima do $\lambda x.f(g\,x)$, por exemplo, diz o seguinte: "...considere a árvore acima do $f(g\,x)$ como um *procedimento* que leva cada valor de x num valor de $f(g\,x)$; vamos supor que os valores de f e de g estejam fixos. Então, por propriedades de $\mathbf{Set}_{\mathcal{A}}$, sabemos que existe *exatamente um* ponto do espaço $\mathbf{E}_{a\to c}$ que corresponde a uma função que se comporta exatamente como esse procedimento; o valor de $\lambda x.f(g\,x)$ vai ser exatamente esse ponto desse espaço de funções." É sem dúvida uma operação estranha, embora conveniente. Pra vermos a que ela corresponde quando passamos para árvore dos tipos, veja primeiro o seguinte: o valor de $f(g\,x)$ depende do de x, mas o de $\lambda x.f(g\,x)$ não, já que em $\lambda x.f(g\,x)$ o x foi transformado em variável[2]. Antes o valor de um nó dependia dos valores de todas as folhas acima desse nó, mas agora algumas dessas folhas podem ser descartadas, e só importam o valor das outras, a estrutura da árvore, e os tipos. Vamos marcar essas folhas cujo valor deixa de importar com colchetes e com um número associado à barra, como na figura abaixo; a idéia é que se estamos abaixo da barra 1 as folhas $[x]^1$ foram "podadas" e não indicam mais dependências, mas acima da barra 1 cada folha $[x]^1$ conta como uma dependência, e a gente ignora os colchetes.

[2]Os $x$zes marcavam lugares onde entrava uma determinada constante, mas agora marcam lugares por onde vão passar *todos os valores possíveis* de x. Compare com o que acontece no capítulo 7, onde falamos sobre os infinitesimais universais.

$$\dfrac{\dfrac{\dfrac{[x]^1 \quad f}{f(x)} \quad g}{g(f(x))}}{\lambda x.g(f(x))}\,1 \qquad\qquad \dfrac{\dfrac{\dfrac{[a]^1 \quad a \to b}{b} \qquad b \to c}{c}}{a \to c}\,1$$

Na árvore dos tipos os colchetes têm uma interpretação muito interessante. Lendo cada tipo como uma proposição verdadeira, a árvore diz que se a é verdade e $a \to b$ é verdade então b é verdade, e se b é verdade e $b \to c$ é verdade então c é verdade, ou seja, a partir de a, de $a \to b$ e $b \to c$ conseguimos deduzir c. Abaixo da barra o a é descartado, e podemos ler o que sobra como "se $a \to b$ é verdade e $b \to c$ é verdade, então $a \to c$ é verdade". Conseguimos uma regra que a partir de uma dedução de $\alpha_1 \wedge \alpha_2 \wedge \ldots \alpha_n \Rightarrow \beta$ nos dá uma dedução de $\alpha_2 \wedge \ldots \alpha_n \Rightarrow (\alpha_1 \to \beta)$ que é só um passo maior que a dedução anterior! É verdade que essa regra soa artificial quando é vista em árvores, mas temos que nos lembrar que estamos tentando carregar o tempo todo a estrutura das demonstrações, não apenas o que foi demonstrado; além disso vamos encontrar uma correspondência entre termos de λ-cálculo e demonstrações, e se identificássemos todas as demonstrações que tivessem as mesmas premissas e a mesma conclusão cairíamos num sistema pobre demais.

Nós vamos estudar esses sistemas em que os tipos são proposições a partir do próximo capítulo; eles são as "interpretações lógicas", em oposição às "interpretações funcionais", mas agora queremos ver duas outras coisas: a primeira, que vamos mencionar só de passagem, é que é possível reconstruir um termo, módulo α-equivalência, a partir da sua árvore de tipos (com a indicação de quais folhas estão associadas a que barras) e dos termos atômicos em folhas não-descarregadas. Por exemplo,

$$\dfrac{\dfrac{\dfrac{[x]^2 \quad \dfrac{[x]^2 \quad (+)}{(+)x}}{(+)x\,x}}{\lambda y.(+)x\,x}\,1}{\lambda x.\lambda y.(+)x\,x}\,2 \qquad \dfrac{\dfrac{\dfrac{[\mathbb{n}]^2 \quad \dfrac{[\mathbb{n}]^2 \quad \mathbb{n} \to (\mathbb{n} \to \mathbb{n})}{\mathbb{n} \to \mathbb{n}}}{\mathbb{n}}}{\mathbb{n} \to \mathbb{n}}}{\mathbb{n} \to \mathbb{n} \to \mathbb{n}} \qquad \dfrac{\dfrac{\dfrac{[\mathbb{n}_2]^2 \quad \dfrac{[\mathbb{n}_2]^2 \quad (+)}{(+)\mathbb{n}_2}}{(+)\mathbb{n}_2\,\mathbb{n}_2}}{\lambda \mathbb{n}_1.(+)\mathbb{n}_2\,\mathbb{n}_2}\,1}{\lambda \mathbb{n}_2.\lambda \mathbb{n}_1.(+)\mathbb{n}_2\,\mathbb{n}_2}\,2 .$$

A segunda coisa é que as noções de redução e equivalência entre termos sugerem noções de redução e equivalência entre deduções. Por exemplo, já que $(\lambda x.f\,x(g\,x))(h\,y) \equiv f(h\,y)(g(h\,y))$ (por uma β-redução), então podemos considerar que as árvores de tipos correspondentes são equivalentes,

$$\dfrac{\dfrac{y\quad h}{hy}\qquad \dfrac{\dfrac{\dfrac{[x]^1\quad g}{gx}\quad \dfrac{[x]^1\quad f}{fx}}{fx(gx)}}{\lambda x.fx(gx)}\,1}{(\lambda x.fx(gx))(hy)} \qquad \dfrac{\dfrac{d\quad d\to a}{a}\qquad \dfrac{\dfrac{\dfrac{[a]^1\quad a\to b}{b}\quad \dfrac{[a]^1\quad a\to(b\to c)}{b\to c}}{c}}{a\to c}\,1}{c}$$

$$\dfrac{\dfrac{\dfrac{y\quad h}{hy}\quad g}{g(hy)}\qquad \dfrac{\dfrac{y\quad h}{hy}\quad f}{f(hy)}}{f(hy)(g(hy))} \qquad \dfrac{\dfrac{\dfrac{d\quad d\to a}{a}\quad a\to b}{b}\qquad \dfrac{\dfrac{d\quad d\to a}{a}\quad a\to(b\to c)}{b\to c}}{c}$$

e vemos que passamos da árvore de tipos de $\varphi = (\lambda x.f\,x(g\,x))(h\,y)$ para a de $\varphi' = f(h\,y)(g(h\,y))$ pegando a árvore sobre o a em φ, grudando uma cópia dela em cada uma das folhas a descarregadas pela barra 1, e juntando o c abaixo do $a \to c$ com o c acima do $a \to c$, descartando tudo o que havia entre os dois. É como se os '$[a]^1$'s fossem buracos que esperam uma demonstração de a; em φ a demonstração de a é dada ao lado do $a \to c$, em φ' ela é plugada diretamente nos buracos.

2.7 Atribuição de tipos a termos não-tipados

Tome um combinador, digamos $(\lambda x^a.\lambda y^b.x^a)^{a\to(b\to a)}$, e apague as anotações de tipo das suas variáveis: dá $\lambda x.\lambda y.x$. Vamos supor que variáveis que eram diferentes com as anotações de tipo continuam diferentes mesmo sem as anotações; na verdade o que é importante mesmo é que as ocorrrências das variáveis continuem presas pelos mesmos 'λ's de antes, o que não acontece por exemplo em $\lambda x^a.\lambda x^b.x^a \mapsto \lambda x.\lambda x.x$.

Agora tente pôr as anotações de tipo de volta, de novo tomando o cuidado de fazer com que as ocorrências das variáveis continuem presas aos mesmos 'λ's. Pode haver vários modos de fazer isso: $\lambda x.\lambda y.x$ pode ser tipado como $(\lambda x^a.\lambda y^b.x^a)^{a\to(b\to a)}$, como $(\lambda x^c.\lambda y^d.x^c)^{c\to(d\to c)}$... de fato, como $(\lambda x^\alpha.\lambda y^\beta.\,x^\alpha)^{\alpha\to(\beta\to\alpha)}$ para qualquer escolha de tipos α e β, e de nenhum outro modo. Nesse caso dizemos que $\alpha \to (\beta \to \alpha)$ é o *esquema principal de tipos* do termo não-tipado $\lambda x.\lambda y.x$; esses α e β fazem o papel de *variáveis de tipo* ([HS], p.173).

Poderíamos ter definido a operação de apagar tipos e depois pô-los de volta também para termos com constantes livres; aí nós não apagaríamos os tipos das constantes, e definiríamos uma noção de esquema principal de tipo que valeria para quaisquer termos tipados. O fato é que todo termo de λ-cálculo tipado tem um esquema principal de tipos, não só os combinadores; mas só os termos que são equivalentes a combinadores têm esquemas principais de tipos em que só aparecem variáveis de tipos.

Em λ-cálculo não-tipado há combinadores que não admitem um esquema principal de tipos e que portanto não são termos tipados com tipos apagados. Por exemplo, tome $\lambda x.xx$; se ele fosse tipável o primeiro x no subtermo $x\,x$ teria que ter um tipo da forma $\alpha \to \beta$; aí o segundo teria que ter tipo α e o tipo do primeiro x teria comprimento estritamente maior que o tipo do segundo x, um absurdo.

Na verdade nós só estamos discutindo esquemas principais de tipos para motivar um pouco uma afirmação que vai ser feita na próxima seção: que certas constantes estranhas que vamos introduzir devem ser consideradas como combinadores, exatamente como os **S**, **K** e **I**. Vamos em frente.

2.8 $\wedge$, $\vee$, $\top$ e $\bot$

No modelo canônico $\Lambda_{\mathcal{A}}$ temos constantes que, pelo menos aparentemente, vão nos permitir lidar com tipos contendo os conectivos $\wedge$, $\vee$, $\top$ e $\bot$. Elas são essas aqui: para todos os tipos α, β e γ,

$\mathbf{P}^{\alpha\to(\beta\to\alpha\wedge\beta)}$, a função formadora de pares;

$\pi_1^{\alpha\wedge\beta\to\alpha}$, a projeção na primeira coordenada de um par;

$\pi_2^{\alpha\wedge\beta\to\beta}$, a projeção na segunda coordenada;

$\kappa_1^{\alpha\to\alpha\vee\beta}$, a inclusão natural de $\mathbf{E}_\alpha$ em $\mathbf{E}_\alpha \amalg \mathbf{E}_\beta$;

$\kappa_2^{\beta\to\alpha\vee\beta}$, a inclusão natural de $\mathbf{E}_\beta$ em $\mathbf{E}_\alpha \amalg \mathbf{E}_\beta$;

$\mathbf{D}^{(\alpha\to\gamma)\to((\beta\to\gamma)\to(\alpha\vee\beta\to\gamma))}$, o operador "definição por casos";

$\top^\top$, o único ponto de $\mathbf{E}_\top$;

$\mathbf{V}^{\bot\to\alpha}$, a única função de $\mathbf{E}_\bot = \varnothing$ em $\mathbf{E}_\alpha$.

Para todas as constantes a^α, b^β, $p^{\alpha\wedge\beta}$, $f^{\alpha\to\gamma}$, $g^{\beta\to\gamma}$, $u^{\alpha\vee\beta}$, $t^\top$ e $v^{\bot\to\alpha}$ as seguintes equações valem no modelo $\Lambda_{\mathcal{A}}$:

$$\begin{aligned}
[\pi_1(\mathbf{P}\,a\,b)] &= [a] \\
[\pi_2(\mathbf{P}\,a\,b)] &= [b] \\
[\mathbf{P}(\pi_1 p)(\pi_2 p)] &= [p] \\
[(\mathbf{D}fg)(\kappa_1 a)] &= [f\,a] \\
[(\mathbf{D}fg)(\kappa_2 b)] &= [g\,b] \\
[(\mathbf{D}\kappa_1\kappa_2)u] &= [u] \\
[t] &= [\top] \\
[v] &= [\mathbf{V}]
\end{aligned}$$

Na seção 2.4 nós definimos uma relação de congruência entre termos, $\cong$, que respeitava certas equações simples envolvendo constantes; por exemplo, tínhamos $\mathbf{S}fgx \cong fx(gx)$. Será que podemos definir uma outra relação de congruência, $\cong'$, que respeite equações como as da lista acima? Vamos querer que essas equações valham com a, b, f, g, u, t e v termos quaisquer, i.e., que tenhamos

$$\begin{aligned}
\pi_1(\mathbf{P}\,a\,b) &\cong' a \\
\pi_2(\mathbf{P}\,a\,b) &\cong' b \\
\mathbf{P}(\pi_1 p)(\pi_2 p) &\cong' p \\
(\mathbf{D}fg)(\kappa_1 a) &\cong' f\,a \\
(\mathbf{D}fg)(\kappa_2 b) &\cong' g\,b \\
(\mathbf{D}\kappa_1\kappa_2)u &\cong' u \\
t &\cong' \top \\
v &\cong' \mathbf{V}
\end{aligned}$$

para quaisquer *termos* a, b, p, f, g, u, t e v com os tipos certos; note que a congruência é sintática, i.e., é entre termos, não entre pontos do modelo. Também vamos querer querer que a relação $\cong'$ seja *consistente*, i.e., que ela nunca diga que $x \cong' y$ quando x e y forem duas constantes diferentes; também gostaríamos de ter uma propriedade como normalização forte para $\cong'$,ou seja, gostaríamos que $\cong'$ viesse de uma noção de redução, e aliás se tivéssemos essa noção de redução seria fácil mostrar que $\cong'$ é consistente.

Não é nada óbvio que uma noção de redução dessas exista. Usando a regra para o $\mathbf{V}$ podemos até dar uma "pseudo-demonstração" de que quaisquer duas constantes do mesmo tipo são congruentes: se a_1 e a_2 são duas constantes de tipo α e $\bot$ é uma constante de tipo $\bot$ (um ponto do conjunto vazio $\mathbf{E}_\bot$), então $a_1 \equiv (\lambda\bot.a_1)\bot \cong' \mathbf{V}^{\bot\to\alpha}\bot \cong' (\lambda\bot.a_2)\bot \equiv a_2$. Essa pseudo-demonstração continuaria valendo se trocássemos a constante $\bot$ por qualquer outro termo de tipo $\bot$; para a relação de congruência ser consistente vai ser preciso que não haja nenhum termo de tipo $\bot$.

O fato é que vai existe uma noção de redução razoável, mas vai ser muito mais natural vê-la sobre deduções, que são o tema do próximo capítulo, do que sobre termos; por enquanto vamos nos concentrar em outras coisas. Considere o termo $\mathbf{F}$ ("flip"), definido como $\lambda p.\mathbf{P}(\pi_2 p)(\pi_1 p)$; como vamos considerar que as constantes $\mathbf{P}$, π_1, π_2, κ_1, κ_2, $\mathbf{D}$, $\top$ e $\mathbf{V}$ são combinadores então $\mathbf{F}$ não tem nem variáveis livres nem constantes livres e também é um combinador. O que esse combinador $\mathbf{F}$ faz é pegar um par p, digamos, $p^{\alpha\wedge\beta}$, e retornar um outro par montado com as duas componentes de p na ordem trocada. Vamos ver como isso fica na árvore de formação de $\mathbf{F}$ e na árvore de tipos correspondente. A árvore de formação é essa:

$$\dfrac{\dfrac{\dfrac{[p]^1 \quad \overline{\pi_1}}{\pi_1 p} \qquad \dfrac{\dfrac{[p]^1 \quad \overline{\pi_2}}{\pi_2 p} \quad \overline{\mathbf{P}}}{\mathbf{P}(\pi_2 p)}}{\mathbf{P}(\pi_2 p)(\pi_1 p)}}{\lambda p.\mathbf{P}(\pi_2 p)(\pi_1 p)}\;1$$

Estamos pondo barras acima de π_1, π_2 e $\mathbf{P}$ para ressaltar que eles são combinadores e não constantes ou variáveis das quais o $\mathbf{F}$ dependa. A árvore dos tipos para essa árvore de formação fica enorme, então introduzimos uma abreviação, as "regras F": se C é um combinador então passa a ser permitido formar $C\alpha_1\alpha_2\ldots\alpha_n$ direto a partir de $\alpha_1, \alpha_2, \ldots, \alpha_n$ usando uma barra só,

$$\dfrac{\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_n}{C\alpha_1\alpha_2\ldots\alpha_n}\;\mathrm{F}_C$$

ao invés de sermos obrigados a fazer as aplicações uma por uma. Usando regras

F o combinador **F** tem essas árvores de formação e de tipos,

$$\dfrac{\dfrac{\dfrac{[p]^1}{\pi_2 p}\,\mathrm{F}_{\pi_2} \qquad \dfrac{[p]^1}{\pi_1 p}\,\mathrm{F}_{\pi_1}}{\mathbf{P}(\pi_2 p)(\pi_1 p)}\,\mathrm{F}_{\mathbf{P}}}{\lambda p.\mathbf{P}(\pi_2 p)(\pi_1 p)}\,1 \qquad\qquad \dfrac{\dfrac{\dfrac{[a \land b]^1}{b} \qquad \dfrac{[a \land b]^1}{a}}{b \land a}}{a \land b \to b \land a}\,1$$

que deixam bem mais claro como ele funciona.

A definição do combinador **F** poderia ficar mais expressiva se a escrevêssemos como $\lambda(a,b).\mathbf{P}(\pi_2(a,b))(\pi_1(a,b))$; como a vírgula não faz parte dos símbolos que a gente usa em λ-cálculo o (a,b) bem poderia ser um nome (estranho!) para uma variável. Melhor ainda: se ficar claro pelo contexto que o símbolo a é só uma abreviatura para $\pi_1(a,b)$ e que b é só uma abreviatura para $\pi_2(a,b)$ então o flip vira $\lambda(a,b).\mathbf{P}ba$ — e aí um passo natural seria definir (b,a) como $\mathbf{P}ba$, e o flip poderia ser escrito como $\lambda(a,b).(b,a)$. Um grande problema pra chegar até aí é que é necessário saber quais objetos são os "objetos-base" a partir dos quais os outros têm que ser definidos; se não tomarmos cuidado com isso podemos definir (a,b) como $\mathbf{P}ab$, a como $\pi_1(a,b)$ e b como $\pi_2(a,b)$; uma definição circular. No capítulo 4 a gente vai começar a ver como escapar disso; no fim do capítulo 3 vamos ver um monte de situações em que seria bom ter regras dessas, mas lá a gente vai ter que se virar usando o bom senso.

2.9 Polimorfismo

Nesta seção vamos introduzir, *muito* brevemente, um assunto que parecia estar sempre sendo mencionado nas entrelinhas das seções anteriores: em que sentido dois termos que ficam iguais quando apagamos as suas anotações de tipos são "o mesmo termo"? É possível formalizar melhor o que seja "mudar o tipo" de um termo? O que são as variáveis de tipo? Nós já sabemos que podemos tentar resolver isso esquecendo as anotações de tipo e passando para λ-cálculo não-tipado, mas essa solução é péssima, porque o sistema não-tipado é muito menos bem-comportado que o λ-cálculo tipado; gostaríamos de encontrar um sistema intermediário que fosse mais expressivo que o não-tipado mas que ainda tivesse propriedades de normalização boas.

Existem extensões do λ-cálculo tipado, por exemplo o "sistema F", descrito em [GLT], que admitem variáveis de tipo. Além do λ, que abstrai símbolos atômicos, fazendo com que eles deixem de ser constantes e passem a ser variáveis, o sistema F tem uma operação Λ que abstrai tipos constantes e os transforma em tipos variáveis; ela tem uma operação complementar, correspondente à aplicação usual, que toma um termo construído com Λ e um tipo e substitui as variáveis de tipo presas pelo Λ pelo tipo dado. Por exemplo, tome $\varphi = \Lambda x.\lambda p.\lambda f.\mathbf{P}(\pi_1 p)(f(\pi_2 p))$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[p]^2}{\pi_1 p} \quad \dfrac{\dfrac{[p]^2}{\pi_2 p} \quad [f]^1}{f(\pi_2 p)}}{\mathbf{P}(\pi_1 p)(f(\pi_2 p))}}{\lambda f.\mathbf{P}(\pi_1 p)(f(\pi_2 p))}\;1}{\lambda p.\lambda f.\mathbf{P}(\pi_1 p)(f(\pi_2 p))}\;2}{\Lambda x.\lambda p.\lambda f.\mathbf{P}(\pi_1 p)(f(\pi_2 p))} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[x \land a]^2}{x} \quad \dfrac{\dfrac{[x \land a]^2}{a} \quad [a \to b]^1}{b}}{x \land b}}{(a \to b) \to (x \land b)}\;1}{x \land a \to ((a \to b) \to (x \land b))}\;2}{x \Rightarrow (x \land a \to ((a \to b) \to (x \land b)))}$$

Esse termo φ tem tipo $x \Rightarrow (x \land a \to ((a \to b) \to (x \land b)))$, e o que a seta '$\Rightarrow$' diz é que φ espera primeiro um tipo (um "ponto do espaço de tipos"?) para fazer o papel de x; depois que o φ recebe um tipo desses, digamos, $\mathbf{T}_c$, ele retorna uma função que espera um par de tipo $c \land a$ e uma função de tipo $a \to b$, e que retorna um par de tipo $c \land b$. Ou seja, $\varphi\,\mathbf{T}_c$ tem tipo $c \land a \to ((a \to b) \to (c \land b))$.

Um termo formado com a operação Λ é chamado de um termo *polimórfico*. Nós não vamos ver nem os detalhes mais básicos sobre eles, como quais são exatamente as suas regras de formação e de redução. Basta que o leitor saiba que existe uma regra de abstração de tipos que é *admissível*, i.e., não gera contradições nem igualdades demais; existe um sistema com essa regra (o sistema F) que é fortemente normalizável, e esse sistema já foi muito bem estudado; várias linguagens de programação modernas, como Haskell e ML, são baseadas nele.

As constantes $\mathbf{P}$, π_1, π_2, $\mathbf{D}$, κ_1, κ_2, $\top$ e $\mathbf{V}$ que usamos para introduzir $\land$, $\lor$, $\top$ e $\bot$ no sistema de tipos de termos podem ser vistas como constantes polimórficas, obedecendo certas equações polimórficas, mas não é necessário entendê-las desta forma. No capítulo sobre categorias várias operações naturais vão ter um ar bastante polimórfico, mas lá também vai ser mais conveniente entender a substituição de tipos como algo informal, a ser feito caso a caso. O único ponto do texto em que polimorfismo vai ser realmente importante vai ser na seção sobre parametricidade no capítulo de notas, em que descrevemos um teorema impressionante sobre termos polimórficos, que gostaríamos de saber adaptar para o sistema com a regra da derivada.

Ou seja, esta seção foi só uma curiosidade, e não precisa ser bem entendida.

Chapter 3

Dedução natural

Se cada barra de uma árvore de tipos A é a aplicação de uma das regras abaixo dizemos que essa árvore A é uma *dedução no sistema DN*.

$\dfrac{\alpha \quad \alpha \to \beta}{\beta}\ {\to}\mathrm{E}$	$\begin{array}{c} [\alpha] \quad A \\ \hline\hline \beta \\ \hline \alpha \to \beta \end{array}\ {\to}\mathrm{I}$
$\dfrac{\alpha \land \beta}{\alpha}\ {\land}\mathrm{E}_1 \qquad \dfrac{\alpha \land \beta}{\beta}\ {\land}\mathrm{E}_2$	$\dfrac{\alpha \quad \beta}{\alpha \land \beta}\ {\land}\mathrm{I}$
$\begin{array}{c} \alpha \lor \beta \quad \begin{array}{c} [\alpha] \quad A \\ \hline\hline \gamma \end{array} \quad \begin{array}{c} [\beta] \quad B \\ \hline\hline \gamma \end{array} \\ \hline \gamma \end{array}\ {\lor}\mathrm{E}$	$\dfrac{\alpha}{\alpha \lor \beta}\ {\lor}\mathrm{I}_1 \qquad \dfrac{\beta}{\alpha \lor \beta}\ {\lor}\mathrm{I}_2$
$\dfrac{\bot}{\alpha}\ {\bot}\mathrm{E}$	
	$\dfrac{}{\top}\ {\top}\mathrm{I}$

Essas barras duplas são novidade; vamos explicá-las agora. Considere uma árvore que tem o tipo β como raiz. Se o tipo de cada folha não-descarregada dessa árvore aparece na lista $\alpha_1, \ldots, \alpha_n$, então nós vamos nos permitir abreviar essa árvore como

$$\begin{array}{c}\alpha_1 \quad \dots \quad \alpha_n \\ \hline\hline \beta\end{array} \quad ;$$

a idéia é que uma barra dupla dessas representa *uma dedução de* β a partir das *hipóteses* α_1, ..., α_n. Por exemplo, na figura abaixo nas duas linhas a primeira árvore pode ser abreviada como a segunda,

$$\dfrac{\dfrac{a \quad a \to b}{b} \qquad b \to c}{c} \qquad \begin{array}{c} a \quad a \to b \quad b \to c \\ \hline\hline c \end{array}$$

$$\dfrac{\dfrac{\dfrac{[a]^1 \quad a \to b}{b} \qquad b \to c}{c}}{a \to c}\,1 \qquad \dfrac{\begin{array}{c} [a]^1 \quad a \to b \quad b \to c \\ \hline\hline c \end{array}}{a \to c}\,1$$

e vamos permitir ainda uma outra abreviação: podemos usar uma letra maiúscula para representar os últimos tipos da lista $\alpha_1, \dots, \alpha_n$. Aí a última árvore acima pode ser escrita como

$$\dfrac{\begin{array}{c} [a]^1 \quad A \\ \hline\hline c \end{array}}{a \to c}\,1\,,$$

o que vai ser muito conveniente quando só quisermos indicar que a barra 1 pode descarregar folhas de um determinado tipo.

Vamos considerar que cada folha não descarregada da árvore original está associada a exatamente um dos $\alpha_1, \dots, \alpha_n$; quando marcarmos um desses α_i como descarregado queremos dizer que todas as folhas associadas a ele estão sendo descarregadas. Note que cada α_i pode estar associado a zero, uma ou mais folhas, e que a lista $\alpha_1, \dots, \alpha_n$ pode ter tipos repetidos.

3.1 A interpretação lógica e a interpretação funcional

Cada uma das dez regras da tabela acima faz sentido tanto se interpretada *logicamente*, com cada tipo fazendo o papel de uma proposição que pode ser verdadeira ou falsa, quanto se interpretada *funcionalmente*, com cada tipo α representando um ponto do espaço $\mathbf{E}_\alpha$.

Já vimos as interpretações das regras $\to$E e $\to$I na seção ... . As regras $\wedge$E$_1$, $\wedge$E$_2$, $\wedge$I, $\vee$I$_1$ e $\vee$I$_2$ são bem claras na interpretação lógica, e na interpretação funcional elas correspondem, respectivamente, a tomar a primeira coordenada de um par (F_{π_1}, como regra F), a tomar a segunda coordenada (F_{π_2}), a formar um par ($\mathrm{F}_{\mathbf{P}}$), a fazer a inclusão no membro esquerdo de uma união disjunta (F_{κ_1}) e a fazer a inclusão no membro direito (F_{κ_2}). A regra $\bot$E diz, na interpretação lógica, que se o absurdo for verdadeiro então uma proposição α, que pode ser qualquer uma, também é verdadeira; na interpretação funcional essa regra corresponde à aplicação da função $\mathbf{V}^{\bot\to\alpha}$, a única função indo do conjunto $\mathbf{E}_\bot$, que é vazio, em $\mathbf{E}_\alpha$. A regra $\top$I é uma dedução do verdadeiro a partir de hipótese nenhuma, e na interpretação funcional ela dá o único ponto do espaço $\mathbf{E}_\top$.

A regra $\vee$E é um pouco pior. Na interpretação lógica, se $\alpha \vee \beta$ é verdade e é possível deduzir γ tanto a partir de α e A, pela dedução $\frac{\alpha\ A}{\gamma}$, quanto a partir de β e B, pela dedução $\frac{\beta\ B}{\gamma}$, então a regra $\vee$E nos permite juntar essas duas deduções de γ numa só, que só precisa de $\alpha \vee \beta$, A e B como hipóteses. Na interpretação funcional, supondo que os tipos da lista A são $\alpha_1, \ldots, \alpha_n$ e os da lista B são $\beta_1, \ldots, \beta_m$, precisamos ver $\frac{\alpha\ A}{\gamma}$ como um *procedimento* que constrói, a partir de pontos $a^\alpha, a_1^{\alpha_1}, \ldots, a_n^{\alpha_n}$, um ponto de tipo γ, e $\frac{\beta\ B}{\gamma}$ como um procedimento que constrói a partir de pontos $b^\beta, b_1^{\beta_1}, \ldots, b_m^{\beta_m}$ um ponto de tipo γ. Então a barra $\vee$E pega o ponto de tipo $\alpha \vee \beta$, e, de acordo com se ele é um ponto do membro esquerdo ou do membro direito da união disjunta, aplica o procedimento $\frac{\alpha\ A}{\gamma}$ ou o procedimento $\frac{\beta\ B}{\gamma}$ a ele. Resumindo: na interpretação funcional a regra $\vee$E corresponde a uma definição por casos, ou a um bloco tipo "if/then/else" de uma linguagem de programação.

Observação: é exatamente por causa dessa regra que exigimos que $\mathbf{E}_{\alpha\vee\beta}$ seja a união *disjunta* de $\mathbf{E}_\alpha$ com $\mathbf{E}_\beta$: se usássemos a união usual, não-disjunta, ou haveria uma ambiguidade no resultado quando o ponto de tipo $\alpha\vee\beta$ pertencesse a $\mathbf{E}_\alpha \cap \mathbf{E}_\beta$, ou teríamos que introduzir algum critério artificial para decidir o que fazer com esses pontos; qualquer uma dessas soluções produziria uma outra interpretação funcional, muito menos bem-comportada.

Às vezes vai ser conveniente considerar o sistema DN$'$, em que a regra $\vee$E é substituída pela regra abaixo:

$$\frac{\alpha \vee \beta \quad \alpha \to \gamma \quad \beta \to \gamma}{\gamma}\;\vee\mathrm{E}'$$

Essa regra $\vee\mathrm{E}'$ é considerada "equivalente" (num sentido que vamos ver na seção 5.3) à regra $\vee\mathrm{E}$, já que ocorrências de $\vee\mathrm{E}'$ podem ser transformadas em ocorrências de $\vee\mathrm{E}$ e vice-versa:

$$\frac{\alpha \vee \beta \quad \alpha \to \gamma \quad \beta \to \gamma}{\gamma}\;\vee\mathrm{E}' \qquad \Longrightarrow \qquad \frac{\alpha \vee \beta \quad \dfrac{[\alpha] \quad \alpha \to \gamma}{\gamma} \quad \dfrac{[\beta] \quad \beta \to \gamma}{\gamma}}{\gamma}\;\vee\mathrm{E}$$

$$\frac{\alpha \vee \beta \quad \dfrac{[\alpha] \quad A}{\gamma} \quad \dfrac{[\beta] \quad B}{\gamma}}{\gamma}\;\vee\mathrm{E} \qquad \Longrightarrow \qquad \frac{\alpha \vee \beta \quad \dfrac{\dfrac{[\alpha]^1 \quad A}{\gamma}}{\alpha \to \gamma}\,1 \quad \dfrac{\dfrac{[\beta]^2 \quad B}{\gamma}}{\beta \to \gamma}\,2}{\gamma}\;\vee\mathrm{E}'$$

A regra $\vee\mathrm{E}'$ tem uma interpretação funcional bem mais direta: se o ponto de tipo $\alpha \vee \beta$ for um α, aplique a função $\alpha \to \gamma$ nele; senão, aplique a função $\beta \to \gamma$. Além disso a regra $\vee\mathrm{E}'$ tem uma regra F correspondente: $\mathrm{F}_{\lambda u.\lambda f.\lambda g.(\mathbf{D}fg)u}$. Só que ela é muito pior que a regra $\vee\mathrm{E}$ quando se trata de pôr uma dedução em forma normal, que é o assunto da próxima seção.

3.2 Deduções em forma normal e normalização

Os 'I's e 'E's nos nomes das regras querem dizer "introdução" e "eliminação". Nas regras $\wedge\mathrm{E}_1$, $\wedge\mathrm{E}_2$, $\vee\mathrm{E}$ e $\to\mathrm{E}$ uma das *premissas* (que são os tipos que ficam logo acima da barra que leva o nome da regra) é desmembrada quando vamos em direção à conclusão[1], e no que essa premissa é desmembrada ela perde o seu conectivo central, $\wedge$, $\vee$ ou $\to$; por isso dizemos que o $\wedge$, o $\vee$ ou o $\to$ são *eliminados*. A premissa que é desmembrada é chamada de *premissa maior*, e vamos dizer que uma árvore está em *forma normal* se nenhuma premissa maior dela tiver sido obtida por uma regra de introdução.

Temos um processo para *normalizar* uma dedução que não esteja em forma normal: ele consiste, exceto por um detalhe técnico que vamos deixar para explicar daqui a pouco, em aplicar as transformações (chamadas de *reduções*)

[1]No sentido funcional, em que o α e o β eliminados na $\vee\mathrm{E}$ são obtidos a partir do $\alpha \vee \beta$, e portanto são considerados posteriores a ele.

abaixo, que são uma para cada caso de introdução (na premissa maior) seguida de eliminação. O fato é que para qualquer dedução que a gente tome inicialmente, esse algoritmo — enquanto houver algo para reduzir, reduza, senão pare — sempre termina num número finito de passos e sempre nos leva para a mesma forma normal. A demonstração da unicidade da forma normal sai pelo teorema de Church-Rosser, que apareceu na seção 2.3; referências para o resto podem ser encontradas em [GLT].

As reduções são as seguintes:

$$\dfrac{\dfrac{\begin{array}{c}A\\\hline\hline\alpha\end{array}\quad\begin{array}{c}B\\\hline\hline\beta\end{array}}{\alpha\wedge\beta}}{\alpha} \quad\Longrightarrow\quad \begin{array}{c}A\\\hline\hline\alpha\end{array} \qquad\qquad \dfrac{\dfrac{\begin{array}{c}A\\\hline\hline\alpha\end{array}\quad\begin{array}{c}B\\\hline\hline\beta\end{array}}{\alpha\wedge\beta}}{\beta} \quad\Longrightarrow\quad \begin{array}{c}B\\\hline\hline\beta\end{array}$$

$$\dfrac{\begin{array}{c}A\\\hline\hline\alpha\end{array}\quad \dfrac{\begin{array}{c}[\alpha]^1\quad B\\\hline\hline\beta\end{array}}{\alpha\to\beta}{\scriptstyle 1}}{\beta} \quad\Longrightarrow\quad \begin{array}{c}\begin{array}{c}A\\\hline\hline\alpha\end{array}\quad B\\\hline\hline\beta\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}A'\\\hline\hline\alpha\end{array}}{\alpha\vee\beta}\quad\begin{array}{c}[\alpha]^1\quad A\\\hline\hline\gamma\end{array}\quad\begin{array}{c}[\beta]^1\quad B\\\hline\hline\gamma\end{array}}{\gamma}{\scriptstyle 1} \quad\Longrightarrow\quad \begin{array}{c}\begin{array}{c}A'\\\hline\hline\alpha\end{array}\quad A\\\hline\hline\gamma\end{array}$$

$$\dfrac{\dfrac{\begin{array}{c}B'\\\hline\hline\beta\end{array}}{\alpha\vee\beta}\quad\begin{array}{c}[\alpha]^1\quad A\\\hline\hline\gamma\end{array}\quad\begin{array}{c}[\beta]^1\quad B\\\hline\hline\gamma\end{array}}{\gamma}{\scriptstyle 1} \quad\Longrightarrow\quad \begin{array}{c}\begin{array}{c}B'\\\hline\hline\beta\end{array}\quad B\\\hline\hline\gamma\end{array}$$

O detalhe técnico que faltava é que precisamos de algumas regras de redução extras para o $\vee$, o $\perp$ e o $\top$, que chamamos de *conversões permutativas*. No caso do $\vee$, como a barra do $\vee$E unifica os dois γs de cima no de baixo, sem alterá-los, pode ser que um tipo γ seja obtido por uma introdução acima de uma barra $\vee$E, mas só vá ser usado numa eliminação mais abaixo; aí temos que aplicar uma transformação que consiga puxar o uso do γ na eliminação para mais perto do ponto em que ele é obtido por introdução. Por exemplo, lembrando que $\neg b$ é definido como sendo $b\to\perp$,

$$\dfrac{\dfrac{a \lor \neg b \qquad \dfrac{[a]^1 \quad c}{a \land c} \qquad \dfrac{\dfrac{b \quad [\neg b]^1}{\bot}}{a \land c}}{a \land c}}{a} \qquad \Longrightarrow \qquad \dfrac{a \lor \neg b \qquad \dfrac{\dfrac{[a]^1 \quad c}{a \land c}}{a} \qquad \dfrac{\dfrac{\dfrac{b \quad [\neg b]^1}{\bot}}{a \land c}}{a}}{a}\,1$$

que depois se reduz a

$$\Longrightarrow \quad \dfrac{a \lor \neg b \qquad [a]^1 \qquad \dfrac{\dfrac{\dfrac{b \quad [\neg b]^1}{\bot}}{a \land c}}{a}}{a}\,1 \quad \Longrightarrow \quad \dfrac{a \lor \neg b \qquad [a]^1 \qquad \dfrac{\dfrac{b \quad [\neg b]^1}{\bot}}{a}}{a}\,1\, ,$$

onde essa última transformação aplica uma das conversões permutativas para o $\bot$: como só existe uma função indo de $\mathbf{E}_\bot$ para qualquer outro espaço, já que $\mathbf{E}_\bot$ é vazio, certas deduções que chegam a um γ indiretamente a partir de um $\bot$ são encolhidas a deduções que vão direto de um $\bot$ para um γ. Nas páginas 80 e 81 de [GLT] há uma lista de todas as conversões para o $\lor$ e para o $\bot$; note que lá elas são dadas como reduções em λ-cálculo e que elas usam uma operação, δ, que nós tentamos ao máximo evitar e que é usada para representar o termo construído pela regra $\lor$E: se φ^γ é um termo no qual queremos abstrair x^α e ψ^γ é um termo no qual queremos abstrair y^β, então $\delta\, x^\alpha.\varphi^\gamma\, y^\beta.\psi^\gamma\, u^{\alpha\lor\beta}$ abstrai x^α de φ^γ e y^β de ψ^γ de só uma vez e aplica $u^{\alpha\lor\beta}$ na função $\alpha\lor\beta\to\gamma$ obtida a partir deles; nós teríamos escrito isso como $\mathbf{D}(\lambda x^\alpha.\varphi^\gamma)(\lambda y^\alpha.\psi^\gamma)u^{\alpha\lor\beta}$. A conversão que queremos para o $\top$ não aparece em [GLT]: qualquer dedução que tenha o $\top$ como conclusão se reduz a uma aplicação da regra $\top$I.

O sistema com essas reduções é fortemente normalizável, i.e., o algoritmo de "enquanto puder reduzir reduza, senão pare" sempre termina e o resultado só depende da dedução inicial, não do caminho escolhido.

Como todas as reduções preservam os valores como termos das deduções, acabamos de aprender como normalizar termos de λ-cálculo com tipos contendo $\to$, $\land$, $\lor$, $\top$ e $\bot$: pegue um termo, transforme-o numa dedução, normalize a dedução, transforme de volta num termo. Uma coisa estranha é que às vezes a normalização leva um termo em outro que tem um valor "mais definido": por exemplo, $\pi_1(\mathbf{P}a\bot)$ se reduz a a, mas o valor do primeiro depende do valor de $\bot$, que não existe (esse exemplo não está totalmente correto, mas é bem expressivo) e o segundo não. Dois termos equivalentes vão ser iguais onde eles estiverem definidos; isso vai ficar claro no capítulo 8.

3.3 A terceira interpretação

Tome uma árvore de dedução A em DN e fixe uma valuação clássica[2]. Por essa valuação cada tipo φ de $\mathcal{A}$ corresponde a uma proposição, que ou é verdadeira ou é falsa; se ela for verdadeira vamos definir $\mathbf{E}_{\top_\varphi}$ como sendo um conjunto com um elemento, e se ela for falsa vamos definir $\mathbf{E}_{\top_\varphi}$ como sendo o conjunto vazio. Agora uma proposição φ ser verdadeira corresponde ao espaço $\mathbf{E}_{\top_\varphi}$ ter um ponto, e um modo de mostrarmos que φ é verdadeira é apresentarmos um ponto de tipo $\top_\varphi$.

Veja que o espaço $\mathbf{E}_{\top_{\varphi\wedge\psi}}$ tem um ponto se e só se $\mathbf{E}_{\top_\varphi}$ e $\mathbf{E}_{\top_\psi}$ têm um ponto cada um; ou seja, se $\mathbf{E}_{\top_\varphi \wedge \top_\psi}$ tem um ponto. Do mesmo modo, $\mathbf{E}_{\top_{\varphi\to\psi}}$ tem um ponto se e só se $\mathbf{E}_{\top_\varphi \to \top_\psi}$ tem um ponto. Já o caso do $\vee$ é diferente: se φ e ψ forem verdadeiras então $\top_\varphi \vee \top_\psi$ tem *dois* pontos, enquanto $\top_{\varphi\vee\psi}$ só tem um; para consertar isso temos que trocar a união disjunta por uma união normal, não-disjunta. Esse conserto vai ser feito no capítulo 8; quando estivermos em topoi só vamos ter a interpretação funcional, e essa tradução da interpretação lógica na funcional, que chamamos de *terceira interpretação*, vai ser o modo de interpretar a lógica dentro de um topos.

Os tipos $\top_\varphi$ vão ser chamados de *subobjetos do* $\top$; o nome "subobjeto" também vem da terminologia de topoi, e dizemos que o tipo dos as é um subobjeto do tipo dos bs quando houver um modo natural de considerar o espaço dos as como subconjunto do espaço dos b, ou seja, quando houver uma função natural de inclusão, injetiva, que leve cada a num b.

3.4 Quantificadores

Fixe uma função P que receba dois parâmetros, um de tipo x, outro de tipo y, e retorne um valor de verdade, i.e., um valor de tipo τ, onde $\mathbf{E}_\tau = \{\top, \dashv\}$; podemos considerar que P é uma constante de tipo $x \to (y \to \tau)$. O valor de verdade de Pxy depende dos valores de x e de y, mas o de $\forall x.\exists y.Pxy$ não; e se

[2]Veja a seção 6.1 se não souber o que é uma valuação clássica.

sabemos que $\forall x.\exists y.Pxy$ é verdade e temos um determinado valor para x (veja que estamos usando os mesmos símbolos para denotar variáveis e constantes, como no capítulo 2), então podemos deduzir que $\exists y.Pxy$ é verdade. Só que só queremos poder deduzir $\exists y.Pxy$ a partir de $\forall x.\exists y.Pxy$ na presença de um ponto x, para evitar as contradições que poderiam aparecer se $\mathbf{E}_x$ fosse vazio.

Vamos aproveitar que na interpretação funcional podemos ter nós de todos os tipos e vamos introduzir regras para os quantificadores que envolvem tanto subobjetos do $\top$ quanto outros tipos.

$$\dfrac{\top_{\exists x} \qquad \dfrac{[x]^1 \quad A}{\top_\varphi}}{\top_\varphi}\;1:\exists\mathrm{E} \qquad\qquad \dfrac{x \quad \top_\varphi}{\top_{\exists x.\varphi}}\;\exists\mathrm{I}$$

$$\dfrac{x \quad \top_{\forall x.\varphi}}{\top_\varphi}\;\forall\mathrm{E} \qquad\qquad \dfrac{\dfrac{[x]^1 \quad A}{\top_\varphi}}{\top_{\forall x.\varphi}}\;1:\forall\mathrm{I}$$

A regra $\forall$I vai ter uma restrição: todos os tipos que aparecem em A têm que ser independentes de x. Não vamos dar uma definição formal para "independentes".

É melhor explicar essas regras através do exemplo abaixo, uma demonstração de $\forall x.\exists y.(Pxy \land Pyx)$ a partir de $\forall x.\exists y.Pxy$ e $\forall x.\forall y.(Pxy \to Pyx)$. Ela foi separada em duas partes por razões meio de clareza de exposição, meio tipográficas.

$$\dfrac{y \qquad \dfrac{\top_{Pxy} \qquad \dfrac{\top_{Pxy} \qquad \dfrac{y \qquad \dfrac{x \quad \top_{\forall x.\forall y.(Pxy\to Pyx)}}{\top_{\forall x.(Pxy\to Pyx)}}\;\forall\mathrm{E}}{\top_{(Pxy\to Pyx)}}\;\forall\mathrm{E}}{\top_{Pyx}}}{\top_{Pxy\land Pyx}}}{\top_{\exists y.(Pxy\land Pyx)}}\;\exists\mathrm{I}$$

Dado um ponto x, um ponto y e um ponto que diz que $\forall x.\forall y.(Pxy \to Pyx)$ é verdade, deduzimos que $Pxy \to Pyx$ é verdade. Se além disso temos um ponto dizendo que Pxy é verdade, vemos que $Pxy \land Pyx$ é verdade. Repare que o espaço dos $\top_{Pxy}$ depende dos valores de x e y; um modo bom de entender essas dependências é pensar que ao invés de começarmos com x e y e depois obtermos o espaço $\mathbf{E}_{\top_{Pxy}}$ e um ponto $\top_{Pxy}$ dentro dele, nós começamos com o espaço das

3-uplas "admissíveis" $(x, y, \top_{Pxy})$, ou, melhor ainda, das 4-uplas admissíveis $(x, y, P, \top_{Pxy})$, e então tomamos uma 4-upla desse espaço, e os pontos x, y e $\top_{Pxy}$ são obtidos por projeções nas coordenadas. Desse modo conseguimos empurrar todo o problema para dois lugares: encontrar que coordenadas as uplas devem ter, e filtrar, dentre todas as uplas que têm os tipos certos, as que são admissíveis. Esses dois problemas são resolvíveis, mas só vamos discutí-los bem mais adiante (cap. 8).

No último passo da árvore acima nós pegamos um y, o mesmo que entrou como hipótese na segunda linha, e deduzimos que $\exists y.(Pxy \land Pyx)$ é verdade. Estamos usando a mesma convenção que em λ-cálculo: se duas variáveis com o mesmo nome, no caso y, aparecem livres num termo φ, as duas ficam presas pelo mesmo λ quando passamos para $\lambda y.\varphi$. Repare que o y não aparece livre em $\exists y.(Pxy \land Pyx)$, mas para encontrarmos o ponto $\top_{\exists y.(Pxy \land Pyx)}$ nós usamos um ponto y e um ponto $\top_{Pxy}$, que depende de y no sentido de que só existe para alguns valores de y.

A árvore acima nos deu um ponto $\top_{\exists y.(Pxy \land Pyx)}$ a partir de um x, um y, um $\top_{Pxy}$ e um $\top_{\forall x.\forall y.(Pxy \to Pyx)}$; ela aparece como a barra dupla da árvore abaixo.

$$
\dfrac{
  \dfrac{[x]^2 \quad \top_{\forall x.\exists y.Pxy}}{\top_{\exists y.Pxy}}\;\forall\mathrm{E}
  \qquad
  \dfrac{[x]^2 \quad \dfrac{[y|_{Pxy}]^1}{y} \quad \dfrac{[y|_{Pxy}]^1}{\top_{Pxy}} \quad \top_{\forall x.\forall y.(Pxy \to Pyx)}}{\top_{\exists y.(Pxy \land Pyx)}}
}{
  \dfrac{\top_{\exists y.(Pxy \land Pyx)}}{\top_{\forall x.\exists y.(Pxy \land Pyx)}}\;2:\forall\mathrm{I}
}\;1:\exists\mathrm{E}
$$

Se temos um x e um $\top_{\forall x.\exists y.Pxy}$ nós temos, por $\forall$-eliminação, um $\top_{\exists y.Pxy}$; esse $\top_{\exists y.Pxy}$ afirma que existe um ponto no espaço dos "ys tais que Pxy", que é um subconjunto do espaço dos ys; esses espaços "tais que" vão ser discutidos no fim do próximo capítulo, mas o que nos interessa agora é que a partir de um "y tal que Pxy" (que vamos escrever como $y|_{Pxy}$) nós podemos obter tanto um y, pela inclusão natural $y|_{Pxy} \to y$, quanto um $\top_{Pxy}$, a partir de uma função $y|_{Pxy} \to \top_{Pxy}$ que leva todo ponto de $\mathbf{E}_{y|_{Pxy}}$ no (possível) único ponto de $\top_{Pxy}$; o espaço $\mathbf{E}_{\top_{Pxy}}$ só vai poder ser vazio se o $\mathbf{E}_{y|_{Pxy}}$ também for. Não escrevemos essas duas funções naturais na árvore acima por questões de espaço, mas elas vão ser como teoremas (ou axiomas!) do sistema dedutivo do capítulo sobre topoi, no sentido de que elas podem ser construídas sem hipóteses, i.e., sem dependerem de nenhum ponto de nenhum tipo, e escrevê-las ou não não vai interferir com os passos seguintes da dedução, já que as nossas regras só envolvem os objetos imediatamente acima da barra e as folhas não-descarregadas.

A barra marcada com $1 : \exists\text{E}$ usa o ponto $\top_{\exists y.Pxy}$ para descarregar todas as hipóteses marcadas com $[\;]^1$: se a partir de qualquer ponto $y|_{Pxy}$ e das outras hipóteses conseguimos encontrar um ponto $\top_{\exists y.(Pxy \land Pyx)}$ (que é unico, se existir), então basta que saibamos que existe um ponto $y|_{Pxy}$ para sabermos que existe o ponto $\top_{\exists y.(Pxy \land Pyx)}$; não precisamos ter um ponto $y|_{Pxy}$ específico.

Depois da aplicação da regra $\exists\text{E}$ chegamos a uma construção de um ponto $\top_{\exists y.(Pxy \land Pyx)}$ só a partir de um ponto x, um $\top_{\forall x.\exists y.Pxy}$ e um $\top_{\forall x.\forall y.(Pxy \to Pyx)}$. Como esses são os únicos pontos necessários para chegar ao $\top_{\exists y.(Pxy \land Pyx)}$, já que as funções naturais mencionadas há dois parágrafos não dependem de ponto nenhum, podemos aplicar a regra $\forall\text{I}$ ao x, já que as outras duas hipóteses são independentes dele; com isso descarregamos o x, e obtemos a demonstração que queríamos.

Observação: a dedução que usamos como exemplo é a tradução para o sistema tipado e com símbolos que são ao mesmo tempo constantes e variáveis da que aparece em [Prawitz], p.19. O original é isso aqui:

$$
\dfrac{
  \dfrac{\forall x.\exists y.Pxy}{\exists y.Pay}\,\forall\text{E}
  \qquad
  \dfrac{
    \dfrac{
      [Pab]^1 \qquad \dfrac{[Pab]^1 \qquad \dfrac{\dfrac{\forall x.\forall y.(Pxy \to Pyx)}{\forall y.(Pay \to Pya)}\,\forall\text{E}}{Pab \to Pba}\,\forall\text{E}}{Pba}
    }{Pab \land Pba}
  }{\exists y.(Pay \land Pya)}\,\exists\text{I}
}{
  \dfrac{\exists y.(Pay \land Pya)}{\forall x.\exists y.(Pxy \land Pyx)}\,\forall\text{I}
}\,1:\exists\text{E}
$$

Chapter 4

Categorias

Introdução. Este capítulo é motivado por algumas perguntas muito básicas: O que é que categorias têm a ver com λ-cálculo? Porque as definições de funtores, transformações naturais e adjunções são exatamente como são? Qual é a vantagem de usarmos diagramas? Como trabalhar sem pontos?

Primeiro uma definição formal. Uma *categoria* $\mathbf{C}$ é uma 6-upla $(\mathcal{O}, \mathcal{M}, \mathrm{dom}, \mathrm{cod}, \circ, \mathrm{id})$, onde $\mathcal{O}$ é a classe dos *objetos da categoria*, $\mathcal{M}$ é a classe dos *morfismos*, dom e cod são funções que levam cada morfismo no seu *domínio* e no seu *contradomínio* respectivamente, $\circ$ é a operação de *composição* de morfismos e id é a função que leva cada objeto a da categoria num morfismo com domínio e contradomínio a, chamado de *identidade*. A composição tem que obedecer as seguintes três condições:

1) A composta $g \circ f$ está definida exatamente quando o contradomínio de f e o domínio de g forem o mesmo objeto. Vamos escrever $a \xrightarrow{f} b$ para indicar que f tem domínio a e contradomínio b; se g é $b \xrightarrow{g} c$ vamos escrever a composta tanto como $g \circ f$ quanto como $a \xrightarrow{f} b \xrightarrow{g} c$ ou $a \xrightarrow{g \circ f} c$.

2) A composição é associativa, i.e., $f \circ (g \circ h) = (f \circ g) \circ h$. Isso nos permite escrever compostas de mais de dois morfismos sem termos que nos preocupar com a ordem em que as composições são aplicadas: por exemplo, $h \circ g \circ f$ ou $a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$.

3) A identidade é o elemento neutro da composição: $a \xrightarrow{\mathrm{id}} a \xrightarrow{f} b = a \xrightarrow{f} b = a \xrightarrow{f} b \xrightarrow{\mathrm{id}} b$.

Os dois exemplos mais evidentes de categorias são **Set**, em que os objetos são os conjuntos e os morfismos de A em B são as funções com domínio A

e contradomínio B, e $\mathbf{Set}_\mathcal{A}$; em $\mathbf{Set}_\mathcal{A}$ ao invés de falarmos, por exemplo, no "espaço dos αs" e no "espaço dos '$\alpha \wedge \beta$'s", vamos falar no *objeto* dos αs, $\mathbf{O}_\alpha$, e no *objeto* dos '$\alpha \wedge \beta$'s, $\mathbf{O}_{\alpha\wedge\beta}$; os morfismos de $\mathbf{O}_\alpha$ em $\mathbf{O}_\beta$ vão ser as funções de $\mathbf{E}_\alpha$ em $\mathbf{E}_\beta$. Essa distinção entre espaços e objetos parece ser meramente burocrática, mas em categorias não temos nenhuma regra que nos permita falar dos *pontos* de um objeto; as regras básicas são apenas

$$\frac{\alpha \to \beta}{\mathbf{O}_\alpha} \qquad \frac{\alpha \to \beta}{\mathbf{O}_\beta} \qquad \frac{\alpha \to \beta \quad \beta \to \gamma}{\alpha \to \gamma} \qquad \frac{\mathbf{O}_\alpha}{\alpha \to \alpha}\ ,$$

que correspondem a dom, cod, $\circ$ e id, respectivamente.

Em categorias vamos querer, muito mais do que em λ-cálculo, interpretar árvores como *construções* de objetos; por exemplo,

$$\cfrac{a \xrightarrow{f} b \qquad \cfrac{\cfrac{a \xrightarrow{f} b}{\mathbf{O}_b}}{b \to b}}{a \to b}$$

é uma construção — não muito interessante, admito — de $a \xrightarrow{f} b \xrightarrow{\mathrm{id}} b$ a partir de $a \xrightarrow{f} b$; e, da mesma forma que em λ-cálculo, em geral estamos muito mais interessados no objeto que é o resultado de uma construção ($\sim$ o valor de um termo, ou a sua interpretação no modelo) do que na estrutura da construção em si ($\sim$ o termo, visto sintaticamente); por isso quando escrevermos que uma construção é "igual" à outra vamos querer dizer que o *resultado* das duas é igual. Repare que o resultado de uma construção depende dos valores que pomos nas folhas da árvore, e que isso depende do contexto: no exemplo acima precisamos saber em que categoria estamos e quem é $a \xrightarrow{f} b$.

4.1 Construções naturais e diagramas comutativos

Nós estamos tão interessados em árvores porque elas vão nos permitir definir *construções naturais* sobre diagramas. Não vamos definir formalmente o que seja um diagrama; assim que definirmos o que são diagramas comutativos e objetos bem-definidos vai ficar claro o porquê disso, e também o de não estarmos interessados em diagramas complicados demais. Em particular, todos os nossos diagramas vão ser finitos.

Definição: uma *construção natural* para γ num diagrama D é uma árvore de tipos com o tipo γ na raiz em que toda barra é uma aplicação de uma das regras de construção válidas, e em que todas as folhas da árvore, inclusive as descarregadas, são objetos do diagrama D.

Cada diagrama costuma vir junto com a noção de que alguns dos objetos que aparecem representados nele são *objetos-base*; cada objeto-base tem um *valor* (do tipo adequado) associado a ele, e cada construção natural em que todas as folhas não-descarregadas são objetos-base também pode ser associada a um valor, que chamamos de *resultado* dessa construção natural. É melhor explicar isso com um exemplo.

$$\begin{tikzcd}
a \arrow[r] \arrow[d, shift right, "f"'] \arrow[d, shift left, "g"] \arrow[dr, "\mathrm{id}"] & b \arrow[d] \\
c \arrow[r] & a'
\end{tikzcd}$$

Vamos considerar que no diagrama acima os objetos-base sejam as seis setas. Então temos exatamente uma construção natural para $a \to b$, uma para $b \to a'$ e uma para $c \to a'$, que são tomar as setas $a \to b$, $b \to a'$ e $c \to a'$ dadas; podemos nos referir a elas pelos nomes dos seus tipos, exatamente como fizemos na seção 1.1. Quanto a $a \to a'$, existem quatro construções naturais para ela:

$$\dfrac{a \stackrel{f}{\to} c \quad c \to a'}{a \to a'} \qquad \dfrac{a \stackrel{g}{\to} c \quad c \to a'}{a \to a'} \qquad a \stackrel{\mathrm{id}}{\to} a' \qquad \dfrac{a \to b \quad b \to a'}{a \to a'}$$

os resultados delas são respectivamente $a \stackrel{f}{\to} c \to a'$, $a \stackrel{g}{\to} c \to a'$, $a \stackrel{\mathrm{id}}{\to} a'$ e $a \to b \to a'$. Note que pusemos uma seta identidade indo de a para a'; a existência dela indica que $\mathbf{O}_a$ e $\mathbf{O}'_a$ são o mesmo objeto da categoria, mas que por algum motivo resolvemos representá-los como se fossem objetos diferentes — por exemplo, para que não seja natural obter a seta $a' \stackrel{\mathrm{id}}{\to} a$. Esse tipo de coisa, e mais as "anotações de valor" f, g e id que aparecem em algumas das folhas das construções acima, fazem com que os tipos que aparecem nos nós de uma árvore de construção natural sejam ligeiramente diferentes dos tipos com que estávamos acostumados, mas vamos pular os detalhes disto.

Uma observação: se as regras de construção fossem as regras de construção de termos em λ-cálculo, então uma construção natural para $a \to a'$ seria essa aqui:

$$\dfrac{\dfrac{\dfrac{[a]^1 \quad a \xrightarrow{f} c}{c} \qquad c \to a'}{a'}}{a \to a'}\;1$$

Ela começa supondo a existência de pontos de tipo a, depois ela constrói um ponto de tipo c e um de tipo a', e depois descarrega a hipótese do ponto de tipo a; o resultado dela é exatamente a composta $a \xrightarrow{f} c \to a'$, que já tínhamos visto como obter sem pontos. Vamos voltar várias vezes a esse tema dos pontos, e no próximo capítulo vamos ver algoritmos que permitem traduzir demonstrações "com pontos" para demonstrações "sem pontos" e vice-versa; a moral da história vai ser que pontos são ferramentas úteis que deixam certas construções mais claras, e que vale a pena entender um diagrama como algo que acontece em $\mathbf{Set}_{\mathcal{A}}$; a generalização para outras categorias pode ser feita depois, mecanicamente.

Se os valores de f, g, $c \to a'$, $a \to b$ e $b \to a'$ tivessem sido escolhidos de tal forma que todas as construções naturais de $a \to a'$ dessem o mesmo resultado poderíamos nos referir a esse resultado como sendo o "valor de $a \to a'$", sem que precisássemos especificar qual construção natural foi usada para obtê-lo. Essa idéia nos leva às seguintes definições:

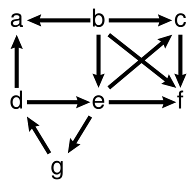
Def.: um tipo γ está *bem-definido* (num diagrama D) quando todas as construções naturais para γ dão o mesmo resultado.

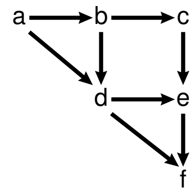
Def.: um diagrama é *comutativo* quando todo tipo que tenha uma construção natural está bem-definido.

A graça disso é que muitas vezes é fácil ver que um diagrama comuta ou que certos objetos dele estão bem-definidos. Nos dois lemas abaixo os objetos-base são todas as setas que aparecem explicitamente no diagrama.

4.2 Dois lemas sobre diagramas

Lema do diagrama planar. se um diagrama D está desenhado em forma planar, cada uma das suas "células" tem exatamente uma "fonte" e um "poço" e todas as suas células comutam, então o diagrama todo comuta. Para entender o que queremos dizer com "planar", "células", "fontes" e "poços" veja o diagrama abaixo à esquerda:



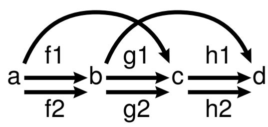


Ele é não-planar porque as setas $b \to f$ e $e \to c$ se cruzam; ele não obedece as condições sobre fontes e poços porque a fronteira da célula *abed* é formada pelas setas $a \leftarrow b \to e \leftarrow d \to a$: b e d são "fontes", porque as duas setas apoiadas em cada um deles estão saindo deles, e a e e são "poços", já que as setas apoiadas neles estão indo para eles; além disso a célula *deg* não tem nem fontes nem poços.

Digamos que no diagrama à direita as células *abd*, *bced* e *def* comutem, ou seja, $a \to b \to d = a \to d$, $b \to c \to e = b \to d \to e$ e $d \to e \to f = d \to f$. Então o lema diz que esse diagrama inteiro comuta, e a demonstração — que vale para o caso geral — é simplesmente "passar por cima das células". Tome duas letras α e β nesse diagrama que representem objetos da categoria e tais que existam construções naturais para $\alpha \to \beta$, i.e., exista um caminho de α para β; digamos, $\alpha = a$ e $\beta = f$. Temos que mostrar que quaisquer duas construções naturais para $a \to f$ dão o mesmo resultado; por exemplo, omitindo as setas para encurtar a notação, que $adf = abcef$. Uma demonstração disso: $adf = abdf = abdef = abcef$.

Note que não foi nem necessário exigir explicitamente que a célula exterior do diagrama, *abcefd*, comutasse: a comutatividade dela é conseqüência do resto.

Lema da sombra. Considere o diagrama abaixo, em que temos uma seqüência de letras, (a, b, c, d), com pelo menos uma seta horizontal indo de cada uma para a seguinte, e entre algumas letras temos "arcos" ($a \to c$, $b \to d$).



Se todos os arcos vão na mesma direção que as setas horizontais, se a "sombra" dos arcos cobre todas as setas horizontais (considere que o sol está no topo da página) e se para cada arco $\alpha \to \beta$ temos que a seta $\alpha \to \beta$ está bem-definida, então a seta $a \to d$, onde a é a primeira letra da seqüência e d é a última, está bem-definida. Por exemplo, vamos mostrar que $a \xrightarrow{f_1} b \xrightarrow{g_1} c \xrightarrow{h_1} d = a \xrightarrow{f_2} b \xrightarrow{g_2} c \xrightarrow{h_2} d$:

$$\begin{aligned} a \xrightarrow{f_1} b \xrightarrow{g_1} c \xrightarrow{h_1} d &= a \longrightarrow c \xrightarrow{h_1} d \\ &= a \xrightarrow{f_2} b \xrightarrow{g_2} c \xrightarrow{h_1} d \\ &= a \xrightarrow{f_2} b \longrightarrow d \\ &= a \xrightarrow{f_2} b \xrightarrow{g_2} c \xrightarrow{h_2} d \end{aligned}$$

As setas que aparecem sem nome acima são arcos.

4.3 Monics, epis, isos

Uma seta $b \xrightarrow{g} c$ induz uma operação, que vamos chamar de *composição com $b \xrightarrow{g} c$ à direita*, que leva cada seta $\alpha \to b$ (para qualquer tipo α) na composta $\alpha \to b \xrightarrow{g} c$. Se essa operação é injetiva, i.e., não existe um tipo α e duas setas diferentes $\alpha \xrightarrow{f_1} b$ e $\alpha \xrightarrow{f_2} b$ com $\alpha \xrightarrow{f_1} b \xrightarrow{g} c$, então dizemos que $b \xrightarrow{g} c$ é um *monomorfismo*, ou, pra encurtar, um *monic*; se uma seta é um monic nós nos permitimos usar '$\hookrightarrow$' ao invés de '$\to$' para escrevê-la. Seguindo essa idéia, $\mathbf{E}_{a \hookrightarrow b}$ é o espaço dos monics indo de a em b; $\mathbf{E}_{a \hookrightarrow b} \subset \mathbf{E}_{a \to b}$.

Em **Set** os monics são as funções injetivas.

Observação: tradicionalmente monics são indicados por '$\rightarrowtail$', não '$\hookrightarrow$'; '$\hookrightarrow$' é usado para indicar setas *injetivas*, que têm uma outra definição em categorias; veja [G], pp.123–124, para a definição e algumas propriedades. Como nós vamos estar em **Set** quase todo o tempo nós vamos ignorar (de propósito!) a distinção entre '$\rightarrowtail$' e '$\hookrightarrow$' e escrever sempre '$\hookrightarrow$'.

Se uma seta $b \xrightarrow{g} c$ tem uma *inversa à direita* $c \xrightarrow{h} b$, ou seja, se essa $c \xrightarrow{h} b$ é tal que $b \xrightarrow{g} c \xrightarrow{h} b = b \xrightarrow{\mathrm{id}} b$, então dizemos que $b \xrightarrow{g} c$ é um *split monic*. É trivial ver que todo split monic é um monic.

Em **Set** os únicos monics que não são split monics são os que têm domínio vazio e contradomínio não-vazio.

Epis, ou *epimorfismos*, são o dual de monics, i.e., eles são definidos como os monics, mas invertendo a direção de todos os morfismos da definição: $c \to b$ é um epimorfismo se a operação que leva cada $b \to \alpha$ em $c \to b \to \alpha$ é injetiva para todo objeto $\mathbf{O}_\alpha$. *Split epis* são o dual de split monics.

Em **Set** os epis são as funções sobrejetivas, e se vale o axioma da escolha então todo epi é um split epi.

Digamos que uma seta $b \to c$ tenha inversas laterais pelos dois lados, ou seja, que essa $b \to c$ seja ao mesmo um split epi e um split monic. Então é possível ver, usando o lema da sombra, que todas essas inversas laterais são iguais: se f_1 e f_2 são inversas pela esquerda e g_1 e g_2 são inversas pela direita, então, pelo diagrama abaixo

$$\begin{tikzcd}
\text{b'} \arrow[r, shift left, "\text{f1}"] \arrow[r, shift right, "\text{f2}"'] \arrow[rr, bend left, "\text{id}"] & \text{a} \arrow[r, "\text{g}"] \arrow[rr, bend left, "\text{id}"] & \text{b} \arrow[r, shift left, "\text{h1}"] \arrow[r, shift right, "\text{h2}"'] & \text{a'}
\end{tikzcd}$$

temos $\mathrm{id}_a \circ f_1 = \mathrm{id}_a \circ f_2 = g_1 \circ \mathrm{id}_b = g_2 \circ \mathrm{id}_b = f_1 = f_2 = g_1 = g_2$, onde id_a é a identidade $a \overset{\mathrm{id}}{\to} a$ e id_b é $b \overset{\mathrm{id}}{\to} b$.

Uma seta $a \to b$ que tenha inversa pelos dois lados é chamada de *isomorfismo*, e vamos nos permitir escrevê-la como $a \leftrightarrow b$. Essa notação deixa implícito que existe uma seta $b \to a$ natural, e obviamente vamos escolhê-la como sendo a inversa de $a \to b$.

Dois objetos de uma categoria que tenham um isomorfismo entre eles são ditos *isomorfos*. Nós vamos usar esse termo num sentindo mais estrito: quando dissermos que dois objetos são isomorfos vai ficar implícito que temos um isomorfismo especial, "natural", entre eles.

4.4 Funtores

Um *funtor* $F : \mathbf{C} \Rightarrow \mathbf{D}$ é algo que leva cada objeto $\mathbf{O}_a$ da categoria $\mathbf{C}$ num objeto da categoria $\mathbf{D}$, que geralmente vamos chamar de $\mathbf{O}_{a^F}$, cada morfismo de $\mathbf{C}$ num morfismo de $\mathbf{D}$ e que obedece as seguintes condições:

1) morfismos de α em β vão em morfismos de α^F em β^F;

2) a imagem de cada morfismo identidade é um morfismo identidade;

3) para quaisquer dois morfismos f e g de $\mathbf{C}$ tais que $f \circ g$ exista, i.e., tais que o objeto-destino de g seja o objeto-origem de f, temos $(f \circ g)^F = (f^F \circ g^F)$.

O que essa condição $(f \circ g)^F = (f^F \circ g^F)$ diz, em termos de diagramas, é que se $a \to b$ e $b \to c$ são os objetos-base do diagrama abaixo

$$
\begin{tikzcd}
 & b \arrow[dr] \arrow[dd, Rightarrow] & \\
a \arrow[ur] \arrow[rr] \arrow[dd, Rightarrow] & & c \arrow[dd, Rightarrow] \\
 & bF \arrow[dr] & \\
aF \arrow[ur] \arrow[rr] & & cF
\end{tikzcd}
$$

então as duas construções naturais para a seta $a^F \to c^F$ dão o mesmo resultado:

$$
\dfrac{\dfrac{a \to b}{a^F \to b^F} \quad \dfrac{b \to c}{b^F \to c^F}}{a^F \to c^F}
\;=\;
\dfrac{\dfrac{g}{g^F} \quad \dfrac{f}{f^F}}{f^F \circ g^F}
\;=\;
\dfrac{\dfrac{g \quad f}{f \circ g}}{(f \circ g)^F}
\;=\;
\dfrac{\dfrac{a \to b \quad b \to c}{a \to c}}{a^F \to c^F}
$$

ou seja, tanto faz primeiro aplicar o funtor e depois fazer a composição quanto compor primeiro e aplicar o funtor depois. Note que como a composição de morfismos é associativa nós sempre temos $(f_1 \circ \ldots \circ f_n)^F = f_1^F \circ \ldots \circ f_n^F$.

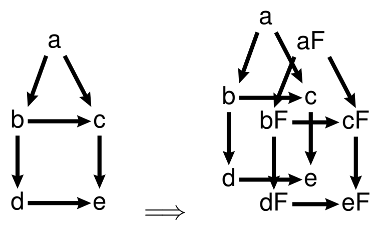
Um exemplo de funtor: digamos que $F : \mathbf{Set}_{\mathcal{A}} \Rightarrow \mathbf{Set}_{\mathcal{A}}$ leve cada tipo α em $\alpha^F = \alpha \wedge \alpha$ e cada função $f^{\alpha \to \beta}$ em $(f^F)^{\alpha^F \to \beta^F}$, onde essa f^F é definida como $f^F = \lambda p.\mathbf{P}(f(\pi_1 p))(f(\pi_2 p))$, ou, muito mais convenientemente, como $f^F(a_1, a_2) := (f(a_1), f(a_2))$[1]. É trivial verificar que esse F é mesmo um funtor.

Dois exemplos de não-funtores: defina os "pseudo-funtores" G e H que levam cada tipo α em $\alpha \wedge \mathbb{n}$ por $f^G(a, n) = (f(a), 2n)$ e $f^H(a, n) = (f(a), 0)$; G não obedece a condição sobre as composições e nem G nem H obedecem a condição sobre as identidades, daí nem G nem H são funtores.

A condição $(f \circ g)^F = (f^F \circ g^F)$ sobre funtores é imposta para fazer com que certo tipo de diagrama comute. Digamos que temos um funtor $F : \mathbf{C} \Rightarrow \mathbf{D}$ e que o diagrama E represente alguns objetos e morfismos da categoria $\mathbf{C}$. Forme o diagrama E' com duas cópias disjuntas de E, a segunda representando a imagem

[1] Essa definição mais curta está no mesmo estilo das definições de **S**, **K** e **I** na seção 2.4 e da definição do "flip" **F** na seção 2.8.

de cada objeto e cada morfismo de E pelo funtor F, como no exemplo abaixo:



então para cada seta $\alpha \to \beta$ bem-definida no diagrama E vamos ter $\alpha \to \beta$ e $\alpha^F \to \beta^F$ bem-definidas no diagrama E'; e, como corolário fácil disso, se E comuta então E' também comuta.

Outros exemplos de funtores: vamos descrever um certo funtor de $\mathbf{Set}_{\mathcal{A}}$ em $\mathbf{Set}_{\mathcal{A}}$, sem nome, que leva cada tipo α no tipo $i \to \alpha$, para um certo tipo i fixo. Cada morfismo $a \to b$ tem que ser levado num morfismo $(i \to a) \to (i \to b)$; como o espaço $\mathbf{E}_{i\to a}$ é o conjunto das funções que levam is em as, uma escolha natural para a ação desse funtor sobre os morfismos é ele levar cada função $i \to a$ na sua composta com $a \to b$ à direita, $i \to a \to b$. Essa ação pode ser descrita no sistema DN como a árvore abaixo:

$$\dfrac{\dfrac{\dfrac{\dfrac{[i]^1 \quad [i \to a]^2}{a} \qquad a \to b}{b}}{i \to b}\;1}{(i \to a) \to (i \to b)}\;2$$

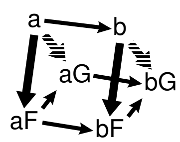
é bem fácil verificar que essa escolha para a ação sobre os morfismos dá um funtor.

É possível provar (por exemplo usando o teorema que aparece em [TS], pp.216–222) que essa é a única ação "natural" sobre morfismos, no sentido de que há uma construção natural que obtém o $(i \to a) \to (i \to b)$ só a partir de $a \to b$; daí num certo sentido esse é o único funtor "natural" que leva cada objeto $\mathbf{O}_\alpha$ em $\mathbf{O}_{i\to\alpha}$. Numa situação dessas, ou em situações em que o funtor fica claro a partir do contexto se dissermos como ele age sobre os objetos da categoria, vamos denotar o funtor só pela sua ação sobre os objetos: esse funtor passa a ser chamado de $\mathbf{O}_\alpha \Rightarrow \mathbf{O}_{i\to\alpha}$ ou de $\alpha \Rightarrow i \to \alpha$; o funtor "identidade", que leva cada objeto em si mesmo e cada morfismo em si mesmo, pode ser chamado de $\alpha \Rightarrow \alpha$; um funtor $F : \mathbf{C} \Rightarrow \mathbf{D}$ pode ser chamado de $\alpha \Rightarrow \alpha^F$, e com isso às vezes podemos nos dispensar de dar nomes para as categorias domínio e contradomínio do funtor F; é fácil verificar que a composta de dois funtores $\alpha \Rightarrow \alpha^F$ e $\beta \Rightarrow \beta^G$ é um funtor, e podemos chamar o resultado da composta de $\alpha \Rightarrow \alpha^{FG}$.

Às vezes vai ser mais conveniente usar uma outra notação curta para descrever funtores, em que só indicamos a categoria-domínio e a categoria-contradomínio. Alguns exemplos: $\mathbf{Set}_{\mathcal{A}} \Rightarrow \mathbf{Set}$ esquece as anotações de tipo e leva cada $\mathbf{O}_\alpha$ em $\mathbf{E}_\alpha$; $\mathbf{Vect}_{\mathbb{R}} \Rightarrow \mathbf{Set}$, onde $\mathbf{Vect}_{\mathbb{R}}$ é a categoria em que os objetos são os espaços vetoriais sobre $\mathbb{R}$ e os morfismos são as transformações lineares, esquece a estrutura de espaço vetorial de cada espaço e leva cada um no conjunto dos seus pontos; $\mathbf{Set} \Rightarrow \mathbf{Vect}_{\mathbb{R}}$ leva cada conjunto B no "espaço vetorial gerado por B", que é o em que os elementos de B fazem o papel de base.

4.5 Transformações naturais

Se F e G são dois funtores indo de $\mathbf{C}$ para $\mathbf{D}$, dizemos que T é uma *transformação natural de F em G* se T leva cada objeto $\mathbf{O}_\alpha$ de $\mathbf{C}$ num morfismo $\alpha^F \to \alpha^G$ de $\mathbf{D}$ de tal forma que o diagrama abaixo sempre comuta se $a \to b$ é o único objeto-base:



ou seja, para qualquer $a \xrightarrow{f} b$ em $\mathbf{C}$ sempre temos $a^F \xrightarrow{f^F} b^F \to b^G = a^F \to a^G \xrightarrow{f^G} b^G$, onde as setas sem nome são as obtidas a partir de $\mathbf{O}_a$ e $\mathbf{O}_b$ aplicando a transformação natural.

A condição $\alpha^F \to \alpha^G \to \beta^G = \alpha^F \to \beta^F \to \beta^G$ é imposta para que certos diagramas comutem, como no caso dos funtores. Tome um diagrama E numa categoria $\mathbf{C}$ e forme um diagrama E' com três cópias disjuntas de E, a "original", a imagem por F e a imagem por G, e ponha em E' também as setas obtidas pela transformação natural, i.e., uma seta $\alpha^F \to \alpha^G$ para cada objeto $\mathbf{O}_\alpha$ de $\mathbf{C}$ que aparece em E. Se o diagrama E comutava então E' vai comutar.

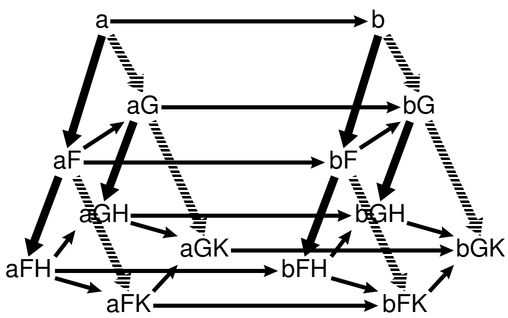
Vamos usar para transformações naturais uma notação curta parecida com a que usamos em funtores: se ficar claro quem é a seta $\alpha^F \to \alpha^G$ a partir de

$\mathbf{O}_\alpha$, $\alpha \Rightarrow \alpha^F$ e $\alpha \Rightarrow \alpha^G$ então vamos escrever a transformação natural que vai de F em G como $\alpha \xrightarrow{\bullet} (\alpha^F \to \alpha^G)$.

Existe uma transformação natural "identidade" que vai de um funtor nele mesmo: $\alpha \xrightarrow{\bullet} (\alpha^F \xrightarrow{\text{id}} \alpha^F)$. A composta de uma transformação natural $\alpha \xrightarrow{\bullet} (\alpha^F \to \alpha^G)$ com uma $\alpha \xrightarrow{\bullet} (\alpha^G \to \alpha^H)$ é uma transformação natural $\alpha \xrightarrow{\bullet} (\alpha^F \to \alpha^H)$, e há dois modos de compor transformações naturais e funtores, um em que o funtor é aplicado antes, nos objetos, e outro em que ele é aplicado depois, nos morfismos:

$$\dfrac{\dfrac{\mathbf{O}_\alpha}{\mathbf{O}_{\alpha^F}}\; F}{\alpha^{FH} \to \alpha^{FK}}\;\text{TN} \qquad \dfrac{\dfrac{\mathbf{O}_\alpha}{\alpha^F \to \alpha^G}\;\text{TN}}{\alpha^{FH} \to \alpha^{GH}}\; H$$

Um bom modo de entender tudo isso é passar direto para um caso muito mais complicado. Digamos que temos dois funtores de $\mathbf{C}$ em $\mathbf{D}$, F e G, dois funtores de $\mathbf{D}$ em $\mathbf{E}$, H e K, uma transformação natural de F em G e outra de H em K. As duas construções de uma seta $\alpha^{FH} \to \alpha^{GK}$ a partir de um objeto $\mathbf{O}_\alpha$ dão o mesmo resultado, e a operação que leva cada $\mathbf{O}_\alpha$ em $\alpha^{FH} \to \alpha^{GK}$ é uma transformação natural, que chamamos de $\alpha \xrightarrow{\bullet} (\alpha^{FH} \to \alpha^{GK})$ e que costuma ser chamada de *produto de Godement* de F, G, H e K. Para ver que tudo isto é verdade basta verificar que o diagrama abaixo comuta para qualquer $a \to b$, o que é bem fácil uma vez que se supere o choque inicial.



4.6 Iniciais e terminais

Um objeto $\mathbf{O}_a$ de uma categoria é dito *inicial* se para todo objeto $\mathbf{O}_x$ na mesma categoria houver exatamente uma seta $a \to x$. Dois objetos iniciais da mesma categoria são necessariamente isomorfos: se $\mathbf{O}_a$ e $\mathbf{O}_{a'}$ são iniciais existe uma única seta $a \to a'$, uma única $a' \to a$ e uma única $a \to a$, a identidade; daí $a \to a' \to a = a \stackrel{id}{\to} a$, e, similarmente, $a' \to a \to a' = a' \stackrel{id}{\to} a'$, e $a \to a'$ e $a' \to a$ são a inversa uma da outra.

Objetos *terminais* têm a definição dual à de objetos iniciais: um objeto $\mathbf{O}_t$ é terminal se para todo objeto $\mathbf{O}_a$ da categoria existe exatamente uma seta $a \to t$.

Em **Set** o conjunto vazio é o único inicial e os terminais são exatamente os singletons. Em $\mathbf{Set}_\mathcal{A}$ nós temos um objeto inicial preferido, $\mathbf{O}_\perp$, e um terminal preferido, $\mathbf{O}_\top$.

4.7 Produtos

Fixe dois objetos, $\mathbf{O}_a$ e $\mathbf{O}_b$. Um objeto $\mathbf{O}_p$, junto com duas setas $p \to a$ e $p \to b$, chamadas de *projeções*, é um produto de $\mathbf{O}_a$ e $\mathbf{O}_b$ se a seguinte coisa acontece:

$$\begin{tikzcd} x \arrow[r] \arrow[d] \arrow[dr] & p \arrow[dl] \arrow[d] \\ a & b \end{tikzcd}$$

para qualquer objeto $\mathbf{O}_x$ e para quaisquer setas $x \to a$ e $x \to b$ existe exatamente uma seta $x \to p$ que faz com que o diagrama acima comute, i.e., tal que $x \to a = x \to p \to a$ e $x \to b = x \to p \to b$; ou seja, há uma bijeção natural entre as setas $x \to p$ e os pares formados por uma seta $x \to a$ e uma $x \to b$; um lado da bijeção, $(x \to p) \to (x \to a,\, x \to b)$, é obtido por composição com as projeções, e o outro lado pela condição "$\exists!\, x \to p$".

Dois produtos para $\mathbf{O}_a$ e $\mathbf{O}_b$ são isomorfos: passe para a categoria em que os objetos são triplas formadas por um objeto, um morfismo indo desse objeto em $\mathbf{O}_a$ e outro indo desse objeto em $\mathbf{O}_b$, e em que um morfismo $(\mathbf{O}_p,\, p \to a,\, p \to b) \to (\mathbf{O}_q,\, q \to a,\, q \to b)$ é uma seta $p \to q$ tal que um diagrama como o acima comuta, i.e., tal que $p \to a = p \to q \to a$ e $p \to b = p \to q \to b$; nessa categoria os produtos de $\mathbf{O}_a$ e $\mathbf{O}_b$ são objetos terminais, e portanto são isomorfos. O diagrama abaixo deve deixar isso claro:

$$\begin{tikzcd} x \arrow[rr] \arrow[dr] \arrow[ddr] \arrow[ddrr] & & q \arrow[dl, shift left] \arrow[ddl] \arrow[dd] \\ & p \arrow[ur, shift left] \arrow[d] \arrow[dr] & \\ & a & b \end{tikzcd}$$

Se $(\mathbf{O}_p,\, p \to a,\, p \to b)$ e $(\mathbf{O}_q,\, q \to a,\, q \to b)$ são produtos para $\mathbf{O}_a$ e $\mathbf{O}_b$, então há uma única escolha razoável (i.e., fazendo tudo comutar) para as setas $p \to q$ e $q \to p$; elas vão ser necessariamente a inversa uma da outra. Para qualquer $(\mathbf{O}_x,\, x \to a,\, x \to b)$ há exatamente uma $x \to p$ e uma $x \to q$ razoáveis, e o diagrama acima comuta.

Nos diagramas acima a seta $x \to p$ é chamada de *a fatoração de* $(\mathbf{O}_x,\, x \to a,\, x \to b)$ *através de* $(\mathbf{O}_p,\, p \to a,\, p \to b)$. Nós vamos considerar que encontrar a fatoração através de um produto é uma "operação construtiva"; para nós uma operação é construtiva quando corresponde a uma construção natural, então o que falta é introduzir uma regra de construção que represente a operação de fatorar através de um produto. A regra seria algo como

$$\frac{x \to a \quad x \to b \quad (\mathbf{O}_p,\, p \to a,\, p \to b)}{x \to p} \quad ,$$

e teríamos que impôr alguma condição extra que garantisse que o $(\mathbf{O}_p,\, p \to a,\, p \to b)$ é realmente um produto; a situação fica mais fácil em $\mathbf{Set}_\mathcal{A}$, já que em $\mathbf{Set}_\mathcal{A}$ quaisquer dois objetos $\mathbf{O}_\beta$ e $\mathbf{O}_\gamma$ têm um produto natural, $(\mathbf{O}_{\beta\wedge\gamma},\, \beta\wedge\gamma \to \beta,\, \beta\wedge\gamma \to \gamma)$, onde as setas são as projeções nas coordenadas, que apareceram na seção 2.8. As regras ficam sendo essas:

$$\frac{\alpha \to \beta \quad \alpha \to \gamma}{\alpha \to \beta\wedge\gamma}\times \qquad \frac{}{\beta\wedge\gamma \to \beta}\pi \qquad \frac{}{\beta\wedge\gamma \to \gamma}\pi'$$

— já que o produto usa setas π_1 e π_2, que são consideradas como "naturais" (e nós vamos chamá-las de *projeções naturais*) nós introduzimos regras que as constróem a partir do nada. Se quiséssemos ser bem burocráticos poderíamos exigir que as regras π e π' tivessem um $\mathbf{O}_\beta$ e um $\mathbf{O}_\gamma$ acima da barra, mas não vamos fazer isso; estamos querendo chegar a um sistema de regras de construção, o sistema CCC (seção 4.9), em que as regras não envolvem objetos, só morfismos. A idéia nesse sistema não é que cada regra, por exemplo a π', é uma função π' que recebe os argumentos acima da barra (no caso, nenhum) e que retorna a seta embaixo da barra, $\beta\wedge\gamma \to \gamma$, já magicamente com o tipo certo; uma regra é uma operação que recebe os *valores* das setas acima da barra e mais o "nome" da seta abaixo da barra, i.e., o tipo dessa seta, e atribui um valor para a seta de baixo.

Dizemos que uma categoria *tem produtos binários*, ou *tem produtos finitos*, se para quaisquer dois objetos $\mathbf{O}_a$ e $\mathbf{O}_b$ dela existe uma tripla $(\mathbf{O}_p,\, p \to a,\, p \to b)$

que é um produto de $\mathbf{O}_a$ e $\mathbf{O}_b$. Em $\mathbf{Set}_{\mathcal{A}}$ nós temos uma situação privilegiada quanto a isso: não só cada produto binário existe, como temos uma operação que dá explicitamente para quaisquer objetos $\mathbf{O}_a$ e $\mathbf{O}_b$ um produto para eles; sem essa operação toda vez que precisássemos usar um produto numa construção teríamos que escolher um dos produtos possíveis. Isto vai ser mais discutido na seção 4.10.

Em $\mathbf{Set}_{\mathcal{A}}$ também temos *coprodutos binários*: coprodutos têm a definição dual à de produtos, e há uma bijeção natural entre os pares de setas $(a \to x,\ b \to x)$ e as setas $a \vee b \to x$; as setas que fazem o papel das "coprojeções" são as inclusões naturais $a \to a \vee b$ e $b \to a \vee b$, que correspondem às constantes κ_1 e κ_2 da seção 2.8. As regras para coprodutos vão ser as seguintes:

$$\frac{\alpha \to \gamma \quad \beta \to \gamma}{\alpha \vee \beta \to \gamma}\,\vee\mathrm{I}_{\mathrm{CCC}} \qquad \frac{}{\alpha \to \alpha \vee \beta}\,\kappa \qquad \frac{}{\beta \to \alpha \vee \beta}\,\kappa'$$

As regras para o produto em $\mathbf{Set}_{\mathcal{A}}$ obedecem as três equações abaixo, que correspondem às três equações sobre o $\wedge$ da seção 2.8:

$$\dfrac{\dfrac{\alpha \to \beta \quad \alpha \to \gamma}{\alpha \to \beta \wedge \gamma} \quad \dfrac{}{\beta \wedge \gamma \to \beta}}{\alpha \to \beta} \quad = \quad \alpha \to \beta$$

$$\dfrac{\dfrac{\alpha \to \beta \quad \alpha \to \gamma}{\alpha \to \beta \wedge \gamma} \quad \dfrac{}{\beta \wedge \gamma \to \gamma}}{\alpha \to \gamma} \quad = \quad \alpha \to \gamma$$

$$\dfrac{\dfrac{\alpha \to \beta \wedge \gamma}{\overline{\alpha \to \beta}} \quad \dfrac{\alpha \to \beta \wedge \gamma}{\overline{\alpha \to \gamma}}}{\alpha \to \beta \wedge \gamma} \quad = \quad \alpha \to \beta \wedge \gamma$$

onde as barras duplas são fazer a composição com as projeções naturais; as regras para o coproduto obedecem equações duais a essas. Não é muito difícil ver que essas equações implicam que $(\mathbf{O}_{\beta\wedge\gamma},\ \beta\wedge\gamma \to \beta,\ \beta\wedge\gamma \to \gamma)$ é um produto de $\mathbf{O}_\beta$ e $\mathbf{O}_\gamma$ e que as equações duais implicam que $(\mathbf{O}_{\beta\wedge\gamma},\ \beta \to \beta\vee\gamma,\ \gamma \to \beta\vee\gamma)$ é um coproduto de $\mathbf{O}_\beta$ e $\mathbf{O}_\gamma$; isso vai ser usado na seção 4.10, quando estivermos discutindo como usar essas mesmas regras fora de $\mathbf{Set}_{\mathcal{A}}$.

Essas regras nos permitem definir um funtor $\alpha \Rightarrow \alpha \wedge c$ de um modo ainda mais natural que o que usamos para definir o funtor $\alpha \Rightarrow i \to \alpha$ na seção 4.4: é possível obter a seta $\alpha \wedge c \to \beta \wedge c$ a partir de $\alpha \to \beta$ usando só regras "boas", que já avisamos que vão ser interpretáveis em outras categorias que não $\mathbf{Set}_{\mathcal{A}}$:

$$\dfrac{\dfrac{\overline{\alpha \wedge c \to \alpha} \quad \alpha \to \beta}{\alpha \wedge c \to \beta} \qquad \overline{\alpha \wedge c \to c}}{\alpha \wedge c \to \beta \wedge c}$$

4.8 Adjunções

Uma *adjunção* entre os funtores $\mathbf{C} \overset{R}{\Rightarrow} \mathbf{D}$ e $\mathbf{D} \overset{L}{\Rightarrow} \mathbf{C}$ é uma bijeção (sem nome por enquanto), para cada objeto $\mathbf{O}_a$ de $\mathbf{C}$ e cada objeto $\mathbf{O}_b$ de $\mathbf{D}$, entre os morfismos $a \to b^L$ (em $\mathbf{C}$) e os morfismos $a^R \to b$ (em $\mathbf{D}$),

$$\begin{tikzcd}
a \arrow[r, Rightarrow] \arrow[d] & aR \arrow[d] \\
bL \arrow[r, phantom, "\leftrightarrow" above=12pt] & b \arrow[l, Rightarrow]
\end{tikzcd}$$

que seja "natural"; nesse caso "natural" quer dizer que faz os dois diagramas abaixo comutarem:

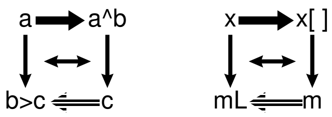
$$\begin{tikzcd}
a \arrow[r, Rightarrow] \arrow[d] & aR \arrow[d] \\
b \arrow[r, Rightarrow] \arrow[d] & bR \arrow[d] \\
cL & c \arrow[l, Rightarrow]
\end{tikzcd}
\qquad
\dfrac{\dfrac{a \to b}{a^R \to b^R} \quad b^R \to c}{a^R \to c} \quad = \quad \dfrac{\dfrac{a \to b \quad b \to c^L}{a \to c^L}}{a^R \to c}$$

$$\begin{tikzcd}
b \arrow[r, Rightarrow] \arrow[d] & bR \arrow[d] \\
cL \arrow[d] & c \arrow[l, Rightarrow] \arrow[d] \\
dL & d \arrow[l, Rightarrow]
\end{tikzcd}
\qquad
\dfrac{b \to c^L \quad \dfrac{c \to d}{c^L \to d^L}}{b \to d^L} \quad = \quad \dfrac{\dfrac{b^R \to c \quad c \to d}{b^R \to d}}{b \to d^L}$$

Vamos dizer que R é uma *adjunta à esquerda* de L, e que L é uma *adjunta à direita* de R. A notação vai ser $R \operatorname{adj} L$; essa notação não deixa claro qual é a bijeção, mas vamos considerar que toda vez que dizemos $R \operatorname{adj} L$ uma bijeção fica implícita.

O tipo de adjunção que é mais familiar para matemáticos é o em que R é um funtor de **Set** em outra categoria que leva cada conjunto numa certa estrutura livre gerada por ele, e L é o funtor de "esquecimento", que esquece a estrutura extra. Por exemplo, há uma adjunção entre os funtores **Set** $\Rightarrow$ $\mathbf{Vect}_{\mathbb{R}}$ e $\mathbf{Vect}_{\mathbb{R}}$ $\Rightarrow$ **Set**, outra entre **Set** $\Rightarrow$ **Grp** e **Grp** $\Rightarrow$ **Set**, onde **Grp** é a categoria dos grupos... a que vamos preferir para exemplos é a entre **Set** $\Rightarrow$ **Mon** e **Mon** $\Rightarrow$ **Set**, onde **Mon** é a categoria dos *monóides*; um monóide é um conjunto com uma operação '$\cdot$', que é associativa e tem um elemento neutro. Um modo de representar o monóide livre gerado por um conjunto A é considerá-lo como sendo o conjunto das seqüências ("listas") de comprimento finito ≥ 0 de elementos de A, com a operação $\cdot$ sendo a concatenação: $[a_1, a_2, a_3] \cdot [\,] \cdot [a_4, a_5] = [a_1, a_2, a_3, a_4, a_5]$; o elemento neutro é a lista de comprimento 0. O funtor R, que nesse caso vamos chamar de $[\,]$, leva cada função $f : A \to B$ na função $f^{[\,]} : A^{[\,]} \to B^{[\,]}$ que aplica f a cada elemento de uma lista: $f^{[\,]}([a_1, a_2]) = [f(a_1), f(a_2)]$. É fácil verificar que $f^{[\,]}(s) \cdot f^{[\,]}(s') = f^{[\,]}(s \cdot s')$ e portanto $f^{[\,]}$ é um morfismo de monóides. Note que há muitos monóides que não são a imagem pelo funtor $[\,]$ de nenhum objeto de **Set**; qualquer grupo, em particular, é um monóide.

A adjunção que vai nos interessar mais é entre dois funtores de $\mathbf{Set}_{\mathcal{A}}$ em $\mathbf{Set}_{\mathcal{A}}$; no lugar do funtor R temos o funtor $\alpha \Rightarrow \alpha \land b$ e no lugar do funtor L temos $\gamma \Rightarrow b \to \gamma$. Essa adjunção e a anterior estão representadas na figura abaixo.



Como em geral é complicado descrever bijeções como essas e garantir que elas obedecem as duas condições de naturalidade do início da seção nós vamos aprender um modo mais simples de obter adjunções.

Teorema: digamos que temos:

1) o funtor L,

2) "metade" do funtor R, i.e., só a sua ação sobre os objetos, e

3) uma função que leva cada $\mathbf{O}_a$ num morfismo $a \to a^{RL}$ e que obedeça a seguinte propriedade: para quaisquer objetos $\mathbf{O}_a$ de $\mathbf{C}$ e $\mathbf{O}_b$ de $\mathbf{D}$ a construção natural que leva setas $a^R \to b$ em setas $a \to b^L$ no diagrama abaixo é uma

bijeção.

$$\begin{tikzcd} aR \arrow[rr] \arrow[d, Rightarrow, dashed] & & b \arrow[d, Rightarrow, dashed] \\ aRL \arrow[rr] & & bL \\ & a \arrow[ul] \arrow[ur] & \end{tikzcd}$$

Então essa bijeção $(a \to b^L) \leftrightarrow (a^R \to b)$ é uma adjunção entre o funtor R e o funtor L, onde a "outra metade" do funtor R (a sua ação sobre os morfismos) é obtida por uma construção natural no diagrama abaixo:

$$\begin{tikzcd} aR \arrow[rr] \arrow[d, Rightarrow, dashed] & & bR \arrow[d, Rightarrow, dashed] & \\ aRL \arrow[rr] & & bRL & \\ & a \arrow[ul] \arrow[ur] \arrow[rr] & & b \arrow[ul] \end{tikzcd}$$

onde agora $\mathbf{O}_b$ é um objeto da categoria $\mathbf{C}$.

Observação: para que a operação $(a^R \to b) \to (a \to b^L)$ do primeiro diagrama seja uma bijeção é necessário (e suficiente!) que para qualquer $a \to b^L$ exista exatamente um $a^R \to b$ que seja levado nele; vamos escrever como

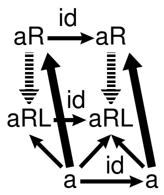
$$\frac{a \to b^L}{a^R \to b}$$

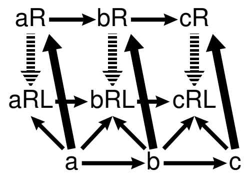
a operação que obtém o $a^R \to b$ que é levado num certo $a \to b^L$. Com essa operação podemos escrever a ação do funtor R sobre os morfismos como

$$\cfrac{a \to b \quad \cfrac{\cfrac{a \to b}{\mathbf{O}_b}}{b \to b^{RL}}}{\cfrac{a \to b^{RL}}{a^R \to b^R}} \quad .$$

Vamos chamar tanto a operação $(a^R \to b) \to (a \to b^L)$ quanto a sua inversa de *transposições*.

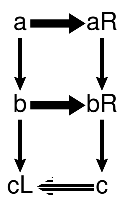
Demonstração do teorema: primeiro temos que ver que o R definido desse jeito é realmente um funtor. O diagrama abaixo à esquerda mostra que identidades são levadas em identidades, e, de brinde, que a transposta da seta $a \to a^{RL}$ é a identidade. O diagrama abaixo à direita mostra que R respeita a composição: a transposta de $a \to b \to c \to c^{RL}$ tem que ser a composta $a^R \to b^R \to c^R$, já que $a \to a^{RL} \to b^{RL} \to c^{RL} = a \to b \to c \to c^{RL}$ e sabemos que a $a^R \to c^R$ que satisfaz isso é única. Este tipo de argumento vai ser bastante usado daqui por diante.

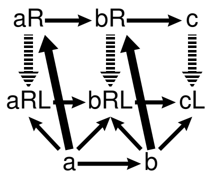




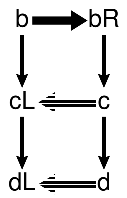
Note que os diagramas acima comutam no sistema de regras que inclui os funtores, as transposições e a regra que obtém $a \to a^{RL}$ a partir de $\mathbf{O}_a$ *se impusermos uma condição extra sobre as construções naturais*: que uma construção nesse sistema só seja considerada natural se os objetos das categorias $\mathbf{C}$ e $\mathbf{D}$ que aparecem na árvore forem só os que aparecem no diagrama. Nós precisamos disso para evitar morfismos apoiados em $a^{RLR}, a^{RLRL}, \ldots$; é possível mostrar por exemplo que em geral há dois morfismos naturais diferentes de tipo $a^{RL} \to a^{RLRL}$ mas que os morfismos de tipo $a^{(RL)^n} \to a^{RL}$ estão bem-definidos; isso envolve estudar a *mônada* gerada pela adjunção, o que não vai nos interessar.

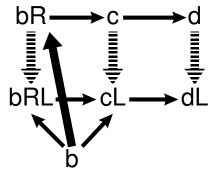
Agora podemos ver que a bijeção $(a \to b^L) \leftrightarrow (a^R \to b)$ é natural no sentido da seção anterior; para ver que ela obedece a condição à esquerda abaixo basta verificar que o diagrama abaixo à direita comuta:



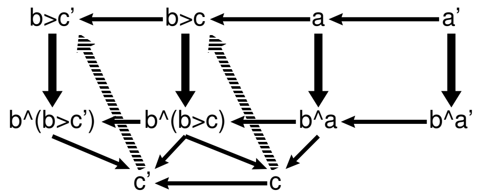


e a mesma coisa no par de diagramas abaixo. O argumento é o mesmo que usamos para demonstrar que o funtor R respeita a composição.





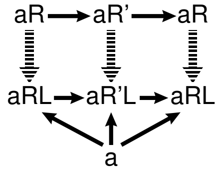
Pra deixar as idéias mais claras vamos ver como isso fica no caso da adjunção que nos interessa mais, a em que R é $\alpha \Rightarrow \alpha \to b$ e L é $\gamma \Rightarrow b \to \gamma$. O diagrama abaixo comuta:



Como nesse caso é mais natural usar a seta $a^{LR} \to a$ — que é $a \wedge (a \to b) \to b$ — do que a $a \to a^{RL}$ nós invertemos a direção dos morfismos e fizemos R e L trocarem de papel um com o outro pro diagrama ficar numa forma mais familiar.

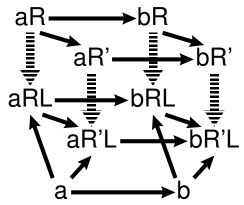
Teorema: se $R \operatorname{adj} L$ e $R' \operatorname{adj} L$ então existe uma transformação natural $\alpha \xrightarrow{\bullet} \alpha^R \leftrightarrow \alpha^{R'}$ que leva cada objeto $\mathbf{O}_\alpha$ num isomorfismo. Ou seja, adjuntas à esquerda para um certo funtor são essencialmente únicas; por dualidade, adjuntas à direita também.

Demonstração: nós conhecemos $a \to a^{RL}$ e $a \to a^{R'L}$ e sabemos, pelas suas propriedades, que existem uma única $a^R \to a^{R'}$ e uma única $a^{R'} \to a^R$ que fazem o diagrama abaixo comutar:



Também existe uma única $a^R \to a^R$ tal que $a \to a^{RL} \to a^{RL} = a^{RL}$; como a identidade $a^R \to a^R$ faz isso concluímos que $a^R \to a^{R'} \to a^R$ é a identidade, e, por um argumento similar trocando Rs e R's, que $a^{R'} \to a^R \to a^{R'}$ é a identidade; daí $\alpha \xrightarrow{\bullet} a^R \leftrightarrow a^{R'}$ realmente leva cada objeto num isomorfismo.

Falta mostrar que $\alpha \stackrel{\bullet}{\to} a^R \to a^{R'}$ é realmente uma transformação natural, ou seja, que o diagrama



comuta. É evidente que a parte de baixo comuta, e $a^R \to a^{R'} \to b^{R'} = a^R \to b^R \to b^{R'}$ porque só existe uma seta $a^R \to b^{R'}$ que é levada por L numa $a^{RL} \to b^{R'L}$ tal que $a \to a^{RL} \to b^{R'L} = a \to b^{R'L}$.

4.9 O sistema CCC

Uma *categoria cartesiana fechada* é uma categoria que tem um terminal, um inicial, produtos binários, coprodutos binários e em que cada funtor $\alpha \Rightarrow \alpha \wedge b$ tem uma adjunta à direita; vamos abreviar "categoria cartesiana fechada" como "CCC" (de *cartesian closed category*). Já vimos que $\mathbf{Set}_{\mathcal{A}}$ é uma CCC.

Uma *dedução no sistema* CCC é uma em que cada barra é uma aplicação de uma das regras abaixo:

$$\frac{}{\alpha \to \alpha}\ \mathrm{id} \qquad \frac{\alpha \to \beta \quad \beta \to \gamma}{\alpha \to \gamma}\ \circ$$

$$\frac{}{\alpha \to \top}\ \top_{\mathrm{CCC}} \qquad \frac{}{\bot \to \alpha}\ \bot_{\mathrm{CCC}}$$

$$\frac{\alpha \to \beta \quad \alpha \to \gamma}{\alpha \to \beta \wedge \gamma}\ \wedge\mathrm{I}_{\mathrm{CCC}} \qquad \frac{}{\alpha \wedge \beta \to \alpha}\ \pi \qquad \frac{}{\alpha \wedge \beta \to \beta}\ \pi'$$

$$\frac{\alpha \to \gamma \quad \beta \to \gamma}{\alpha \vee \beta \to \gamma}\ \vee\mathrm{I}_{\mathrm{CCC}} \qquad \frac{}{\alpha \to \alpha \vee \beta}\ \kappa \qquad \frac{}{\beta \to \alpha \vee \beta}\ \kappa'$$

$$\frac{\alpha \wedge \beta \to \gamma}{\alpha \to (\beta \to \gamma)}\ \mathrm{cur} \qquad \frac{}{\alpha \wedge (\alpha \to \beta) \to \beta}\ \mathrm{ev}$$

Uma dedução dessas pode ser interpretada funcionalmente como uma construção de uma seta com o tipo da raiz a partir de setas com os tipos das folhas da árvore; note que esse sistema não tem regras com descargas, então não precisamos falar de folhas descarregadas e folhas não-descarregadas.

Essas deduções estão sujeitas a certas “equações”, que dizem, como as reduções para dedução natural, que certas subárvores de uma dedução podem ser trocadas por outras equivalentes sem alterar o resultado final da construção. Nós já vimos todas essas equações, e é evidente que elas são obedecidas em $\mathbf{Set}_{\mathcal{A}}$:

A composição é associativa.

A identidade é o elemento neutro da composição.

Para cada tipo α existe exatamente uma seta de tipo $\alpha \to \top$ e exatamente uma de tipo $\bot \to \alpha$.

Há uma bijeção natural entre setas $\alpha \to \beta \land \gamma$ e pares de setas $\alpha \to \beta$; idem para setas $\alpha \lor \beta \to \gamma$ e pares de setas $\alpha \to \gamma$ e $\beta \to \gamma$.

A operação de “currying” (a regra ‘cur’) e a de “uncurrying” (que vamos explicar daqui a pouco) são, para cada tipo β, as duas transpostas de uma adjunta do funtor $\alpha \to \alpha \land \beta$, em que as setas ‘ev’ (“evaluation”) fazem o papel das setas $a^{LR} \to a$ da seção anterior. Basta exigir que a regra ‘cur’ seja a inversa da operação abaixo,

$$\dfrac{\dfrac{\dfrac{}{\alpha\land\beta\to\beta} \quad \dfrac{\dfrac{}{\alpha\land\beta\to\alpha} \quad \alpha\to(\beta\to\gamma)}{\alpha\land\beta\to(\beta\to\gamma)}}{\alpha\land\beta\to\beta\land(\beta\to\gamma)} \qquad \dfrac{}{\beta\land(\beta\to\gamma)\to\gamma}}{\alpha\land\beta\to\gamma}$$

que é a construção do uncurrying a partir do funtor $\alpha \to \alpha \land \beta$ e do ‘ev’.

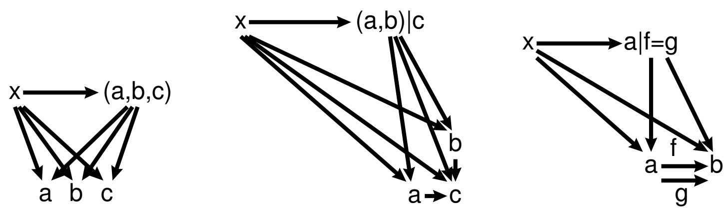
Vamos dizer que duas construções A e B (agora vistas como árvores, não só como o resultado) são CCC-*equivalentes* se uma pode ser obtida a partir da outra por uma série de operações em que trocamos uma subárvore de uma construção por outra árvore que é considerada igual a ela por uma das equações que descrevemos acima. Duas construções CCC-equivalentes dão o mesmo resultado em categorias cartesianas fechadas.

No próximo capítulo vamos ver que CCC-equivalência é essencialmente o mesmo que equivalência em DN, que por sua vez era a melhor noção de equivalência que tínhamos para estudar o λ-cálculo com tipos que admitissem as regras de formação $\land$, $\lor$, $\top$ e $\bot$; vamos ver também uma tradução entre construções em DN e construções em CCC que vai nos permitir interpretar λ-cálculo (com tipos com $\land$, $\lor$, $\top$ e $\bot$!) em qualquer categoria cartesiana fechada.

4.10 Limites

Vamos introduzir rapidamente alguns conceitos muito interessantes mas que só serão usados de novo no capítulo 8.

Em **Set** temos as seguintes situações muito parecidas com produtos:

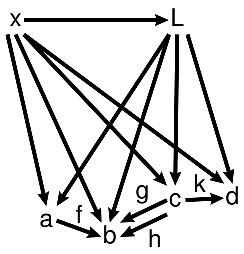


Primeira figura: há uma bijeção natural entre triplas de setas $x \to a$, $x \to b$ e $x \to c$ e setas que levam cada x numa tripla formada por um a, um b e um c; o espaço das triplas (a, b, c), junto com as projeções naturais, é um produto de $\mathbf{E}_a$, $\mathbf{E}_b$ e $\mathbf{E}_c$. Em $\mathbf{Set}_{\mathcal{A}}$ tínhamos nos restringido a produtos de dois objetos.

Segunda figura: há uma bijeção natural entre as triplas de setas $x \to a$, $x \to b$ e $x \to c$ que comutam com as setas $a \to c$ e $b \to c$, i.e., tais que $x \to c = x \to a \to c = x \to b \to c$, e as setas que vão de x para $\mathbf{E}_{(a,b)|_c}$, o espaço dos pares (a, b) tais que o a e o b são levados no mesmo c; as setas indo de $(a, b)|_c$ para a, b são as projeções naturais e a que vai para c é a composta $(a, b)|_c \to a \to c$ ou $(a, b)|_c \to b \to c$, tanto faz. Dizemos que o espaço $\mathbf{E}_{(a,b)|_c}$ junto com as projeções naturais dele em a e em b (costumamos descartar a que vai para c já que ela pode ser recuperada a partir das outras) é um *pullback* para $a \to c$ e $b \to c$.

Terceira figura: $\mathbf{E}_{a|_{f=g}}$ é o espaço dos 'a's que são levados no mesmo b por f e por g, i.e., os 'a's tais que $f(a) = g(a)$. Os pares de setas $x \to a$ e $x \to b$ tais que $x \to b = x \to a \xrightarrow{f} b = x \to a \xrightarrow{g} b$ estão em bijeção com as setas $x \to a|_{f=g}$. Dizemos que o espaço $\mathbf{E}_{a|_{f=g}}$, junto com a função de inclusão $a|_{f=g} \to a$, é um *equalizador* para f e g.

No caso geral nós temos uma classe de objetos da categoria, O_B, e uma classe de morfismos da categoria, M_B, tais que O_B contém os domínios e os contradomínios de todos os morfismos de M_B e talvez outros objetos. No exemplo da figura abaixo O_B é formado pelos objetos a, b, c e d e M_B é formado por f, g, h e k. Um *cone* sobre B (B é a *base* do cone) é um objeto, o "vértice" do cone, e uma seta indo dele para cada objeto da base, com a condição de que todos os $x \to \alpha$, onde α é um objeto da base, estão bem-definidos no diagrama formado pela base e pelo cone; essa condição é equivalente a exigir que para qualquer morfismo $\alpha \to \beta$ da base temos $x \to \alpha \to \beta = x \to \beta$.



Um *limite* para uma base B é um cone que é terminal na categoria dos cones sobre B, no sentido da seção 4.7; i.e., um cone com vértice L (digamos) tal que para qualquer outro cone existe exatamente um morfismo do vértice desse cone para L fazendo tudo comutar, onde "tudo comutar" só quer dizer que toda seta $x \to \alpha$, onde x é o vértice do cone e α é um objeto da base, está bem definida, mesmo com agora sendo permitido ir primeiro de x para L e só depois para a base.

Dizemos que um limite para uma base que só tem finitos objetos e finitos morfismos é um limite *finito*. Se uma categoria tem um objeto terminal, produtos binários e equalizadores então ela tem todos os limites finitos, i.e., qualquer base finita tem um limite. Vamos mostrar como construir o limite para a base do diagrama acima, e o leitor se encarrega de verificar todos os detalhes e de passar para o caso geral.

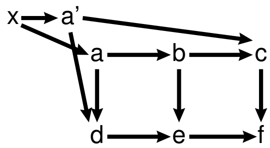
O espaço $\mathbf{E}_L$ vai ser um subconjunto do espaço $\mathbf{E}_{(a,b,c,d)}$, mais precisamente o conjunto das 4-uplas (a, b, c, d) tais que todos os modos de obter a, b, c e d a partir de (a, b, c, d) dêem o mesmo resultado. Basta exigir que $b = f(a) = g(c) = h(c)$ e que $d = k(c)$, onde a, b, c e d são obtidos a partir de (a, b, c, d) pelas projeções naturais; ou seja, o que queremos é que $(b, b, b, d) = (f(a), g(c), h(c), k(c))$, i.e., os (a, b, c, d) bons são os que são levados no mesmo ponto por duas funções $(a, b, c, d) \to (b, b, b, d)$; essas funções são exatamente $\lambda(a, b, c, d).(b, b, b, d)$ e $\lambda(a, b, c, d).(f(a), g(c), h(c), k(c))$, se quisermos usar a notação da seção 2.8. É possível montar espaços isomorfos a (a, b, c, d) e a (b, b, b, d) em $\mathbf{Set}_{\mathcal{A}}$ e as duas funções entre eles só usando as regras de produtos binários; o limite de B pode ser obtido tomando o equalizador dessas duas funções. Em certas situações pode ser necessário tomar um produto 0-ário, e é aí que entra o terminal: ele faz o papel de limite 0-ário, ou de limite para uma base vazia.

Conclusão, se acrescentamos ao sistema CCC uma regra que obtém equalizadores passamos a ser capazes de construir qualquer limite finito. Isso vai ser feito quando passarmos para topoi no capítulo 8.

Até agora toda vez que nós mostramos explicitamente um limite L em **Set** ou $\mathbf{Set}_{\mathcal{A}}$ para uma base B a justificativa para ele ser um limite era do tipo "é óbvio que existe exatamente uma função $x \to L$ adequada, já que cada ponto x só pode ir em um ponto L"; essa justificativa pode ser reformulada em termos mais categóricos. Dizemos que um objeto $\mathbf{O}_\alpha$ é um *gerador* para uma categoria se para quaisquer duas setas entre os mesmos objetos dessa categoria, $\beta \xrightarrow{f} \gamma$ e $\beta \xrightarrow{g} \gamma$, se $f \neq g$ há uma seta $\alpha \to \beta$ capaz de distingüí-las, i.e., tal que $\alpha \to \beta \xrightarrow{f} \gamma \neq \alpha \to \beta \xrightarrow{g} \gamma$; qualquer singleton é um gerador em **Set** ou em $\mathbf{Set}_{\mathcal{A}}$. Para verificar que uma seta $x \to L$ é a fatoração de um cone $x \to B$ (estamos usando um abuso de linguagem evidente) por $L \to B$ basta verificar que para toda seta $1 \to x$ toda seta $1 \to \alpha$ está bem-definida, onde $\mathbf{O}_1$ é um gerador e os α são os objetos da base.

Em algumas categorias que vão nos interessar mais tarde não temos um objeto gerador, mas temos uma *família de geradores*, uma classe de objetos tais que se dois morfismos $\alpha \to \beta$ são diferentes então existe um morfismo indo de um dos geradores da família em α que distingue esses dois morfismos. É evidente que a classe de todos os objetos de uma categoria é uma família de geradores, mas os casos interessantes vão ser quando as famílias de geradores forem pequenas.

Muitas demonstrações em topoi ou em categorias — não as que vamos dar, que vão ser pouquíssimas, mas as da literatura sobre o assunto — envolvem provar que dois objetos são isomorfos mostrando que eles são limites sobre a mesma base. Por exemplo, para mostrar que o retângulo formado por dois pullbacks colados é um pullback,



podemos mostrar que existe um isomorfismo natural entre a e a' no diagrama acima, onde a' é um pullback para $c \to f$ e $d \to f$; para isso usamos uma seqüência de bijeções naturais. Escrevendo como $x \to (\alpha, \dots, \omega)$ cones cuja base é formada pelos objetos $\alpha, \dots, \omega$ e pelos morfismos entre eles herdados do diagrama acima, temos a seguinte seqüência de bijeções:

$$\begin{array}{c}
x \to a' \\ \hline
x \to (c, d, f) \\ \hline
x \to (c, d, e, f) \\ \hline
x \to (c, b, d, e, f) \quad * \\ \hline
x \to (a, c, b, d, e, f) \\ \hline
x \to a
\end{array}$$

onde a bijeção da barra marcada com ‘$*$’ é um pouco mais difícil de verificar que as outras; a partir daí encontramos o isomorfismo entre a e a'.

Chapter 5

O teorema da dedução

5.1 As regras T

Para mostrarmos que o sistema DN e o sistema CCC são equivalentes nós vamos precisar aprender a interpretar barras duplas como barras normais. Digamos que A seja uma árvore que obtém β a partir de $\alpha_A, \ldots, \alpha_K$; vai ser melhor usar letras maiúsculas ao invés de números nos índices dessas hipóteses. Por exemplo, seja A a árvore abaixo,

$$\dfrac{\dfrac{x \wedge y}{y} \qquad \dfrac{\dfrac{x \wedge y}{x} \quad x \to (y \to z)}{y \to z}}{z} \qquad ,$$

uma dedução em DN de $\beta = z$ a partir de $\alpha_A = x \wedge y$ e $\alpha_B = x \to (y \to x)$; cada folha não-descarregada de A está associada a uma das letras maiúsculas usadas como índices, embora não tenhamos indicado isso na árvore.

Então a última barra dupla na árvore abaixo à esquerda pode ser entendida como uma "aplicação do teorema em DN provado por A"; nós formalizamos isso dizendo que ela é uma aplicação da regra $\mathrm{T}^{\mathsf{A}}_{\mathrm{DN}}$:

$$\begin{array}{c} \begin{array}{c} V \\ \hline\hline x \wedge y \end{array} \quad \begin{array}{c} W \\ \hline\hline x \to (y \to z) \end{array} \\ \hline\hline z \end{array} \qquad\qquad \begin{array}{c} \begin{array}{c} V \\ \hline\hline x \wedge y \end{array} \quad \begin{array}{c} W \\ \hline\hline x \to (y \to z) \end{array} \\ \hline z \end{array} \; \mathrm{T}^{\mathsf{A}}_{\mathrm{DN}}$$

e isso é só uma abreviatura para a árvore abaixo, em que a última barra é uma "aplicação de um teorema em DN dado explicitamente"; a regra nesse caso é $\mathrm{Te}_{\mathrm{DN}}$.

$$\dfrac{\dfrac{V}{x \land y} \quad \dfrac{W}{x \to (y \to z)} \quad \dfrac{\dfrac{[x \land y]^{1A}}{y} \quad \dfrac{\dfrac{[x \land y]^{1A}}{x} \quad [x \to (y \to z)]^{1B}}{y \to z}}{z}}{z}\ 1 : \mathrm{Te}_{\mathrm{DN}}$$

Vamos ter uma regra de redução para eliminar barras $\mathrm{Te}_{\mathrm{DN}}$; ela diz que uma aplicação de uma regra $\mathrm{Te}_{\mathrm{DN}}$ pode ser substituída por uma dedução de A com as deduções dos $\alpha_A, \ldots, \alpha_K$ grudadas nas folhas correspondentes de A. A árvore acima pode ser reduzida, por uma aplicação dessa regra, a essa árvore aqui:

$$\dfrac{\dfrac{\dfrac{V}{x \land y}}{y} \quad \dfrac{\dfrac{\dfrac{V}{x \land y}}{x} \quad \dfrac{W}{x \to (y \to z)}}{y \to z}}{z}$$

Uma dedução em que cada barra é ou uma aplicação de uma das regras do sistema DN ou uma aplicação de uma regra T_{DN} (i.e., $\mathrm{Te}_{\mathrm{DN}}$ ou, se estivermos numa árvore escrita com abreviações, $\mathrm{T}^{\mathsf{A}}_{\mathrm{DN}}$, onde A é uma dedução em DN) é considerada uma *dedução no sistema* DN+T_{DN}. Como consideramos, por um pequeno abuso de linguagem, que um sistema dedutivo é dado pelas suas regras de dedução, vamos considerar que o *sistema* DN+T_{DN} é dado pelas regras de DN e mais pela regra T_{DN}.

5.2 Traduções

Cada dedução em DN já é uma dedução em DN+T_{DN}; uma dedução em DN+T_{DN} pode ser traduzida para uma dedução em DN aplicando a regra de redução a cada barra T_{DN}; é fácil ver que a ordem das reduções não importa e que o resultado final (uma dedução em DN) é sempre o mesmo. Isso nos dá uma *função de tradução* indo de deduções em DN+T_{DN} em deduções em DN, e já tínhamos uma tradução DN $\to$ DN+T_{DN}. A composta DN $\to$ DN+T_{DN} $\to$ DN é a identidade, mas a composta DN+T_{DN} $\to$ DN $\to$ DN+T_{DN} só age como a identidade para as deduções que já estavam em DN.

Da mesma forma que definimos as regras T_{DN} podemos definir as regras $\mathrm{T}_{\mathrm{CCC}}$, com a diferença de que os "teoremas" aplicados pelas regras $\mathrm{T}_{\mathrm{CCC}}$ têm que ser dados por deduções no sistema CCC. Também definimos o sistema DN+$\mathrm{T}_{\mathrm{CCC}}$ e as traduções CCC $\to$ DN+$\mathrm{T}_{\mathrm{CCC}}$ e DN+$\mathrm{T}_{\mathrm{CCC}}$ $\to$ CCC do modo natural.

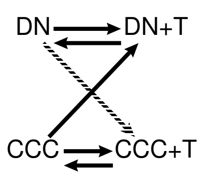
Agora podemos definir uma tradução de CCC para DN. Cada barra das que aparecem na tabela das regras do sistema CCC na seção 4.9 pode ser entendida como uma certa regra T_{DN}; por exemplo,

$$\dfrac{\alpha\land\beta\to\gamma}{\alpha\to(\beta\to\gamma)}\,\mathrm{cur} \quad\Longrightarrow\quad \dfrac{\alpha\land\beta\to\gamma \qquad \dfrac{\dfrac{\dfrac{\dfrac{[\alpha]^2\quad[\beta]^1}{\alpha\land\beta}\qquad[\alpha\land\beta\to\gamma]^{3A}}{c}}{\beta\to\gamma}\,1}{\alpha\to(\beta\to\gamma)}\,2}{\alpha\to(\beta\to\gamma)}\,3:\mathrm{Te}_{\mathrm{DN}} \quad ;$$

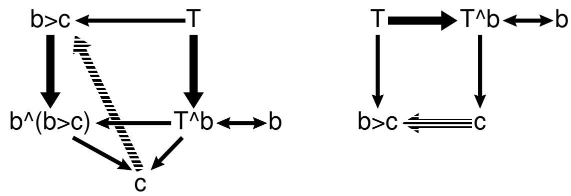
fixe a dedução em DN associada a cada regra de CCC, por exemplo escolhendo um cada caso a única dedução adequada que está em forma normal, e temos uma tradução bem-definida indo de CCC em DN+T_{DN}. Compondo-a com a tradução DN+T_{DN} $\to$ DN obtemos uma função de tradução CCC $\to$ DN; é fácil, embora um pouco trabalhoso, verificar que duas construções equivalentes (i.e., CCC-equivalentes, como definimos na seção 4.9) para um morfismo em CCC são levadas em duas deduções em DN que correspondem a termos equivalentes.

Como temos um modo de normalizar deduções em DN temos um algoritmo que diz se duas construções em CCC são CCC-equivalentes, mesmo sem nunca termos falado de normalização em CCC: passe as duas construções para DN, normalize e compare o resultado. Na verdade falta verificar uma coisa para ver que esse algoritmo funciona: falta ver que duas construções em CCC só são levadas em construções equivalentes em DN se forem CCC-equivalentes; vamos ver na próxima seção que isso é verdade.

Já temos traduções entre DN e DN+T_{DN}, entre CCC e DN+$\mathrm{T}_{\mathrm{CCC}}$ e uma extra indo de CCC em DN+T_{DN}. Parece que deveria dar para construir a tradução que falta, DN $\to$ DN+$\mathrm{T}_{\mathrm{CCC}}$,



de um modo parecido com como constuímos a tradução CCC $\to DNT$, mas há dois problemas: primeiro, que todos os tipos que aparecem numa dedução em CCC têm o '$\to$' como conectivo central, já que eles são morfismos; seria preciso transformar todo tipo num tipo da forma $\alpha \to \beta$. Há um modo óbvio de fazer isso: há um isomorfismo natural entre setas de tipo $\alpha \to \beta$ e setas de tipo $\top \to (\alpha \to \beta)$, que pode ser obtido em CCC a partir do diagrama abaixo;



trocamos αs por bs e βs por cs para deixá-lo com um aspecto mais familiar, e o isomorfismo natural entre b e $\top \wedge b$ é bastante evidente (veja a seção 4.10). Fazendo essa tradução $\alpha \Rightarrow \top \to \alpha$ em cada nó de uma dedução em DN nós *quase* obtemos uma dedução em DN+$\mathrm{T_{CCC}}$: todas as regras de DN podem ser vistas como regras $\mathrm{T_{CCC}}$, exceto pelas duas regras que envolvem descargas, $\vee$E e $\to$I; nós ainda não fazemos a menor idéia de como interpretar em DN+$\mathrm{T_{CCC}}$ as descargas que vêm da dedução em DN, e o que sabemos sobre eliminar as descargas das regras $\mathrm{T_{CCC}}$ não ajuda muito.

Podemos melhorar a situação um pouco lembrando que cada ocorrência da regra $\vee$E pode ser substituída por uma ocorrência da regra $\vee$E$'$ e vice-versa, como vimos na seção 3.1; é fácil montar traduções DN $\to$ DN$'$ e DN$'$ $\to$ DN, e o sistema DN$'$ só tem uma regra ruim, $\to$I, já que $\vee$E$'$ não envolve descargas.

Um exemplo simples dessa tradução DN$'-\{\to\mathrm{I}\} \to$ DN+$\mathrm{T_{CCC}}$ funcionando:

$$\dfrac{\dfrac{a\wedge b}{b} \qquad \dfrac{a\wedge b}{a}}{b\wedge a} \qquad \Longrightarrow \qquad \dfrac{\dfrac{\top\to a\wedge b}{\top\to b}\,\mathrm{T_{CCC}} \qquad \dfrac{\top\to a\wedge b}{\top\to a}\,\mathrm{T_{CCC}}}{\top\to b\wedge a}\,\mathrm{T_{CCC}}$$

5.3 O teorema da dedução

Para podermos lidar com as descargas nas regras $\to$I na tradução DN$'$ $\to$ CCC vamos ter que consertar a tradução que tínhamos e que ia de DN$'-\{\to\mathrm{I}\}$ para DN+$\mathrm{T_{CCC}}$ pondo '$\top \to$' na frente de cada tipo. Vamos passar a escrever a seta do '$\top \to$' como '$\Rightarrow$' para ressaltar que ela é diferente das outras e vamos introduzir um símbolo novo, '$\bigwedge$', que só aparece à esquerda do '$\Rightarrow$'. O algoritmo para traduzir uma dedução em DN$'$ para uma CCC vai ter duas partes: na primeira ele usa uma tradução parecida com a dos '$\top\ \to$'s para obter uma árvore em que os tipos estão escritos com '$\bigwedge$'s e '$\Rightarrow$'s; na segunda parte essa árvore intermediária é consertada e vira uma dedução em CCC.

Vamos dizer que um nó de uma árvore *depende* de uma certa folha descarregada da árvore se esse nó está entre essa folha e a barra em que ela é descarregada.

Na primeira parte da tradução faça tudo igual a antes, mas ao invés de '$\top\ \to$' ponha em cada nó '$\alpha_1 \bigwedge \ldots \bigwedge \alpha_n\ \Rightarrow$', se $\alpha_1, \ldots, \alpha_n$ é uma lista, *sem repetições*, dos tipos que aparecem nas folhas descarregadas das quais esse nó depende; se o nó não depende de nenhuma folha descarregada use $\top$ ao invés de $\alpha_1 \bigwedge \ldots \bigwedge \alpha_n\ \Rightarrow$, e além disso ponha uma barra acima de cada nó que corresponde a uma folha descarregada da árvore original. É mais fácil entender isso seguindo o exemplo abaixo, em que a dedução em DN$'$ é uma demonstração de que $((a \to b) \wedge (b \to c)) \to (a \to c)$ é um teorema; como ela iria ficar grande demais nós escrevemos $a \to b$ como ab, $b \to c$ como bc e $(a \to b) \wedge (b \to c)$ como abc.

$$\dfrac{\dfrac{\dfrac{\dfrac{[a]^1 \quad \dfrac{[abc]^2}{ab}}{b} \qquad \dfrac{[abc]^2}{bc}}{c}}{ac}\,{\scriptstyle 1}}{abc \to ac}\,{\scriptstyle 2} \qquad \Longrightarrow \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a_1 \Rightarrow a} \quad \dfrac{\dfrac{}{abc_2 \Rightarrow abc}}{abc_2 \Rightarrow ab}}{a_1 \bigwedge abc_2 \Rightarrow b} \qquad \dfrac{\dfrac{}{abc_2 \Rightarrow abc}}{abc_2 \Rightarrow bc}}{a_1 \bigwedge abc_2 \Rightarrow c}}{abc_2 \Rightarrow ac}\,{\scriptstyle 1}}{\top \Rightarrow abc \to ac}\,{\scriptstyle 2}$$

Repare que cada tipo à esquerda do '$\Rightarrow$' na árvore da direita ganhou uma anotação em subescrito que diz a que barra de descarga ele está associado; dois tipos com subescritos diferentes são considerados diferentes e devem aparecem separados na lista à esquerda do '$\bigwedge$'. Isto não é muito importante enquanto estivermos vendo os tipos só como proposições, mas vai ser fundamental para fazer com que essa tradução funcione também na interpretação funcional.

Agora a segunda parte da tradução. Pegue a árvore intermediária e interprete '$\bigwedge$'s como '$\wedge$'s e '$\Rightarrow$'s como '$\to$'s; é preciso pôr parênteses de alguma forma nas expressões '$\alpha_1 \bigwedge \ldots \bigwedge \alpha_n$'; escolha qualquer um dos modos possíveis. Agora interprete cada barra como uma regra $\mathrm{T}_{\mathrm{CCC}}$. À primeira vista parece que pode haver muitas deduções em CCC não equivalentes que servem como o teorema escondido, mas isso não é verdade: se tomarmos o cuidado de respeitar os números em subescrito e de usar no lado direito dos '$\Rightarrow$' as mesmas regras naturais que usamos na tradução com os '$\top \to$'s a tradução funciona. Por exemplo, em

$$\frac{a_1 \wedge a_2 \Rightarrow b \quad a_2 \wedge a_3 \Rightarrow b \to c}{a_1 \wedge a_2 \wedge a_3 \Rightarrow c}$$

só temos uma "projeção" natural que a partir de $a_1 \wedge a_2 \wedge a_3$ monta o par com a_1 e a_2 e só uma que monta o par com a_2 e a_3. Compondo essas projeções com $a_1 \wedge a_2 \Rightarrow b$ e com $a_2 \wedge a_3 \Rightarrow b \to c$ obtemos

$$\frac{a_1 \wedge a_2 \wedge a_3 \Rightarrow b \quad a_1 \wedge a_2 \wedge a_3 \Rightarrow b \to c}{a_1 \wedge a_2 \wedge a_3 \Rightarrow c} \quad ,$$

que pode ser tratado da mesma maneira que tratamos o caso com '$\top \to$'s, que era

$$\frac{\dfrac{\top \to b \quad \top \to (b \to c)}{\top \to (b \wedge (b \to c))} \quad \dfrac{}{b \wedge (b \to c) \xrightarrow{\text{ev}} c}}{\top \to b} \quad ;$$

substitua cada $\top$ por $a_1 \wedge a_2 \wedge a_3$ e pronto. Quanto às barras que correspondem às aplicações de $\to$I com descargas, o que elas fazem é passar o α_i com o índice certo para o lado direito do '$\Rightarrow$', como na operação de currying.

Chapter 6

A álgebra dos valores de verdade

6.1 Valuações

Digamos que $A, B, C \ldots$ sejam as nossas *proposições atômicas* e $\mathbb{P}$ seja o conjunto das proposições formadas a partir dessas, $\top$, $\bot$, $\wedge$, $\vee$ e $\to$. Seja Ω o conjunto dos *valores de verdade* do sistema, que por enquanto não tem nenhuma estrutura especial. O conjunto das proposições atômicas vai ser chamado de $\mathbb{P}_{\text{atom}}$.

Uma *valuação* em $\mathbb{P}$ é uma função $[\cdot] : \mathbb{P} \to \Omega$ que obedeça uma condição que diz que "tudo que é derivável a partir de hipóteses verdadeiras é verdadeiro"; formalmente, se existe uma derivação de β com $\alpha_1, \ldots, \alpha_n$ ($n \geq 0$) como hipóteses e $[\alpha_1] = [\top]$, …, $[\alpha_n] = [\top]$, então $[\beta] = [\top]$.

Repare que se β é um teorema intuicionista então $[\beta] = [\top]$ em qualquer valuação, já que existe uma derivação de β a partir de um conjunto vazio de hipóteses.

Vamos definir o *conjunto de verdades* de uma valuação como sendo $\{\, \alpha \in \mathbb{P} \mid [\alpha] = [\top] \,\}$.

Exemplos de valuações (com $\Omega = \{\, \top, \dashv \,\}$):

1) a que diz que "toda proposição é verdade": $[\alpha] = \top$ para todo α.

2) A que associa $\top$ às proposições que são teoremas intuicionistas e $\dashv$ às outras.

3) Se $V : \mathbb{P}_{\text{atom}} \to \{\, \top, \dashv \,\}$ é uma atribuição qualquer de verdadeiros e falsos

para as variáveis atômicas, a extensão natural dele para $\mathbb{P}$, usando $[\top] = \top$, $[\bot] = \dashv$ e as tabelas clássicas para $\wedge$, $\vee$ e $\to$, que vamos chamar de $[\cdot]_V$, é uma valuação. Os $[\cdot]$s que são $[\cdot]_V$s para alguma V são chamados de *valuações clássicas*.

4) Para $S \subset \mathbb{P}$ defina $\mathrm{conseq}(S)$ como sendo o conjunto das proposições que são deriváveis usando só elementos de S como hipóteses. A função que associa $\top$ às proposições em $\mathrm{conseq}(S)$ e $\dashv$ às outras é uma valuação, a valuação "menos verdadeira" tornando os membros de S verdadeiros.

6.2 Valuações algébricas

Definição: uma valuação é *algébrica* se para $\star = \wedge, \vee, \to$ o valor de $[\alpha \star b]$ só depende dos valores de $[\alpha]$ e $[\beta]$; formalmente, se $([\alpha] = [\alpha']$ e $[\beta] = [\beta'])$ implica em $[\alpha \star \beta] = [\alpha' \star \beta']$. A idéia é que nas valuações algébricas as operações $\wedge$, $\vee$, $\to$ são dadas por funções $\Omega^2 \to \Omega$, e a partir delas e de $[\top]$ e $[\bot]$ a gente reconstrói a valuação inteira a partir dos valores das variáveis atômicas; aliás, a partir de $[\top]$, $[\bot]$ e de operações $\wedge, \vee, \to$ "boas" qualquer escolha de valores para as variáveis atômicas se estende a uma valuação algébrica.

Definição: uma *álgebra de valores de verdade* é uma 7-upla $(\Omega, \top, \bot, \wedge, \vee, \to)$, onde $\top, \bot \in \Omega$ e $\wedge, \vee, \to: \Omega^2 \to \Omega$, que sempre induz valuações algébricas; isto é, tal que para toda escolha de $\mathbb{P}_{\mathrm{atom}}$ e para qualquer escolha de valores para as variáveis atômicas, $\varphi : \mathbb{P}_{\mathrm{atom}} \to \Omega$, a extensão natural de φ para uma função $\mathbb{P} \to \Omega$ é uma valuação algébrica. É bem fácil ver que para qualquer valuação algébrica tal que a sua $\mathbb{P} \to \Omega$ seja sobrejetiva a 7-upla $(\Omega, [\top], [\bot], \wedge, \vee, \to)$, onde $\wedge$, $\vee$ e $\to$ são os induzidos pela valuação, é uma álgebra de valores de verdade; além disso temos um jeito natural de gerar uma valuação algébrica a partir de uma álgebra de valores de verdade: basta tomar $\mathbb{P}_{\mathrm{atom}} = \Omega$.

6.3 Valuações induzem valuações algébricas

Um fato surpreendente é que para qualquer valuação V existe uma valuação algébrica V', única módulo certas tecnicalidades, que tem o mesmo conjunto de verdades que V; vamos demonstrar isso a seguir.

Para $\alpha, \beta \in \mathbb{P}$ vamos dizer que $\alpha \le \beta$ ("α é tão ou menos verdadeiro quanto β") quando $[\alpha \to \beta] = [\top]$. A relação $\le$ é reflexiva porque $\alpha \to \alpha$ é sempre um teorema intuicionista, e é transitiva porque porque se $[\alpha \to \beta] = [\beta \to \gamma] = [\top]$ então $[\alpha \to \gamma] = [\top]$:

$$\dfrac{[\alpha]^1}{\alpha \to \alpha}\,1 \qquad\qquad \dfrac{\dfrac{\dfrac{[\alpha]^1 \quad \alpha \to \beta}{\beta} \qquad \beta \to \gamma}{\gamma}}{\alpha \to \gamma}\,1$$

Isso gera uma relação de equivalência '$\sim$' entre os membros de $\mathbb{P}$, em que $\alpha \sim \beta$ se e só se $\alpha \le \beta$ e $\beta \le \alpha$), ou, equivalentemente, se $[(\alpha \to \beta) \land (\beta \to \alpha)] = [\top]$; vamos abreviar $(\alpha \to \beta) \land (\beta \to \alpha)$ como $\alpha \leftrightarrow \beta$.

Vamos definir $[\cdot]' : \mathbb{P} \to \mathcal{P}(\mathbb{P})$ por $[\alpha]' = \{\, \beta \mid \beta \sim \alpha \,\}$; esse $[\cdot]'$ vai dar a valuação algébrica que queremos. Primeiro vamos mostrar que $[\cdot]'$ é uma valuação algébrica:

1) $[\cdot]'$ é uma valuação: basta ver que o conjunto de verdades de $[\cdot]'$ é o mesmo que o de $[\cdot]$. Uma proposição α é verdadeira por $[\cdot]'$, i.e., $[\alpha]' = [\top]'$, se e só se $[\alpha \to \top] = [\top]$ e $[\top \to \alpha] = [\top]$, ou seja, se $\alpha \to \top$ e $\top \to \alpha$ estão ambas no conjunto de verdades de $[\cdot]$; mas $\alpha \to \top$ é um teorema intuicionista e $\top \to \alpha$ é equivalente a α:

$$\dfrac{\overline{\top}}{\alpha \to \top}\,1 \qquad \dfrac{\alpha}{\top \to \alpha}\,1 \qquad \dfrac{\overline{\top} \quad \top \to \alpha}{\alpha}$$

2) $[\cdot]'$ é algébrica. Para isso é preciso provar que se $[\alpha_1]' = [\alpha_2]'$ e $[\beta_1]' = [\beta_2]'$ então $[\alpha_1 \star \beta_1]' = [\alpha_2 \star \beta_2]'$, para $\star = \land, \lor, \to$, ou seja, que se $\alpha_1 \leftrightarrow \alpha_2$ e $\beta_1 \leftrightarrow \beta_2$ são verdade em $[\cdot]$ então os $(\alpha_1 \star \beta_1) \leftrightarrow (\alpha_2 \star \beta_2)$ são verdade em $[\cdot]$ para $\star = \land, \lor, \to$. As demonstrações são fáceis mas são diferentes para cada conectivo. A título de exemplo, o cerne da demonstração para o $\lor$ é

$$\dfrac{\dfrac{[\alpha_1 \lor \beta_1]^2 \qquad \dfrac{\dfrac{[\alpha_1]^1 \quad \alpha_1 \to \alpha_2}{\alpha_2}}{\alpha_2 \lor \beta_2} \qquad \dfrac{\dfrac{[\beta_1]^1 \quad \beta_1 \to \beta_2}{\beta_2}}{\alpha_2 \lor \beta_2}}{\alpha_2 \lor \beta_2}\,1}{\alpha_1 \lor \beta_1 \to \alpha_2 \lor \beta_2}\,2 \quad .$$

Agora queremos ver que num certo sentido $[\cdot]'$ é a única valuação algébrica compatível com $[\cdot]$. Já que não interessa o caso em que uma $[\cdot]'$ tem valores de verdade que não são assumidos por nenhuma proposição de $\mathbb{P}$ (porque não poderíamos dizer nada sobre esses valores de verdade só a partir do que sabemos sobre $[\cdot]$), considere que os valores de verdade de $[\cdot]'$ são os conjuntos de proposições que têm o mesmo $[\cdot]'$-valor; vamos chamar o conjunto desses valores de $\mathbb{P}_{/[\cdot]'}$, o "quociente de $\mathbb{P}$ por $[\cdot]'$". Da mesma forma vamos definir $\mathbb{P}_{/\leftrightarrow}$ como o quociente de $\mathbb{P}$ pela relação que diz que α e β são equivalentes se $[\alpha \leftrightarrow \beta] = [\top]$, ou $[\alpha \leftrightarrow \beta]' = [\top]'$, tanto faz.

Para simplificar um pouco vamos esquecer $[\cdot]'$ e passar a só discutir uma valuação algébrica $[\cdot]$, para ver em que sentido ela pode ser recuperada a partir dos seus valores de verdade.

Digamos que α e α' estejam na mesma $\mathbb{P}_{/[\cdot]}$-classe e β e β' estejam na mesma $\mathbb{P}_{/[\cdot]}$-classe. Se $\alpha \leftrightarrow \beta$ for verdade mas $\alpha' \leftrightarrow \beta'$ não, vamos ter $[(\alpha \to \beta) \land (\beta \to \alpha)] = [\top]$ e $[(\alpha' \to \beta') \land (\beta' \to \alpha')] \neq [\top]$, o que é um absurdo já que a valuação é algébrica. Considere o caso particular em que α, α', β e β' estão todos na mesma $\mathbb{P}_{/[\cdot]}$-classe e em que $\alpha = \beta$; como $\alpha \leftrightarrow \alpha$ é um teorema então $\alpha \leftrightarrow \beta$ é verdade, e daí $\alpha' \leftrightarrow \beta'$ é verdade. Conclusão: numa valuação algébrica elementos da mesma $\mathbb{P}_{/[\cdot]}$-classe estão na mesma $\mathbb{P}_{/\leftrightarrow}$-classe, i.e., cada $\mathbb{P}_{/[\cdot]}$-classe está contida dentro de uma $\mathbb{P}_{/\leftrightarrow}$-classe.

A recíproca não é verdadeira se não impusermos condições extras. Se só exigimos que a valuação seja algébrica é possível fazer uma certa bagunça nas $\mathbb{P}_{/[\cdot]}$-classes e continuar com uma valuação algébrica; por exemplo, tome uma proposição atômica A que não seja verdadeira. Mude a valuação da seguinte forma: o valor de A passa a ser um ponto novo que acrescentamos a Ω, mas o valor de qualquer proposição de comprimento maior do que 1 passa a ser o valor calculado como se o A ainda tivesse o valor antigo. Assim o valor de A muda, mas o valor de por exemplo $A \land A$, ou de qualquer proposição que não contenha A, continua o mesmo. Essa nova valuação tem o mesmo conjunto de verdades que a antiga, também é algébrica, mas provavelmente tem um valor de verdade a mais que a anterior. Podemos até tomar um caso extremo disso: defina a operação '$*$' sobre proposições, tal que $\varphi \mapsto \varphi^*$ substitui todas as "subproposições verdadeiras" de φ por $\top$. Por exemplo, se $\varphi = ((A \to B) \lor C) \land D$ e A e $(A \to B) \lor C$ são verdadeiras (i.e., $[A] = [(A \to B) \lor C] = [\top]$) mas B e $A \to B$, C e φ não são, então $\varphi^* = \top \land D$. Crie uma valuação nova, em que duas proposições φ e ψ têm o mesmo valor se e só se $\varphi^* = \psi^*$; essa valuação é algébrica.

Vamos considerar que o caso mais interessante é quando as $\mathbb{P}_{/[\cdot]}$-classes são exatamente as $\mathbb{P}_{/\leftrightarrow}$-classes. Quando uma valuação obedece essa condição vamos dizer que ela é *bem-comportada*; é evidente que uma valuação induz uma única valuação bem-comportada, e já vimos que uma valuação bem-comportada é algébrica. É só nesse sentido que valuações induzem valuações algébricas.

Numa valuação bem-comportada é possível recuperar o comportamento das operações $\wedge$, $\vee$, $\rightarrow$, $\top$ e $\perp$ da álgebra só a partir da relação de ordem '$\leq$' que definimos por $\alpha \leq \beta \iff [\alpha \rightarrow \beta] = [\top]$.

Vamos voltar a chamar $\mathbb{P}_{/\leftrightarrow}$ de Ω e vamos usar $a, b, \ldots, z$ como nomes de elementos de Ω; um *sup* de um conjunto $S \subset \Omega$ é um elemento $z \in S$ que é tão ou mais verdadeiro que todos os elementos de S, i.e., tal que $\forall a \in S\ a \leq z$ (defina a relação '$\leq$' em Ω a partir dela em $\mathbb{P}$ do modo óbvio). Um *inf* de $S \in \Omega$ é um elemento a tão ou menos verdadeiro que todos os elementos de S, no mesmo sentido.

Qualquer subconjunto de Ω tem no máximo um sup, já que se tivesse dois, w e z, teríamos $w \leq z$ e $z \leq w$, e tomando representantes α e β para w e z (i.e., $\alpha \in w$, $\beta \in z$) veríamos que $\alpha \leftrightarrow \beta$ é verdade e portanto $w = z$. Pelo mesmo argumento qualquer subconjunto de Ω tem no máximo um inf.

Defina $\Omega_{\rightarrow z} := \{\, x \in \Omega \mid x \leq z \,\}$ e $\Omega_{a \rightarrow} := \{\, x \in \Omega \mid a \leq x \,\}$. Temos $a = \sup \Omega_{\rightarrow a} = \inf \Omega_{a \rightarrow}$, e podemos recuperar a a partir de $\Omega_{\rightarrow a}$ ou de $\Omega_{a \rightarrow}$.

Para quaisquer $\alpha, \beta, \gamma, \delta$ em $\mathbb{P}$, temos

$\alpha \leq \top$,

$\perp \leq \delta$,

$\alpha \leq \beta \wedge \gamma \iff \alpha \leq \beta$ e $\alpha \leq \gamma$,

$\beta \vee \gamma \leq \delta \iff \beta \leq \delta$ e $\gamma \leq \delta$,

$\alpha \leq (\beta \rightarrow \gamma) \iff \alpha \wedge \beta \leq \gamma$;

as demonstrações são muito fáceis. Por exemplo, para o lado '$\Leftarrow$' da terceira linha,

$$\dfrac{\dfrac{\dfrac{\dfrac{[\alpha]^1 \quad \alpha \rightarrow \beta}{\beta} \quad \dfrac{[\alpha]^1 \quad \alpha \rightarrow \gamma}{\gamma}}{\beta \wedge \gamma}}{\alpha \rightarrow \beta \wedge \gamma}\,1}{} .$$

Daí temos, se $b = [\beta]$ e $c = [\gamma]$,

$\Omega_{\rightarrow \top} = \Omega$,

$\Omega_{\bot\to} = \Omega,$
$\Omega_{\to b\land c} = \Omega_{\to b} \cap \Omega_{\to c},$
$\Omega_{b\lor c\to} = \Omega_{b\to} \cap \Omega_{c\to},$
$\Omega_{\to(b\to c)} = \{\, a \in \Omega \mid a \land b \leq c \,\},$

e portanto

$\top = \sup \Omega,$
$\bot = \inf \Omega,$
$b \land c = \sup(\Omega_{\to b} \cap \Omega_{\to c}),$
$b \lor c = \inf(\Omega_{b\to} \cap \Omega_{c\to}),$
$b \to c = \sup\{\, a \in \Omega \mid a \land b \leq c \,\}.$

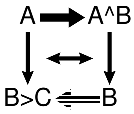
Na próxima seção vamos ver como ordens parciais que têm os sups e infs necessários induzem álgebras de valores de verdade. Repare que nos dois últimos blocos de equações os $\land$, $\lor$, $\to$, $\top$ e $\bot$ eram as operações da álgebra, não os símbolos usados para formar proposições.

6.4 Álgebras de Heyting

Álgebras de Heyting são a versão categórica das álgebras de valores de verdade. Vamos usá-las para mostrar que topologias "são" álgebras de valores de verdade e que certas sentenças válidas classicamente não podem ser teoremas intuicionistas.

Definições: uma categoria é uma *preordem* se para quaisquer dois objetos A e B dela existe no máximo uma seta indo de A para B; dizemos que $A \leq B$ quando existe um morfismo $A \to B$. Uma *álgebra de Heyting* é uma categoria cartesiana fechada que é também uma preordem; uma categoria é *esqueletal* se ela não tem dois objetos diferentes isomorfos.

Teorema: se uma categoria $\mathbf{C}$ é uma álgebra de Heyting esqueletal então os objetos de $\mathbf{C}$ formam uma álgebra de valores de verdade, com $\land$, $\lor$, $\to$, $\top$ e $\bot$ definidos da maneira óbvia: $A \land B$ é o produto de A e B, $A \lor B$ é o coproduto de A e B, $\top$ é o terminal, $\bot$ é o inicial; para definir $A \to B$, repare que como a categoria é esqueletal cada funtor $A \Rightarrow A \land B$ (para B um objeto fixo) tem exatamente uma adjunta à esquerda; chame a adjunta de $C \Rightarrow B \to C$.



Demonstração: como **C** é esqueletal só existe um terminal, um inicial, um produto e um coproduto para quaisquer dois objetos; as operações $\wedge$, $\vee$, $\to$, $\top$ e $\perp$ estão bem-definidas, mas falta ver que elas respeitam deduções, ou seja, que se existe uma dedução em DN de β a partir de $\alpha_1, \ldots, \alpha_n$ e o valor de cada α_1 é $\top$ então o tipo β é levado no $\top$. Transforme essa dedução numa dedução de $\alpha_1 \wedge \ldots \wedge \alpha_n \to \beta$ a partir de nenhuma hipótese (ponha parênteses em $\alpha_1 \wedge \ldots \wedge \alpha_n$ de qualquer modo) e aplique a transformação DN $\to$ CCC do capítulo 5. Examine de novo a tabela das regras do sistema CCC na seção 4.9 e veja que para cada barra da tabela se os morfismos acima da barra existem então o morfismo de baixo também existe; releia a dedução em CCC como uma construção de um morfismo $\alpha_1 \wedge \ldots \wedge \alpha_n \to \beta$ a partir de morfismos que sempre existem, como identidades, projeções, coprojeções, 'ev's e setas indo para o terminal ou saindo do inicial. Exemplo: vamos ver que se $a \wedge b$ é verdade então $b \wedge a$ também é verdade. Tomamos a dedução abaixo à esquerda e a transformamos na dedução abaixo à direita, que diz que o morfismo $a \wedge b \to b \wedge a$ sempre existe; se $a \wedge b$ é verdade existe uma seta $\top \to a \wedge b$, e compondo-a com $a \wedge b \to b \wedge a$ obtemos $\top \to b \wedge a$, que diz que $b \wedge a$ é verdade.

$$\dfrac{\dfrac{[a \wedge b]^1}{b} \quad \dfrac{[a \wedge b]^1}{a}}{b \wedge a} \qquad\qquad \dfrac{\dfrac{}{a \wedge b \to b} \quad \dfrac{}{a \wedge b \to a}}{a \wedge b \to b \to a}$$

Com isso vemos que o morfismo $\alpha_1 \wedge \ldots \wedge \alpha_n \to \beta$ existe; se cada α_i for $\top$ (e eles são por hipótese) então $\alpha_1 \wedge \ldots \wedge \alpha_n = \top$ e temos um morfismo $\top \to \beta$; como temos $\beta \to \top$ e a categoria é esqueletal então $\beta = \top$.

Teorema: seja X um espaço topológico e $\mathcal{O}_X$ a sua topologia, i.e., o conjunto dos seus abertos. Considere a categoria, que também vamos chamar de $\mathcal{O}_X$, em que os objetos são os abertos de X e os morfismos são as inclusões. Essa categoria é uma álgebra de Heyting esqueletal.

Demonstração: $\mathcal{O}_X$ é trivialmente uma preordem esqueletal; além disso o conjunto vazio é inicial, X é terminal, $O_1 \cap O_2$ é o produto de O_1 e O_2 e

$O_1 \cup O_2$ é o coproduto; a única dificuldade é encontrar a adjunta para o produto. Seguindo o mesmo diagrama de cima, temos que ter $A \subset (B \to C)$ exatamente quando tivermos $(A \cap B) \subset C$, onde A, B e C são abertos de X. Mas $A \cap B$ está contido em C exatamente quando A não tiver interseção com $B \backslash C$, e isso acontece exatamente quando A estiver contido no complemento de $B \backslash C$; como A é aberto, isso é equivalente a A estar no interior do complemento de $B \backslash C$. Moral, $B \to C$ corresponde a $\mathrm{Int}(X \backslash (B \backslash C)) = \mathrm{Int}((X \backslash B) \cup C)$. Observe que se estivéssemos na topologia que diz que todos os subconjuntos de X são abertos teríamos $B \to C = \mathrm{Int}((X \backslash B) \cup C) = (X \backslash B) \cup C$; compare com a definição clássica do '$\to$' como $P \to Q := \neg P \vee Q$.

Seja P um conjunto parcialmente ordenado por uma relação $\twoheadrightarrow$; quando temos $a \twoheadrightarrow b$ dizemos que "b está adiante de a". Um subconjunto A de P é considerado *fechado para adiante* quando todo ponto p de P que esteja adiante de um ponto de a também está em A. Por exemplo, considere que P é formado pelos pontos a, b, c e d, e que a relação '$\twoheadrightarrow$' é gerada por $a \twoheadrightarrow b \twoheadrightarrow c$ e $a \twoheadrightarrow d$. Se representamos esses pontos nas posições $\begin{smallmatrix} & \mathrm{a} & \\ \mathrm{b} & & \mathrm{d} \\ \mathrm{c} & & \end{smallmatrix}$, i.e., com adiante sendo para baixo, e representamos subconjuntos de P usando 1s e 0s nessas posições para indicar pontos que pertencem e pontos que não pertencem ao subconjunto, então estes são os conjuntos fechados para adiante: $\begin{smallmatrix} & 0 & \\ 0 & & 0 \\ 0 & & \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 0 & & 0 \\ 1 & & \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 1 & & 0 \\ 1 & & \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 0 & & 1 \\ 0 & & \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 0 & & 1 \\ 1 & & \end{smallmatrix}, \begin{smallmatrix} & 0 & \\ 1 & & 1 \\ 1 & & \end{smallmatrix}, \begin{smallmatrix} & 1 & \\ 1 & & 1 \\ 1 & & \end{smallmatrix}$. Não é difícil verificar que a "topologia dos fechados para adiante" sobre um P parcialmente ordenado qualquer, em que os abertos são exatamente os conjuntos fechados para adiante, é realmente uma topologia; o interior de um conjunto é o maior fechado para adiante contido naquele conjunto, e $A \to B$ é o interior de $(P \backslash A) \cup B$. Como caso particular, $\neg A = A \to \bot =$ interior do complemento de A.

Teorema: $A \vee \neg A$, $B \to \neg\neg B$ e $\neg(C \wedge D) \to (\neg C \vee \neg D)$ não são teoremas intuicionistas.

Demonstração: se eles fossem teoremas eles valeriam $\top$ em qualquer álgebra de Heyting, mas

$$\dfrac{A \qquad \dfrac{A}{\neg A}}{A \vee \neg A} \qquad \dfrac{\begin{smallmatrix} 0 \\ 1\,0 \end{smallmatrix} \qquad \dfrac{\begin{smallmatrix} 0 \\ 1\,0 \end{smallmatrix}}{\begin{smallmatrix} 0 \\ 0\,1 \end{smallmatrix}}}{\begin{smallmatrix} 0 \\ 1\,1 \end{smallmatrix}} \qquad\qquad \dfrac{B \qquad \dfrac{\dfrac{B}{\neg B}}{\neg\neg B}}{B \to \neg\neg B} \qquad \dfrac{\begin{smallmatrix} 0 \\ 1 \end{smallmatrix} \qquad \dfrac{\dfrac{\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}}{\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}}}{\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}}}{\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}}$$

$$\dfrac{\dfrac{\dfrac{C \quad D}{C \land D}}{\neg(C \land D)} \qquad \dfrac{\dfrac{C}{\neg C} \quad \dfrac{D}{\neg D}}{\neg C \lor \neg D}}{\neg(C \land D) \to (\neg C \lor \neg D)} \qquad \dfrac{\dfrac{\dfrac{\overset{0}{1\,0} \quad \overset{0}{0\,1}}{\overset{0}{0\,0}}}{\overset{1}{1\,1}} \qquad \dfrac{\dfrac{\overset{0}{1\,0}}{\overset{0}{0\,1}} \quad \dfrac{\overset{0}{0\,1}}{\overset{0}{1\,0}}}{\overset{0}{1\,1}}}{\overset{0}{1\,1}} \quad .$$

Chapter 7

Análise não-standard

Primeiro uma exposição informal. Vamos construir um universo novo a partir do antigo; o antigo vai ser chamado de "standard" e o novo às vezes de "não-standard" (quando ele for construído como um $\mathbf{Set}^{\mathcal{U}}$), às vezes de "semi-standard" (quando for construído como um $\mathbf{Set}^{\mathcal{F}}$). Os pontos do universo novo vão ser seqüências de pontos do universo antigo módulo uma certa relação de equivalência, e queremos que essa relação de equivalência seja tal que para sabermos se duas seqüências são equivalentes baste olhar para o conjunto dos índices em que os valores das duas coincidem: se esse conjunto de índices é "grande" (onde para nós um conjunto de índices "grande" é a princípio simplesmente um conjunto que pertence ao conjunto dos conjuntos de índices considerados como grandes) então as duas seqüências são consideradas equivalentes; senão, não.

Vamos ver que as propriedades que esse conjunto dos conjuntos grandes de índices (vamos chamá-lo de $\mathcal{F}$) tem que obedecer para induzir uma relação de equivalência não trivial são exatamente as que fazem com que ele seja um *filtro*; o conceito de filtro sobre um espaço já foi um dos conceitos mais fundamentais de topologia, mas ele foi aos poucos sendo substituído pelo de topologia sobre um espaço. No nosso caso o conceito de filtro vai ser mais útil, por um lado porque ele se presta melhor para analisar continuidade pontual e limites, e por outro lado porque filtros com uma certa propriedade de maximalidade são o que chamamos de *ultrafiltros*, e ultrafiltros têm uma propriedade lógica muito interessante: se o conjunto $\mathcal{F}$ for escolhido como sendo um ultrafiltro, o universo novo vai obedecer *exatamente a mesma lógica de primeira ordem que o antigo*; se $\mathcal{F}$ for só um filtro o universo novo vai obedecer uma lógica mais fraca que o antigo, com mais valores de verdade, mas que não chega a ser tão fraca quanto a lógica intuicionista pura: a lógica do universo novo vai ser sempre booleana, e vamos ter uma descrição explícita dos seus valores de verdade.

Tanto no caso em que o $\mathcal{F}$ é um ultrafiltro quanto no caso em que ele é só um filtro vamos ter cópias dos objetos do universo antigo dentro do novo, as classes de equivalência das seqüências constantes, e elas vão obedecer propriedades parecidas com as que obedeciam no universo original. Além disso quando $\mathcal{F}$ não é trivial vamos ter objetos que se comportam como infinitesimais; por exemplo, as seqüências tendendo a zero vão se comportar (nos sentidos adequados, é claro) como pontos diferentes de zero mas infinitamente próximos de zero.

Mudando o conjunto tomado como conjunto de índices para as seqüências as seqüências podem deixar de ser seqüências (porque o conjunto-índice deixa de ser ordenado) e passar a ser só funções; isso aparentemente complica as coisas, mas vamos ver como certos conjuntos de índices não só têm filtros naturais, como vão aparecer certos objetos que são *infinitesimais naturais*, e que têm certas propriedades de universalidade. Um resultado bem conhecido de análise não-standard [1]é que uma função (standard) f é contínua no 0 se e só se ela leva todos os pontos infinitamente próximos do 0 em pontos infinitamente próximos de $f(0)$; nós vamos ver que isso é equivalente à imagem do infinitesimal natural por ela ser um ponto infinitamente próximo de $f(0)$, ou seja, não só a direção da quantificação se inverteu (ao invés de algo como "a imagem inversa de cada aberto é um aberto" temos "a imagem de cada infinitesimal é um infinitesimal") como um quantificador desapareceu, e da mesma forma que aprendemos que uma variável $x \in \mathbb{R}$ se comporta como um número real universal vamos começar a ver que esse infinitesimal natural se comporta como um real infinitamente pequeno universal.

[1]Há várias construções — que não são equivalentes entre si — para esses universos ampliados que obedecem a mesma lógica de primeira ordem que o universo usual; a que vamos usar é uma versão, simplificada porque só vamos tratar de sentenças tipadas, do "modelo gerado por ultrapotência" que aparece em [Davis] e [SL]. Uma construção que, a grosso modo, é como um limite de ultrapotências é a dos *enlargements*, descrita em [Robinson] e [SL]. A "*Internal Set Theory*" de Edward Nelson ([Nelson], [Lutz]) é uma abordagem axiomática, em que se usa o teorema da compacidade para demonstrar a existência de um modelo. Nós vamos ser simplistas por conveniência e dizer que a "análise semi-standard" trata do universo novo ("semi-standard") quando $\mathcal{F}$ é um filtro, e a "análise não-standard" trata só dos casos em que $\mathcal{F}$ é um ultrafiltro.

7.1 Hiperpontos

Vamos usar sempre o símbolo $\mathbb{I}$ para denotar o conjunto-índice. Até fixarmos bem as idéias vamos ter $\mathbb{I} = \mathbb{N}$; o universo antigo é $\mathbf{Set}_{\mathcal{A}}$, e o universo novo formado por seqüências de pontos do antigo é $\mathbf{Set}_{\mathcal{A}}^{\mathbb{I}}$, que vamos abreviar como $\mathbf{Set}^{\mathbb{I}}$. Se α é um tipo de $\mathbf{Set}_{\mathcal{A}}$, então um *ponto de tipo α de* $\mathbf{Set}^{\mathbb{I}}$ é uma seqüência de pontos de tipo α, i.e., é uma função de $\mathbb{I}$ em $\mathbf{E}_\alpha$, ou, se preferirmos, um ponto de tipo $\mathtt{i} \to \alpha$.

O nosso primeiro quociente vai identificar seqüências que só difiram num conjunto finito de índices, i.e., que coincidam num subconjunto *cofinito* de $\mathbb{I} = \mathbb{N}$. O conjunto dos conjuntos cofinitos de $\mathbb{N}$ vai ser chamado de $\mathcal{N}$. $\mathcal{N}$ é um *filtro* sobre $\mathbb{I}$; a definição de filtro vai ser dada daqui a pouco. Em geral vamos deixar um filtro fixo, e vamos chamá-lo de $\mathcal{F}$; por enquanto $\mathcal{F} = \mathcal{N}$.

Os pontos de $\mathbf{Set}^{\mathbb{I}}$ — que são seqüências indexadas por $\mathbb{I}$ — são chamados de *pré-hiperpontos*. Chamamos de $\mathbf{Set}^{\mathcal{F}}$ o quociente de $\mathbf{Set}^{\mathbb{I}}$ pelo filtro $\mathcal{F}$: dois pré-hiperpontos são *$\mathcal{F}$-iguais* se eles coincidem num conjunto *$\mathcal{F}$-grande* — por enquanto, num subconjunto cofinito de $\mathbb{N}$. Os pontos de $\mathbf{Set}^{\mathcal{F}}$, que chamamos de *hiperpontos*, são classes de equivalência de pré-hiperpontos $\mathcal{F}$-iguais. É melhor seguir com um exemplo: $(1, 1/2, 1/3, \dots)$ é um pré-hiperponto de tipo $\mathtt{r}$, que também dizemos que é um "pré-hiperponto de $\mathbb{R}$", ou que é um "pré-hiper-$\mathtt{r}$". O hiperponto correspondente a ele é o conjunto dos pré-hiperpontos $\mathcal{F}$-iguais a ele, i.e., o conjunto das seqüências que só diferem dele num conjunto finito de índices. Como ficaria confuso demais falar sempre dos hiperpontos como conjuntos de pré-hiperpontos nós vamos preferir pegar um pré-hiperponto como representante da classe dos pré-hiperpontos $\mathcal{F}$-iguais a ele, e dizer que a classe de equivalência dele é "ele visto como hiperponto".

Nós temos uma função natural indo de $\mathbf{Set}_{\mathcal{A}}$ em $\mathbf{Set}^{\mathbb{I}}$, a que leva cada ponto α de $\mathbf{Set}_{\mathcal{A}}$ na seqüência constante $(\alpha, \alpha, \alpha, \dots)$, e temos uma função natural de $\mathbf{Set}^{\mathbb{I}}$ em $\mathbf{Set}^{\mathcal{F}}$, o quociente. Usando essas funções obtemos um pré-hiperponto e um hiperponto a partir de cada ponto α de $\mathbf{Set}_{\mathcal{A}}$; dizemos que os pontos de $\mathbf{Set}_{\mathcal{A}}$ são pontos *standard*, e os pré-hiperpontos e hiperpontos obtidos dessa maneira são chamados de *pré-hiperpontos standard* e *hiperpontos standard*. Seqüências não-constantes não são pré-hiperpontos standard, e não é difícil ver que $(1, 2, 3, \dots)$ visto como hiperponto também não é standard.

$\mathbf{Set}^{\mathbb{I}}$ e $\mathbf{Set}^{\mathcal{F}}$ também são universos de pontos tipados, como $\mathbf{Set}_{\mathcal{A}}$. Até agora só tínhamos mencionado pré-hiperpontos $(a_1, a_2, \dots)$ em que cada a_i era do mesmo tipo α, mas podia ser que os a_is tivessem tipos diferentes. Um exemplo natural é o seguinte: sejam $\epsilon = (1, 1/2, 1/3, \dots)$, $\epsilon/2 = (1/2, 1/4, 1/6, \dots)$; $\epsilon/2$ é um pré-hiperponto de $[0, \epsilon)$, já que $(1/2 \in [0, 1),\ 1/4 \in [0, 1/2),\ 1/6 \in [0, 1/3),\ \dots) = (\top, \top, \top, \dots)$. O que estávamos chamando de um "pré-hiperponto de tipo α" na verdade é um pré-hiperponto de tipo $(\alpha, \alpha, \alpha, \dots)$, e $(\alpha, \alpha, \alpha, \dots)$ é um *pré-hipertipo standard*, e é "α visto como pré-hipertipo"; mas nós vamos usar bastante o abuso de linguagem segundo o qual esse pré-hipertipo é simplesmente "α", e segundo o qual $(a, a, a, \dots)$ e $(a, a, a, \dots)$ visto como hiperponto são simplesmente "a".

Repare que o conjunto dos hipertipos é um quociente do conjunto dos pré-hipertipos, e que dois pré-hiperpontos com tipos diferentes podem ser levados pelo quociente no mesmo hiperponto.

Temos bons motivos para considerar que $\epsilon = (1, 1/2, 1/3, \dots)$ é um infinitesimal. Se o comparamos com 0 índice a índice vemos que ele é sempre maior do que 0, i.e., obtemos uma seqüência de valores de verdade que é verdadeira em todo índice; como ela é igual a $\top$ visto como pré-hiperponto (e ela vista como hiperponto é igual a $\top$ visto como hiperponto) dizemos que $(0 < \epsilon) =_{\mathbb{I}} \top$ e que $(0 < \epsilon) =_{\mathcal{F}} \top$, ou que $0 < \epsilon$ é "$\mathbb{I}$-verdade" e "$\mathcal{F}$-verdade". Se comparamos ϵ com qualquer real standard $\delta > 0$, i.e., com $(\delta, \delta, \delta, \dots)$, vemos que o resultado é uma seqüência de valores de verdade que a partir de um certo ponto é sempre verdade; $\epsilon < \delta$ pode não ser $\mathbb{I}$-verdadeira, já que pode falhar nos primeiros índices, mas é $\mathcal{F}$-verdadeira, e isso para qualquer δ standard > 0. Esse ϵ está entre 0 e qualquer standard maior do que 0, então (por definição, digamos) ϵ é um infinitesimal. Repare que isso deixa de ser verdade se permitirmos que δ seja um pré-hiper-$\mathbb{r}$ qualquer maior que 0: $0 < \epsilon/2 < \epsilon$ em todo índice. Repare também que entre pré-hiper-$\mathbb{r}$s e entre hiper-$\mathbb{r}$s a relação '$<$' não é muito bem-comportada. Se tomamos $a = (0, 1, 0, 1, \dots)$ e $b = (1, 0, 1, 0, \dots)$ então $(a < b) = (\top, \dashv, \top, \dashv, \dots)$, que nem coincide com $\top$ num conjunto grande (i.e., cofinito) de índices nem coincide com $\dashv$ num conjunto grande de índices; o valor de verdade de $a < b$ é um valor de verdade intermediário entre $\top$ e $\dashv$. Isto vai ser consertado daqui a pouco, quando passarmos para ultrafiltros, mas um pouco depois vamos ver que o sistema quebrado vai ser ainda mais útil que o consertado.

7.2 Filtros

Fixe um conjunto índice $\mathbb{I}$ qualquer. Que propriedades o conjunto $\mathcal{F} \subset \mathcal{P}(\mathbb{I})$ deve ter para que a relação de "$\mathcal{F}$-igualdade" que tínhamos antes — dois pré-hiperpontos são $\mathcal{F}$-iguais quando o conjunto de índices em que eles coincidem pertence a $\mathcal{F}$ — seja uma relação de equivalência?

Vamos continuar chamando "pré-hiperpontos" de "seqüências", e vamos usar a, b e c como nomes de seqüências e A, B e C como nomes de subconjuntos de $\mathbb{I}$.

Queremos que toda seqüência seja equivalente a ela mesma, daí $\mathbb{I} \in \mathcal{F}$. E se $a =_{\mathcal{F}} b$ e $b =_{\mathcal{F}} c$ vamos querer que valha $a =_{\mathcal{F}} c$; digamos que a e b coincidem no conjunto A e b e c coincidem no conjunto B. Então a e c coincidem pelo menos na interseção de A e B, e aliás podemos arranjar as coisas para que eles coincidam em qualquer conjunto contendo a interseção de A e B; daí para quaisquer conjuntos A e B em $\mathcal{F}$ qualquer conjunto contendo a interseção de A e B tem que estar em $\mathcal{F}$, e $\mathcal{F}$ é fechado por interseções finitas e por "aumentar conjuntos".

Se $\varnothing \in \mathcal{F}$ então todo subconjunto de $\mathbb{I}$ está em $\mathcal{F}$, e todas as seqüências são equivalentes. Isso não é uma situação muito interessante, e em geral vamos nos restringir aos casos em que $\varnothing \notin \mathcal{F}$. Obs: se $A, B \in \mathcal{F}$ e $A \cap B = \varnothing$ então $\varnothing \in \mathcal{F}$.

Quando $\mathcal{F} \subset \mathcal{P}(\mathbb{I})$ for fechado por interseções finitas e por aumentar conjuntos, e $\mathbb{I} \in \mathcal{F}$ e $\varnothing \notin \mathcal{F}$ vamos dizer que $\mathcal{F}$ é um *filtro*, ou um *filtro sobre* $\mathbb{I}$.

Alguns exemplos de filtros:

0) $\{\, \mathbb{I} \,\}$ é um filtro. No quociente por $\mathbb{I}$ cada seqüência só é equivlanete a si mesma; daí o sentido de usar a notação $[\alpha]_{\mathbb{I}}$ para falar de α como pré-hiperponto.

1) Se $\mathbb{I}$ é infinito o conjunto dos subconjuntos cofinitos de $\mathbb{I}$ é um filtro. Se $\mathbb{I}$ é não-enumerável o conjunto dos subconjuntos coenumeráveis de $\mathbb{I}$ também é um filtro.

2) Defina $\uparrow A$ como o conjunto dos subconjuntos de $\mathbb{I}$ que contêm A. Se $A \neq \varnothing$ então $\uparrow A$ é um filtro. Se $A = \{\, a \,\}$ e $\mathcal{F} = \uparrow A$ então os hiperpontos e hipertipos de $\mathbf{Set}^{\mathcal{F}}$ se identificam naturalmente com pontos e tipos de $\mathbf{Set}_{\mathcal{A}}$, já que a é o único índice que importa.

3) Seja $A_1 \supset A_2 \supset \ldots$ uma seqüência decrescente de conjuntos não-vazios com $\mathbb{I} \supset A_1$. Então o conjunto $\{\, X \subset \mathbb{I} \mid \exists i\; X \supset A_i \,\}$ é um filtro.

4) Seja $\mathcal{B}$ uma família de subconjuntos de $\mathbb{I}$ tal que nenhuma interseção finita de membros de $\mathcal{B}$ seja vazia; o conjunto dos subconjuntos de $\mathbb{I}$ que contêm alguma interseção finita de membros de $\mathcal{B}$ é um filtro. Vamos chamar o conjunto das interseções finitas de membros de $\mathcal{B}$ de $\bigcap_{\text{fin}} \mathcal{B}$; fica implícito que a interseção de zero conjuntos de $\mathcal{B}$ é $\mathbb{I}$. O filtro gerado a partir de $\mathcal{B}$ é $\uparrow(\bigcap_{\text{fin}} \mathcal{B})$; dizemos que $\mathcal{B}$ é uma *pré-base* e $\bigcap_{\text{fin}} \mathcal{B}$ é uma *base* para esse filtro.

5) Se X é um espaço topológico e $x_0 \in X$ então o conjunto das vizinhanças de X (i.e., dos subconjuntos de X que contêm algum aberto contendo x_0) é um filtro sobre X, o *filtro das vizinhanças de* x_0.

6) Uma *vizinhança furada* do ponto $x_0 \in X$, onde X é um espaço topológico, é um subconjunto de $X \backslash \{\, x_0 \,\}$ tal que a sua união com $\{\, x_0 \,\}$ é uma vizinhança de x_0. Se $\{\, x_0 \,\}$ não for um aberto de X então o conjunto das vizinhanças furadas de x_0 é um filtro sobre $X \backslash \{\, x_0 \,\}$, o *filtro das vizinhanças furadas de* x_0. O filtro $\mathcal{N}$ é o filtro das vizinhanças furadas do infinito em $\mathbb{N}^*$, a compactificação de Alexandroff de $\mathbb{N}$ (visto como espaço discreto) por um ponto no infinito; é possível dar uma definição bem mais elementar da topologia de $\mathbb{N}^*$. Leve $\mathbb{N} \cup \{\, \infty \,\} = \{\, 0, 1, 2, \ldots, \infty \,\}$ em $A = \{\, 1, \frac{1}{2}, \frac{1}{3}, \ldots, 0 \,\} \subset \mathbb{R}$ por $n \mapsto \frac{1}{n+1}$, $\infty \mapsto 0$. Faça A herdar a topologia de $\mathbb{R}$ e puxe a topologia de volta para $\mathbb{N} \cup \{\, \infty \,\}$; agora cada natural é aberto e fechado, e as vizinhanças do ∞ são os conjuntos que têm todos os naturais acima de algum N. A é compacto, e $\mathbb{N} \cup \{\, \infty \,\}$ também.

7) Se $\mathcal{F}'$ é um filtro sobre $\mathbb{I}' \subset \mathbb{I}$ então o conjunto dos subconjuntos de $\mathbb{I}$ que contêm algum conjunto de $\mathcal{F}'$, $\uparrow\mathcal{F}'$, é um filtro sobre $\mathbb{I}$.

8) Se $\mathbb{I} = \mathbb{I}' \times \mathbb{I}''$, $\mathcal{F}'$ é um filtro sobre $\mathbb{I}'$ e $\mathcal{F}''$ é um filtro sobre $\mathbb{I}''$, então $\{\, A' \times A'' \mid A' \in \mathcal{F}', A'' \in \mathcal{F}'' \,\}$ é uma pré-base (na verdade uma base) para um filtro, $\mathcal{F}$, sobre $\mathbb{I}$; dizemos que esse $\mathcal{F}$ é o filtro *produto* de $\mathcal{F}'$ e $\mathcal{F}''$.

Observação: cada filtro corresponde a uma medida finitamente aditiva sobre $\mathbb{I}$, μ_F, completa (i.e., em que todo subconjunto de um conjunto de medida 0 é mensurável e tem medida 0), que só assume os valores 0 e 1 e tal que $\mu_{\mathcal{F}}(\mathbb{I}) = 1$; basta interpretar os conjuntos grandes como os conjuntos de medida 1, os complementos de conjuntos grandes como os de medida 0, e os outros como não-mensuráveis.

7.3 Ultrafiltros

Além de dizer que um conjunto de índices é "grande" quando ele pertence a $\mathcal{F}$, vamos dizer que um conjunto A de índices é *pequeno* quando o complemento de A pertence a $\mathcal{F}$, e vamos dizer que A é *médio* quando ele não for nem grande nem pequeno. Se $\mathcal{F}$ é um filtro e A é um conjunto $\mathcal{F}$-médio então $\mathcal{F}' := \uparrow(\bigcap_{\text{fin}}(\mathcal{F} \cup \{ A \}))$ é um filtro que contém $\mathcal{F}$ (i.e., todo conjunto $\mathcal{F}$-grande é $\mathcal{F}'$-grande) e para o qual A é grande. Quando temos qualquer situação dessas em que $\mathcal{F}$ e $\mathcal{F}'$ são filtros sobre $\mathbb{I}$ e $\mathcal{F} \subset \mathcal{F}'$ dizemos que $\mathcal{F}'$ é um *refinamento* de $\mathcal{F}$; quando refinamos um filtro os conjuntos grandes continuam grandes, os pequenos continuam pequenos, mas pode ser que alguns médios passem a ser grandes, e outros, os complementos destes, passam a ser pequenos.

Se o único refinamento de um filtro $\mathcal{F}$ é o próprio $\mathcal{F}$ — i.e., se ele divide os subconjuntos de $\mathbb{I}$ só entre grandes e pequenos, sem médios — dizemos que $\mathcal{F}$ é um *ultrafiltro*, e nos permitimos escrever $\mathcal{F}$ como $\mathcal{U}$.

Se vale o axioma da escolha então qualquer filtro pode ser refinado até um ultrafiltro. Uma esboço da demonstração disso: indexe os subconjuntos de $\mathbb{I}$ por um ordinal, obtendo uma "seqüência" $(I_j)_{j \in J}$, que pra simplificar vamos considerar que começa de 1. Tente acrescentar os conjuntos da seqüência a $\mathcal{F}$, um por vez: defina que $\mathcal{F}_0 = \mathcal{F}$, que $\mathcal{F}_{j+1} = \uparrow \bigcap_{\text{fin}}(\mathcal{F}_j \cup \{ I_j \})$ quando isso for um filtro, $\mathcal{F}_{j+1} = \mathcal{F}_j$ quando não, e que quando j não tem antecessor e é maior que 0 então $\mathcal{F}_j$ é a união de todos os $\mathcal{F}_k$ com $k < j$. A união de todos os $\mathcal{F}_j$ é um filtro que não pode ser refinado a nenhum filtro diferente de si mesmo e que portanto é um ultrafiltro.

Qualquer $\mathbf{Set}^{\mathcal{U}}$ tem exatamente dois hipervalores de verdade, $\top$ e $\dashv$ (vistos como hiperpontos). Tome um pré-hipervalor de verdade; ele induz uma partição de $\mathbb{I}$ em dois conjuntos disjuntos, $I_\top$ e $I_\dashv$; é fácil ver que eles não podem ser ambos grandes, porque aí a sua interseção, que é o conjunto vazio, teria que ser grande; também não podem ser ambos pequenos, porque aí a interseção dos seus complementos teria que ser grande. Como não existem conjuntos $\mathcal{U}$-médios então ou $I_\top$ é grande e $I_\dashv$ é pequeno ou $I_\top$ é pequeno e $I_\dashv$ é grande. Uma adaptação desse argumento mostra que para qualquer partição de um conjunto $\mathcal{U}$-grande I em finitos conjuntos disjuntos, $I_1, \ldots, I_n$, exatamente um dos I_i é grande, e os outros são pequenos. Voltando aos valores de verdade, qualquer pré-hiperponto que tomemos como representante de um hipervalor de verdade é um pré-hipervalor de verdade num conjunto grande de índices (nos outros índices ele pode assumir tipos estranhos), e esse conjunto grande é a união disjunta de um $I_\top$ e um $I_\dashv$, com as definições naturais; daí ou ele é $\top$ num conjunto grande ou $\dashv$ num conjunto grande. Um argumento similar mostra que para qualquer tipo α de $\mathbf{Set}_\mathcal{A}$ com $\mathbf{E}_\alpha$ finito qualquer hiper-α é um a visto como hiperponto; se $\mathcal{U}$ é um refinamento de $\mathcal{N}$ então $(0, 1, 2, \ldots)$ visto como hiperponto não é standard, ou seja, para esse $\mathcal{U}$ existem mais hipernaturais (ou hiperreais) do que os que vêm do universo standard.

Já vimos que se um ultrafiltro é da forma $\uparrow\{\, a \,\}$ então todos os hipertipos correspondem a tipos standard e todos os hiperpontos correspondem a pontos standard, já que só o índice a interessa quando comparamos duas seqüências; nesse caso $\mathbf{Set}^\mathcal{U} = \mathbf{Set}_\mathcal{A}$. É fácil ver que um ultrafiltro $\mathcal{U}$ não é da forma $\uparrow\{\, a \,\}$ (um ultrafiltro dessa forma é chamado de *principal*) se e somente se a interseção dos conjuntos $\mathcal{U}$-grandes for vazia. Nesse caso — que é o caso interessante — dizemos que o ultrafiltro é *não-principal*, e aí para todo tipo α com $\mathbf{E}_\alpha$ infinito existem hiper-'α's que não são standard[2].

Um dos problemas com ultrafiltros é que não é possível encontrar explicitamente um ultrafiltro não-principal. Uma idéia ingênua seria que poderíamos encontrar um ultrafiltro não-principal a partir de uma pré-base enumerável bem escolhida; essa idéia não funciona, e vamos dar uma esboço da demonstração disso. Digamos que $A_1, A_2, \ldots$ seja uma pré-base para um filtro sobre $\mathbb{I}$. Defina a seqüência $B_1, B_2, \ldots$ por $B_i = \bigcap_{j \leq i} A_j$; essa seqüência é uma base para o mesmo filtro. Agora remova os elementos repetidos dela: obtemos uma seqüência $C_1, C_2, \ldots,$ que pode ser finita ou infinita. Se ela for finita e terminar em C_n então o filtro é $\uparrow C_n$, e é principal. Se ela for infinita então $(C_1 \backslash C_2) \cup (C_3 \backslash C_4) \cup \ldots$ é um conjunto de índices tal que nem ele nem o seu complemento estão no filtro, e portanto o filtro não é um ultrafiltro.

É possível dar uma demonstração correta, mas muitíssimo mais pesada, de que não dá pra encontrar explicitamente ultrafiltros não-principais. Vamos chamar de UF a sentença de lógica de primeira ordem que diz que todo filtro pode ser refinado a um ultrafiltro. UF é uma conseqüência de ZFC, mas existem modelos de ZF+¬UF e de ZF+UF+¬AC. Há demonstrações disso no

[2]Podem acontecer exceções quando a cardinalidade do conjunto índice é incrivelmente grande, i.e., quando $|\mathbb{I}|$ é o que chamamos de um cardinal *mensurável*. Mas esse caso não nos interessa.

capítulo 7 de [Jech], mas só estamos mencionando estes fatos para convencer o leitor de que a técnica mais comum de demonstrar coisas sobre o universo standard usando o universo não standard, que vamos descrever num instante, é ruim. A técnica é a seguinte: pelo teorema da transferência, que vamos ver na próxima seção, toda sentença standard de primeira ordem que obedece uma certa condição, a de ter "quantificadores limitados", é verdadeira no universo standard ("U") se e só se ela for verdadeira no universo não-standard ("U'") construído a partir dele usando um ultrafiltro. Daí podemos provar algo em U' a partir de sentenças standard verdadeiras "transferidas" para U', e em U' podemos usar infinitesimais e outros hiperelementos na demonstração; se a sentença que demonstramos ser verdadeira em U' é uma sentença standard com quantificadores limitados, então ela vale também em U. Mas isso não diz nada sobre como trazer a *demonstração* de volta para U; como U' é construído usando (vamos ser bem informais) um pouquinho do axioma da escolha, é possível que só dê para trazer a demonstração para U fazendo com que ela use explicitamente o axioma da escolha, ou, pior ainda, que para expulsarmos dela essa dependência do axioma da escolha tenhamos que transformá-la bastante; nesse caso a demonstração em U ficaria bem diferente da demonstração em U', talvez muito maior, e talvez praticamente ilegível.

Demonstrações usando infinitesimais podem ser muito elegantes, mas não podemos exigir sempre que os nossos interlocutores entendam análise não-standard... e se queremos que a nossa demonstração final esteja em U e não em U' precisamos ter um processo que traduza demonstrações em U' para demonstrações em U, que preserve a estrutura (e portanto a elegância!) das demonstrações feitas em U' e mantenha a demonstração traduzida tão elementar quanto possível; um processo de tradução desses não deve funcionar sempre, e ele vai quebrar pelo menos nos lugares onde a demonstração original usava o axioma da escolha de modo escondido. O máximo que devemos esperar de um processo "bom" de tradução, então, é que ele quebre de um modo claro e instrutivo onde precisar quebrar, e que conheçamos uma classe — suficientemente grande para ser útil — de demonstrações nas quais ele esteja definido e funcione.

7.4 O teorema da transferência

Obs.: nesta seção vamos discutir, sem demonstrar quase nada, a relação entre $\mathbf{Set}_{\mathcal{A}}$, $\mathbf{Set}^{\mathbb{I}}$, $\mathbf{Set}^{\mathcal{F}}$ e $\mathbf{Set}^{\mathcal{U}}$. Leitores que nunca tenham tido contato com análise não-standard vão provavelmente achar mais interessante ler as seções 7.5 e 7.6 antes dessa para terem mais motivação.

O teorema da transferência diz que qualquer sentença verdadeira em $\mathbf{Set}_{\mathcal{A}}$ continua verdadeira quando interpretada em $\mathbf{Set}^{\mathcal{U}}$, mas não temos como entender o que isso quer dizer enquanto não soubermos reinterpretar sentenças de $\mathbf{Set}_{\mathcal{A}}$ em $\mathbf{Set}^{\mathcal{U}}$. O primeiro conceito que precisamos é o de *operações internas*; uma operação é *interna* quando sua ação pode ser descrita índice a índice. Vamos começar com um exemplo: queremos ter em $\mathbf{Set}^{\mathcal{F}}$ uma operação de soma de reais que venha da soma de reais em $\mathbf{Set}^{\mathbb{I}}$, que por sua vez vem da soma de reais em $\mathbf{Set}_{\mathcal{A}}$. Em $\mathbf{Set}^{\mathbb{I}}$ a soma vai receber duas seqüências de reais, somá-las em cada índice em separado e retornar uma terceira seqüência de reais. Quando nós levamos essa operação para $\mathbf{Set}^{\mathcal{F}}$ vamos querer que ela esteja definida exatamente quando seus dois argumentos são hiperreais, e que — digamos que os dois hiperreais se chamem $^*a$ e $^*b$, e o resultado se chame $^*c$ — se $^*a$ e $^*b$ são pre-hiperreais $\mathtt{i} \to a$ e $\mathtt{i} \to b$ vistos como hiperreais então $^*c$ seja $\mathtt{i} \to c$ visto como hiperreal, onde $\mathtt{i} \to c$ é obtido a partir de $\mathtt{i} \to a$ e $\mathtt{i} \to b$ somando as seqüências índice a índice. Usando a notação da seção 2.8, as três operações são

$$\frac{a \quad b}{c}\,\mathrm{F}_{(+)} \qquad \frac{\mathtt{i} \to a \quad \mathtt{i} \to b}{\mathtt{i} \to c}\,\mathrm{F}_{\mathtt{i}\to(+)} \qquad \frac{^*a \quad ^*b}{^*c}\,\mathrm{F}_{^*(+)}$$

onde a segunda está escrita de forma meio enganosa; como a operação é a mesma em todo índice ela é na verdade

$$\dfrac{\dfrac{\dfrac{[\mathtt{i}]^1 \quad \mathtt{i} \to a}{a} \quad \dfrac{[\mathtt{i}]^1 \quad \mathtt{i} \to b}{b}}{c}\,\mathrm{F}_{(+)}}{\mathtt{i} \to c}\,1 \qquad ,$$

e quando escrevemos $\mathrm{F}_{\mathtt{i}\to(+)}$ estamos querendo indicar que o mesmo funtor que leva cada tipo α em $\mathtt{i} \to \alpha$ leva $\mathrm{F}_{(+)}$ em $\mathrm{F}_{\mathtt{i}\to(+)}$. Voltando ao assunto, quando mudamos os representantes de $^*a$ e $^*b$ estamos mudando $\mathtt{i} \to a$ e $\mathtt{i} \to b$ de tal forma que os valores de a e b só mudam em conjuntos pequenos de índices; vamos chamar estes conjuntos de A e B. Se os novos $\mathtt{i} \to a$ e $\mathtt{i} \to b$ continuam sendo pré-hiperreais podemos continuar fazendo a soma índice a índice do modo usual e vamos obter um $\mathtt{i} \to c$ que difere do antigo no máximo em $A \cup B$, que por estar contida na união de finitos conjuntos pequenos também é um conjunto pequeno. Conclusão, o novo $\mathtt{i} \to c$ visto como hiperponto é igual ao antigo, e os representantes escolhidos para ${}^*a$ e ${}^*b$ são irrelevantes.

Podia ser também que tivéssemos trocado o $\mathtt{i} \to a$ e o $\mathtt{i} \to b$ de tal forma que os novos tivessem tipos diferentes para alguns índices; se isso tivesse acontecido só em conjuntos pequenos de índices o resultado ainda deveria ser o mesmo quando visto como hiperponto. Repare que o $\mathtt{i} \to c$ é obtido aplicando uma *seqüência de operações* $\mathtt{i} \to (+)$ às seqüências $\mathtt{i} \to a$ e $\mathtt{i} \to b$; nesse exemplo a seqüência $\mathtt{i} \to (+)$ é constante, mas não precisava ser, e também poderíamos tê-la modificado num conjunto pequeno de índices sem mudar o valor do resultado visto como hiperponto. Vamos ter dois modos de permitir essas mudanças em conjuntos pequenos: o primeiro, pior, seria modificar a operação $\mathtt{i} \to (+)$ para que o seu tipo combinasse sempre com os tipos dos parâmetros; o segundo, que é o que vamos usar, é admitir que $\mathtt{i} \to c$ seja uma função parcial, definida exatamente nos índices em que a operação faz sentido. É fácil ver que já que queremos montar árvores com várias aplicações de operações vamos ter que permitir que todos os ‘$\mathtt{i} \to \ldots$’, exceto talvez os das folhas, sejam funções parciais; como as árvores são finitas o resultado só não vai estar definido numa união de finitos conjuntos pequenos de índices, que é um conjunto pequeno.

Agora que já sabemos interpretar a aplicação de funções em $\mathbf{Set}^{\mathcal{F}}$, mesmo no caso em que a função ou seus parâmetros não são standard,

$$\frac{a_1 \quad \ldots \quad a_n \quad f}{f(a_1, \ldots, a_n)}\,\mathrm{F_{app}} \qquad \frac{\mathtt{i} \to a_1 \quad \ldots \quad \mathtt{i} \to a_n \quad \mathtt{i} \to f}{\mathtt{i} \to f(a_1, \ldots, a_n)}\,\mathrm{F}_{\mathtt{i}\to\mathrm{app}}$$

$$\frac{{}^*a_1 \quad \ldots \quad {}^*a_n \quad {}^*f}{{}^*f({}^*a_1, \ldots, {}^*a_n)}\,\mathrm{F}_{{}^*\mathrm{app}}$$

sabemos interpretar expressões que sejam formadas por aplicação de funções e pelos símbolos lógicos $\land$, $\lor$, $\to$, $\neg$, $\top$ e $\bot$, já que $\land$, $\lor$ e $\to$ podem ser vistos como constantes de tipo $\tau, \tau \to \tau$, $\neg$ como uma constante de tipo $\tau \to \tau$ e $\top$ e $\bot$ como constantes de tipo τ, onde τ é o tipo dos valores de verdade. Por exemplo, $P(x) \land \neg Q(x, f(y))$ pode ser interpretada a partir de

$$\dfrac{\dfrac{x \quad P}{P(x)} \qquad \dfrac{\dfrac{x \quad \dfrac{y \quad f}{f(y)} \quad Q}{Q(x, f(y))} \quad \tau \overset{\neg}{\to} \tau}{\neg Q(x, f(y))} \qquad \tau, \tau \overset{\wedge}{\to} \tau}{P(x) \land \neg Q(x, f(y))} \quad .$$

A abstração pode ser interpretada exatamente como λ-cálculo. Por exemplo, a hiperfunção de tipo ${}^*(x \to \tau)$ que leva cada ${}^*x$ no hipervalor de verdade de (vamos começar a omitir estrelas e a usar uma árvore só para as três versões) $P(x) \land \neg Q(x, f(y))$ vem de

$$\dfrac{\dfrac{\dfrac{[x]^1 \quad P}{P(x)} \qquad \dfrac{\dfrac{[x]^1 \quad \dfrac{y \quad f}{f(y)} \quad Q}{Q(x, f(y))} \quad \tau \overset{\neg}{\to} \tau}{\neg Q(x, f(y))} \qquad \tau, \tau \overset{\wedge}{\to} \tau}{P(x) \land \neg Q(x, f(y))}}{\lambda x.(P(x) \land \neg Q(x, f(y)))}\, 1 \quad ;$$

essa árvore — até logo acima da última barra — pode ser vista como uma operação que leva um x num t, um $\mathbb{i} \to x$ num $\mathbb{i} \to \tau$ (exceto por um conjunto pequeno de índices) ou um ${}^*x$ num ${}^*\tau$; o fato é que essa operação $(\mathbb{i} \to x) \to (\mathbb{i} \to \tau)$ vem de uma operação interna $\mathbb{i} \to (x \to \tau)$, obtida lendo índice a índice a operação descrita pela árvore, que vai fazer sentido num conjunto grande de índices.

Usando a abstração temos um modo de interpretar os quantificadores *limitados*, que são aqueles que só quantificam sobre os pontos de um conjunto; por exemplo, em $\forall x.P(x)$ o '$\forall$' não é limitado, mas em $\forall x \in A.P(x)$ ele é. Seja P uma função de tipo $x \to \tau$; quando aplicamos "$\forall x$" a ela — e como estamos sempre usando variáveis tipadas "$\forall x$" é uma abreviatura para "$\forall x \in \mathbf{E}_x$", e os nossos quantificadores já são implicitamente limitados — e obtemos o valor de verdade de $\forall x.P(x)$, estamos passando de uma função de tipo $x \to \tau$ para um τ, ou seja, aplicar o quantificador '$\forall x$' corresponde a aplicar uma função $(x \to \tau) \overset{\forall_x}{\to} \tau$; essa função retorna $\top$ no caso em que a função $x \to \tau$ dada leva todo x em $\top$, e $\dashv$ em todos os outros casos. O quantificador '$\exists x$' corresponde a uma outra função do mesmo tipo, $(x \to \tau) \overset{\exists_x}{\to} \tau$, que só retorna $\dashv$ se a função $x \to \tau$ leva todo x em $\dashv$. Como para cada tipo x as funções associadas aos quantificadores '$\forall x$' e '$\exists x$' são constantes de $\mathbf{Set}_{\mathcal{A}}$ podemos encontrar constantes correspondentes em $\mathbf{Set}^{\mathbb{I}}$ e em $\mathbf{Set}^{\mathcal{F}}$ — até para tipos não-standard. Isso dá uma interpretação para os quantificadores, mas não é nada evidente que ela tenha as mesmas propriedades que os quantificadores usuais, nem mesmo que ela vá ser bem-comportada sob as regras de dedução da seção 3.4.

Pra piorar em $\mathbf{Set}^{\mathcal{U}}$ há uma segunda interpretação natural para os quantificadores, que parece não ter nada a ver com essa. Nessa nova interpretação $\forall {}^*x\,.\,{}^*P({}^*x)$ retorna ${}^*\top$ se e só se todo ${}^*x$ é levado em ${}^*\top$ por ${}^*P$, e $\exists\,{}^*x\,.\,{}^*P({}^*x)$ retorna ${}^*\top$ se e só se existe algum ${}^*x$ com ${}^*P({}^*x) = {}^*\top$. Em um $\mathbf{Set}^{\mathcal{F}}$ interpretações como essas soariam um pouco artificiais já que elas privilegiariam dois valores de verdade, $\top$ e $\dashv$ como hiperpontos, dentre muitos hipervalores de verdade, mas em $\mathbf{Set}^{\mathcal{U}}$ os hipervalores de verdade são só ${}^*\top$ e ${}^*\dashv$.

Todos os resultados que ficaram pendentes nesta seção — o teorema da transferência, a internalidade da abstração e a equivalência das duas interpretações para os quantificadores em $\mathbf{Set}^{\mathcal{U}}$ — são corolários triviais do teorema de Łós ou de lemas usados na sua demonstração. Demonstrações para ele podem ser encontradas em [SL] e em [Davis]; nós vamos dar só o seu enunciado, que é o seguinte: seja φ uma sentença em $\mathbf{Set}^{\mathcal{U}}$ de tipo ${}^*\tau$ tal que todos os seus quantificadores são limitados e as suas constantes são só, digamos, ${}^*a, \dots, {}^*z$. Escolha pré-hiperpontos $\mathbb{i} \to a, \dots, \mathbb{i} \to z$ como representantes para essas constantes; o valor de φ vai ser ${}^*\top$ se e só se o valor na interpretação via pré-hiperpontos for $\top$ num conjunto grande de índices.

7.5 Infinitesimais

Seja $X = \mathbf{E}_x$ um espaço topológico e x_0 um ponto de X. Para encurtar certas definições nós vamos chamar x_0 de o *ponto zero* do espaço X. Um espaço com um ponto escolhido para fazer o papel de ponto zero é chamado de um *espaço com ponto zero*. Em espaços vetoriais vamos considerar que o ponto zero é o próprio zero do espaço vetorial, exceto quando especificarmos explicitamente outro ponto zero.

Um *infinitesimal* de X, ou *um ponto infinitamente próximo de x_0 em X*, é um hiper-x que está $\mathcal{F}$-dentro de toda vizinhança standard de x_0. Um exemplo:

se $X = \mathbb{R}$, $x_0 = 0$, $\mathbb{I} = \mathbb{N}$ e $\mathcal{F} = \mathcal{N}$ então $(1, 1/2, 1/3, \dots)$ visto como hiperponto é um infinitesimal.

Um hiperreal é *infinito* quando seu módulo for $\mathcal{F}$-maior que qualquer real standard R, e *finito* quando existir um real R standard tal que o seu módulo seja $\mathcal{F}$-menor que R. Com os mesmos $\mathbb{I}$ e $\mathcal{F}$ de antes $(1, 2, 3, \dots)$ e $(1, -2, 3, -4, \dots)$ são infinitos, $(0, 1, 0, 1, \dots)$ é finito e $(1, 1/2, 3, 1/4, 5, \dots)$ não é nem finito nem infinito nem infinitesimal.

O *infinitesimal natural* de um espaço topológico X com ponto zero x_0 é construído da seguinte forma: tome como conjunto-índice o próprio X (i.e., $\mathbb{I} = X$), como $\mathcal{F}$ o filtro das vizinhanças de x_0 e como o (pré-)infinitesimal a função "identidade" indo de $\mathbb{I}$ em X, que leva cada ponto x nele próprio, mas muda o nome do tipo. Esse pré-hiperponto $\mathtt{i} \to x$ é claramente um infinitesimal, já que para cada vizinhança V de x_0 ele está $\mathcal{F}$-dentro de V pela própria definição de $\mathcal{F}$: os conjuntos grandes são exatamente as vizinhanças de x_0.

Lembre que a definição de continuidade pontual em espaços topológicos é que uma função $f : X \to Y$ é *contínua no ponto* $x_0 \in X$ (X e Y são espaços topológicos) se para toda vizinhança V de $f(x_0)$ a imagem inversa de V por f, $f^{-1}(V)$, for uma vizinhança de x_0; traduzindo isso para a nossa notação, se $X = \mathbf{E}_x$ e $Y = \mathbf{E}_y$ são espaços topológicos com pontos zero — e a nossa convenção vai ser que o ponto zero de um $\mathbf{E}_\alpha$ vai ser sempre chamado de α_0 e que uma função $\alpha \to \beta$ entre dois espaços com ponto zero leva o ponto zero de um no ponto zero de outro — então $x \to y$ é contínua no ponto zero se e só se a imagem inversa de toda vizinhança de y_0 for uma vizinhança de x_0.

Filtros induzem topologias: é fácil verificar que para qualquer filtro $\mathcal{F}$ sobre $\mathbb{I}$ o conjunto $\mathcal{F} \cup \{ \varnothing \}$ obedece todas as condições para ser uma topologia sobre $\mathbb{I}$: é fechado por interseções finitas, por uniões arbitrárias, contém o $\varnothing$ e o $\mathbb{I}$. Essa topologia parece estranhíssima à primeira vista mas ela é ótima para estudar continuidade pontual. Para cada um dos nossos espaços com ponto zero defina a *topologia das vizinhanças do zero* nesse espaço como a induzida pelo filtro das vizinhanças do zero; se o espaço original era, digamos, $\mathbf{E}_x$, o espaço com os mesmos pontos mas com a topologia das vizinhanças do zero vai ser chamado de $\mathbf{E}_{x'}$; as funções naturais $x \to x'$ e $x' \to x$ são a identidade nos pontos mas mudam a topologia; em geral nem $x \to x'$ nem $x' \to x$ são contínuas. Vamos considerar que $\mathbf{E}_{\mathtt{i}'}$ é $\mathbb{I}$ com a topologia induzida pelo filtro $\mathcal{F}$.

Teorema: digamos que para um pré-hiperponto $\mathbb{i} \to x$ toda vizinhança de x_0 tenha imagem inversa não-vazia. Então $\mathbb{i} \to x$ é infinitesimal se e só se $\mathbb{i}' \to x'$ for contínua.

Demonstração: abertos de $\mathbf{E}_{x'}$ são ou vizinhanças de x_0 ou o conjunto vazio; abertos de $\mathbf{E}_{\mathbb{i}'}$ são ou conjuntos grandes ou o vazio. A condição de que toda vizinhança de x_0 tem imagem inversa não-vazia garante que o caso vazio e o caso não-vazio não se misturam quando olhamos para imagens inversas de abertos de $\mathbf{E}_{x'}$: a imagem inversa do vazio é o vazio, a imagem inversa de um aberto não-vazio é um conjunto não-vazio, não necessariamente aberto.

Com isso $\mathbb{i} \to x$ é infinitesimal se e só se a imagem inversa de cada vizinhança de x_0 é um conjunto grande, ou seja, se a imagem inversa de cada aberto não-vazio de $\mathbf{E}_{x'}$ é um aberto não-vazio de $\mathbf{E}_{\mathbb{i}'}$, ou seja, se $\mathbb{i}' \to x'$ é contínua.

Repare que só precisamos da condição de que as vizinhanças de x_0 têm imagem inversa não-vazia para demonstrar a direção ($\Leftarrow$); mesmo sem essa condição $\mathbb{i} \to x$ ser infinitesimal implica em que $\mathbb{i}' \to x'$ é contínua.

Teorema: $x \to y$ é contínua em x_0 se e só se $x' \to y'$ é contínua.

Demonstração: ($\Rightarrow$): se $x \to y$ é contínua em x_0 a imagem inversa de toda vizinhança de y_0 é uma vizinhança de y_0, e portanto a imagem inversa de todo aberto não-vazio de $\mathbf{E}_{y'}$ é um aberto não-vazio de $\mathbf{E}_{x'}$. A imagem inversa do vazio é o vazio, e portanto todo aberto de $\mathbf{E}_{y'}$ é puxado por $(x \to y)^{-1}$ a um aberto de $\mathbf{E}_{x'}$, e daí $x' \to y'$ é contínua.

($\Leftarrow$): se $x' \to y'$ é contínua cada aberto não-vazio de $\mathbf{E}_{y'}$ vai num aberto de $\mathbf{E}_{x'}$, que sabemos que não é vazio porque o x_0 vai em y_0 por $x \to y$, e portanto cada vizinhança de y_0 vai numa de x_0 e $x \to y$ é contínua em x_0.

Teorema: se $\mathbb{i} \to x$ é um infinitesimal e $x \to y$ é contínua em x_0 então $\mathbb{i} \to y = \mathbb{i} \to x \to y$ é um infinitesimal; funções contínuas levam infinitesimais em infinitesimais.

Demonstração: como $x \to y$ é contínua em a imagem inversa de cada vizinhança de y_0 é uma vizinhança de x_0; como $\mathbb{i} \to x$ é infinitesimal a imagem inversa de qualquer vizinhança de x_0 é um conjunto grande de índices.

7.6 Infinitesimais naturais

Dizemos que um infinitesimal $\mathbb{i} \to x$ *percebe continuidade pontual* quando para todo espaço $\mathbf{E}_y$ com ponto 0 e toda função $x \to y$ que leve x_0 em y_0 tenhamos que $\mathbb{i} \to y = \mathbb{i} \to x \to y$ não é um infinitesimal quando $x \to y$ não for contínua em x_0. Com um infinitesimal $\mathbb{i} \to x$ que perceba continuidade pontual podemos verificar que uma $x \to y$ é contínua em x_0 só verificando que $\mathbb{i} \to y$ é um infinitesimal: $x \to y$ é contínua em x_0 se e só se $\mathbb{i} \to y$ é um infinitesimal.

Não é difícil ver que um infinitesimal $\mathbb{i} \to x$ percebe continuidade pontual se e só se a imagem de $\mathbb{I}$ por $\mathbb{i} \to x$ é uma vizinhança de x_0 e a imagem inversa de qualquer não-vizinhança de x_0 é um conjunto de índices não-grande. Se $\mathbb{I} = \mathbf{E}_x$ e o filtro em $\mathbb{I}$ é o filtro de vizinhanças de x_0 então a "identidade" $\mathbb{i} \to x$ é um infinitesimal que percebe continuidade pontual; se $\mathbb{I} = \mathbf{E}_x \times \mathbf{E}_y \times \mathbf{E}_z$ e o filtro é o das vizinhanças de (x_0, y_0, z_0) na topologia produto então os $\mathbb{i} \to x$, $\mathbb{i} \to y$, $\mathbb{i} \to z$, $\mathbb{i} \to (x, y)$, $\mathbb{i} \to (x, z)$, $\mathbb{i} \to (y, z)$, $\mathbb{i} \to (x, y, z)$ naturais são infinitesimais que percebem continuidade pontual; se $\mathbb{I}$ fosse só uma vizinhança de (x_0, y_0, z_0) ao invés de ser o espaço $\mathbf{E}_{(x,y,z)}$ inteiro esses '$\mathbb{i} \to \dots$' também seriam infinitesimais que percebem continuidade pontual.

Essas construções mostram que não só existem hiperpontos infinitamente próximos de qualquer x_0 que não são equivalentes a x_0, como que esses hiperpontos podem ser construídos explicitamente, e a construção natural tem certas propriedades boas: a de perceber continuidade pontual, e uma outra, que não vamos discutir muito: *universalidade*. Como um infinitesimal natural $\mathbb{i} \to x$ é a identidade, todo infinitesimal $\mathbb{j} \to x$ se fatora de exatamente um modo através de $\mathbb{i} \to x$, i.e., existe exatamente uma seta $\mathbb{j} \to \mathbb{i}$ com $\mathbb{j} \to x = \mathbb{j} \to \mathbb{i} \to x$; acaba que uma demonstração usando o infinitesimal natural gera, de modo facílimo, todas as demonstrações semelhantes usando infinitesimais quaisquer.

Na verdade a construção que nos leva ao infinitesimal natural é um caso particular de uma construção bem mais geral, que é bem conhecida em análisa não-standard com o nome de *teorema da concorrência*. Uma relação $(a, b) \to \tau$ é dita *concorrente* quando para todo subconjunto finito do espaço dos as, $A = \{ a_1, \dots, a_n \}$, existe um b ("b_A") tal que $a_1 R b_A, \dots, a_n R b_A$. Toda relação concorrente induz um filtro $\mathcal{F}$ sobre $\mathbb{I} = \mathbf{E}_b$ tal que o pré-hiper-b natural, a identidade $\mathbb{i} \to b$, que vamos chamar de $b^\natural$, obedece $aRb^\natural$ para todo a standard, ou seja, $aRb^\natural$ é $\mathcal{F}$-verdade para todo a standard.

O filtro $\mathcal{F}$ é definido do seguinte modo: para cada $a \in \mathbf{E}_a$ defina $B_a := \{\, b \mid aRb \,\}$; pela hipótese da relação ser concorrente nenhuma interseção finita de B_as é vazia, então os B_as são a pré-base de um filtro; seja $\mathcal{F}$ esse filtro, i.e., $\mathcal{F} = \uparrow \bigcap_{\text{fin}} \{\, B_a \mid a \in \mathbf{E}_a \,\}$.

Exemplo: digamos que cada a seja um aberto de X contendo x_0, que cada b seja um ponto de X e que a relação R seja '$\ni$'. Então $b^\natural$ é o infinitesimal natural de X.

Outro exemplo: os 'a's ainda são abertos de X contendo x_0, mas agora os 'b's são vizinhanças de x_0 e R é '$\supset$'; $b^\natural$ vai ser uma vizinhança infinitamente pequena de x_0, i.e., uma que está $\mathcal{F}$-contida em qualquer aberto contendo x_0.

7.7 Um exemplo de tradução

Vamos ver um exemplo muito simples de uma demonstração usando contas com infinitesimais e a sua tradução para uma demonstração clássica. Digamos que queremos ver que $\lim_{n\to\infty}(1+\frac{x}{n})^n = e^x$:

$$\begin{aligned}
\log(1+x/\omega_0)^{\omega_0} &= \omega_0\ \log(1+x/\omega_0) \\
&= \omega_0\ (\log 1 + ((\log' 1) + \epsilon_1)\ x/\omega_0) \\
&= \omega_0\ (0 + (1+\epsilon_1)\ x/\omega_0) \\
&= \omega_0\ (1+\epsilon_1)\ x/\omega_0 \\
&= (1+\epsilon_1)\ x
\end{aligned}$$

$$\begin{aligned}
(1+x/\omega_0)^{\omega_0} &= e^{((1+\epsilon_1)\ x)} \\
&= e^{(x+\epsilon_1 x)} \\
&= e^{(x+\epsilon_2)} \\
&= e^x + \epsilon_3.
\end{aligned}$$

Começamos trocando n por um (hiper)natural infinitamente grande, ω_0. Queremos ver que $(1+\frac{x}{\omega_0})^{\omega_0}$ está infinitamente próximo de e^x, então tomamos o logaritmo dos dois lados e vemos que existe (pela série de Taylor do logaritmo no ponto 1; vamos ver os detalhes daqui a pouco) um infinitesimal ϵ_1 que obedece a segunda igualdade acima, depois, porque x é finito, um infinitesimal ϵ_2 que obedece a penúltima igualdade e, porque a exponencial é contínua no ponto x, um ϵ_3 que obedece a última igualdade.

[Falta explicar melhor que os hiperpontos se comportam como pontos por dois motivos: índice a índice e pelo teorema da transferência. Falta grudar a parte anterior nisso que vem agora e bater a parte sobre cálculo até uma certa precisão.]

Fixe um infinitesimal ϵ. Vou dizer que eu *sei α até ordem n* quando eu tiver uma aproximação α' de α que difira de α por um valor da forma ϵo, onde o é um infinitesimal; ou seja, quando existir um infinitesimal o tal que $|\alpha' - \alpha| = \epsilon^n o$. Se eu sei α até ordem 0 eu conheço α módulo um erro infinitesimal qualquer; se eu conheço α até ordem 2 eu conheço α módulo um erro da forma $\epsilon^2 o$, onde o pode ser qualquer infinitesimal.

Se uma função $f : x \to y$ standard é duas vezes derivável em um ponto standard x_0 e dx é um infinitesimal de ordem 2, i.e., um da forma $\epsilon^2 O$, onde O é um finito qualquer, então podemos usar as derivadas de f em x_0 para obter uma aproximação de ordem 2 para $f(x_0 + dx)$. Vamos definir $y_0 := f(x_0)$, $y_{x0} := f'(x_0)$, $y_{xx0} := f''(x_0)$, $x_1 := x_0 + dx$, $y_1 := f(x_1)$, $y_{x1} := f'(x_1)$ (se fizer sentido), e assim por diante; aí $y_0 + y_{x0}dx + \frac{y_{xx0}}{2}dx^2$ é uma aproximação de ordem 2 para y_1, e para verificar isso basta ver que a condição standard

$$\lim_{dx\to 0} \frac{f(x_0+dx) - \left(f(x_0) + f'(x_0)\,dx + \frac{f''(x_0)}{2}\,dx^2\right)}{|dx|^2} = 0$$

é equivalente à seguinte: para todo dx infinitesimal existe um infinitesimal o tal que

$$f(x_0+dx) - \left(f(x_0) + f'(x_0)\,dx + \frac{f''(x_0)}{2}\,dx^2\right) = o\,|dx|^2 \,.$$

Também podemos passar o o para junto do $\frac{f''(x_0)}{2}$, e aí temos, já usando as definições para $x_1, y_{x0}, \ldots,$

$$y_1 = y_0 + y_{x0}\,dx + \left(\frac{y_{xx0}}{2} + o'\right)\,dx^2,$$

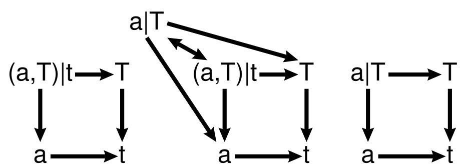
onde esse o' é um outro infinitesimal, que talvez tenha outro tipo.

Chapter 8

Topoi

Dizemos que um objeto $\mathbf{O}_\tau$ de uma categoria é um *objeto classificador* se ele se comporta como o espaço dos valores de verdade de **Set** e de $\mathbf{Set}_\mathcal{A}$, no sentido de que para qualquer espaço $\mathbf{O}_a$ há uma bijeção natural entre as funções que levam cada a num valor de verdade e os subconjuntos do espaço de as; vamos dar a definição formal aos poucos. Em $\mathbf{Set}_\mathcal{A}$ para qualquer par de setas $a \to c$ e $b \to c$ podemos formar o pullback delas, que é um espaço $(a,b)|_c$ junto com as projeções naturais $(a,b)|_c \to a$ e $(a,b)|_c \to b$; $\mathbf{E}_{(a,b)|_c}$ é definido como sendo o conjunto dos pares formados por um a e um b que são levados no mesmo c.

Se formamos o pullback de uma função qualquer $a \to \tau$ com a inclusão natural $\top \hookrightarrow \tau$, obtemos um espaço $(a,\top)|_\tau$ e duas projeções naturais; uma é $(a,\top)|_\tau \to \top$, que é trivial, já que existe exatamente uma seta $x \to \top$ para todo tipo x; a outra, $(a,\top)|_\tau \to a$, é praticamente a inclusão natural $a|_\top \hookrightarrow a$, onde $\mathbf{E}_{a|_\top}$ é o conjunto dos as que são levados em $\top \in \mathbf{E}_\tau$ pela função $a \to \tau$; não é difícil ver que existem bijeções naturais $(a,\top)|_\tau \leftrightarrow (a|_\top,\top) \leftrightarrow a|_\top$, e que o diagrama do meio, abaixo, comuta se pomos nele as setas naturais; portanto o diagrama abaixo à direita também é um pullback. Dissemos que $(a,\top)|_\top \hookrightarrow a$ é “praticamente” $a|_\top \hookrightarrow a$ porque usando a bijeção $(a,\top)|_\top \leftrightarrow a|_\top$ cada uma das inclusões determina a outra.



Definição: um *topos* é uma categoria que tem limites finitos, colimites finitos, exponenciais e um objeto classificador. A definição formal de objeto classificador é a seguinte: um objeto $\mathbf{O}_\tau$, junto com um morfismo $\top \hookrightarrow \tau$, onde $\top$ é um terminal, é um objeto classificador se para todo monic $a' \hookrightarrow a$ existe exatamente uma seta $a \to \tau$, que chamamos de *função característica* de $a' \hookrightarrow a$, tal que o quadrado formado por $a' \to a$, $a \to \tau$, $\top \to \tau$ e $a' \to \top$ é um pullback:

$$\begin{tikzcd}
\mathsf{a'} \arrow[r] \arrow[d] & \mathsf{T} \arrow[d] \\
\mathsf{a} \arrow[r] & \mathsf{t}
\end{tikzcd}$$

Numa categoria com limites sempre podemos tomar tomar o pullback de duas setas com o mesmo codomínio, e portanto sempre podemos fazer a operação $(a \to \tau) \to (a' \hookrightarrow a)$. A seta $a' \hookrightarrow a$ que obtemos desse modo só é única módulo isomorfismo: se $a'' \hookrightarrow a$ é outra seta obtida do mesmo modo então existe exatamente um isomorfismo $a' \leftrightarrow a''$ tal que $a'' \hookrightarrow a = a'' \to a' \hookrightarrow a$. O que acontece no caso em que a parede direita do quadrado é $\top \hookrightarrow \tau$, o classificador, é que também podemos ir na outra direção: $(a' \hookrightarrow a) \to (a \to \tau)$, e, exceto pelo fato de que o '$a' \to$' só está determinado módulo isomorfismo, o classificador dá, para qualquer tipo a, uma bijeção entre setas $a \to \tau$ e monics com codomínio a.

Primeiro fato importante: Set e os $\mathbf{Set}_\mathcal{A}$, $\mathbf{Set}^\mathbb{I}$, $\mathbf{Set}^\mathcal{F}$ e $\mathbf{Set}^\mathcal{U}$ são topoi. Em $\mathbf{Set}^\mathbb{I}$ o $\mathbf{O}_\tau$ é o conjunto dos pré-hipervalores de verdade, i.e., das funções de $\mathbb{I}$ em $\{\top, \dashv\}$; em $\mathbf{Set}^\mathcal{F}$ e $\mathbf{Set}^\mathcal{U}$ o $\mathbf{O}_\tau$ é o $\mathbf{O}_\tau$ de $\mathbf{Set}^\mathbb{I}$ módulo o quociente pela relação de equivalência.

Vamos dizer que um *ponto de tipo* α num topos é um morfismo $\top \to \alpha$. Um *subobjeto de* $\top$ é um objeto $\top'$ tal que a seta $\top' \to \top$ (que como $\top$ é um terminal sempre existe e é única) é um monic.

Um pré-hipertipo α em $\mathbf{Set}^\mathbb{I}$ é uma seqüência de tipos standard $(\alpha_1, \alpha_2, \alpha_3, \dots)$; vamos fingir que $\mathbb{I} = \mathbb{N}$ para simplificar a notação. Um ponto a de tipo α em $\mathbf{Set}^\mathbb{I}$ é uma seqüência $(a_1, a_2, \dots)$ em que $a_1 \in \mathbf{E}_{\alpha_1}$, $a_2 \in \mathbf{E}_{\alpha_2}$, ...; quando vemos esse ponto como um morfismo ele é a seqüência de funções $(\top \mapsto a_1, \top \mapsto a_2, \dots)$.

É facil ver que um subobjeto $\top'$ de $\top$ em $\mathbf{Set}^\mathbb{I}$ é um pré-hipertipo $(\top'_1, \top'_2, \dots)$, onde cada $\mathbf{E}_{\top'_i}$ é ou um singleton ou um conjunto vazio. Se todos os $\mathbf{E}_{\top'_i}$ singletons forem $\{\top\}$, então a seqüência $(\top'_1, \top'_2, \dots)$ é uma seqüência de '$\top$'s e '$\bot$'s; os pré-hipertipos $\top$ e $\bot$ de $\mathbf{Set}^{\mathbb{I}}$ são as seqüências constantes de tipos $(\top, \top, \top, \dots)$ e $(\bot, \bot, \bot, \dots)$. Se $\top'$ é uma seqüência não-constante de '$\top$'s e '$\bot$'s, como por exemplo $\top' = (\top, \bot, \top, \bot, \dots)$, então quando virmos $\top'$ como hipertipo em $\mathbf{Set}^{\mathcal{U}}$ ele vai ser equivalente a $(\top, \top, \top, \dots)$ ou $(\bot, \bot, \bot, \dots)$ vistos como hipertipos, mas isso não é necessariamente verdade em $\mathbf{Set}^{\mathcal{F}}$; se $\mathcal{F} = \mathcal{N}$ então o $\top'$ que tomamos como exemplo está entre $\bot$ e $\top$: temos $\bot \hookrightarrow \top' \hookrightarrow \top$, mas não temos setas $\top \to \top'$ ou $\top' \to \bot$. O fato é que em qualquer topos a categoria formada pelos subobjetos de $\top$, com os morfismos herdados do topos, é uma álgebra de Heyting, que dizemos que é a *álgebra dos valores de verdade* daquele topos. Em $\mathbf{Set}$, $\mathbf{Set}_{\mathcal{A}}$ e $\mathbf{Set}^{\mathcal{U}}$ a álgebra dos valores de verdade tem exatamente dois objetos, $\top$ e $\dashv$, se identificamos os objetos isomorfos; nos $\mathbf{Set}^{\mathbb{I}}$ e $\mathbf{Set}^{\mathcal{F}}$ a álgebra dos valores de verdade costuma ser bem maior.

Dizemos que um morfismo $\top' \to \alpha$, onde $\top'$ é um subobjeto do $\top$, é um $\top'$-*subponto de* α, ou simplesmente um *subponto de* α; subpontos de α em $\mathbf{Set}^{\mathbb{I}}$ correspondem a funções parciais de $\mathbb{I}$ nos $\mathbf{E}_{\alpha_i}$, definidas exatamente nos índices em que o $\mathbf{E}_{\top'_i}$ é um singleton. Um subponto de α é *trivial* quando ele é da forma $\bot \to \alpha$; se alguns $\mathbf{E}_{\alpha_i}$ são vazios e outros não então em $\mathbf{Set}^{\mathbb{I}}$ existem subpontos não-triviais de α em mas não existem pontos de tipo α.

Para quaisquer dois morfismos diferentes $\alpha \xrightarrow{f} \beta$ e $\alpha \xrightarrow{g} \beta$ existe um subponto $\top' \hookrightarrow \alpha$ que os distingue, i.e., tal que $\top' \hookrightarrow \alpha \xrightarrow{f} \beta \neq \top' \hookrightarrow \alpha \xrightarrow{g} \beta$; ou seja, a classe dos subobjetos do $\top$ é uma família de geradores, no sentido da seção 4.10. Um topos com essa propriedade de que os subobjetos do $\top$ formam uma família de geradores é dito *subextensional*, e um topos em que o $\top$ é um gerador é dito *extensional*. $\mathbf{Set}$, $\mathbf{Set}_{\mathcal{A}}$ e $\mathbf{Set}^{\mathcal{U}}$ são extensionais, $\mathbf{Set}^{\mathbb{I}}$ e $\mathbf{Set}^{\mathcal{F}}$ são subextensionais; veja [Bell], p.146 em diante, para algumas conseqüências da subextensionalidade ou extensionalidade de um topos. $\mathbf{Set}$, $\mathbf{Set}_{\mathcal{A}}$, $\mathbf{Set}^{\mathbb{I}}$, $\mathbf{Set}^{\mathcal{F}}$ e $\mathbf{Set}^{\mathcal{U}}$ também são *booleanos*; confira [Bell], p.147, ou qualquer outro texto sobre topoi citado na bibliografia.

Existem topoi que não são nem booleanos nem subextensionais; os $\mathbf{Set}^{\mathbf{P}}$, as categorias das transformações naturais de $\mathbf{P}$ em $\mathbf{Set}$, onde $\mathbf{P}$ é uma pré-ordem, são topoi e são não-booleanos quando em $\mathbf{P}$ há morfismos entre objetos não-isomorfos; os $\mathbf{Set}^{\mathbf{M}}$, construídos da mesma forma mas como $\mathbf{M}$ sendo um monóide visto como categoria, também são sempre topoi, e só são subextensionais para certos $\mathbf{M}$ triviais. Não vamos entrar em detalhes.

Segundo fato importante: é possível interpretar não só λ-cálculo como também qualquer sentença em lógica de primeira ordem com quantificadores limitados (e com funções) em qualquer topos; essa interpretação respeita tudo que pode ser demonstrado via deduções no sistema DN com quantificadores, num sentido que vamos ver daqui a pouco.

O axioma do classificador estabelece uma bijeção natural entre valores de verdade de um topos, i.e., pontos de tipo τ, ou seja, setas $\top \to \tau$, e subobjetos de $\top$ ("$\top'$"); em princípio o subobjeto associado a em τ só está definido módulo isomorfismo, mas em $\mathbf{Set}^{\mathbb{I}}$, por exemplo, se exigimos que cada tipo singleton da seqüência $(\top'_1, \top'_2, \dots)$ seja $\top$ então o $\top'$ fica bem-definido; os $\top'_i$ são $\top$ quando $\top_i \mapsto \top$ e $\bot$ quando $\top_i \mapsto \dashv$. Usando a terceira interpretação, que apareceu na seção 3.3, cada proposição φ está associada a um subobjeto de $\top$, $\top\varphi$; usando a bijeção natural vamos associá-la também a um valor de verdade, τ_φ.

Chapter 9

Onde nós gostaríamos de chegar

Este capítulo é na verdade apenas um apêndice com algumas anotações, assumidamente incompletas, sobre possíveis direções em que a teoria pode ser desenvolvida.

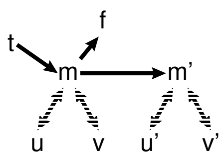
Variedades

Considere uma variedade M, com cartas $(\varphi_i)_{i \in I}$, onde cada carta φ_i é um homeomorfismo indo de um aberto $U_i \subset \mathbb{R}^n$ em M. As cartas também vão ser chamadas de *parametrizações*. Vamos querer considerar funções suaves indo de $\mathbb{R}$ em M, de M em $\mathbb{R}$ e de M em M', onde M' é uma outra variedade; pra simplificar vamos ter só uma função de cada um desses tipos: $\gamma : \mathbb{R} \to M$, $f : M \to \mathbb{R}$, $F : M \to M'$. As variedades M e M' vão ser respectivamente o "espaço dos ms" e o "espaço dos m's"; o $\mathbb{R}$ que é o domínio de γ é o "espaço dos ts" e o outro, o contradomínio de f, vai ser o "espaço dos fs" (lembre de como os físicos costumam dizer "$y = y(x)$": o y função é uma função de tipo $x \to y$).

Vamos estar interessados em estudar a vizinhança de um certo ponto m_0 de M, e muitas vezes vamos querer usar parametrizações para isso. Como vamos querer falar também de *mudanças de parametrização* vamos considerar duas parametrizações ao mesmo tempo, φ_i e φ_j; vamos dizer que uma parametrização φ_i é *adequada* (ao m_0) quando ela cobre uma vizinhança de m_0, i.e., quando

$m_0 \in \varphi_i(U_i)$. Queremos que tanto φ_i quanto φ_j sejam adequadas, mas exceto por isso elas são parametrizações arbitrárias; na verdade φ_i e φ_j vão ser variáveis que assumem valores no espaço das parametrizações adequadas.

Uma vez escolhidas essas duas parametrizações podemos definir os domínios das funções que vamos usar como sendo os seguintes: $\mathbf{E}_m := \varphi_i(U_i) \cup \varphi_j(U_j)$; $\mathbf{E}_u := \varphi_i^{-1}(\mathbf{E}_m)$, e $\mathbf{E}_v := \varphi_j^{-1}(\mathbf{E}_m)$. As restrições das funções φ_i e φ_j a $\mathbf{E}_u$ e $\mathbf{E}_v$ dão bijeções entre $\mathbf{E}_u$ e $\mathbf{E}_v$ e $\mathbf{E}_m$. As funções que nos interessam passam a ser as desse diagrama aqui:



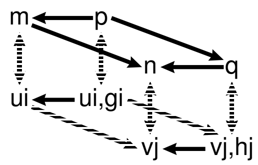
Cada seta cinza $\alpha \to \beta$ nesse diagrama representa uma função *parcial* de $\mathbf{E}_\alpha$ em $\mathbf{E}_\beta$, definida num aberto contendo o ponto α_0. Como por enquanto queremos evitar funções parciais estamos discutindo como restringir cada espaço $\mathbf{E}_\alpha$.

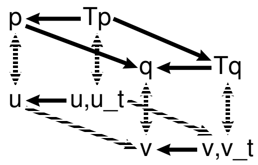
Quando estivermos considerando a função $t \to m$ provavelmente vamos ter um t preferido, t_0; o m_0 é escolhido como sendo a imagem dele, e o $\mathbf{E}_t$ fica definido como sendo a imagem inversa de $\mathbf{E}_m$. Quando estivermos interessados na função $m \to m'$ vamos querer tomar duas parametrizações em M' adequadas a m'_0 (onde obviamente m'_0 é a imagem de m_0), $\varphi'_{i'}$ e $\varphi'_{j'}$. Os espaços $\mathbf{E}_m$, $\mathbf{E}_u$ e $\mathbf{E}_v$ vão ter que ser ajustados para que a imagem do $\mathbf{E}_m$ caiba dentro de $\mathbf{E}_{m'}$.

Imagino que não vá ser difícil demais encontrar um algoritmo que, para problemas formulados em termos de diagramas como o acima, escolha domínios bons e traduza uma demonstração feita com ts, ms, us, vs, m's, u's, v's e fs numa demonstração feita em notação convencional. E repare que se usarmos infinitesimais, no sentido do capítulo 8, o que esse diagrama diz é que para cada seta $\alpha \to \beta$ *todo ponto infinitamente próximo de* α_0 *está sendo levado num ponto infinitamente próximo de* β_0, e para as setas $\alpha \leftrightarrow \beta$ temos ainda mais: cada uma delas dá uma *bijeção* entre os pontos infinitamente próximos de α_0 e os infinitamente próximos de β_0. Antes íamos de vizinhanças suficientemente pequenas de α_0 em vizinhanças suficientemente pequenas (mas também suficientemente grandes, para conterem a imagem das vizinhanças anteriores) de β_0; agora basta nos restringimos a pontos infinitamente próximos dos pontos

0 e a funções contínuas e os problemas de domínio evaporam na nossa mão, e só se materializam de novo na hora de traduzir a demonstração para uma convencional.

Fibrados





Fluxos

Como derivar uma função $f : p \to \mathbb{r}$ com relação a um campo vetorial $X : p \to (d\mathbb{r} \multimap dp)$? Bom, a função f induz uma outra função, $j_1 f$ ou jf (o "jato de ordem 1 de f"), que é de tipo $(p, dp) \to (\mathbb{r}, d\mathbb{r})$ mas que na verdade vem de uma função $p \to (\mathbb{r}, (dp \multimap d\mathbb{r}))$; vamos preferir essa segunda forma porque ela deixa em evidência a dependência linear de $d\mathbb{r}$ com relação a dp para p fixo, que vai ser muito importante. Descartando o $\mathbb{r}$ dela e compondo o $d\mathbb{r} \multimap dp$ que vem de X com o $dp \to d\mathbb{r}$ de jf obtemos $p \to (d\mathbb{r} \multimap dp \multimap d\mathbb{r})$, ou seja, $p \to (d\mathbb{r} \multimap d\mathbb{r})$; pela estrutura multiplicativa de $\mathbb{R}$ temos uma bijeção natural $\mathbb{r} \leftrightarrow (d\mathbb{r} \multimap d\mathbb{r})$, que é até linear em cada direção; usando-a obtemos uma função $p \to \mathbb{r}$, que é a derivada f_X que queríamos.

O teorema da função inversa

$$\frac{x \quad x \to y \quad \ldots}{y \to x}\ \text{TFI}$$

$$\dfrac{\dfrac{t \quad \dfrac{t \to x \land y}{t \to x}}{x \to t}\,\mathrm{TFI} \qquad t \to x \land y}{x \to y}$$

$$\dfrac{\dfrac{x \land y \quad x \land y \to f}{f} \qquad \dfrac{\dfrac{x \land y \quad \dfrac{x \land y \to f}{x \land y \to x \land f}}{x \land f \to x \land y}\,\mathrm{TFI}}{x \land f \to y}}{x \to y}$$

Uma migalha de lógica linear

O isomorfismo de Curry-Howard funciona tão bem em $\mathbf{Set}_{\mathcal{A}}$ que vamos tentar aplicá-lo a $\mathbf{Vect}_{\mathbb{R}}$, a categoria em que os objetos são os espaços vetoriais sobre $\mathbb{R}$ e os morfismos são as transformações lineares. As transformações lineares de um espaço vetorial em outro formam um espaço vetorial, então temos alguma esperança de que as coisas funcionem.

Considere a construção/dedução abaixo à esquerda. Se todos os espaços envolvidos são iguais a $\mathbb{R}$ podemos interpretar a aplicação como a multiplicação de dois números em $\mathbb{R}$; a árvore à direita mostra uma interpretação dessas.

$$\dfrac{\dfrac{\dfrac{[a]^1 \quad a \to b}{b} \quad \dfrac{[a]^1 \quad a \to (b \to c)}{b \to c}}{c}}{a \to c}\,1 \qquad\qquad \dfrac{\dfrac{\dfrac{[x]^1 \quad 2}{2x} \quad \dfrac{[x]^1 \quad 3}{3x}}{6x^2}}{x \mapsto 6x^2?}\,1$$

Só que a função que obtemos, $x \mapsto 6x^2$, é bilinear, não linear, e portanto não pode ser representada por um ponto de $\mathbb{R}$... um jeito de tentar resolver isso é passar para uma categoria em que cada ponto de tipo $x \to y$ seja um polinômio em x, ou uma série de potências que convirja suficientemente bem; por exemplo, cada $x \to y$ é uma função que seja analítica em todo o domínio $\mathbf{E}_x$. Vamos nos referir a elas como "polinômios" só pra encurtar. O espaço das '$x \to y$'s ainda é um espaço vetorial, mas a descrição de um espaço $\mathbf{E}_{x \to y}$ ficou bem mais complicada; pra piorar, dentro de $\mathbf{E}_{x \to y}$ ainda temos um subespaço cujos pontos queremos destacar: o espaço das transformações *lineares* de $x$zes em ys. Repare que temos um modo de separar a seta '$\to$' em duas operações para isolar toda a não-linearidade em uma delas; se $\mathbf{E}_x = \mathbb{R}$ então uma função bem-comportada que leve $x$zes em ys é linear sobre o espaço das uplas infinitas $(1, x, x^2, x^3, \dots)$; se $\mathbf{E}_x$ tem mais de uma dimensão precisamos de produtos tensoriais, e usamos as uplas $(1, x, x \otimes x, x \otimes x \otimes x, \dots)$. Aí, usando '!' para a operação que leva cada x na upla correspondente e '$\multimap$' para indicar dependência linear, temos que cada $x \to y$ pode ser expresso como um $!x \multimap y$. Isso fica mais claro se pensamos em termos de séries de potências:

$$\dfrac{\dfrac{dx}{!dx} \quad !dx \multimap dy}{dy} \qquad \dfrac{\dfrac{dx}{(dx, dx^2, \dots)} \quad (y_x, \frac{y_{xx}}{2}, \dots)}{dy = y_x dx + \frac{y_{xx}}{2} dx^2 + \dots}$$

Estamos excluindo as zerésimas potências das uplas por conveniência.

Se ao invés de usarmos as operações '!' nós usamos as operações '$!_n$', onde por exemplo $!_3 dx := (dx, dx^2, dx^3, 0, 0, \dots)$, então obtemos aproximações até uma certa ordem do resultado original. Se temos um infinitesimal fixo, ϵ, e $|dx|$ é um número finito vezes ϵ, então uma aproximação até ordem n de um valor é uma cujo valor difere de um finito vezes ϵ do valor verdadeiro. Repare que em $!dx$ a ordem (nesse sentido) de cada membro da upla é maior que a do anterior; se só queremos calcular uma aproximação até ordem n de um valor que é o resultado de uma conta expressa por uma árvore, podemos anotar na raiz da árvore que ali queremos ordem n de precisão, e podemos subir com as anotações de precisão até as folhas da árvore; uma árvore com essas anotações passa a dizer coisas como "se eu sei dy até a ordem 3 e sei os primeiros três termos de $!dy \multimap dz$, o primeiro até ordem 2, o segundo até ordem 1 e o terceiro até ordem 0, então eu sei dz até ordem 3"... de novo, como na seção ([?]), temos uma situação em que a árvore original descreve uma situação simplificada, muito mais bem-comportada, e somos capazes de refinar essa árvore para que ela descreva uma situação real; além disso a árvore nos permite não fazer as contas até o último minuto: ela diz que é possível calcular um certo objeto a partir de certos outros e nós sabemos preencher os detalhes quando for necessário e encontrar uma expressão para esse objeto em termos dos anteriores; não é preciso mostrar explicitamente a expressão para ele para mostrarmos que sabemos quem ele é.

Outra coisa: em $\mathbf{Set}_\mathcal{A}$ tínhamos uma bijeção natural entre pares de funções $a \to b$ e $a \to c$ e funções $a \to b \wedge c$, e outra entre funções $a \to (b \to c)$ e função

$a \wedge b \to c$; tínhamos dois modos diferentes de fazer o $\wedge$ aparecer a partir do tipos construídos apenas com o '$\to$'.

$$\frac{a \to b \quad a \to c}{a \to b \wedge c} \qquad \frac{a \to (b \to c)}{a \wedge b \to c}$$

Se tentamos trocar esses '$\to$'s por '$\multimap$'s vemos que os dois '$\wedge$'s vão corresponder a coisas diferentes; um modo rápido, mas meio furado, de ver isso é olhar para as dimensões dos espaços. Digamos que $\mathbf{E}_a = \mathbb{R}^2$, $\mathbf{E}_b = \mathbb{R}^3$ e $\mathbf{E}_c = \mathbb{R}^4$; então $\dim \mathbf{E}_{a \multimap b} = 6$, $\dim \mathbf{E}_{a \multimap c} = 8$ e $\dim \mathbf{E}_{(a \multimap b,\, a \multimap c)} = 14 = \dim \mathbf{E}_a \cdot (\dim \mathbf{E}_b + \dim \mathbf{E}_c)$, mas $\dim \mathbf{E}_{b \multimap c} = 12$ e $\dim \mathbf{E}_{a \multimap (b \multimap c)} = 24 = \dim \mathbf{E}_a \cdot (\dim \mathbf{E}_b \cdot \dim \mathbf{E}_c)$. O primeiro $\mathbf{E}_{a \wedge b}$ vai ter a dimensão do produto direto de $\mathbf{E}_a$ e $\mathbf{E}_b$, o segundo a dimensão do produto tensorial de $\mathbf{E}_a$ e $\mathbf{E}_b$.

$$\frac{a \to b \quad a \to c}{a \to b \wedge c} \qquad \frac{a \to (b \to c)}{a \wedge b \to c}$$

O $\top$ também se duplica: o terminal, tal que para todo tipo α havia exatamente uma seta $\alpha \to \top$, passa a ter como espaço associado o $\mathbb{R}^0 = \{0\}$; o $\top$ tal que havia uma uma bijeção natural entre 'α's e '$\top \to \alpha$'s, para qualquer tipo α, passa a ter como espaço associado o $\mathbb{R}$; cada α vai na função $\mathbb{r} \multimap \alpha$ que leva 1 em α.

O que aconteceu foi que "enfraquecemos a lógica" de novo; passamos a exigir que ela obedecesse menos equações, já que vamos querer ter $\oplus \neq \otimes$ e $0 \neq 1$; com isso passamos a ter ainda mais modelos.

Essa lógica com '!', '$\multimap$', '$\otimes$' e '$\oplus$' se chama *lógica linear*; consulte [Girard] e [BPS] para introduções sérias a ela, ainda com ênfase na interpretação em espaços vetoriais. Ela tem vários outros modelos também.

9.1 Parametricidade

Um *teorema de parametricidade* é um que diz que qualquer termo A de tipo α que obedeça certas condições, como ser um combinador ou só usar certas constantes, obedece uma propriedade P_α, que não é totalmente trivial e que só depende do tipo α. Em [Wadler] há um teorema desses para um certo sistema de λ-cálculo polimórfico que admite uma regra de formação de tipos '[ ]', onde um ponto[1] de tipo $\alpha^{[\,]}$ é uma lista finita de pontos de tipo α.

[1]Modelos para λ-cálculo tipado polimórfico são difíceis de obter, como como os modelos para λ-cálculo não-tipado; veja [Wadler], p.4, para referências. Quando falamos de pontos

Vamos ver o enunciado do teorema. Precisamos de duas abreviações: α^2 é (α', α'') e α^P é $(\alpha', \alpha'') \to \tau$; α' e α'' não são necessariamente iguais. Para cada tipo α vamos obter uma relação R_α de tipo α^P, e para qualquer termo de tipo α tal que nenhuma das constantes que aparecem nele é polimórfica vamos ter $R_\alpha AA = \top$; "todo termo está relacionado consigo mesmo". Em geral vamos nos referir à relação R_α pelo tipo dela, α^P.

A relação α^P é construída mais ou menos indutivamente:

1) Se α é um tipo atômico então α^P é a relação que diz que cada ponto a^α só está relacionado consigo mesmo: $\alpha^P a_1^\alpha a_2^\alpha = \top$ se $a_1^\alpha = a_2^\alpha$, $\dashv$ se não. Repare que nesse caso os tipos α' e α'' ficam iguais.

2) A relação $(\alpha, \beta)^P$ é montada a partir de duas relações α^P e β^P. Se as relações α^P e β^P são $(\alpha', \alpha'') \to \tau$ e $(\beta', \beta'') \to \tau$ respectivamente, a relação $((\alpha', \beta'), (\alpha'', \beta'')) \to \tau$ que queremos responde $\top$ para $((a^{\alpha'}, b^{\beta'}), (a^{\alpha''}, b^{\beta''}))$ se e só se as relações α^P e β^P que já tínhamos respondem $\top$ para $(a^{\alpha'}, a^{\alpha''})$ e $(b^{\beta'}, b^{\beta''})$; "dois pares estão relacionados se e só se seus primeiros membros estão relacionados e seus segundos membros também".

3) A relação $\alpha^{[\,]P}$ é montada a partir de relação α^P da seguinte forma: se α^P é de tipo $(\alpha', \alpha'') \to \tau$, duas listas $(a_1^{[\,]})^{\alpha'[\,]}$ e $(a_2^{[\,]})^{\alpha''[\,]}$ estão relacionadas se e só se elas têm o mesmo comprimento e elementos correspondentes delas estão relacionados.

4) A relação $(\alpha \to \beta)^P$ é montada a partir de α^P e β^P e ela diz que duas funções, $(f')^{\alpha' \to \beta'}$ e $(f'')^{\alpha'' \to \beta''}$, estão relacionadas se e só se cada par $(a_1^{\alpha'}, a_2^{\alpha''})$ relacionado por α^P é levado num par $(b_1^{\beta'}, b_2^{\beta''}) = (f' a_1, f'' a_2)$ relacionado por β^P. Em notação muito curta isso é $(\alpha \to \beta)^P (\alpha \to \beta)^2 = \forall \alpha^2 . (\alpha^P \alpha^2 \to \beta^P \beta^2)$.

5) A relação $(\alpha \Rightarrow \alpha^F)^P$ é obtida a partir de F, onde F se comporta como a ação de um funtor sobre objetos, como na seção 4.4 (veja também a seção 2.9).

---

estamos obviamente nos referindo a pontos do modelo, uma lista de pontos é a representação da lista dentro do modelo, etc.

$$\dfrac{[\mathbf{T}_{a^2}]^1 \quad [(a \Rightarrow \dots)^2]^1 \quad \dfrac{[\mathbf{T}_{b^2}]^3 \quad [(b \Rightarrow \dots)^2]^3 \quad \dfrac{\dfrac{[a^P]^1 \quad [b^P]^2}{(a \to b)^P} \quad \dfrac{\dfrac{[a^P]^1}{a^{[\,]P}} \quad \dfrac{[b^P]^2}{b^{[\,]P}}}{(a^{[\,]} \to b^{[\,]})^P}}{((a \to b) \to (a^{[\,]} \to b^{[\,]}))^P}}{(b \Rightarrow ((a \to b) \to (a^{[\,]} \to b^{[\,]})))^P}\,3}{(a \Rightarrow b \Rightarrow ((a \to b) \to (a^{[\,]} \to b^{[\,]})))^P}\,1$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[(a \wedge b)^2]^1}{a^2} \quad a^P}{a^{P2}} \quad \dfrac{\dfrac{[(a \wedge b)^2]^1}{b^2} \quad b^P}{b^{P2}}}{a^{P2} \wedge b^{P2}}}{(a \wedge b)^2 \to a^{P2} \wedge b^{P2}}\,1}{(a \wedge b)^P}\,\eta \qquad \dfrac{\dfrac{\dfrac{[a^{[\,]2}]^1 \quad a^P}{a^{[\,]P2}}}{a^{[\,]2} \to a^{[\,]P2}}\,1}{a^{[\,]P}}\,\eta$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[a^2]^1 \quad a^P}{a^{P2}} \quad \dfrac{\dfrac{[a^2]^1 \quad [(a \to b)^2]^3}{b^2} \quad b^P}{b^{P2}}}{a^{P2} \to b^{P2}}}{\forall a^2.(a^{P2} \to b^{P2})}\,1}{(a \to b)^2 \to \forall a^2.(a^{P2} \to b^{P2})}\,3}{(a \to b)^P}\,\eta \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\mathbf{T}_{a^2}]^3 \quad [(a \Rightarrow a^F)^2]^4}{a^{F2}} \quad \dfrac{[a^P]^1}{a^{FP}}}{a^{FP2}}}{\forall a^P.a^{FP2}}\,1}{\forall \mathbf{T}_{a^2}.\forall a^P.a^{FP2}}\,3}{(a \Rightarrow a^F)^2 \to \forall \mathbf{T}_{a^2}.\forall a^P.a^{FP2}}\,4}{(a \Rightarrow a^F)^P}\,\eta$$

Bibliografia

[Borceux]: F. Borceux: *Handbook of Categorical Algebra*, Cambridge University Press, 1994 (3 vols.).

[BPS]: R. F. Blute, P. Panangaden, R. A. G. Seely: *Fock Space: A Model of Linear Exponential Types*, 1993. Preprint.

[Bell]: J. L. Bell: *Toposes and Local Set Theories, an Introduction*, Oxford University Press, 1988.

[Dummett]: M. Dummett: *Elements of Intuitionism*, Oxford University Press, 1977.

[Freyd72]: P. Freyd, *Aspects of Topoi*, Bull. Austral. Math. Soc. **7**, pp.1–76, 1972.

[Girard]: J. Y. Girard: *Geometry of Interaction, III: Accomodating the Additives*, 1995. Preprint.

[GLT]: J. Y. Girard, Y. Lafont e P. Taylor, *Proofs and Types*, Cambridge University Press, 1989.

[Goldblatt]: Goldblatt, R. I.: *Topoi: the Categorial Analysis of Logic.* North-Holland, 1984 (2ª ed.).

[HS]: J. R. Hindley e P. Seldin: *Introduction to Combinators and λ-Calculus*, Cambridge University Press, 1986.

[Jech]: T. J. Jech: *The Axiom of Choice*, North-Holland, 1973.

[Johnstone]: P. T. Johnstone: *Topos Theory*, Academic Press, 1977.

[LS]: Lambek, J. e Scott, P. J.: *Introduction to Higher Order Categorical Logic*, Cambridge University Press, 1986.

[Nelson]: E. Nelson: *Internal Set Theory: a New Approach to Nonstandard Analysis*, Bull. Am. Math. Soc. **83**, pp.1165–1195, 1977.

[Prawitz]: D. Prawitz, *Natural Deduction, a Proof-Theoretical Study*, Almqvist & Wiksell, 1965.

[Robinson]: A. Robinson, *Non-standard Analysis*, North-Holland, 1974 (2nd ed.).

[SL]: K. D. Stroyan, W. A. J. Luxembourg: *Introduction to the Theory of Infinitesimals*, Academic Press, 1979.

[TS]: A. S. Troelstra e H. Schwichtenberg: *Basic Proof Theory*, Cambridge University Press, 1996.

[Wadler]: P. Wadler: *Theorems for Free!*, 1989. Preprint.