Lógica

aula 20: O significado das regrinhas H1 e H2

Liduína não tinha como saber disso, claro.

Mas, era hoje que ela ia fazer a sua grande descoberta.

Quer dizer, no primeiro dia ela só conseguiu fazer manipulações mecânicas, o que é um tanto frustrante mesmo.

Mas, isso é importante!

Quer dizer, é assim que a gente aprende a usar as regras.

Ou melhor, as regras são usadas mecanicamente mesmo.

A gente usa elas sem pensar (depois que aprende).

E daí, a gente pode pensar em outra coisa ...

Quer dizer, isso foi o que Liduína entendeu no dia seguinte.

Pra que as regras serviam, e como elas eram usadas pra fazer outra coisa.

Você só entende o que é um martelo na hora em que você esquece que ele está na sua mão ...

Na verdade, as coisas só começaram a fazer sentido para Liduína quando ela parou de prestar atenção nas coisas, e passou a prestar atenção nos movimentos que ela estava fazendo com as coisas.

Quer dizer, quando ela passou a ver o jogo lógico de Hilbert com um *jogo de transformações*.

Foi daí que ela viu que a regrinha H2 servia pra fazer a operação de distribuição.

Isto é, para transformar coisas no formato

$$\mathtt{p} \;\rightarrow\; (\mathtt{q} \rightarrow \mathtt{r})$$

em coisas assim

$$(\mathtt{p} \rightarrow \mathtt{q}) \;\rightarrow\; (\mathtt{p} \rightarrow \mathtt{r})$$

E ela viu também que haviam dois truques de transformação que podia ser feitos com a regrinha H1.

O primeiro era colocar uma coisa qualquer na frente de uma coisa que a gente já construiu

$$\mathtt{p} \quad \Longrightarrow \quad \mathtt{q} \rightarrow \mathtt{p}$$

o que era a mesma coisa que a regra R2 do Ubiratan.

E o segundo era contruir uma coisa pra depois poder tirar ela da frente de uma outra coisa

$$\texttt{q} \to \texttt{p} \quad \Longrightarrow \quad \texttt{p} \qquad\qquad \text{, se a gente consegue construir } \texttt{q}$$

o que era um caso particular da regra `R3` do Ubiratan.

Mas não foi só isso!

Quer dizer, ela viu também que as regras que a gente constrói no jogo de Hilbert podem ser transformadas em regras de transformação (acoplando elas ao MP), pra depois serem usadas para construir outras regras.

E foi dessa maneira que ela descobriu

- a sua regra `L1`

$$\texttt{q} \to \texttt{r} \quad \stackrel{\texttt{L1}}{\Longrightarrow} \quad (\texttt{p} \to \texttt{q}) \to (\texttt{p} \to \texttt{r})$$

- a regra da inversão

$$\texttt{X} \to (\texttt{Y} \to \texttt{Z}) \quad \stackrel{\texttt{Inv}}{\Longrightarrow} \quad \texttt{Y} \to (\texttt{X} \to \texttt{Z})$$

- a regra da transitividade

$$\texttt{p} \to \texttt{q} \quad \text{e} \quad \texttt{q} \to \texttt{r} \quad \stackrel{\texttt{Trans}}{\Longrightarrow} \quad \texttt{p} \to \texttt{r}$$

Mas, apesar de ter aprendido e entendido tanta coisa, ainda haviam coisas confusas na cabeça de Liduína.

1. *Eu já num sei mais sei eu tou transformando construções.*

   Ou se eu tou construindo transformações.

   Ou se eu tou fazendo as duas coisas.

   Ou se eu só tou fazendo é confusão mesmo ...

   Quer dizer, no primeiro dia era tudo muito claro: eu tava construindo construções.

   Mas daí começou essa história de transformação.

   E ontem, já no final, parecia que eu tava transformando transformações.

   E então bagunçou tudo na minha cabeça ...

   Certo.

   Mas tem uma coisa que eu já entendi.

   Quer dizer, se eu quero entender o que eu tou fazendo, é só eu olhar o que eu tou fazendo.

   E pra fazer isso, a melhor coisa é eu começar do começo ...

No começo eu tava realmente construindo construções.

Usando as regrinhas `H1` *e* `H2` *do Hilbert, que deixam a gente construir coisas do nada.*

Quer dizer, do nada não ...

Elas vem das intuições lógicas do Hilbert ...

Mas isso só Deus sabe o que é ...

E o Hilbert também ...

Talvez ...

Mas o Hilbert já morreu ...

E agora eu tou aqui, quebrando a cabeça, tentando entender o que ele fez ...

Mas, deixa eu me concentrar de novo.

Então, eu tava construindo construções até que eu percebi que as regras de construção também são regras de transformação.

Quer dizer, quando a gente acopla elas ao Modus Ponens.

E foi daí que eu vi um novo jeito de fazer construções.

Quer dizer, eu vi que pra construir uma coisa, a gente pode construir primeiro uma outra coisa, e depois transformar essa outra coisa na coisa que a gente quer.

E isso foi uma boa coisa, porque foi desse jeito que eu consegui construir a minha regrinha `L1`*, que depois virou uma regra de transformação.*

Mas deixa eu ir mais devagar.

Nessa hora eu já tava transformando construções.

E eu tava fazendo isso como uma outra maneira de construir construções.

Quer dizer, a gente pode raciocinar de duas maneira diferentes:

- *montando e desmontando coisas, tentando juntar as partes da coisa que a gente quer*
- *transformando as coisas umas nas outras, tentando deixar as coisas cada vez mais parecidas com a coisa que a gente quer*

Hmm ...

Legal isso ...

Mas depois ficou mais legal ainda.

Quer dizer, depois eu vi que dava para juntar várias transformações e pensar nelas como uma transformação só, mais complicada.

Como um programa complicado, que a gente constrói com vários programas simples.

Mas então, se eu não tou enganada, nessa hora eu já tava construindo transformações.

E foi nessa hora que começou a enrolada ...

Quer dizer, ainda não foi nessa hora não.

Porque essa história de fazer transformações em sequência e depois empacotar elas em uma nova transformação ainda é uma ideia simples.

Mas, se não é nessa hora que as coisas ficam enroladas ...

Então só pode ser na hora que a gente pensa em transformar transformações em outras transformações ...

Pronto.

Agora Liduína tinha conseguido encontrar um foco para o seu pensamento.

E tudo o que ela precisava em seguida era de um bom exemplo.

2. *Hmm ...*

   Essa transformação `Inv` *que eu tenho aqui*

$$\mathtt{X} \ \rightarrow \ (\mathtt{Y} \rightarrow \mathtt{Z}) \qquad \xRightarrow{\mathtt{Inv}} \qquad \mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Z})$$

   até parece legal, mas ela é mó páia ...

   Quer dizer, ela deixa eu inverter a ordem da primeira e da segunda partes da regra, o que é legal ...

   Mas eu não consigo inverter a segunda e a terceira partes de uma regra com ela, o que é mó páia ...

   Então, eu queria transformar a minha transformação `Inv` *numa transformação* `Inv 2.0` *que fizesse por exemplo assim*

$$\mathtt{X} \ \rightarrow \ \big(\mathtt{Y} \ \rightarrow \ (\mathtt{Z} \rightarrow \mathtt{W})\big) \qquad \xRightarrow{\text{Inv 2.0}} \qquad \mathtt{X} \ \rightarrow \ \big(\mathtt{Z} \ \rightarrow \ (\mathtt{Y} \rightarrow \mathtt{W})\big)$$

   E agora eu tenho um problema pra resolver.

3. *Hmm ...*

   Eu podia lembrar o jeito que eu fiz o `Inv`

$$\begin{aligned} & \mathtt{X} \ \rightarrow \ (\mathtt{Y} \rightarrow \mathtt{Z}) \\ & \quad \xRightarrow{\mathtt{Dist}} \quad (\mathtt{X} \rightarrow \mathtt{Y}) \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Z}) \\ & \quad \xRightarrow{\mathtt{L1}} \quad \big(\mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Y})\big) \ \rightarrow \ \big(\mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Z})\big) \\ & \quad \xRightarrow{\mathtt{R3}} \quad \mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Z}) \end{aligned}$$

E daí, eu podia tentar fazer parecido, pra sair daqui

$$\mathtt{X} \;\to\; \big(\mathtt{Y} \;\to\; (\mathtt{Z} \to \mathtt{W})\big)$$

e chegar aqui

$$\mathtt{X} \;\to\; \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{W})\big)$$

4. *Hmm ...*

   Mas não dá.

   O X *tá atrapalhando ...*

   (1) *Quer dizer, eu ia ter que fazer a distribuição na parte de dentro*

   $$\mathtt{X} \;\to\; \big(\mathtt{Y} \;\to\; (\mathtt{Z} \to \mathtt{W})\big)$$

   $$\xRightarrow{\mathtt{Dist}} \quad \mathtt{X} \;\to\; \big((\mathtt{Y} \to \mathtt{Z}) \;\to\; (\mathtt{Y} \to \mathtt{W})\big)$$

   o que eu não posso fazer ...

   (2) *Mas, se eu pudesse, daí então eu fazia a transformação* L1 *na parte de dentro*

   $$\mathtt{X} \;\to\; \big((\mathtt{Y} \to \mathtt{Z}) \;\to\; (\mathtt{Y} \to \mathtt{W})\big)$$

   $$\xRightarrow{\mathtt{L1}} \quad \mathtt{X} \;\to\; \Big(\big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{W})\big)\Big)$$

   o que eu também não posso fazer ...

   (3) *Mas, se eu pudesse, daí então eu fazia a distribuição do* X

   $$\mathtt{X} \;\to\; \Big(\big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{W})\big)\Big)$$

   $$\xRightarrow{\mathtt{Dist}} \quad \Big(\mathtt{X} \;\to\; \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big)\Big) \;\to\; \Big(\mathtt{X} \;\to\; \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{W})\big)\Big)$$

   o que eu até posso fazer ...

   (4) *Nessa hora, eu ia ter que aplicar a regra* R3*, e pra isso eu ia precisar construir*

   $$\mathtt{X} \to \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big)$$

   Mas isso é fácil

   $$\xRightarrow{\mathtt{H1}} \quad \mathtt{Z} \to (\mathtt{Y} \to \mathtt{Z}) \quad \xRightarrow{\mathtt{R2}} \quad \mathtt{X} \to \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big)$$

   (5) *Então, aplicando a regra* R3 *eu chegaria em*

   $$\mathtt{X} \to \big(\mathtt{Z} \;\to\; (\mathtt{Y} \to \mathtt{W})\big)$$

   Deu certo!

Quer dizer, não deu ...

Porque o `X` *tá atrapalhando ...*

Droga!

Liduína estava brava com razão.

Quer dizer, não era só que o `X` estava atrapalhando a construção da transformação `Inv 2.0`.

Mas ele também estava atrapalhando a aplicação do `Dist` na parte de dentro da regra.

E também estava atrapalhando a aplicação de `L1` na parte de dentro da regra.

Em outras palavras, não era só a transformação `Inv` que era mó páia ...

A transformação `Dist` também era mó páia ...

E a transformação `L1` também era mó páia ...

Quando Liduína viu isso, ela ficou muito contrariada.

Mas, o que ela não percebeu é que ela estava raciocinando pra trás.

Só que, dessa vez, no jogo das transformações ...

5. *A culpa é toda do Hilbert!*

   Quer dizer, que as transformações `Inv` *e* `L1` *sejam mó páia pode até ser culpa minha, que fui eu quem inventei ...*

   Mas, se a transformação `Dist` *é mó páia, então a culpa é do Hilbert.*

   Essa transformação vem da regra `H2` *...*

   E quem inventou a regra `H2` *foi o Hilbert ...*

   Então a culpa é dele.

   Na verdade, pensando bem ...

   Esse jogo lógico é todo invenção do Hilbert ...

   Então, se a gente for ver, a culpa é toda dele mesmo ...

   Eu não tenho nada a ver com isso!

   E se for pra eu continuar pensando desse jeito, então eu ia achar que não tem intuição nenhuma do Hilbert por trás das regrinhas `H1` *e* `H2` *...*

   Mas eu acho que tem ...

   É a intuição lógica que todo mundo tem ...

   Mas que ninguém sabe que tem ...

   E que eu tava tentando descobrir qual é ...

   hmm ...

   Então é melhor eu voltar pros meus raciocínios ...

6. *Hmm ...*

   A transformação `Dist` *que eu tenho é mó páia, mas ela não precisa ser ...*

   Quer dizer, eu podia tentar transformar ela numa transformação legal ...

   Uma transformação que faz a distribuição na parte de dentro da regra.

   Hmm ...

   Deixa eu ver ...

   (1) *A transformação* `Dist` *que eu tenho é essa aqui*

   $$\mathtt{p} \rightarrow (\mathtt{q} \rightarrow \mathtt{r}) \quad \xRightarrow{\mathtt{Dist}} \quad (\mathtt{p} \rightarrow \mathtt{q}) \rightarrow (\mathtt{p} \rightarrow \mathtt{r})$$

   (2) *E ela vem da regra* `H2`

   $$\big(\mathtt{p} \rightarrow (\mathtt{q} \rightarrow \mathtt{r})\big) \rightarrow \big((\mathtt{p} \rightarrow \mathtt{q}) \rightarrow (\mathtt{p} \rightarrow \mathtt{r})\big)$$

   quando eu acoplo o Modus Ponens nela.

   Certo.

   (3) *A transformação* `Dist` *que eu queria é essa aqui*

   $$\mathtt{p} \rightarrow \big(\mathtt{q} \rightarrow (\mathtt{r} \rightarrow \mathtt{s})\big) \quad \xRightarrow{\mathtt{Dist\ 2.0}} \quad \mathtt{p} \rightarrow \big((\mathtt{q} \rightarrow \mathtt{r}) \rightarrow (\mathtt{q} \rightarrow \mathtt{s})\big)$$

   que faz a distribuição na parte de dentro da regra.

   (4) *Então, ela deveria vir de uma regra assim*

   $$\Big(\mathtt{p} \rightarrow \big(\mathtt{q} \rightarrow (\mathtt{r} \rightarrow \mathtt{s})\big)\Big) \rightarrow \Big(\mathtt{p} \rightarrow \big((\mathtt{q} \rightarrow \mathtt{r}) \rightarrow (\mathtt{q} \rightarrow \mathtt{s})\big)\Big)$$

   quando eu acoplasse o Modus Ponens nela.

   (5) *Mas, é muito fácil conseguir essa regra!*

   Quer dizer, uma vez que eu tenho isso

   $$\big(\mathtt{q} \rightarrow (\mathtt{r} \rightarrow \mathtt{s})\big) \rightarrow \big((\mathtt{q} \rightarrow \mathtt{r}) \rightarrow (\mathtt{q} \rightarrow \mathtt{s})\big)$$

   que eu já tenho, porque isso é só a regra `H2` *com as letras trocadas.*

   (6) *É só eu aplicar a minha transformação* `L1`

   $$\xRightarrow{\mathtt{L1}} \quad \Big(\mathtt{p} \rightarrow \big(\mathtt{q} \rightarrow (\mathtt{r} \rightarrow \mathtt{s})\big)\Big) \rightarrow \Big(\mathtt{p} \rightarrow \big((\mathtt{q} \rightarrow \mathtt{r}) \rightarrow (\mathtt{q} \rightarrow \mathtt{s})\big)\Big)$$

   pra colocar o `p` *na frente dos dois lados.*

   Consegui!!

7. *E não foi tão difícil assim.*

Então eu acho que eu consigo fazer a mesma coisa com L1.

Quer dizer, transformar ela numa transformação que transforma o lado de dentro de uma regra.

Deixa eu ver ...

(1) *A transformação* L1 *que eu tenho é essa aqui*

$$\mathtt{q} \to \mathtt{r} \quad \overset{\text{L1}}{\Longrightarrow} \quad (\mathtt{p} \to \mathtt{q}) \ \to \ (\mathtt{p} \to \mathtt{r})$$

(2) *E ela vem de uma regra que eu construí ontem*

$$(\mathtt{q} \to \mathtt{r}) \ \to \ \big((\mathtt{p} \to \mathtt{q}) \ \to \ (\mathtt{p} \to \mathtt{r})\big)$$

quando eu acoplo o Modus Ponens nela.

Certo.

(3) *A transformação* L1 *que eu queria é essa aqui*

$$\mathtt{p} \ \to \ (\mathtt{r} \to \mathtt{s}) \quad \overset{\text{L1 2.0}}{\Longrightarrow} \quad \mathtt{p} \ \to \ \big((\mathtt{q} \to \mathtt{r}) \ \to \ (\mathtt{q} \to \mathtt{s})\big)$$

que coloca uma coisa qualquer na frente dos dois lados na parte de dentro.

(4) *Então, ela deveria vir de uma regra assim*

$$\big(\mathtt{p} \ \to \ (\mathtt{r} \to \mathtt{s})\big) \ \to \ \Big(\mathtt{p} \ \to \ \big((\mathtt{q} \to \mathtt{r}) \to (\mathtt{q} \to \mathtt{s})\big)\Big)$$

quando eu acoplasse o Modus Ponens nela.

(5) *Mas, é muito fácil conseguir essa regra!*

Quer dizer, uma vez que eu tenho isso

$$(\mathtt{r} \to \mathtt{s}) \ \to \ \big((\mathtt{q} \to \mathtt{r}) \ \to \ (\mathtt{q} \to \mathtt{s})\big)$$

que eu já tenho, porque isso é só a regra de L1 *com as letras trocadas.*

(6) *Então é só aplicar a minha transformação* L1 *em cima da regra de* L1

$$\overset{\text{L1}}{\Longrightarrow} \quad \big(\mathtt{p} \ \to \ (\mathtt{r} \to \mathtt{s})\big) \ \to \ \Big(\mathtt{p} \ \to \ \big((\mathtt{q} \to \mathtt{r}) \to (\mathtt{q} \to \mathtt{s})\big)\Big)$$

pra colocar o p *na frente dos dois lados.*

Consegui de novo!!

E dessa vez foi engraçado, porque eu usei a transformação L1 *que eu tinha pra transformar a transformação* L1 *na transformação* L1 *que eu queria.*

8. *hi hi hi*

   Eu já entendi como esse negócio funciona ...

   E agora vai ser moleza fazer a mesma coisa com o `Inv` *...*

   Deixa eu ver ...

   (1) *A transformação* `Inv` *que eu tenho é essa aqui*

$$\mathtt{X} \rightarrow (\mathtt{Y} \rightarrow \mathtt{Z}) \quad \stackrel{\mathtt{Inv}}{\Longrightarrow} \quad \mathtt{Y} \rightarrow (\mathtt{X} \rightarrow \mathtt{Z})$$

   (2) *E ela vem de uma ...*

   hmm ...

   Não, ela não vem de regra nenhuma ...

   Quer dizer, ela é o empacotamento de 3 transformações

$$\stackrel{\mathtt{Inv}}{\Longrightarrow} \quad = \quad \stackrel{\mathtt{Dist}}{\Longrightarrow} \ + \ \stackrel{\mathtt{L1}}{\Rightarrow} \ + \ \stackrel{\mathtt{R3}}{\Rightarrow}$$

   Como se fosse um programinha ...

   Mas, se eu sei como empacotar transformações mais simples pra construir uma transformação mais complicada.

   Eu não sei empacotar regras mais simples pra construir uma regra mais complicada.

   Agora lascou-se ...

Liduína sentia que dessa vez o problema era mais sério.

E que a questão estava na base da relação entre construção e transformação.

Ela estava com medo de não conseguir avançar mais ...

Mas agora já era tarde demais para desistir ...

O negócio era manter a calma e respirar fundo ...

E, quando ela fez isso ...

Uma luzinha apareceu lá no fundo da sua cabeça ...

9. *Eu acho que eu tou tendo uma intuição ...*

   E se eu desempacotasse o `Inv`*?*

   Quer dizer, claro, nesse caso eu ia obter as 3 transformações que foram usadas para construir ele

$$\stackrel{\mathtt{Dist}}{\Longrightarrow} \ + \ \stackrel{\mathtt{L1}}{\Rightarrow} \ + \ \stackrel{\mathtt{R3}}{\Rightarrow}$$

   Certo.

Mas daí, eu continuo desmontando ...

Quer dizer, cada transformação dessas é o acoplamento de uma regra com o Modus Ponens

$$\big(\texttt{H2} + \texttt{MP}\big) \quad + \quad \big(\texttt{Ex1} + \texttt{MP}\big) \quad + \quad \big(\texttt{H1} + \texttt{MP}\big)$$

onde Ex1 *é a regra do exemplo de ontem, que está por trás de* L1*.*

Legal.

Agora tá tudo desmontado.

E então, eu posso remontar de um jeito diferente ...

Por exemplo, assim

$$\Big[\big(\texttt{H2} + \texttt{Ex}\big) \; + \; \texttt{MP} \; + \; \texttt{MP}\Big] \quad + \quad \big(\texttt{H1} + \texttt{MP}\big)$$

E aqui eu encontro o meu problema de frente ...

Quer dizer, o meu problema é construir uma regra R? *que eu não sei qual é ...*

Que seja a combinação de H2 *e* Ex

$$\texttt{R?} \quad \equiv \quad \texttt{H2} \overset{?}{\odot} \texttt{Ex1}$$

de modo que, se ela for acoplada duas vezes ao Modus Ponens, isso tem o mesmo efeito que as transformações Dist *e* L1 *em sequência*

$$\Big[\texttt{R?} \; + \; \texttt{MP} \; + \; \texttt{Mp}\Big] \quad \equiv \quad \xRightarrow{\texttt{Dist}} \quad + \quad \xRightarrow{\texttt{L1}}$$

Certo.

E daí, se isso der certo ...

Então eu posso repetir a mesma coisa outra vez ...

Quer dizer, então eu vou ter isso

$$\big(\texttt{R?} + \texttt{MP} + \texttt{MP}\big) \quad + \quad \big(\texttt{H1} + \texttt{MP}\big)$$

Que eu posso remontar assim

$$\big(\texttt{R?} + \texttt{H1}\big) \; + \; \texttt{MP} \; + \; \texttt{MP} \; + \; \texttt{MP}$$

E daí, se eu combinar R? *e* H1 *assim*

$$\texttt{Rinv} \quad \equiv \quad \texttt{R?} \overset{?}{\odot} \texttt{H1}$$

Eu vou conseguir construir a regra do Inv *...*

que eu vou usar aplicando o Modus Ponens 3 vezes ...

$$\texttt{Rinv} \ + \ \texttt{MP} \ + \ \texttt{MP} \ + \ \texttt{MP}$$

hmm ...

Essa parte ainda tá meio estranha ...

10. *Mas o meu problema tá claro.*

Só que é difícil pensar nele assim desse jeito ...

Porque H2 *e* Ex *são regras meio complicadas ...*

Então, eu vou pensar de outro jeito ...

Quer dizer, suponha que eu tenho uma transformação TA *que veio de uma regra* RA

$$\overset{\texttt{TA}}{\Longrightarrow} \ = \ \texttt{RA} + \texttt{MP}$$

E suponha que eu tenho uma transformação TB *que veio de uma regra* RB

$$\overset{\texttt{TB}}{\Longrightarrow} \ = \ \texttt{RB} + \texttt{MP}$$

E suponha que as transformações TA *e* TB *foram empacotadas para construir a transformação* TC

$$\overset{\texttt{TC}}{\Longrightarrow} \ = \ \overset{\texttt{TA}}{\Longrightarrow} + \overset{\texttt{TB}}{\Longrightarrow}$$

Certo.

O problema é construir uma regra RC *que é a regra daonde vem a transformação* TC

(só que dessa vez a transformação veio primeiro ...)

Algo assim

$$\overset{\texttt{TC}}{\Longrightarrow} \ = \ \texttt{RC} + \texttt{MP}$$

A minha ideia é que ela vai ser uma combinação das regras RA *e* RB

$$\texttt{RC} \ = \ \texttt{RA} \overset{?}{\odot} \texttt{RB}$$

Mas, eu ainda não sei que combinação é essa.

Deixa eu pensar então ...

(1) *As transformações* TA *e* TB *acontecem uma depois da outra ...*

Então, aquilo que sai de uma, é aquilo que entra na outra ...

Quer dizer, a coisa é assim

$$\texttt{X} \xrightarrow{\texttt{TA}} \texttt{Y} \xrightarrow{\texttt{TB}} \texttt{Z}$$

Mas, isso significa que as regras RA *e* RB *são assim*

$$\texttt{RA} \;=\; \texttt{X} \;\rightarrow\; \texttt{Y} \qquad\qquad e \qquad\qquad \texttt{RB} \;=\; \texttt{Y} \;\rightarrow\; \texttt{Z}$$

Quer dizer, elas tem uma parte em comum ...

Hmm ...

(2) *Então, talvez seja aqui que eu engato uma na outra, pra fazer a combinação ...*

Tipo assim

$$\underbrace{\texttt{X} \rightarrow (\texttt{Y}}_{\texttt{RA}} \overbrace{\rightarrow \texttt{Z})}^{\texttt{RB}}$$

Hmm ...

E agora?

(3) *Hmm ...*

E se eu fizesse o Dist*?*

Bom, então, eu ia obter isso

$$\texttt{X} \;\rightarrow\; (\texttt{Y} \rightarrow \texttt{Z}) \quad \xRightarrow{\texttt{Dist}} \quad (\texttt{X} \rightarrow \texttt{Y}) \;\rightarrow\; (\texttt{X} \rightarrow \texttt{Z})$$

E ...

Hmm ...

(4) *Ali do lado direito*

$$\texttt{X} \;\rightarrow\; (\texttt{Y} \rightarrow \texttt{Z}) \quad \xRightarrow{\texttt{Dist}} \quad (\texttt{X} \rightarrow \texttt{Y}) \;\rightarrow\; \underbrace{(\texttt{X} \rightarrow \texttt{Z})}_{*}$$

Eu já tenho uma coisa que tem a cara da regra RC.

Quer dizer, uma coisa que acoplada ao Modus Ponens transforma X *em* Z

$$\texttt{X} \quad \xRightarrow{(\texttt{X}\rightarrow\texttt{Z}) \,+\, \texttt{MP}} \quad \texttt{Z}$$

que é aquilo que a transformação TC *faz.*

(5) *Mas, ainda tem uma coisa na frente dela*

$$\underbrace{(\texttt{X} \rightarrow \texttt{Y})}_{*} \;\rightarrow\; (\texttt{X} \rightarrow \texttt{Z})$$

Essa coisa é a regra RA*, que eu tou assumindo que eu já tenho.*

Logo, eu posso aplicar Modus Ponens pra obter

$$\mathtt{X} \;\rightarrow\; \mathtt{Z}$$

que é a regra RC.

Legal!

(6) *Mas, isso aqui não caiu do céu pra mim*

$$(\mathtt{X} \rightarrow \mathtt{Y}) \;\rightarrow\; (\mathtt{X} \rightarrow \mathtt{Z})$$

Isso apareceu quando eu apliquei Dist *nisso aqui*

$$\mathtt{X} \;\rightarrow\; (\mathtt{Y} \rightarrow \mathtt{Z})$$

Mas, Dist *é só* H2 *acoplada com Modus Ponens.*

Então, tudo começa quando eu construo isso aqui

$$\big(\mathtt{X} \;\rightarrow\; (\mathtt{Y} \rightarrow \mathtt{Z})\big) \;\;\rightarrow\;\; \big((\mathtt{X} \rightarrow \mathtt{Y}) \;\rightarrow\; (\mathtt{X} \rightarrow \mathtt{Z})\big)$$

que é uma coisa construída com a regrinha H2.

E as coisas construídas com H2 *caem do céu sim ...*

Pelo menos no jogo lógico do Hilbert ...

(7) *Mas agora, eu preciso passar disso aqui*

$$\big(\mathtt{X} \;\rightarrow\; (\mathtt{Y} \rightarrow \mathtt{Z})\big) \;\;\rightarrow\;\; \big((\mathtt{X} \rightarrow \mathtt{Y}) \;\rightarrow\; (\mathtt{X} \rightarrow \mathtt{Z})\big)$$

pra isso

$$(\mathtt{X} \rightarrow \mathtt{Y}) \;\rightarrow\; (\mathtt{X} \rightarrow \mathtt{Z})$$

o que eu consigo fazer aplicando Modus Ponens, se eu tiver isso aqui

$$\mathtt{X} \;\rightarrow\; (\mathtt{Y} \rightarrow \mathtt{Z})$$

Mas, isso não cai do céu pra mim ...

Hmm ...

É, não cai mesmo ...

Mas isso caiu

$$\mathtt{Y} \rightarrow \mathtt{Z}$$

Quer dizer, isso é a regra RB *que eu tou assumindo que eu já tenho.*

E isso é uma outra coisa que cai do céu

$$(\mathtt{Y} \to \mathtt{Z}) \;\to\; \big(\mathtt{X} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big)$$

porque essa é uma coisa que dá pra construir com a regrinha H1.

E daí, eu posso aplicar Modus Ponens pra obter isso

$$\mathtt{X} \;\to\; (\mathtt{Y} \to \mathtt{Z})$$

que é a coisa que eu preciso pra poder aplicar Modus Ponens na outra regra.

E agora acabou!

(6) *Quer dizer, recapitulando tudo*

- *eu usei* H2 *pra construir a regra*

$$\big(\mathtt{X} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big((\mathtt{X} \to \mathtt{Y}) \;\to\; (\mathtt{X} \to \mathtt{Z})\big)$$

- *e eu usei* H1 *pra construir a regra*

$$(\mathtt{Y} \to \mathtt{Z}) \;\to\; \big(\mathtt{X} \;\to\; (\mathtt{Y} \to \mathtt{Z})\big)$$

- *depois, eu apliquei Modus Ponens na segunda regra (porque eu tenho* $\mathtt{RB} : \mathtt{Y} \to \mathtt{Z}$*), pra obter*

$$\mathtt{X} \;\to\; (\mathtt{Y} \;\to\; \mathtt{Z})$$

- *e eu usei isso isso pra aplicar Modus Ponens na primeira regra, pra obter*

$$(\mathtt{X} \;\to\; \mathtt{Y}) \;\to\; (\mathtt{X} \;\to\; \mathtt{Z})$$

- *e depois eu apliquei Modus Ponens outra vez nisso (porque eu tenho* $\mathtt{RA} : \mathtt{X} \to \mathtt{Y}$*), pra conseguir*

$$\mathtt{X} \to \mathtt{Z}$$

- *que é a regra* RC *que eu queria*

Hmm ...

É que nem um dominó ...

Onde as peças vão caindo uma depois da outra, aplicando Modus Ponens ...

Ou um circuito, onde a gente monta as caixinhas pro sinal ir passando ...

Será que era isso que o professor de circuitos lógicos tava explicando ...

E eu nunca entendi ...

Caráááááácoles!!!!!!!!

Entendi!!!!!!

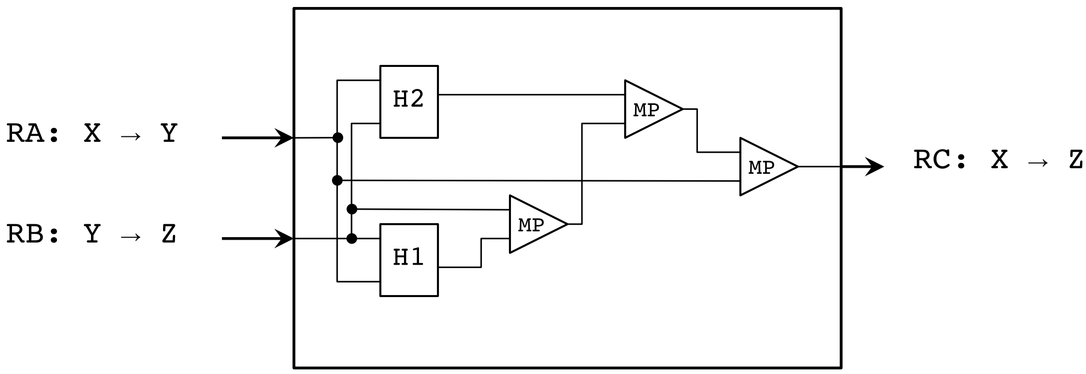
Ahhhh!!!!

Num acredito!!

Pronto.

Liduína tinha descoberto o significado das regrinhas do Hilbert.

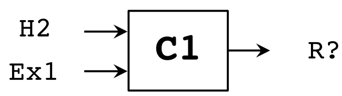
E pra poder ver melhor as coisas, ela fez o seguinte desenho



Quer dizer, as regrinhas `H1` e `H2` eram usadas para fazer o empacotamento (ou a combinação) de regras, para construir regras que produzem o efeito combinado das regras originais.

Liduína resolveu chamar o circuito de construção acima de `C1`.

E depois ela utilizou ele para construir a combinação das regras que estavam por trás de `Dist` e `L1`



onde

$$\mathtt{R?}: \ \big(\mathtt{X} \ \rightarrow \ (\mathtt{Y} \rightarrow \mathtt{Z})\big) \ \rightarrow \ \Big(\big(\mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Y})\big) \ \rightarrow \ \big(\mathtt{Y} \ \rightarrow \ (\mathtt{X} \rightarrow \mathtt{Y})\big)\Big)$$

é a regra que faz o `Dist` e aplica `L1`, em um só passo.

Mas, logo em seguida, Liduína descobriu que teria que construir um outro circuito.

11. *Legal.*

    Eu já resolvi a metade do meu problema.

    Agora, só falta a outra metade.

    Quer dizer, eu preciso de uma regra que tenha o efeito combinado dessas duas transformações

$$\xRightarrow{\mathtt{Dist-L1}} \quad + \quad \xRightarrow{\mathtt{R3}}$$

Mas aqui a situação é diferente.

Quer dizer, R3 *não é uma transformação que transforma uma coisa em outra coisa.*

Mas, R3 *é uma transformação que tira a primeira parte de uma alguma coisa*

$$\mathtt{T} \;\to\; \mathtt{q} \quad \stackrel{\mathtt{R3}}{\Longrightarrow} \quad \mathtt{q}$$

onde T *é uma coisa que a gente já tem (ou que a gente consegue construir).*

No caso do meu exemplo, R3 *está tirando uma coisa que dá pra construir com* H1*.*

Certo.

Agora eu preciso construir uma regra que tem o efeito combinado de Dist-L1 *e de* R3*.*

E a ideia é que eu vou usar H2 *e* H1 *pra fazer isso ...*

Além do Modus Ponens, claro.

Hmm ...

12. *Pensando em transformações é fácil.*

Quer dizer, primeiro eu posso fazer um Dist *na regra de* Dist-L1

$$\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \Big(\big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)\Big)$$

$$\xRightarrow{\mathtt{Dist}} \quad \Big(\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})\big)\Big) \;\to\; \Big(\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)\Big)$$

Daí eu guardo isso e faço a seguinte construção com H1

$$\big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})\big) \;\to\; \Big(\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})\big)\Big)$$

e faço essa outra construção com H1

$$\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})$$

Pronto, agora com dois Modus Ponens eu resolvo o problema.

Primeiro eu aplico Modus Ponens com as duas últimas regras pra obter

$$\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Y})\big)$$

E depois eu uso isso pra aplicar Modus Ponens na regra que foi o resultado do Dist*, pra obter*

$$\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \;\to\; \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)$$

que já é a regra que tem o efeito combinado de Dist-L1 *e* R3*.*

E que também já é a regra do Inv*.*

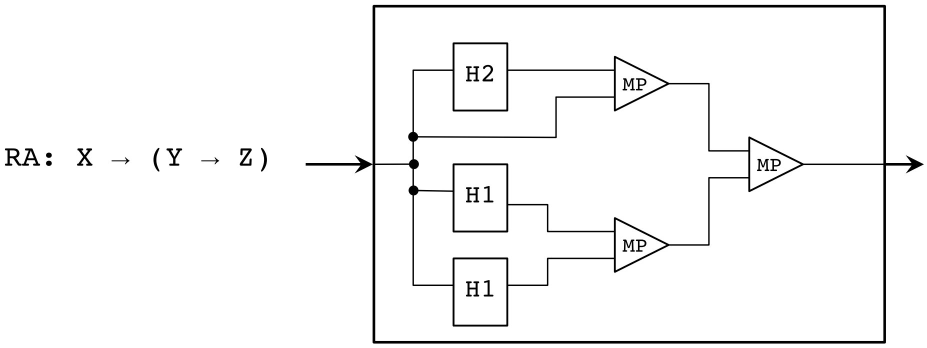
13. *Examinando bem o que eu fiz, eu vejo que esse raciocínio pode ser aplicado sempre que eu quero construir uma regra que combina dois passos de transformação desse tipo*

$$\mathtt{X} \quad \overset{\mathtt{TA}}{\Longrightarrow} \quad \mathtt{Y} \to \mathtt{Z} \quad \overset{\mathtt{R3}}{\Longrightarrow} \quad \mathtt{Z}$$

onde `Y` *é uma coisa que eu posso construir com* `H1`*.*

Além disso, se eu quiser o circuito que constrói a regra desses dois passos, então o `Dist` *que eu apliquei no início vai virar uma caixinha de* `H2`*.*

A coisa ia ficar assim



14. *Legal.*

Agora, finalmente, eu tenho a regra do `Inv`

$$\big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \ \to\ \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)$$

A boa notícia é que eu já sei que eu posso aplicar a transformação `L1` *nessa regra*

$$\overset{\mathtt{L1}}{\Longrightarrow} \quad \Big(\mathtt{W} \ \to\ \big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big)\Big) \ \to\ \Big(\mathtt{W} \ \to\ \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)\Big)$$

e daí, acoplando o Modus Ponens a essa regra, eu obtenho a transformação `Inv 2.0` *que eu tanto queria*

$$\mathtt{W} \ \to\ \big(\mathtt{X} \to (\mathtt{Y} \to \mathtt{Z})\big) \quad \xrightarrow{\mathtt{Inv2.0}} \quad \mathtt{W} \ \to\ \big(\mathtt{Y} \to (\mathtt{X} \to \mathtt{Z})\big)$$