Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Formalização do Cálculo *Pointfree* em estilo relacional

Thiago Mendonça Ferreira Ramos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. rer. nat. Mauricio Ayala-Rincón

Brasília
2014

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Homero Luiz Piccolo

Banca examinadora composta por:

Prof. Dr. rer. nat. Mauricio Ayala-Rincón (Orientador) — CIC/UnB
Prof. Dr. Vander Ramos Alves — CIC/UnB
Prof. Dr. Flávio Leonardo Cavalcante Moura — CIC/UnB

CIP — Catalogação Internacional na Publicação

Ramos, Thiago Mendonça Ferreira.

Formalização do Cálculo *Pointfree* em estilo relacional / Thiago Mendonça Ferreira Ramos. Brasília : UnB, 2014.
149 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2014.

1. Lógica em Ciência da Computação, 2. Álgebra Relacional,
3. Dedução Formal, 4. Formalização em Assistente de Prova

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Formalização do Cálculo *Pointfree* em estilo relacional

Thiago Mendonça Ferreira Ramos

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. rer. nat. Mauricio Ayala-Rincón (Orientador)
CIC/UnB

Prof. Dr. Vander Ramos Alves
CIC/UnB

Prof. Dr. Flávio Leonardo Cavalcante Moura
CIC/UnB

Prof. Dr. Homero Luiz Piccolo
Coordenador do Bacharelado em Ciência da Computação

Brasília, 10 de dezembro de 2014

Dedicatória

Agradecimentos

Agradeço a todos que contribuiram para a construção dessa monografia.

Resumo

A Lógica de Primeira Ordem provê técnicas para demonstrar e expressar propriedades de correção sobre sistemas. Entretanto, especificação e formalização são técnicas que tomam muito tempo na construção de *software* e *hardware*. Logo, o objetivo é estudar um mecanismo alternativo de especificação e dedução baseado em álgebra relacional, chamado cálculo relacional, e compará-lo com o cálculo de Gentzen, observando tamanhos de provas e de especificações.

Palavras-chave: Lógica em Ciência da Computação, Álgebra Relacional, Dedução Formal, Formalização em Assistente de Prova

Abstract

The First Order Logic provides techniques to demonstrate and express properties of correctness in systems. However, specification and formalization are techniques that take a lot of time in the construction of software and hardware. Thus, the goal is to study an alternative way of deduction and specification based on relational algebra, called relational calculus, and to compare to Gentzen calculus, seeing sizes of proofs and specifications.

Keywords: Logic in Computer Science, Relational Algebra, Formal Deduction, Formalization in Proof Assistant

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Sobre a Monografia

As pessoas desejam que os produtos e serviços adquiridos funcionem corretamente. Se é um carro, irá querer que os freios não falhem. Se for um liquidificador, que não se estilhasse. Se for uma ponte, que não entre em ressonância com a brisa. Se for um prédio com arquitetura côncava e com vidros espelhados, que não convirja os raios solares e derreta um carro que está na frente do prédio. Com *software* não é diferente. Mas no caso de produtos físicos, para atestar o bom funcionamento, faz-se protótipos, testes exaustivos e verificam-se os cálculos. Com programas, é possível fazer testes, entretanto não se consegue verificar tudo. Como agir para verificar a correção de programas?

Primeiramente, modela-se e verifica-se a validade desse modelo segundo um contrato. Após a modelagem, especifica-se propriedades desejadas e formaliza-se em assistentes de prova. Assim, argumenta-se a correção do programa. Em engenharia de software, modularização permite organização para mantenabilidade. Logo, após demonstrada a correção dos módulos, deverá ser provado que componentes estão bem coordenados, pois há problemas que surgem na montagem de componentes. No entanto, criar especificações e construir formalizações em assistentes de prova são tarefas que tomam muito tempo, pois é preciso demonstrar cada passo de prova sem deixar de fazer aquilo que se considera intuitivo.

Chega-se a demorar anos até que se argumente a correção de um código. Na prática, é muito mais rápido construir software verificado com testes a construir por demonstrações. Mas dependendo do caso, pode ser mais arriscado, pois testes não costumam cobrir todos os casos. De quem será a culpa se um foguete explodir por conta de um erro no momento de ligar componentes? E quem poderá substituir as vidas perdidas caso um equipamento de radioterapia superdosar os pacientes, matando-os? Mas, por outro lado, como pode-se diminuir o tempo de verificação por assistente de prova?

Em termos de modelagem formal, há uma ferramenta baseada em *model checking* chamada *Alloy* (ver [8]). A partir dessa ferramenta é possível representar objetos do mundo real ou de um mundo abstrato, fazer especificações e demonstrar propriedades por *model checking*. Também é possível gerar alguns exemplos para automatizar testes de software ou observar alguma inconsistência sobre alguma especificação não pensada. Formalizações e especificações do modelo são feitas por um estilo de lógica que permite usar poucos

quantificadores. Usar poucos quantificadores permite uma leitura mais confortável, mais velocidade nas especificações e, por consequência, menos erros de especificações.

O exemplo da figura 1.1 mostra uma modelagem *Alloy* para casamento heterossexual e monogâmico feito pelo autor.

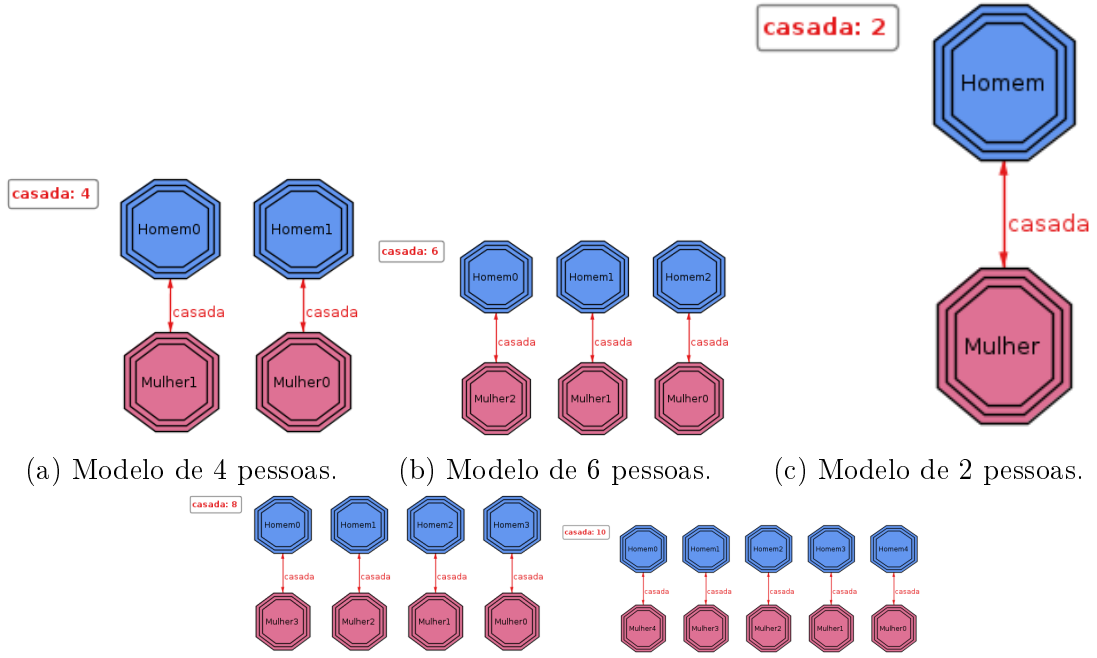
```
abstract sig Pessoa{
   casada: one Pessoa //Cada pessoa é casada com uma única pessoa.
}                     //Entretanto, a relação não é reciproca.

sig Homem, Mulher extends Pessoa{}

fact PessoaComOutra{ // Ninguém está casado consigo.
   (casada & iden) = none->none
}

fact CasamentoInverso { // O casamento é recíproco.
   casada = ~casada
}

fact CasamentoHetero { // O casamento é entre homem e mulher
   casada in (Homem->Mulher + Mulher->Homem)
}



check{                              // Checar segundo o modelo:
   not (Pessoa->Pessoa = casada              // se todos estão casados
        and (all p : Pessoa |  one p.casada) // se é monogamico
        and some Pessoa            // para o modelo com alguma pessoa
        )} for 10                     //Está ok.

run{} for 10
```

Figura 1.1: Modelagem de casamento com restrições baseadas em leis e costumes.

A ferramenta *Alloy* gera exemplos de modelos de acordo com as restrições como visto na figura 1.2.

Há um fato a se observar na especificação em *Alloy* na figura 1.1. Na maioria das afirmações, usa-se nenhum quantificador. Isso ocorre porque a ferramenta usa um estilo de notação baseado em álgebra relacional e teoria dos conjuntos. Esse estilo é o *pointfree*. O estilo *pointfree* para relações, na verdade é uma generalização para o estilo *pointfree* para a programação funcional na linguagem *haskell* já que uma relação é uma generalização de uma função. Para mais detalhes, no livro [11] e no *site* `http://www.haskell.org/haskellwiki/Pointfree` retirado de [16]. Como dito nesse *site*, a palavra *pointfree* vem da ramo da topologia, onde elementos de entrada de funções pode ser modelada como pontos. Como o estilo de programação permite retirar argumentos ela é livre de pontos.

Apesar do *Alloy* ser uma ferramenta útil para especificação, há ainda uma funcionalidade da qual carece. O uso de *model checking* é de suma importância para encontrar contra-exemplos, erros de especificações, gerar casos de testes e demonstrações com máquinas de estados. Entretanto, há o problema da explosão do número de estados. Também é uma ferramenta que não lida bem com problemas aritméticos e que, para certas demonstrações com um número infinito de elementos, não demonstra tudo.



(a) Modelo de 4 pessoas. (b) Modelo de 6 pessoas. (c) Modelo de 2 pessoas.

(d) Modelo de 8 pessoas. (e) Modelo de 10 pessoas.

Figura 1.2: Modelos de casamentos gerados pela ferramenta *Alloy*

Se há uma ferramenta que permite especificação em *pointfree* em verificação em *model checking*, por que não haver uma onde se faça verificação por regras de cálculo? Em Oliveira [12] há regras de cálculo associadas a essa notação conhecida como Cálculo da Álgebra Relacional ou Cálculo de Sistema de Informação. Como a notação permite facilidades de especificações e menos erros, as seguintes questões podem ser formuladas:

- A notação expressa que tipos de problemas?
- A notação ajuda a especificar de maneira mais rápida?
- O cálculo relacional permite formalizar de maneira mais rápida?

Ao longo da construção dessa monografia, espera-se descobrir respostas as questões anteriores.

1.2 Metodologia

Especificações e formalizações foram escritas usando o assistente de prova *Prototype Verification System* ou PVS com o uso dos manuais [18] e [19].

As principais especificações serão de conceitos de álgebra relacional e teoria dos conjuntos, com o reuso de bibliotecas já prontas em PVS sobre o assunto. As principais formalizações serão de regras de cálculo da álgebra relacional para mostrar a correção do método.

Uma vez que foi formalizado correção do cálculo, pretende-se comparar o método tradicional de formalização com o método usando álgebra relacional. Serão comparados as complexidades de prova e o tamanho das especificações.

Capítulo 2

Notação *Pointfree* e a álgebra relacional: semântica

2.1 Sobre o Cálculo da Álgebra Relacional

A notação *pointfree* é um estilo de escrita usado, primeiramente, para descrever relações binárias. Essa notação usa regras de cálculo da álgebra relacional foi idealizada por Augustus De Morgan. Além da teoria da álgebra relacional, outra teoria por trás da notação *pointfree* é a teoria dos conjuntos. A história do desenvolvimento da álgebra relacional é contada em [10] por Maddux. Nela, Tarski axiomatizou uma parte da álgebra relacional em [20] e perguntou se era possível representar qualquer modelo e se era possível deduzir todas as sentenças verdadeiras. Após Lyndon responder "não"a essas perguntas [9], Tarski criou outro cálculo relacional junto com uma álgebra chamada de álgebra cilindrica e demonstrou a equivalência entre ele e a lógica de primeira ordem em [21]. Mais tarde, Codd [2] mostra a equivalência com o cálculo de busca em banco de dados. Assim sem a álgebra relacional, não haveria a linguagem de busca em bancos de dados da forma como existe atualmente, como escreveu Ferferman em [6]:

> Você vê essas brilhantes e grandes torres da *Oracle* na *Highway* 101? Elas nunca teriam sido construídas sem o trabalho de Tarski sobre definições recursivas de satisfatibilidade e verdade.

Oracle é uma empresa que adminintra banco de dados. Recentemente, a teoria foi expandida por Oliveira em [12] e está resumida em [14] por ele. Oliveira e Ferreira também fazem o uso da notação para explicar a semântica da ferramenta *Alloy* e um estudo de caso sobre endereçamento de memória flash em [13].

Antes de começar a explicar sobre definições e propriedades da álgebra relacional, colocar-se-á um exemplo do uso da notação e cálculo para fins comparativos.

Primeiramente, costuma-se usar para descrição de fórmulas matemáticas quantificadores universais e existenciais além de predicados para denotar os objetos de estudo matemático. A sintaxe é semelhante ao que se segue:

$$\langle \forall x : R : S \rangle$$

Acima, R representa restrição de tipo dos quantificadores e S, o que se deseja denotar sobre os objetos matemáticos.

A maneira supracitada de descrição parece simples, entretanto costuma dificultar a escrita, leitura e demonstrações de especificações de problemas. Por exemplo, pode-se imaginar um modelo para descrever a relação de ancestral, A. Portanto, A relaciona duas pessoas:

$$A : Pessoa \longrightarrow Pessoa$$

Ancestral é uma relação transitiva (se alguém é ancestral de Ana e Ramom é ancestral desse alguém então Ramom é ancestral de Ana):

$$\langle \forall x, y, z :: xAy \wedge yAz \Rightarrow xAz \rangle \tag{2.1}$$

E é assimétrica (se alguém é ancestral de outro esse outro não pode ser ancestral desse outro):

$$\langle \forall x, y :: xAy \Rightarrow \neg yAx \rangle \tag{2.2}$$

As especificações acima modelam em parte sobre o funcionamento do conceito de ancestralidade. Mostrar-se-á como seriam essas propriedades na notação *Pointfree*.

Eis aqui a composição de relações. Da mesma maneira que há o conceito de composição de funções (um tipo de relação), pode-se generalizar para composição para relações:

$$xR.Sy \stackrel{def}{=} \langle \exists z :: xRz \wedge zSy \rangle$$

Eis aqui a relação conversa. Lembra a inversão de funções. Se tem que x é ancestral de y temos que y é da descendente de x. Com isso, há a ideia de relação conversa:

$$yA^{o}x \stackrel{def}{=} xAy$$

Para começar, também pode-se descrever relações como conjuntos:

$$xAy \equiv (x, y) \in A$$

Portanto, (2.1) (transitividade) pode ser colocada de outra forma:

$$A.A \subseteq A \tag{2.3}$$

Da mesma maneira, (2.2) (assimetria) pode ser especificado da forma:

$$A^{o} \cap A = \bot \tag{2.4}$$

Acima, $\bot$ representa a relação vazia (nada se relaciona a nada) ou conjunto vazio.

Percebe-se que as duas especificações não usa quantificadores de maneira explícita e é mais compacta.

Daqui, especificar-se-á uma propriedade do modelo supracitado.

Sabe-se que uma pessoa não é ancestral dela própria :

$$\langle \forall x, y : xAy : \neg(x \; id \; y) \rangle \tag{2.5}$$

A relação id é a identidade (ou função identidade).
Logo, a equação 2.1 pode ser reescrita como:

$$A \cap id = \bot \tag{2.6}$$

Mostrar-se-á, a partir do modelo, que uma pessoa não pode ser seu próprio ancestral. Em princípio, a compreensão da demonstração em alguns pontos não será clara. Entretanto ao longo da monografia, ela será esclarecida. Aqui, ela será mostrada apenas com objetivo de comparação. Quer-se mostrar que se a ancestralidade é transitiva e e assimétrica, pode-se inferir que uma pessoa não é ancestral dela mesma:

$\langle$ Tomamos um subconjunto qualquer do conjunto de pessoas que são seus próprios ancestrais. $\rangle$

$$\forall \phi : \phi \subseteq A \cap id$$

$\langle$ Propriedade fundamental dos conjuntos.$\rangle$

$$\equiv \phi \subseteq A \wedge \phi \subseteq id$$

$\langle$ Monotonia do converso em relação a inclusão.$\rangle$

$$\equiv \phi^o \subseteq A^o \wedge \phi \subseteq id$$

$\langle$ Igualdade de conversos de relações correflexivas ($\phi = \phi^o$).$\rangle$

$$\equiv \phi \subseteq A^o \wedge \phi \subseteq id$$

$\langle$ Propriedade fundamental de conjuntos.$\rangle$

$$\Longrightarrow \phi \subseteq A^o \cap A$$

$\langle$ Assim $\rangle$

$$\forall \phi : \phi \subseteq A \cap id \Longrightarrow \phi \subseteq A^o \cap A$$

$\langle$ Propriedade de inclusão de conjuntos.$\rangle$

$$\equiv A \cap id \subseteq A^o \cap A$$

$\langle$ Axioma da ancestralidade $A^o \cap A = \bot$.$\rangle$

$$\Longrightarrow A \cap id \subseteq \bot$$

$\langle$ Propriedade dos conjuntos vazios.$\rangle$

$$\equiv A \cap id = \bot$$

Essa prova anterior, organizada em forma de árvore, fica da seguinte forma:

$$A^o \cap A = \bot,\ A.A \subseteq A \vdash A \cap id = \bot$$

$$\dfrac{\dfrac{\dfrac{\overbrace{\phi \subseteq A}^{1,\phi}}{\phi \subseteq A} \quad \dfrac{\dfrac{\dfrac{[\phi \subseteq A \cap id]}{\{\phi \subseteq A\}^1}\cap e}{\phi^o \subseteq A^o}\, o\subseteq \quad \dfrac{\dfrac{[\phi \subseteq A \cap id]}{\phi \subseteq id}\cap e}{\phi^o = \phi}\, id}{\phi \subseteq A^o} =}{\phi \subseteq A^o \cap A}\cap i}{A \cap id \subseteq A^o \cap A}\subseteq i,\phi \qquad A \cap A^o = \bot}{\dfrac{A \cap id \subseteq \bot}{A \cap id = \bot} = \bot} = \tag{2.7}$$

Provar-se-á usando lógica de primeira ordem por dedução natural. As regras de inferência foram retiradas de [1] e [7].

$$\forall x \forall y(A(x,y) \Rightarrow \neg A(y,x)), \forall x \forall y \forall z(A(x,y) \wedge A(y,z) \Rightarrow A(x,z)) \vdash \forall x \neg A(x,x)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\forall x \forall y(A(x,y) \Rightarrow \neg A(y,x))}{\forall y(A(x_0,y) \Rightarrow \neg A(y,x_0))}\forall e}{A(x_0,x_0) \Rightarrow \neg A(x_0,x_0)}\forall e \quad [A(x_0,x_0)]^{\not\phi}}{\neg A(x_0,x_0)} \Rightarrow e \quad [A(x_0,x_0)]^{\not\phi}}{\bot}\neg e}{\neg A(x_0,x_0)}\neg i, a}{\forall x \neg A(x,x)}\forall i \tag{2.8}$$

E agora em sistema de Gentzen (descrito em [1]).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A(x_0,x_0) \vdash A(x_0,x_0), \bot}Ax}{\vdash \neg A(x_0,x_0), A(x_0,x_0)}R\Rightarrow \quad \dfrac{}{\neg A(x_0,x_0) \vdash \neg A(x_0,x_0)}Ax}{A(x_0,x_0) \Rightarrow \neg A(x_0,x_0) \vdash \neg A(x_0,x_0)}L\Rightarrow}{\forall y(A(x_0,y) \Rightarrow \neg A(y,x_0)) \vdash \neg A(x_0,x_0)}L\forall}{\forall x \forall y(A(x,y) \Rightarrow \neg A(y,x)) \vdash \neg A(x_0,x_0)}L\forall}{\forall x \forall y(A(x,y) \Rightarrow \neg A(y,x)) \vdash \forall x \neg A(x,x)}R\forall}{\forall x \forall y(A(x,y) \Rightarrow \neg A(y,x)), \forall x \forall y \forall z(A(x,y) \wedge A(y,z) \Rightarrow A(x,z)) \vdash \forall x \neg A(x,x)}LW \tag{2.9}$$

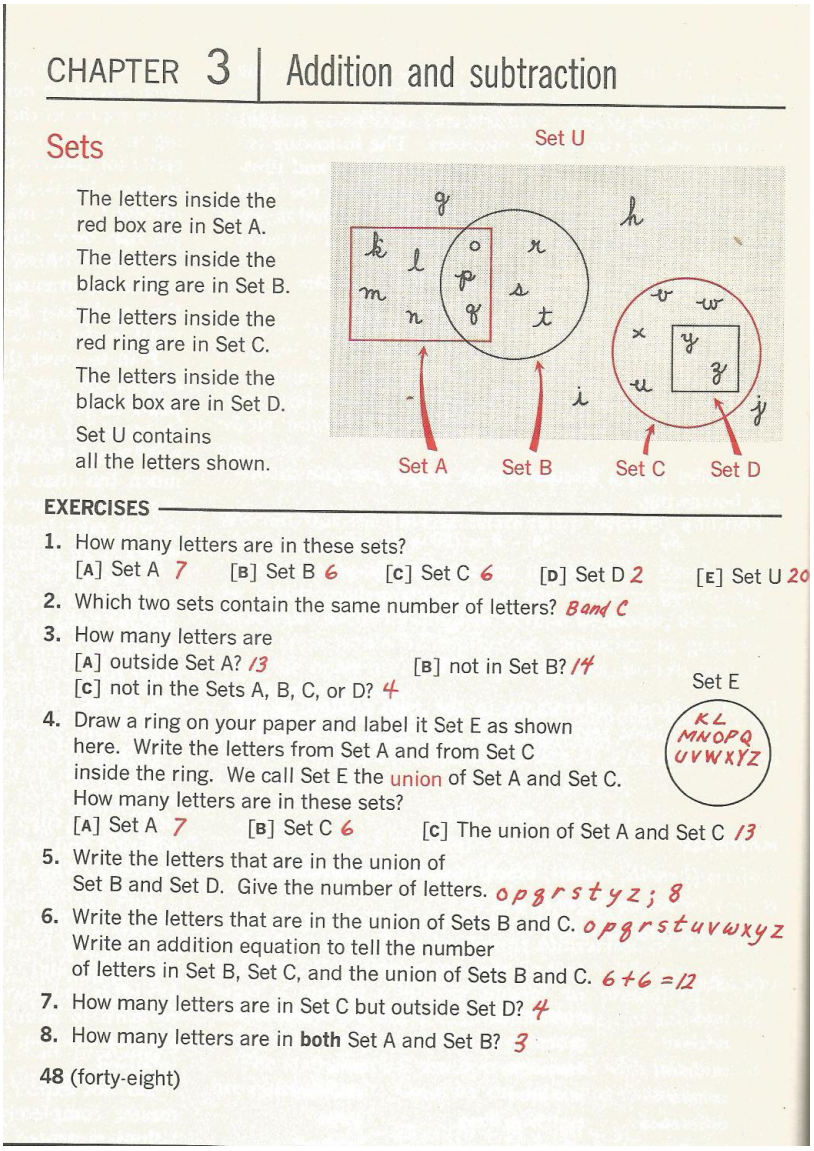
Dentre as três formas de se fazer a demonstração, percebe-se que a classificação da menos complexa para mais complexa é: dedução natural (2.8), cálculo de Gentzen (2.9) e cálculo relacional (2.7). Apesar de dedução natural ter sido o menos complexo nesses exemplos, o que está implementado em alguns sistemas de prova como o PVS (manual em [18]) é o cálculo de Gentzen, mas de maneira otimizada.

Apesar de tudo, nota-se que a prova onde os nós da árvore estão mais compactos foi a feita por cálculo relacional (2.7). Isso ocorreu, porque não se usou de maneira explícita os quantificadores. Com isso facilita notação e leitura de especificações. Outra observação importante é a facilidade de compreensão de algumas partes das fórmulas nos nós das árvores. Se for perguntado a uma criança qualquer do ensino fundamental por volta dos 8 anos, um bom estudante e amante dos números o que significa "$\cap$", provavelmente ela

vai desenhar dois círculos que se cruzam, desenhar objetos que não se repetem em cada um deles e irá apontar o que há em comum aos dois. Se for feita as mesmas perguntas a um pré-adolescente por volta dos 13 sob condições semelhantes a criança de 8, ele já vai poder responder também o que é "$\subseteq$".

Uma situação diferente ocorrerá se for perguntado sobre "$\forall$", "$\exists$", "$\vee$", "$\wedge$", "$\neg$"e "$\Rightarrow$", onde possivelmente haverá respostas como "'A' de cabeça para baixo", "'E' espelhado", "Letra 'v'", "'v' de cabeça para baixo", "um olho piscando", "Siga à direita"ou "Não sei". Isso é um indício de que há mais facilidade de se lidar com teoria dos conjuntos a usar lógica de predicados.

Noções básicas de teoria dos conjuntos são ensinadas a crianças ao mesmo tempo que a somar desde antes de 1964 como mostra a figura 2.1 retirada de [17].



CHAPTER 3 | Addition and subtraction

Sets

The letters inside the red box are in Set A.
The letters inside the black ring are in Set B.
The letters inside the red ring are in Set C.
The letters inside the black box are in Set D.
Set U contains all the letters shown.

Set U

Set A Set B Set C Set D

EXERCISES

1. How many letters are in these sets?
[A] Set A 7 [B] Set B 6 [C] Set C 6 [D] Set D 2 [E] Set U 20
2. Which two sets contain the same number of letters? B and C
3. How many letters are
[A] outside Set A? 13 [B] not in Set B? 14
[C] not in the Sets A, B, C, or D? 4
4. Draw a ring on your paper and label it Set E as shown here. Write the letters from Set A and from Set C inside the ring. We call Set E the union of Set A and Set C. How many letters are in these sets?
[A] Set A 7 [B] Set C 6 [C] The union of Set A and Set C 13

Set E

KL MNOPQ UVWXYZ

5. Write the letters that are in the union of Set B and Set D. Give the number of letters. opgrstyz; 8
6. Write the letters that are in the union of Sets B and C. opgrstuvwxyz
Write an addition equation to tell the number of letters in Set B, Set C, and the union of Sets B and C. 6+6=12
7. How many letters are in Set C but outside Set D? 4
8. How many letters are in **both** Set A and Set B? 3

48 (forty-eight)

Figura 2.1: Exercícios sobre teoria dos conjuntos para crianças

Ainda hoje, o básico de teoria dos conjuntos costuma ser ensinada à crianças do ensino fundamental conforme pode ser visto em [3, 5], entretanto a lógica costuma ser tratada

em nível universitário. Mas porque se ensina às crianças noções básicas de teoria dos conjuntos antes de lógica de predicados? Afinal, matemática funciona por uma linguagem (como escrito em [4]) e crianças tem facilidade de aprender linguagens, portanto, não seria possível ensinar lógica de predicados a crianças? O fato é que isso não ocorre. Mas uma vez ensinado teoria dos conjuntos, operações sob elas acabam por ficarem de mais fácil manipulação. Logo, demonstrações que usam um método baseada nessa teoria, por hipótese, seriam mais rápidas, pois tem-se mais facilidades com linguagens que se aprendem quando criança do que com linguagens que se aprendem quando se é adulto.

Mas será que o cálculo baseado em *pointfree* torna as demonstrações mais fáceis de compreender e rápidas? Segundo [12] a resposta é sim. Mas é preciso ter certeza, e perguntar o que matemáticos e cientistas da computação pensam e avaliar como elas lidam com cada uma das maneiras de pensar. E ainda há outra pergunta a ser respondida: dada uma semântica à notação *pointfree*, o cálculo associado é completo e correto? Ela tem o mesmo poder de expressividade ou maior que a lógica de predicados? É o que se espera descobrir.

2.2 Definições Básicas

Os principais conceitos apresentados a seguir são baseados e refinados a partir de conceitos retirados do relatório [12], do livro em construção [11] e complementados por [24] e por [22].

Definição 1: o que é tipo? Segundo o livro [15] sistema de tipos é um método sintático tratável para prover a ausência de certos comportamentos de programas pela classificação de frases concordando com as espécies de valores que eles computam. A partir dessa definição, pode-se dizer que tipo é uma classificação para valores de mesma espécie. Pode-se dizer também que tipos fornecem abstrações sobre classes de objetos. Segundo o livro [23], seja T um conjunto de tipos que é contruído a partir de um conjunto contável de tipos P_0, P_1, P_2, ... e do operador $\times$ da seguinte maneira:

- Os tipos variáveis pertencem a T;
- Se A e B pertencem a T, então $A \times B$ também pertence.

Acima, o operador $\times$ é o produto cartesiano entre os tipos. Ou seja, se a é de tipo A e b é de tipo B, (a, b) é de tipo $A \times B$. Na reaidade, nesse trabalho a semântica adotada para os tipos é a semântica de conjuntos. Por exemplo, quando se coloca A como um tipo, na verdade significa conjunto de todos os elementos do tipo A. Assim $A \times B$ na verdade significa produto cartesiano entre conjunto de todos os elementos de tipo A e B.

Definição 2: o que é relação binária e uma endo-relação? Dado dois tipos, uma relação binária (ou simplesmente relação ou **Verbo**) será um subconjunto do produto cartesiano desses tipos. Uma endo-relação será quando esses dois tipos forem iguais. Se há pessoas (modeladas por $Pessoas$) e direitos (idem $Direitos$), pode-se estabelecer uma relação de posse (modelada por P) entre eles:

$$P \subseteq Pessoas \times Direitos$$

Quatro alternativas a notação acima são:

$$P : Pessoas \longleftarrow Direitos$$

$$P : Direitos \longrightarrow Pessoas$$

$$Direito \overset{P}{\longleftarrow} Pessoas$$

$$Pessoas \overset{P}{\longrightarrow} Direito$$

O conjunto expresso pelo tipo que está a esquerda será chamado de **Conjunto da Esquerda**, **Esquerda** ou **Sujeito** e o conjunto a direita, **Conjunto da Direita**, **Direita** ou **Objeto**. Assim, ter-se-á a seguinte relação:

$$Sujeto \overset{Verbo}{\longleftarrow} Predicado$$

Como exemplo de endo-relação, há as relações amorosas (aqui, notadas por A) são entre pessoas, e portanto, uma endo-relação.

$$Pessoas \overset{A}{\longleftarrow} Pessoas$$

E eis aqui a especificação no assistente de prova, PVS;

```
relation[X,Y: TYPE]     : TYPE = setof[[X,Y]]
endo_relation[X: TYPE] : TYPE = relation[X,X]
```

Figura 2.2: Definição de relação e endo-relação em PVS

Definição 3: o que é topo? A relação topo (notada como $\top$) é a relação entre todos os elementos com todos os elementos de dois tipos. Se os tipos forem *Pessoas* e *Direitos*, a relação topo modelaria o ideal para a Declaração Universal dos Direitos Humanos, pois cada pessoa deve ter todos os direitos e cada direito deve pertencer a todas as pessoas.

Mas com dois tipos arbitrários, Q e W, por exemplo, a definição mais geral será:

$$\top \overset{def}{=} Q \times W$$

Especificação em PVS:

Definição 4: o que é base? A relação base (notada como $\bot$) é a relação em que nenhum elemento se relaciona com nenhum elemento entre dois tipos. Nesse caso, é uma outra forma de expressar o conjunto vazio.

$$\bot \overset{def}{=} \emptyset$$

Especificação em PVS:

Definição 5: o que é relação identidade? Relação identidade (função identidade, notada como id) é uma endo-relação (relação entre elementos de mesmo tipo) em que um elemento se relaciona consigo. Exemplificando com o tipo *Pessoas* e supondo que todos tenham boa autoestima, a relação de amor próprio seria uma outra forma de função identidade.

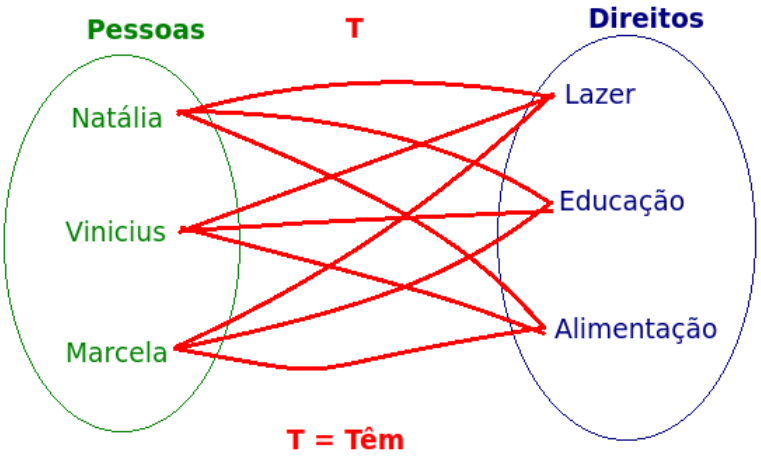


Figura 2.3: A relação topo: todos devem ter todos os direitos.

```
Top : relation[A,B] = fullset
```

Figura 2.4: Definição de topo em PVS

$$id\ x \stackrel{def}{=} x$$

A relação identidade é formalizada em PVS colocando um abre e fecha parêntesis no sinal de iguadade.

Definição 6: operadores de relações. Os operadores de relações são funções que transformam relações em outras ou relações de ordem superior (relações de relações). Elas são:

- Intersecção $(\_ \cap \_) : x(A \cap B)y \stackrel{def}{=} xAy \wedge xBy$
- União $(\_ \cup \_) : x(A \cup B)y \stackrel{def}{=} xAy \vee xBy$
- Inclusão $(\_ \subseteq \_) : A \subseteq B \stackrel{def}{=} \langle \forall x, y :: xAy \Rightarrow xBy \rangle$
- Composição $(\_ \cdot \_) : eA \cdot Bh \stackrel{def}{=} \langle \exists x :: eAx \wedge xBh \rangle$
- Complemento $(\text{-}) : e\overline{A}x \stackrel{def}{=} \neg(eAx)$
- Converso $(\_^o) : eA^o x \stackrel{def}{=} xAe$
- Igualdade $(\_ = \_) : A = B \stackrel{def}{=} \langle \forall a, b :: aAb \Leftrightarrow aBb \rangle$

Em PVS todos os conceitos acima já estão formalizados.

Como dito na introdução deste capítulo, intersecção, união, complemento e inclusão são conceitos básicos aprendidos. Mas composição e converso precisam de exemplos.

Seja a relação *Leram* o relacionamento entre os tipos *Pessoas* e *Livros*. E seja também a relação *Contem* a relação entre os tipos *Livros* e *Assuntos*. Sejam elas modelos

```
Botton : relation[A,B] = emptyset
```

Figura 2.5: Definição de base em PVS

verossímeis de relações e objetos do mundo real. Para que uma pessoa aprenda algum assunto, ela deve ter lido algum livro sobre ele. Ou seja, para que uma pessoa aprenda um assunto tem que existir algum livro que esteja relacionado com ele. Logo, a relação *Aprenderam* pode ser definida por composição de relações:

$$Aprenderam \stackrel{def}{=} Leram \cdot Contem$$

A figura 2.6 representa o modelo descrito acima.

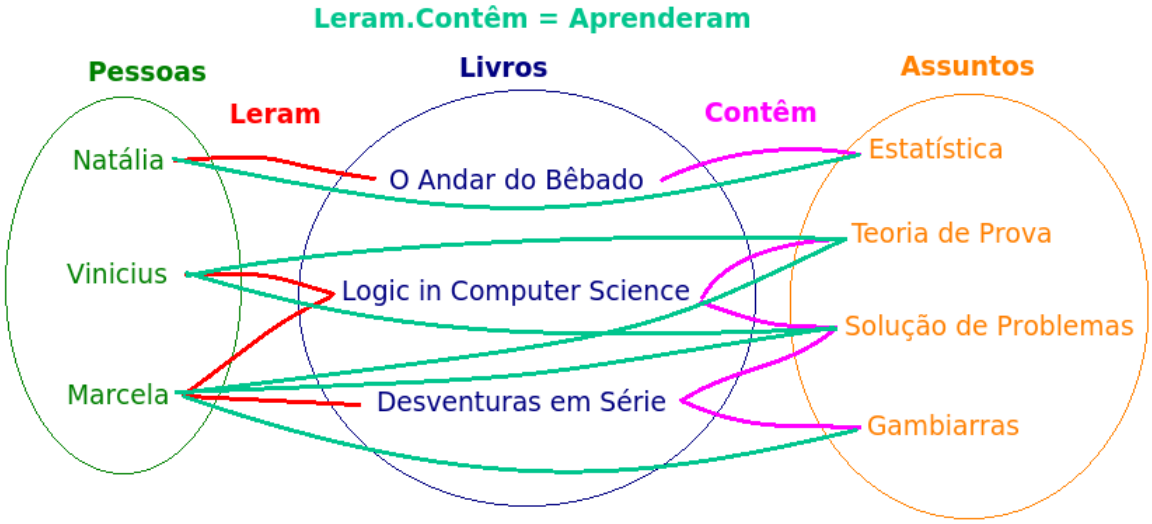


Figura 2.6: Composição de relações: pessoas que leem sobre um assunto o aprendem.

Na língua portuguesa, verbos transitivos possuem a voz ativa, em que o elemento mais importante é o sujeito, e a voz passiva, que exalta o objeto da oração. Para o exemplo anterior, a relação *Foram Lidos Por*, é o converso de *Leram*. O exemplo de converso pode ser lido na figura 2.7.

Definição 7: *kernel* e imagem. A imagem de uma relação (notada como Img) é outra relação que corresponde elementos da **Direita** que se relacionam com algum elemento em comum do **Esquerda**. Usando a notação *pointfree* fica:

$$ImgP \stackrel{def}{=} P \cdot P^o$$

Por exemplo, pessoas possuem mesmas contas em bancos na figura 2.8.

Já o *kernel* de uma relação (notada como Ker), relaciona elementos da **Esquerda** que tem em comum elementos da **Direita**. Pode ser especificado da seguinte forma:

$$KerP \stackrel{def}{=} P^o \cdot P$$

Como exemplo temos contas bancárias que possuem um mesmo dono na figura 2.9.

Definição 8: domínio e contradomínio (alcance) O domínio (notado como δ) de uma relação representa quais são os elementos da **Direita** que estão relacionados.

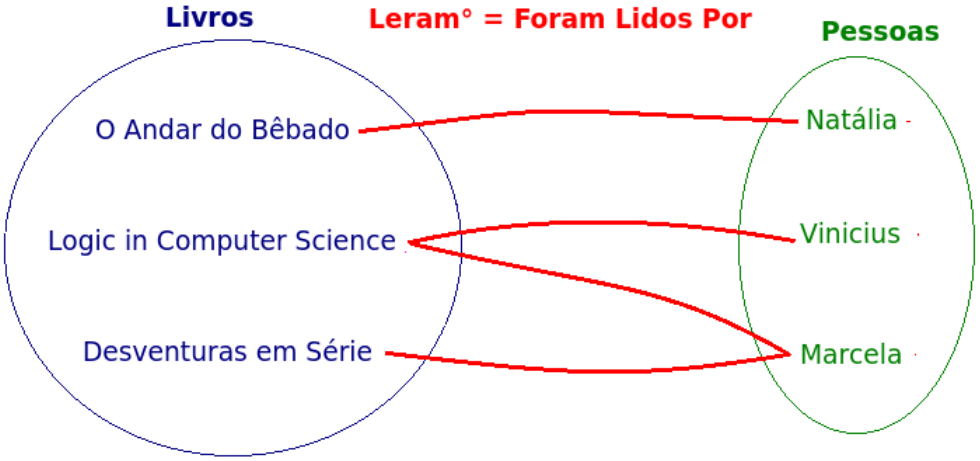


Figura 2.7: Converso de relações: livros foram lidos por pessoas.

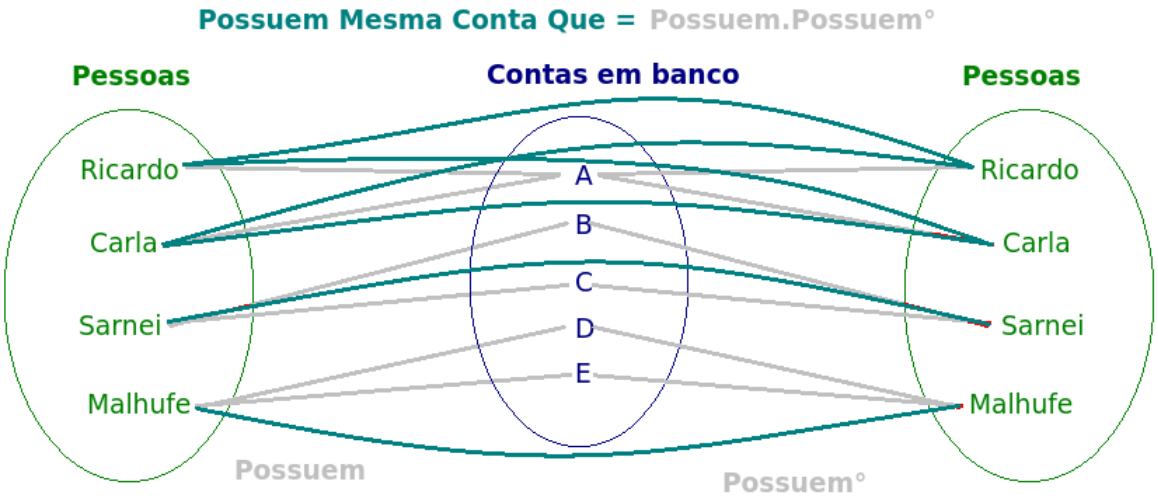


Figura 2.8: Imagem: pessoas que possuem conta em comum.

$$\delta R \stackrel{def}{=} KerR \cap id$$

Por sua vez, o contradomínio (ou alcance, com notação ρ) representa elementos da **Esquerda** de uma relação que estão relacionados. Logo:

$$\rho R \stackrel{def}{=} ImgR \cap id$$

Definição 9: taxonomia e hierarquia das relações. Relações podem possuir características em comum. Essas caracteríticas, de tão comum, podem ser agrupadas em definições que permitem estudar melhor as propriedades e permitem criar padrões para especificações de modelagem de banco de dados, por exemplo.

Assim, há as seguintes definições para relações:

- R Inteira $\stackrel{def}{=} id \subseteq KerR$

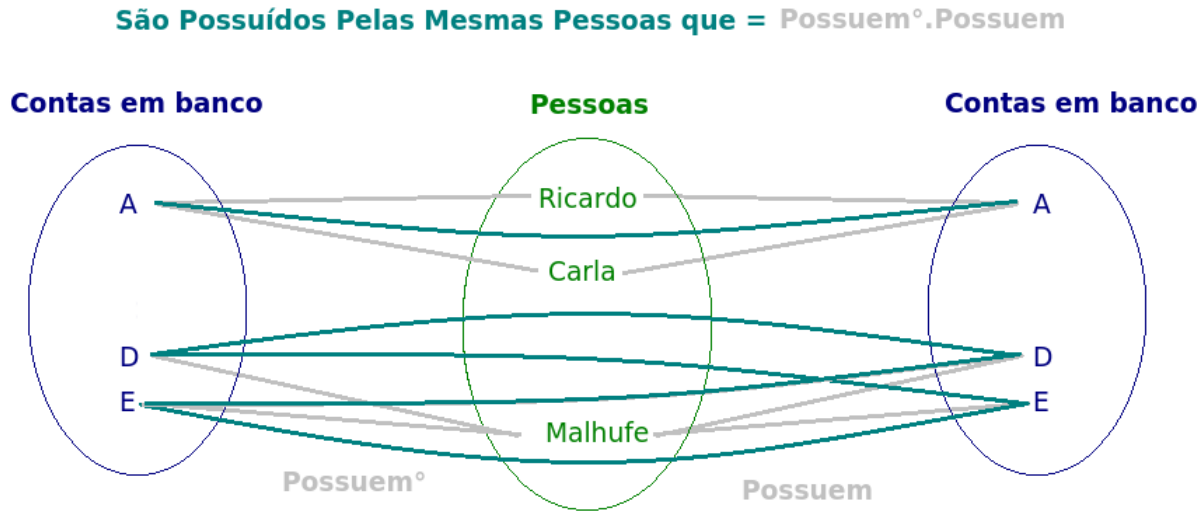


Figura 2.9: Kernel: contas que são possuídas por pelo menos uma pessoa em comum.

"Todos os elementos da **Direita** estão relacionados."

- R Sobrejetiva $\stackrel{def}{=} id \subseteq ImgR$

  "Todos os elementos da **Esquerda** estão relacionados."

- R Simples $\stackrel{def}{=} ImgR \subseteq id$

  "Elementos da **Esquerda** se relaciona a no máximo um da **Direita**."

- R Injetiva $\stackrel{def}{=} KerR \subseteq id$

  "Elementos da **Direita** se relaciona a no máximo um da **Esquerda**."

- R Aleatória quando não segue nenhum dos padrões anteriores.

Na figura há um diagrama sobre a hierarquia das relações. Os conceitos não definidos são construídos a partir das setas. Por exemplo, uma Função é uma relação Inteira e Simples, pois as setas direcionadas a função vem dessas relações. Pelo diagrama, há conceitos ainda não definidos:

- Analogia
- Atuação
- Instância
- Total
- Oligarquia
- Aleatória

Segundo o dicionário Aurélio online, (disponível em `http://www.dicionariodoaurelio.com/`) o conceito de analogia é:

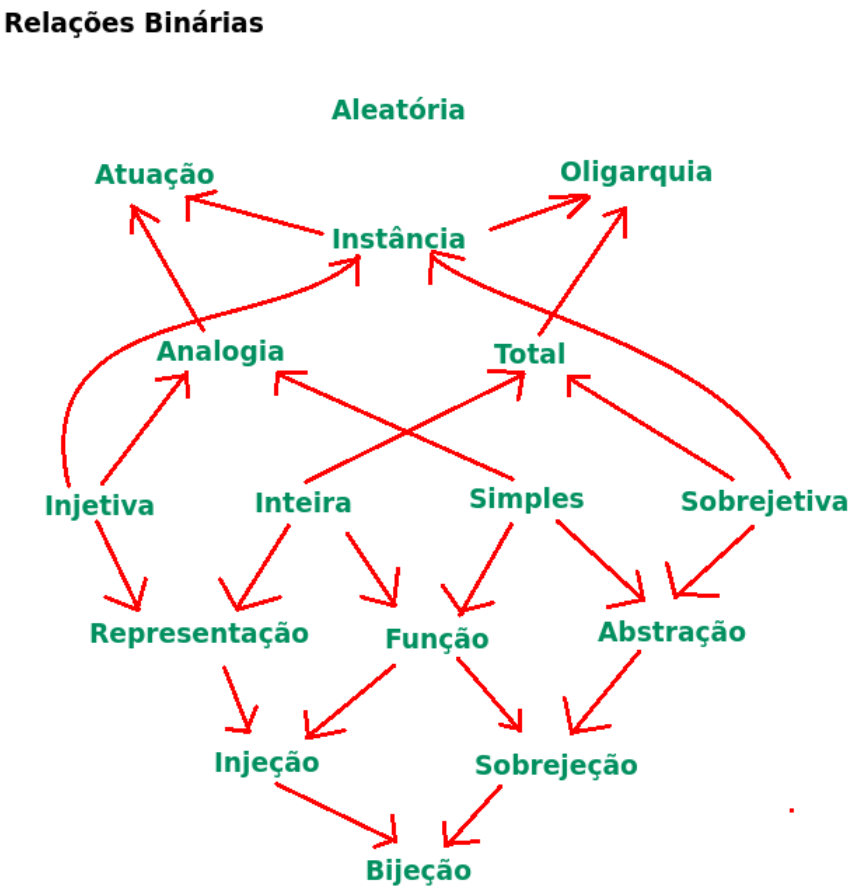


Figura 2.10: Hierarquia de relações binárias

> s.f. Relação, semelhança de uma coisa com outra: analogia de formas, de gostos. // Filosofia. Analogias da experiência, princípios que governam a ligação entre os fenômenos, segundo Kant. // Por analogia, de acordo com as relações que existam entre as coisas: raciocinar por analogia.

Mostrar-se-á um exemplo de analogia retirada do livro [4], *The Math Gene*. Ela foi criada em 1970 pelo psicólogo britânico Peter Wason.

> Há quatro cartas em uma mesa. Em um lado de cada uma há uma letra, de outro, um número. Os lados virados para cima são:
>
> E K 4 5
>
> Quais cartas é suficiente virar para verificar a seguinte regra: se em um lado há uma vogal, de outro há um número par?
>
> A situação acima corresponde a situação logo a seguir.
>
> Em uma festa, há jovens bebendo. E em uma mesa há quatro jovens. O primeiro está bebendo cerveja. O segundo, um refrigerante. Ambos estão com as identidades viradas para baixo, e não se pode ver as idades. O terceiro e o quarto, ambos com identidades viradas para cima, possuem maior e menor idade respectivamente, mas nos copos podem ter bebida alcoólica ou não.

Tem-se:

Cerveja Refrigerante Maior Menor

Para verificar quais deles não podem beber, quais identidades e bebidas precisam ser verificadas?

A analogia com as duas situações é:

- Vogal corresponde à bebida alcoólica;
- Consoante corresponde à bebida não alcoólica
- Número par corresponde à maior de idade
- Número ímpar corresponde à menor de idade

Logo, há uma relação *Correspondem a* entre os objetos acima:

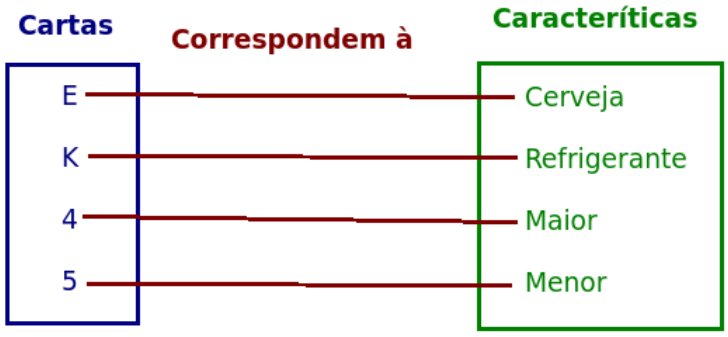


Figura 2.11: Solução por analogia

Para resolver a primeira questão, basta resolver a segunda. Assim, deve-se pedir para que o garoto que bebe cerveja mostre a identidade e verificar se o menor está bebendo bebida alcoólica. Com efeito, para saber se a regra sobre cartas foi obedecida, basta virar a carta com E e com 5.

Portanto, quando há uma analogia, há uma semelhança. Para que haja essa semelhança, os elementos da **Direita** não podem colidir com os da **Esquerda** e vice-eversa. Logo, é uma relação simples e injetiva.

Atuação foi usado como uma metáfora para exercer um papel. Quando há uma peça, os elementos da peça tentam representar elementos do mundo real. E existe uma correspondência direta, ou seja, uma analogia. E todos os elementos de uma peça representam algo do mundo real. Assim, há uma relação sobrejetiva.

A palavra instância vem da orientação a objetos. Nele há os elementos que classificam características e ações do mundo real. Esses elementos são chamados de classes. Quando essas caraterísticas são escolhidas, têm-se objetos. A ação de criar objetos é denotada de instanciar. Todos os objetos possuem uma classe, portanto há uma relação de sobrejeção. Em orientação a objetos, para uma modelagem bem feita, objetos são instâncias de uma única classe. Logo, há uma relação de injeção.

Uma relação é total, quando todos os elementos em uma relação estão relacinados de alguma forma. Logo ela precisa ser sobrejetiva e inteira.

Oligarquia vem do grego *ologoi* que significa "de poucos"e *arkos* que é "poder". Em uma oligarquia, todos estão relacionados, portanto uma relação total. E na relação de poder, os subordinados possuem apenas um superior, ou seja, é uma relação injetiva.

E, por fim, quando algo não segue um padrão é dito aleatório. Logo, uma relação sem padrão é aleatória.

Para endo-relações, há os seguintes conceitos

- R Conexa $\stackrel{def}{=} R \cup R^o = \top$ "Dois elementos sempre estarão relacionados."
- R Antissimétrica $\stackrel{def}{=} R \cap R^o \subseteq id$ "Ao tentar inverter os elementos da relação, e continuar relacionado significa que são iguais."
- R Reflexiva $\stackrel{def}{=} id \subseteq R$ "Todos os elementos estão relacionados consigo mesmo."
- R Irreflexiva $\stackrel{def}{=} R \cap id = \bot$ "Nenhum elemento está relacionado consigo."
- R Transitiva $\stackrel{def}{=} R.R \subseteq R$ "Se um elemento está relacionado com um segundo e este com um terceiro, então o primeiro está relacionado com um terceiro."
- R Simétrica $\stackrel{def}{=} R = R^o$ "O relacionamento é recíproco."
- R Densa $\stackrel{def}{=} R \subseteq R.R$
- R Cotransitiva (ou Comparativa) $\stackrel{def}{=} \overline{R}.\overline{R} \subseteq \overline{R}$
- R Assimétrica $\stackrel{def}{=} R \cap R^o = \bot$
- R Reflexiva à Direita $\stackrel{def}{=} \delta R \subseteq R$
- R Reflexiva à Esquerda $\stackrel{def}{=} \rho R \subseteq R$
- R Fracamente Simétrica $\stackrel{def}{=} R \subseteq R^o \cup id$
- R Euclidiana (à Direita) $\stackrel{def}{=} KerR \subseteq R$
- R Euclidiana à Esquerda $\stackrel{def}{=} ImgR \subseteq R$
- R Estreita $\stackrel{def}{=} R \cup id = \top$
- R Tricotomia $\stackrel{def}{=} R \cup R^o \cup id = \top$
- R Quasi-reflexiva $\stackrel{def}{=} \delta(R \cup R^o) \subseteq R$
- R Antitransitiva $\stackrel{def}{=} R.R \cap R = \bot$

Logo, pode-se construir o seguinte diagrama da figura 2.12.

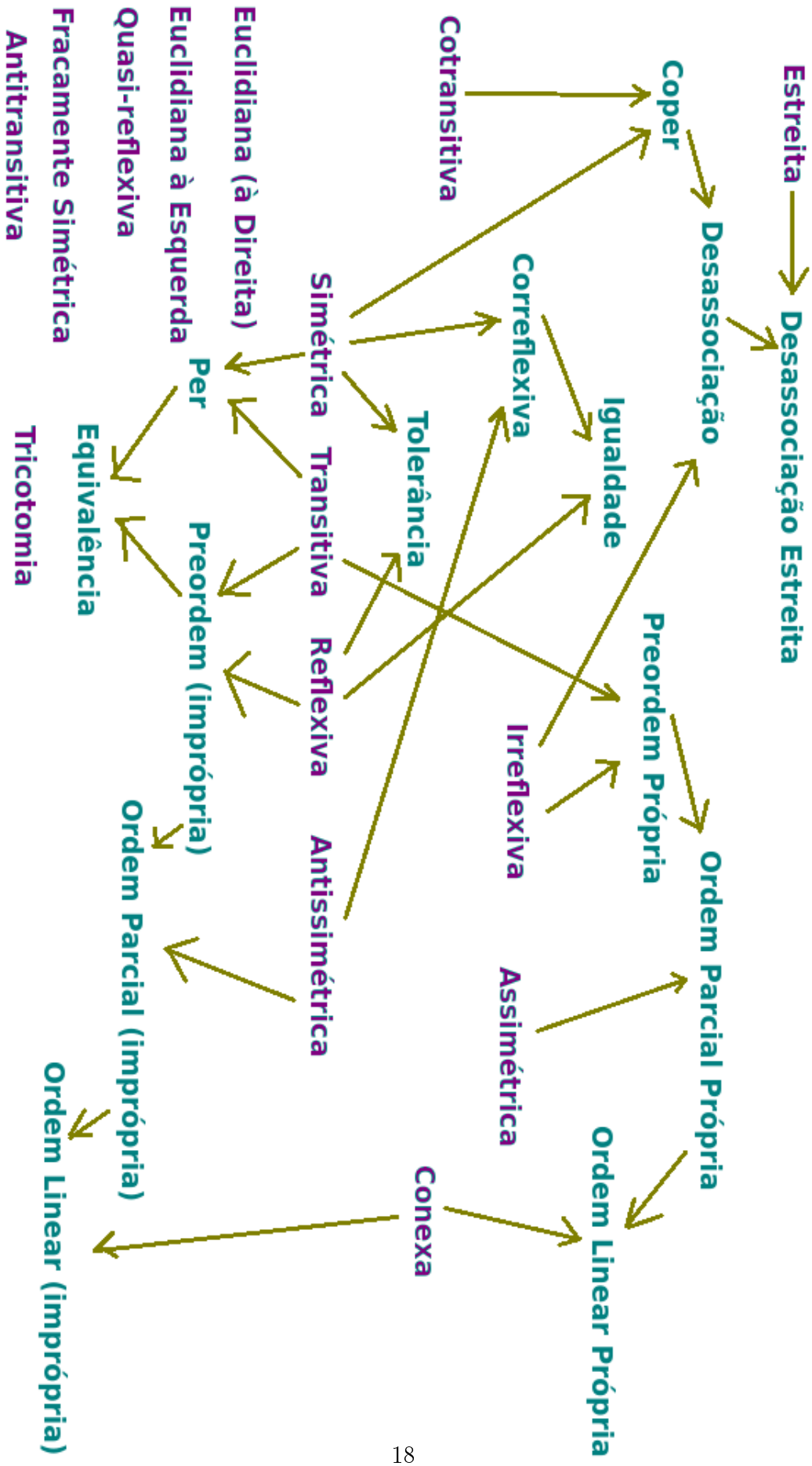


Figura 2.12: Diagrama de taxonomia de endo-relações

Nela, as endo-relações de conceitos básicos estão em azul-escuro. As endo-relações que usam esses conceitos estão em ciano-escuro. As setas entrando em uma determinada palavra indica que a conjunção dos elementos que estão ligados formam o conceito dessa palavra. Por exemplo, uma ordem linear é uma ordem parcial que é conexa. Por sua vez, ordem parcial é uma preordem antissimétrica. Por fim, uma preordem é uma relação transitiva e reflexiva.

Vale ressaltar que o conceito de relação correflexiva é: R Correflexiva $\stackrel{def}{=} R \subseteq id$. Mas dizer que uma relação é correflexiva é equivalente a dizer que ela é simétrica e antissimétrica como será demonstrado nos próximos capítulos. As relações correflexivas modelam predicados unários, relacionando elementos que obedecem ao predicado com ele mesmo. Portanto, tem-se a última definição.

Definição 10: transformador unário-binário. Seja p um predicado unário. O transformador Φ_p é uma correflexiva definido, para todos os elementos m e n, como:

$$m \Phi_p n \stackrel{def}{=} m \ id \ n \wedge p \ n$$

Capítulo 3

Propriedades do Cálculo de Sistema de Informação

3.1 Introdução

Em uma lógica, depois de definida uma linguagem e um significado associado, ou seja, uma semântica, é preciso criar regras de inferência a partir de propriedades observadas. A esse conjunto de regras, dá-se o nome de regras de inferência. A partir do momento que há regras, é preciso mostrar que cada passo de inferência está correto. Essa propriedade que deve ser imposta á lógica é chamada correção.

Como o objetivo da lógica é o estudo das proposições, que é tudo aquilo que pode ser classificado como verdade, então a semântica em lógica está associado com a ideia de verdade e falsidade.

Quando consegue se demonstrar tudo o que é verdade com as regras de inferência, tem-se a propriedade de completude, que também deve ser imposta a lógica.

Assim, há símbolos associados a cada uma da forma de se inferir dado um conjunto de premissas Ψ e uma conclusão ψ:

- observando-se as regras de inferência: $\Psi \vdash \psi$
- observando-se a semântica: $\Psi \models \psi$

Logo, será necessária provar a completude e correção do cálculo.

A organização das regras de inferência dar-se no formato de árvore:

$$\frac{\text{Premissa}_1 \quad \ldots \quad \text{Premissa}_N}{\text{Conclusão}} \text{ regra de inferência}$$

Na figura acima a quantidade de premissas pode ser zero ou mais.

Para suposições, usar-se-á colchetes e uma etiqueta:

$$[\text{fórmula}]^{\text{etiqueta}}$$

Haverá momentos em que, dentro de premissas, será necessário criar conjuntos novos (que não aparecem em lugar nenhum de uma questão) em premissas. Esses conjuntos funcionarão como etiqueta:

[fórmula com um conjunto novo chamado etiqueta]

Quando há uma premissa, ela pode ser usada várias vezes na prova. Entretanto, quando há uma regra de conclusão, ela deve ser eliminada (cortada) em todas as ocorrências da árvore acima:

$$\frac{\cancel{[\text{fórmula}]^{\text{etiqueta}}}}{\begin{array}{c}\vdots\\ \text{Conclusão}\end{array}}\ \text{regra de conclusão, etiqueta}$$

A etiqueta não pode ser usada em nenhuma outra parte da demonstração uma vez eliminada em uma fórmula.

Para que uma demonstração esteja correta, todas as fórmulas com etiqueta devem ser eliminadas.

Em uma prova, pode-se acabar repetindo passos de demonstração. Para abreviar essas repetições usa-se o seguinte recurso:

$$\begin{array}{c}\vdots\\ \{\text{fórmula}\}^{\text{número}}\end{array}$$

$$\overbrace{\text{número},\text{etiqueta}_1,\cdots,\text{etiqueta}_n}^{\text{fórmula}}$$

Onde as etiquetas de 1 a n são etiquetas da prova da fórmula ainda não descarregadas. Depois de usar esse recurso de reuso de subprova, ao se descarregar fórmulas, deve-se descarregar todas as etiquetas do reuso.

A principal idéia das próximas seções é axiomatizar propriedades relacionadas à notação *pointfree* através de regras de inferência. Não é uma idéia inovadora, pois Tarski [20] axiomatizou o cálculo relacional com as seguintes regras (traduzidas pois Tarski usa símbolos diferentes e símbolos extras que podem ser representados pela linguagem escolhida), onde A, B e C são relações quaisquer:

1. $(A = B \wedge A = C) \Rightarrow A = C$
2. $A = B \Rightarrow (A \cup C = B \cup C \wedge A \cap C = B \cap C)$
3. $A \cap B = B \cap A \wedge A \cup B = B \cup A$
4. $(A \cup B) \cap C = (A \cap C) \cup (B \cap C) \wedge (A \cap B) \cup C = (A \cup C) \cap (B \cup C)$
5. $A \cup \bot = A \wedge A \cap \top = A$
6. $A \cup \overline{A} = \top \wedge A \cup \overline{A} = \bot$
7. $\neg(\top = \bot)$
8. $R^{oo} = R$
9. $(A.B)^o = B^o.A^o$

10. $A.(B.C) = (A.B).C$
11. $A.id = A$
12. $A.\top = \top \vee \top.\overline{A} = \top$
13. $(A.B).C^o = \bot \Rightarrow (B.A).C^o = \bot$

A principal diferença entre as axiomatizações de Tarski e as regras de inferência produzidas aqui é que nem tudo da axiomatização de Tarski pode ser provado. Por exemplo, "$\neg(\top = \bot)$"não é uma sentença que pode ser provada, pois $\top$ é o produto cartesiano entre tipos quaisquer e mas um deles pode ser vazio. Se ele for vazio, então $\top = \bot$.

Tarski pergunta em seu trabalho se qualquer proposição verdadeira pode ser demonstrada, ou seja, se os axiomas são completos. A resposta é "não". Lyndon responde a essa pergunta em [9].

Da mesma maneira pode se questionar se o cálculo explicado mais a frente é completo e também que tipo de problemas pode se expressar com a linguagem da álgebra relacional. Entretanto o foco do trabalho é verificar a facilidade de expressão e de prova usando linguagem e cálculo respectivamente.

Em especificações formais, verifica-se que a linguagem mais usada é a da lógica de primeira ordem. Mas supondo que a notação *pointfree* não descreva todos os problemas da lógica de primeira ordem, se descoberto que a notação *pointfree* é melhor (ou seja, que as pessoas sentem menos dificuldade para expressar os problemas possíveis de se especificar) a notação *pointfree* será mais útil para alguns tipos de problemas.

Da mesma forma, se o cálculo relacional mostrar-se incompleto mas as pessoas usam menos esforço cognitivo para usá-lo ele seria útil para resolver alguns problemas. Portanto, de qualquer forma, se a notação *pointfree* facilita a expressão de subproblemas expressos pela lógica de primeira ordem e o cálculo relacional facilita a prova de alguns teoremas elas podem ser usadas de maneira complementar a notação e cálculo da lógica de primeira ordem.

Portanto, poder de expressão da notação e completude do cálculo são problemas importantes, mas o que se deseja é busca de técnicas que tornem mais rápido especificação e demonstração sejam elas complementares ou substitutivas às já existentes.

3.2 Propriedades, Regras e Demonstrações

Definição 11: regras básicas e intuitivas de conjuntos e igualdade. As seguintes derivações são corretas para relações A, B, C e fórmula ϕ. A maioria delas está implementada como comandos de prova ou lemas da biblioteca de PVS *sets_ lemas*.

- Introdução da igualdade:

$$\frac{A \subseteq B \quad B \subseteq A}{A = B} = i$$

```
subset_antisymmetric: LEMMA subset?(a, b) AND subset?(b, a) IMPLIES a = b
```

Figura 3.1: Introdução da igualdade

- Substituição ($\phi[A/B]$ denota substituir na fórmula ϕ algumas ocorrências de conjuntos A por B, implementado com regra de prova):

$$\frac{\phi \quad A = B}{\phi[A/B]} =$$

$$\frac{\phi \quad A = B}{\phi[B/A]} =$$

- Reflexibilidade da igualdade (implementado como comando de prova):

$$\frac{}{A = A}\ id \subseteq (=)$$

- Eliminação ($\cup\, e$) e introdução ($\cup\, i$) da união:

$$\frac{A \cup B \subseteq C}{A \subseteq C} \cup e$$

$$\frac{A \cup B \subseteq C}{B \subseteq C} \cup e$$

$$\frac{A \subseteq C \quad B \subseteq C}{A \cup B \subseteq C} \cup i$$

```
subset_union: LEMMA subset?(union(a, b), c) = (subset?(a, c) AND subset?(b, c))
```

Figura 3.2: Eliminação e Introdução da união

- Introdução da intersecção:

$$\frac{A \subseteq B \quad A \subseteq C}{A \subseteq B \cap C} \cap i$$

```
intersection_lower_bound: LEMMA
  subset?(c, a) and subset?(c, b) IMPLIES subset?(c, intersection(a, b))
```

Figura 3.3: Introdução da intersecção

- Igualdade com o fundo (ou conjunto vazio):

$$\frac{A \subseteq \bot}{A = \bot} = \bot$$

```
emptyset_min: LEMMA subset?(a, emptyset) IMPLIES a = emptyset
```

Figura 3.4: Igualdade com o fundo

- Igualdade com o topo:

$$\frac{\top \subseteq A}{A = \top} =\top$$

```
fullset_max: LEMMA subset?(fullset, a)  IMPLIES a = fullset
```

Figura 3.5: Igualdade com o topo

- Igualdade do duplo complemento:

$$\frac{}{\overline{\overline{A}} = A}\ \overline{\overline{()}}$$

```
complement_complement: LEMMA complement(complement(a)) = a
```

Figura 3.6: Igualdade do duplo complemento

- Os conjuntos estão no topo:

$$\frac{}{\overline{A \subseteq \top}}\ \subseteq \top$$

- O fundo está nos conjuntos:

$$\frac{}{\overline{\bot \subseteq A}}\ \bot \subseteq$$

- Troca da inclusão:

$$\frac{A \subseteq B}{\overline{B} \subseteq \overline{A}}\ \overline{()} \subseteq \overline{()}$$

- Transitividade da inclusão:

$$\frac{A \subseteq B \quad B \subseteq C}{A \subseteq C}\ \subseteq \tag{3.1}$$

- Distributividade da união em relação a intersecção:

$$\frac{}{(C \cup A) \cap (C \cup B) = C \cup (A \cap B)}\ \cup\cap$$

```
subset_fullset: LEMMA subset?(a, fullset)
```

Figura 3.7: Os conjuntos estão no topo

```
subset_emptyset: LEMMA subset?(emptyset, a)
```

Figura 3.8: O fundo está nos conjuntos

```
subset_complement: LEMMA
  subset?(complement(a), complement(b)) IFF subset?(b, a)
```

Figura 3.9: Troca da inclusão

- Distributividade da intersecção em relação a união:

$$\overline{(C \cap A) \cup (C \cap B) = C \cap (A \cup B)}\ {}^{\cap\cup}$$

- Igualdade do duplo complemento:

$$\overline{\overline{\overline{A}} = A}\ {}^{\overline{\overline{()}}}$$

Cada uma das propriedades capturadas pelas regras acima estão demonstradas em PVS. As regras abaixo não foram especificadas em nenhuma biblioteca em PVS. Logo, elas foram formalizadas no arquivo *Extra_ Sets*.

- Eliminação da igualdade:

$$\frac{A = B}{A \subseteq B} = e$$

$$\frac{A = B}{B \subseteq A} = e$$

- Eliminação da intersecção:

$$\frac{A \subseteq B \cap C}{A \subseteq B} \cap e$$

$$\frac{A \subseteq B \cap C}{A \subseteq C} \cap e$$

- Igualdade com o fundo:

$$\overline{A \cap \overline{A} = \bot} = \bot$$

- Igualdade com o topo:

$$\overline{A \cup \overline{A} = \top} = \top$$

```
subset_transitive: LEMMA
  subset?(a, b) AND subset?(b, c) IMPLIES subset?(a, c)
```

Figura 3.10: Transitividade da inclusão

```
distribute_union_intersection: LEMMA
  union(a, intersection(b, c)) = intersection(union(a, b), union(a, c))
```

Figura 3.11: Distributividade da união na intersecção

```
distribute_intersection_union: LEMMA
  intersection(a, union(b, c))
      = union(intersection(a, b), intersection(a, c))
```

Figura 3.12: Distributividade da intersecção na união

```
complement_complement: LEMMA complement(complement(a)) = a
```

Figura 3.13: Igualdade do duplo complemento

```
Equality_inclusion1 : LEMMA
A = B => subset?(A,B)

Equality_inclusion2 : LEMMA
A = B => subset?(B,A)
```

Figura 3.14: Eliminação da igualdade

```
Intersection_elimininationl: LEMMA
subset?(N,intersection(A,B)) => subset?(N,A)

Intersection_elimininationl: LEMMA
subset?(N,intersection(A,B)) => subset?(N,B)
```

Figura 3.15: Eliminação da intersecção

```
Emptyset_equality: LEMMA
intersection(A,complement(A)) = emptyset
```

Figura 3.16: Igualdade com o fundo

```
Fullset_equality: LEMMA
union(A,complement(A)) = fullset
```

Figura 3.17: Igualdade com o topo

Além das regras descritas acima, há regras da lógica proposicional como as regras abaixo retiradas do livro *Logic in Computer Science*, referência [7]. Elas estão implementadas em PVS pelo cálculo de Gentzen nas regras de prova. Para as fórmulas ϕ e ψ, onde $false$ é o absurdo, valem:

- Introdução da conjunção:

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i$$

- Eliminação da conjunção:

$$\frac{\phi \wedge \psi}{\phi} \wedge e$$

$$\frac{\phi \wedge \psi}{\psi} \wedge e$$

- Introdução da disjunção:

$$\frac{\phi}{\phi \vee \psi} \vee i$$

$$\frac{\psi}{\phi \vee \psi} \vee i$$

- Eliminação da dupla negação:

$$\frac{\neg\neg\phi}{\phi} \neg\neg\ e$$

- Eliminação da implicação:

$$\frac{\phi \quad \phi \Rightarrow \psi}{\psi} \Rightarrow e$$

- Introdução da implicação:

$$\frac{\begin{array}{c}[\phi]^{\not A} \\ \vdots \\ \psi\end{array}}{\phi \Rightarrow \psi} \Rightarrow i, S$$

- Eliminação da disjunção:

$$\frac{\phi \vee \psi \quad \begin{array}{c}[\phi]^{\not A} \\ \vdots \\ \chi\end{array} \quad \begin{array}{c}[\psi]^{\not B} \\ \vdots \\ \chi\end{array}}{\chi} \vee e, A, B$$

- Eliminação da negação:

$$\frac{\phi \quad \neg\phi}{false} \neg e$$

- Introdução da negação:

$$\frac{\begin{array}{c}[\phi]^{\not A} \\ \vdots \\ false\end{array}}{\neg\phi} \neg i, S$$

Definição 12: regras novas. Cada uma das regras abaixo são regras que não se costuma aprender sobre teoria dos conjuntos.

Introdução da inclusão (onde o conjunto N é uma etiqueta):

$$\frac{\begin{array}{c}\not{[N \subseteq A]} \\ \vdots \\ N \subseteq B\end{array}}{A \subseteq B} \subseteq i, N$$

$$\frac{\begin{array}{c}\not{[A \subseteq N]} \\ \vdots \\ B \subseteq N\end{array}}{B \subseteq A} \subseteq i, N$$

As regras acima capturam as seguintes propriedades:

$$\langle \forall N :: N \subseteq A \Rightarrow N \subseteq B \rangle \equiv A \subseteq B \tag{3.2}$$

$$\langle \forall N :: A \subseteq N \Rightarrow B \subseteq N \rangle \equiv B \subseteq A \tag{3.3}$$

Para mostrar a necessidade, basta substituir N por A. Para suficiência, supor que N é um subconjunto qualquer de A e pela transitividade da relação de inclusão, ele estará em B.

Necessidade:

Partindo de :

$$\langle \forall N :: N \subseteq A \Rightarrow N \subseteq B \rangle$$

Pode-se tomar N como sendo A:

$$A \subseteq A \Rightarrow A \subseteq B$$

Como $A \subseteq A$ é válido, então

$$A \subseteq B$$

Suficiência:

Partindo de:

$$A \subseteq B$$

Tome N um subconjunto qualquer de A. Pela transitividade dos subconjuntos, N também será subconjunto de A. Logo:

$$\langle \forall N :: N \subseteq A \Rightarrow N \subseteq B \rangle$$

Por necessidade e suficiência provou-se 3.2.

A prova para 3.3 é análoga a anterior, ou seja, existe uma regra de reescrita que, aplicada na necessidade e suficiência, gera uma prova válida:

$$\theta = \{N \subseteq A / B \subseteq N, N \subseteq B / A \subseteq N, \tag{3.4}$$

$$"subconjunto"/"superconjunto", A \subseteq A/B \subseteq B."sendo\ A"/"sendo\ B"\} \tag{3.5}$$

Até a formalização em PVS da especificações descrita na figura 3.18 mostra quão análogo eles são. Nas árvores de prova de formalização na figura 3.19 nota-se que elas são de mesma estrutura.

```
Inclusion1 : LEMMA
(FORALL N: subset?(N,A) => subset?(N,B)) <=> subset?(A,B)

Inclusion2 : LEMMA
(FORALL N: subset?(B,N) => subset?(A,N)) <=> subset?(A,B)
```

Figura 3.18: Formalização de propriedade sobre conjunto.

Ao aplicar as regras de reescrita 3.2 na demonstração anterior, produz-se a seguinte prova:

Necessidade:

Partindo de :

$$\langle \forall N :: B \subseteq N \Rightarrow A \subseteq N \rangle$$

Pode-se tomar N como sendo B:

$$B \subseteq B \Rightarrow A \subseteq B$$

Como $B \subseteq B$ é válido, então

$$A \subseteq B$$

Suficiência:

Partindo de:

$$A \subseteq B$$

Tome N um superconjunto qualquer de B. Pela transitividade de superconjuntos, N também será superconjunto de A. Logo:

$$\langle \forall N :: B \subseteq N \Rightarrow A \subseteq N \rangle$$

Igualdade do duplo converso:

$$\overline{A^{oo} = A}^{\,oo}$$

A demonstração dessa regra está no fato que aplicar o duplo converso troca os elementos das duplas duas vezes de lugar e isso os mantêm ordenados.

Suponha (m, n) elemento qualquer de A:

$$(m, n) \in A$$

Que é equivalente a:

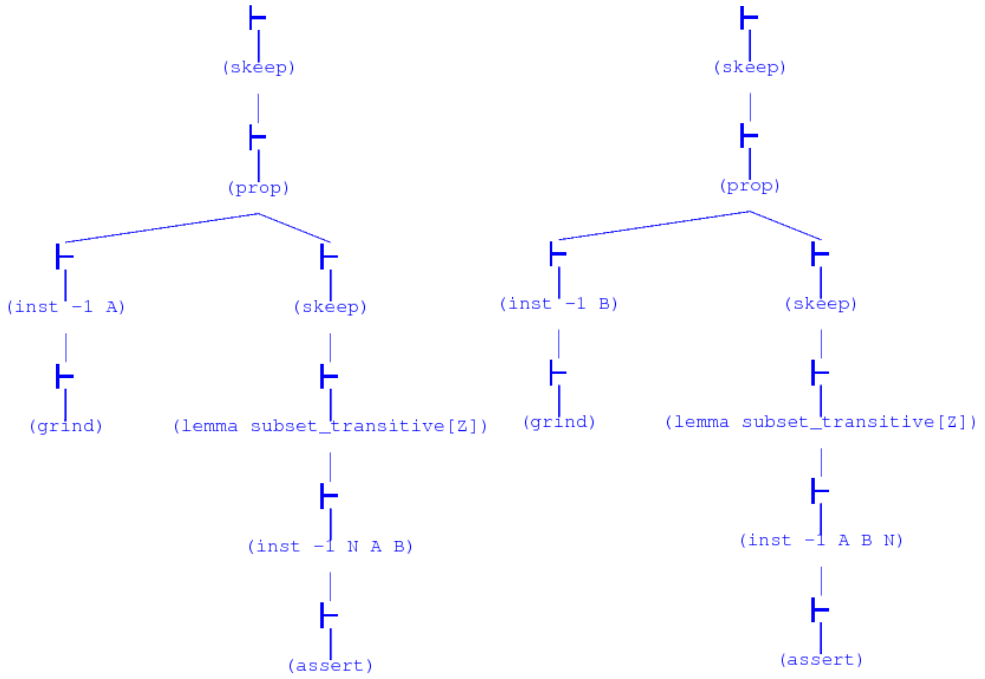


Figura 3.19: Formalização de propriedade sobre conjunto.

```
converse_converse: LEMMA
    converse(converse(K)) = K
```

Figura 3.20: O duplo converso

$$(n, m) \in A^o$$

Que também é equivalente à:

$$(m, n) \in A^{oo}$$

Logo:

$$(m, n) \in A \Leftrightarrow (m, n) \in A^{oo}$$

Por definição:

$$A = A^{oo}$$

Regras de monotonia sob a inclusão:

$$\frac{A \subseteq B}{A^o \subseteq B^o}\ o\subseteq$$

$$\frac{A \subseteq B \quad C \subseteq D}{A.C \subseteq B.D}\ .\subseteq$$

```
converse_inclusion: LEMMA
    subset?(K,L) => subset?(converse(K),converse(L))
```

Figura 3.21: Monotonia do converso em relação a inclusão

```
compose_inclusion : LEMMA
    subset?(K,L) and subset?(M,N) => subset?(K o M,L o N)
```

Figura 3.22: Monotonia da composição em relação a inclusão

Demonstração da monotonia do converso em relação a inclusão:
Usando como premissa $A \subseteq B$, supor para (m, n) elemento qualquer:

$$(m, n) \in A^o$$

Pela definição de converso:

$$(n, m) \in A$$

Como $A \subseteq B$, pela transitividade de $\subseteq$:

$$(n, m) \in B$$

Pela definição de converso:

$$(m, n) \in B^o$$

Assim:

$$(m, n) \in A^o \Rightarrow (m, n) \in B^o$$

Pela definição de inclusão:

$$A^o \subseteq B^o$$

Demonstração da monotonia da composição em relação a inclusão:
Usando como premissas $A \subseteq B$ e $C \subseteq D$, supor para (m, n) qualquer:

$$(m, n) \in A.C$$

Pela definição de composição, há um z tal que

$$(m, z) \in A \wedge (z, n) \in C$$

Por $A \subseteq B$ e $C \subseteq D$ e pela transitividade da inclusão, pode-se concluir que:

$$(m, z) \in B \wedge (z, n) \in D$$

Pela definição de composição:

$$(m, n) \in B.D$$

Logo:

$$(m, n) \in A.C \Rightarrow (m, n) \in B.D$$

Pela definição de inclusão:

$$A.C \subseteq B.D$$

Identidade é o elemento neutro da composição:

$$\frac{}{A.id = A}\ .id$$

$$\frac{}{id.A = A}\ .id$$

```
neutral_identity1 : LEMMA
    K o (=) = K

neutral_identity2 : LEMMA
    (=) o K = K
```

Figura 3.23: Identidade é o elemento neutro da composição

A demonstração da neutralidade da identidade é:
Supor (m, n) um elemento qualquer de A:

$$(m, n) \in A$$

Como $(m, m) \in id$ é valido, a fórmula acima é equivalente a:

$$(m, m) \in id \wedge (m, n) \in A$$

E também a:

$$(m, n) \in A \wedge (n, n) \in id$$

Pois, $(n, n) \in id$. Pela noção de composição, será equivalente a:

$$(m, n) \in id.A$$

E também a:

$$(m, n) \in A.id$$

Logo:

$$(m, n) \in id.A \Leftrightarrow (m, n) \in A \Leftrightarrow (m, n) \in A.id$$

Pela definição de igualdade de conjuntos:

$$id.A = A = A.id$$

Fundo composto é fundo:

$$\frac{}{A.\bot = \bot}\ .\bot$$

$$\frac{}{\bot.A = \bot}\ .\bot$$

```
botton_compose1 : LEMMA
    K o Botton = Botton

botton_compose2 : LEMMA
    Botton o M  = Botton
```

Figura 3.24: Fundo composto é fundo

Para o elemento (m, n):

$$m \bot n$$

Equivale a:

$$m \bot z \wedge z A n$$

E também a:

$$m A a \wedge a \bot n$$

Pois todas as fórmulas são falsas já que $\bot$ é o conjunto vazio . Logo são equivalentes a:

$$m A.\bot n$$

$$m \bot.A n$$

Assim:

$$m A.\bot n \Leftrightarrow m \bot n \Leftrightarrow m \bot.A n \tag{3.6}$$

Portanto:

$$A.\bot = \bot = \bot.A \tag{3.7}$$

Complemento do converso é o converso do complemento:

$$\frac{}{\overline{A^o} = (\overline{A})^o}\ \overline{()^o}$$

```
complement_converse : LEMMA
    complement(converse(K)) = converse(complement(K))
```

Figura 3.25: Complemento do converso é o converso do complemento

Para demonstrar, suponha que (m, n) é um elemento qualquer de $\overline{A^o}$:

$$(m, n) \in \overline{A^o}$$

A fórmula acima é equivalente a:

$$(m, n) \notin A^o$$

Que também é equivalente a:

$$(n, m) \notin A$$

Que equivale a:

$$(n, m) \in \overline{A}$$

Que também equivale a:

$$(m, n) \in (\overline{A})^o$$

Assim:

$$(m, n) \in \overline{A^o} \Leftrightarrow (m, n) \in (\overline{A})^o$$

Portanto:

$$\overline{A^o} = (\overline{A})^o$$

Distribuição do converso em relação a composição:

$$\overline{B^o.A^o = (A.B)^o}\ .^o$$

```
compose_converse : LEMMA
    converse(M) o converse(K) = converse(K o M)
```

Figura 3.26: Sobre composição e converso

Para demonstrar suponha (m, n) elemento qualquer tal que:

$$mB^o.A^on$$

Pela definição de composição, existirá um z na qual a fórmula acima será equivalente a:

$$mB^oz \wedge zA^on$$

Que também será equivalente a:

$$nAz \wedge zBm$$

Reintroduzindo a noção de composição, será equivalente a:

$$nA.Bm$$

E também será equivalente a:

$$m(A.B)^o n$$

Assim:

$$mB^o.A^o n \Leftrightarrow m(A.B)^o n$$

Portanto:

$$B^o.A^o = (A.B)^o$$

Distribuição da composição em relação a união:

$$\frac{}{A.C \cup B.C = (A \cup B).C} \ .\cup$$

$$\frac{}{C.A \cup C.B = C.(A \cup B)} \ .\cup$$

```
union_compose1 : LEMMA
    union(K,L) o M = union(K o M, L o M)

union_compose2 : LEMMA
    K o union(M,N) = union(K o M, K o N)
```

Figura 3.27: União e composição

Para demonstrar, suponha que (m, n) é um elemento qualquer:

$$m(A \cup B).Cn$$

Pela definição de composição, há um z tal que:

$$mA \cup Bz \wedge zCn$$

Será equivalente a:

$$(mAz \vee mBz) \wedge zCn$$

Usando a propriedade distributiva:

$$(mAz \wedge zCn) \vee (mBz \wedge zCn)$$

Reaplicando a definição de composição:

$$mA.Cn \vee mB.Cn$$

Será equivalente a:

$$mA.C \cup B.Cn$$

Assim:

$$m(A \cup B).Cn \Leftrightarrow mA.C \cup B.Cn$$

Portanto:

$$(A \cup B).C = A.C \cup B.C$$

Para demonstrar $C.(A \cup B) = C.A \cup C.B$ é análogo. Suponha que (n, m) é um elemento qualquer:

$$nC.(A \cup B)m$$

Pela definição de composição, há um z tal que:

$$nCz \wedge zA \cup Bm$$

Será equivalente a:

$$nCz \wedge (zAm \vee zBm)$$

Usando a propriedade distributiva:

$$(nCz \wedge zAm) \vee (nCz \wedge zBm)$$

Reaplicando a definição de composição:

$$mC.An \vee mC.Bn$$

Será equivalente a:

$$mC.A \cup C.Bn$$

Assim:

$$mC.(A \cup B)n \Leftrightarrow mC.A \cup C.Bn$$

Portanto:

$$C.(A \cup B) = C.A \cup C.B$$

Nas correflexivas, intersecção e composição são iguais:

$$\frac{A \subseteq id \quad B \subseteq id}{A \cap B = A.B}\ id.\cap$$

```
C : VAR correflexive
D : VAR correflexive
cor_compose_intersection : LEMMA
    intersection(C,D) = C o D
```

Figura 3.28: Sobre composição e intersecção de correflexivas

Para demonstração, sejam A e B relações correflexivas. E seja (m, n) um elemento qualquer de $A \cap B$:

$$m(A \cap B)n$$

A proposição acima é equivalente a:

$$mAn \wedge mBn$$

Como A é correflexiva, $m = n$. Assim, a fórmula acima será equivalente a:

$$mAn \wedge nBn$$

Pela definição de composição:

$$mA.Bn$$

Assim:

$$mA \cap Bn \Leftrightarrow mA.Bn$$

Portanto:

$$A \cap B = A.B$$

Correflexivas são simétricas:

$$\frac{A \subseteq id}{A^o = A}\ id$$

```
C : VAR correflexive
D : VAR correflexive
cor_symmetry : LEMMA
    C = converse(C)
```

Figura 3.29: Correflexivas são simétricas

Para demonstrar, seja A uma correflexiva e (m, n) um elemento qualquer:

$$mAn$$

Será equivalente a

$$nA^om$$

Como A é correflexiva, $m = n$, portanto a proposição acima é equivalente a:

$$mA^on$$

Assim:

$$mAn \Leftrightarrow mA^on$$

Portanto:

$$A = A^o$$

Pré-condição como correflexiva:

$$\frac{A \subseteq id}{B.A = B \cap \top.A} \cap\top.id$$

```
C : VAR correflexive
D : VAR correflexive

cor_pre : LEMMA
    P o C = intersection(P,Top o C)
```

Figura 3.30: Pré-condição como correflexiva

Para demonstrar, partindo do princípio que A é correflexiva, suponha (m, n) elemento qualquer tal que:

$$mB.An$$

Expandindo a definição de composição, há um z tal que a fórmula acima é equivalente a:

$$mBz \wedge zAn$$

Por A ser correflexiva, $n = z$ e a equação acima será equivalente a:

$$mBn \wedge zAn$$

Sem prejudicar a semântica da expressão acima é possível acrescentar a conjunção $\wedge m\top z$. Logo, ela será equivalente a:

$$mBn \wedge m\top z \wedge zAn$$

Pela definição de composição:

$$mBn \wedge m\top.An$$

Que é equivalente a:

$$mB \cap \top.An$$

Logo, vale a seguinte equivalência:

$$mB.An \Leftrightarrow mB \cap \top.An$$

E, portanto:

$$B.A = B \cap \top.A$$

Pós-condição como correflexiva:

$$\frac{A \subseteq id}{A.B = A.\top \cap B}$$

```
C : VAR correflexive
D : VAR correflexive

cor_pos : LEMMA
    C o K = intersection(C o Top,K)
```

Figura 3.31: Pós-condição como correflexiva

Para demonstrar, considerando A uma relação correflexiva, suponha que (m, n) é um elemento qualquer tal que:

$$mA.Bn$$

Pela definição de composição, há um z tal que a fórmula acima é equivalente a:

$$mAz \wedge zBn$$

Como A é uma relação correflexiva, $z = m$, e portanto a fórmula será equivalente a:

$$mAz \wedge mBn$$

Sem perda semântica, pode-se acrescentar à fórmula acima a conjunção $\wedge z \top n$:

$$mAz \wedge z\top n \wedge mBn$$

Pela definição de composição, é equivalente a:

$$mA.\top n \wedge mBn$$

E é equivalente a:

$$mA.\top \cap Bn$$

Assim vale a equivalência:

$$mA.Bn \Leftrightarrow mA.T \cap Bn$$

E, portanto:

$$A.B = A.T \cap B$$

O transformador unário-binário é uma relação correflexiva:

$$\frac{}{\Phi_p \subseteq id} \subseteq id$$

A regra acima foi demonstrada em PVS como uma condição de tipo, como o tipo de retorno na função que transforma predicados unários em correflexivas.

Para demonstrar, suponha (m, n) um elemento qualquer tal que:

```
to_correflexive(M : setof[A]) : correflexive =
    LAMBDA(a,a1): a = a1 and M(a1)
```

Figura 3.32: Transformador é correflexiva

$$m\Phi_p n$$

Por definição do transformador:

$$m\ id\ n \wedge p\ n$$

O que implica em:

$$m\ id\ n$$

Assim, vale a implicação:

$$m\Phi_p n \Rightarrow m\ id\ n$$

Portanto:

$$\Phi_p \subseteq id$$

Conjunção de predicados e composição:

$$\frac{}{\Phi_{p \wedge q} = \Phi_p . \Phi_q}\ \wedge.$$

```
cor_and : LEMMA
    to_correflexive(LAMBDA(a): and(W(a), Z(a))) =
       to_correflexive(W) o to_correflexive(Z)
```

Figura 3.33: Conjunção de predicados e composição

Para demonstrar, suponha que (m, n) é um elemento qualquer tal que:

$$m\Phi_{p \wedge q} n$$

Por definição é equivalente a:

$$m\ id\ n \wedge p\ n \wedge q\ n$$

Que é equivalente a:

$$m\ id\ n \wedge p\ n \wedge m\ id\ n \wedge q\ n$$

Por definição do transformador:

$$m\Phi_p n \wedge m\Phi_q n$$

A fórmula acima é equivalente a:

$$m\Phi_p \cap \Phi_q n$$

Como em correflexivas intersecção e composição são equivalentes:

$$m\Phi_p.\Phi_q n$$

Assim, vale a equivalência:

$$m\Phi_{p\wedge q} n \Leftrightarrow m\Phi_p.\Phi_q n$$

Portanto:

$$\Phi_{p\wedge q} = \Phi_p.\Phi_q$$

Os transformadores, disjunção e união:

$$\frac{}{\Phi_{p\vee q} = \Phi_p \cup \Phi_q} \vee\cup$$

```
cor_or : LEMMA
    to_correflexive(LAMBDA(a): or(W(a), Z(a))) =
       union(to_correflexive(W),to_correflexive(Z))
```

Figura 3.34: Disjunção e união

Para a demonstração, suponha (m, n) um elemento qualquer tal que:

$$m\Phi_{p\vee q} n$$

Por definição do transformador:

$$m\ id\ n \wedge (p\ n \vee q\ n)$$

Pela propriedade distributiva da conjunção em relação a disjunção, será igual a:

$$(m\ id\ n \wedge p\ n) \vee (m\ id\ n \wedge q\ n)$$

Pela definição do transformador, será equivalente a:

$$m\Phi_p n \vee m\Phi_q n$$

Que é equivalente a:

$$m\Phi_p \cup \Phi_q n$$

Logo:

$$m\Phi_{p\vee q} n \Leftrightarrow m\Phi_p \cup \Phi_q n$$

Portanto:

$$\Phi_{p\vee q} = \Phi_p \cup \Phi_q$$

Transformador, negação e complemento:

$$\overline{\Phi_{\neg p} = id \cap \overline{\Phi_p}}^{\neg}$$

Para demonstrar, suponha (m, n) um elemento qualquer tal que:

$$m \Phi_{\neg p} n$$

Pela definição do transformador, será equivalente a:

$$m\ id\ n \wedge \neg p\ n$$

A fórmula acima é equivalente a:

$$m\ id\ n \wedge (\neg p\ n \vee \neg m\ id\ n)$$

Pela lei de De Morgan, será equivalente a:

$$m\ id\ n \wedge \neg(p\ n \wedge m\ id\ n)$$

Pela definição de transformador, é equivalente a:

$$m\ id\ n \wedge \neg m \Phi_p n$$

É equivalente a:

$$m\ id\ n \wedge m \overline{\Phi_p} n$$

E também é equivalente a:

$$m\ id \cap \overline{\Phi_p} n$$

Assim, vale a equivalência:

$$m \Phi_{\neg p} n \Leftrightarrow m\ id \cap \overline{\Phi_p} n$$

Portanto:

$$\Phi_{\neg p} = id \cap \overline{\Phi_p}$$

Transformador, verdade e identidade:

$$\overline{\Phi_{True} = id}\ True\ id$$

```
cor_true : LEMMA
    to_correflexive(LAMBDA(a): TRUE) = (=)
```

Figura 3.35: Verdade e identidade

Para demonstrar, suponha (m, n) elemento qualquer tal que:

$$m \Phi_{True} n$$

Pela definição do transformador, é equivalente a:

$$m\ id\ n \wedge True$$

Que é equivalente a:

$$m\ id\ n$$

Assim, vale a equivalência:

$$m \Phi_{True} n \Leftrightarrow m\ id\ n$$

E, portanto:

$$\Phi_{True} = id$$

Transformador, falsidade e fundo:

$$\frac{}{\Phi_{False} = \bot}\ False\bot$$

```
cor_false : LEMMA
    to_correflexive(LAMBDA(a): FALSE) = Botton
```

Figura 3.36: Falsidade e fundo

Para demonstrar, suponha (m, n) um elemento qualquer tal que:

$$m \Phi_{False} n$$

Pela definição do transformador, é equivalente a:

$$m\ id\ n \wedge False$$

A equação acima é equivalente a $False$, e logo, é equivalente a:

$$m \bot n$$

Assim, vale a equivalência:

$$m \Phi_{False} n \Leftrightarrow m \bot n$$

Portanto:

$$\Phi_{False} = \bot$$

Por último, vale algumas observações. A composição de relações é associativa. Essa propriedade já foi demonstrada na teoria do PVS do arquivo *relations_props2*. Essa propriedade é sub-assumida nas demonstrações de forma que não se usa parênteses para quando há uma sequência de composições.

Algumas regras acima deduzidas são derivadas de outras conforme será demonstrado na parte na seção de regras derivadas do próximo capítulo.

3.3 Resumo

As regras do Cálculo de Sistema de Informação estão resumidas a seguir:

$$\frac{A \subseteq B \quad B \subseteq A}{A = B}\, = i \qquad \frac{\phi \quad A = B}{\phi[A/B]}\, = \qquad \frac{}{A \subseteq \top}\, \subseteq \top$$

$$\frac{A \cup B \subseteq C}{A \subseteq C}\, \cup e \qquad \frac{A \cup B \subseteq C}{B \subseteq C}\, \cup e \qquad \frac{A \subseteq B \quad A \subseteq C}{A \subseteq B \cap C}\, \cap i$$

$$\frac{}{\bot \subseteq A}\, \bot \subseteq \qquad \frac{A \subseteq B}{\overline{B} \subseteq \overline{A}}\, \overline{()} \subseteq \overline{()} \qquad \frac{}{\overline{\overline{A}} = A}\, \overline{\overline{()}}$$

$$\frac{A \subseteq B \quad B \subseteq C}{A \subseteq C}\, \subseteq \qquad \frac{}{(C \cup A) \cap (C \cup B) = C \cup (A \cap B)}\, \cup\cap \qquad \frac{A = B}{A \subseteq B} = e$$

$$\frac{A = B}{B \subseteq A} = e \qquad \frac{A \subseteq B \cap C}{A \subseteq B}\, \cap e \qquad \frac{}{A \cap \overline{A} = \bot}\, = \bot$$

Tabela 3.1: Regras intuitivas sobre conjuntos

$$\frac{\phi \quad \psi}{\phi \wedge \psi}\, \wedge i \qquad \frac{\phi \wedge \psi}{\phi}\, \wedge e \qquad \frac{\phi \wedge \psi}{\psi}\, \wedge e$$

$$\frac{\phi}{\phi \vee \psi}\, \vee i \qquad \frac{\psi}{\phi \vee \psi}\, \vee i \qquad \frac{\phi \vee \psi \quad \begin{array}{c}[\phi]^{A} \\ \vdots \\ \chi\end{array} \quad \begin{array}{c}[\psi]^{B} \\ \vdots \\ \chi\end{array}}{\chi}\, \vee e, A, B$$

$$\frac{\phi}{\neg\neg\phi}\, \neg\neg\, i$$

$$\frac{\phi \quad \phi \Rightarrow \psi}{\psi}\, \Rightarrow e \qquad \frac{\begin{array}{c}[\phi]^{S} \\ \vdots \\ \psi\end{array}}{\phi \Rightarrow \psi}\, \Rightarrow i, S$$

$$\frac{\phi \quad \neg\phi}{false}\, \neg e \qquad \frac{\begin{array}{c}[\phi]^{S} \\ \vdots \\ false\end{array}}{\neg\phi}\, \neg i, S \qquad \frac{false}{\phi}\, false\ e$$

Tabela 3.2: Regras da lógica proposicional

$$\frac{\begin{array}{c}[N \subseteq A] \\ \vdots \\ N \subseteq B\end{array}}{A \subseteq B} \subseteq i, N \qquad \frac{\begin{array}{c}[A \subseteq N] \\ \vdots \\ B \subseteq N\end{array}}{B \subseteq A} \subseteq i, N \qquad \frac{}{A^{oo} = A}\ oo$$

$$\frac{A \subseteq B}{A^o \subseteq B^o}\ o\subseteq \qquad \frac{A \subseteq B \quad C \subseteq D}{A.C \subseteq B.D}\ .\subseteq \qquad \frac{}{A.id = A}\ .id$$

$$\frac{}{A.\bot = \bot}\ .\bot \qquad \frac{}{\overline{A^o} = (\overline{A})^o}\ \overline{()}^o \qquad \frac{}{B^o.A^o = (A.B)^o}\ .^o$$

$$\frac{}{A.C \cup B.C = (A \cup B).C}\ .\cup \qquad \frac{A \subseteq id \quad B \subseteq id}{A \cap B = A.B}\ id.\cap \qquad \frac{A \subseteq id}{A^o = A}\ id$$

$$\frac{A \subseteq id}{B.A = B \cap \top.A}\ \cap\top.id \qquad \frac{}{\Phi_p \subseteq id} \subseteq id \qquad \frac{}{\Phi_{p \wedge q} = \Phi_p.\Phi_q}\ \wedge.$$

$$\frac{}{\Phi_{\neg p} = id \cap \overline{\Phi_p}}\ \neg \qquad \frac{}{\Phi_{True} = id}\ True\ id \qquad \frac{}{\Phi_{False} = \bot}\ False\bot$$

$$\frac{}{\Phi_{p \vee q} = \Phi_p \cup \Phi_q}\ \vee.$$

Tabela 3.3: Regras extras

$$\frac{A \subseteq \bot}{A = \bot} = \bot \qquad \frac{\top \subseteq A}{A = \top} = \top \qquad \frac{}{A = A}\ id \subseteq (=)$$

$$\frac{}{(C \cap A) \cup (C \cap B) = C \cap (A \cup B)}\ \cap\cup \qquad \frac{}{A \cup \overline{A} = \top} = \top \qquad \frac{}{id.A = A}\ .id$$

$$\frac{}{\bot.A = \bot}\ .\bot \qquad \frac{}{C.A \cup C.B = C.(A \cup B)}\ .\cup \qquad \frac{A \subseteq id}{A.B = A.\top \cap B}\ \cap id.\top$$

$$\frac{}{A = A}\ id \subseteq (=)$$

Tabela 3.4: Regras derivadas

Capítulo 4

Comparando técnicas de demonstração de propriedades fundamentais pelo assistente de prova PVS

A partir daqui, demonstrar-se-ão propriedades fundamentais do cálculo *pointfree* de maneira comparada. Cada propriedade será demonstrada duas vezes, uma com o uso do cálculo e outra sem. Far-se-á, também, o reuso de propriedades já demonstradas.

4.1 Lei de Shunting

Abaixo, a Lei de Shunting está definida pela notação *pointfree* por duas sequentes, onde $f\ function$ denota que f é função:

$$f\ function \vdash (f.R \subseteq S \Leftrightarrow R \subseteq f^o.S) \tag{4.1}$$

$$f\ function \vdash (R \subseteq S.f \Leftrightarrow R.f^o \subseteq S) \tag{4.2}$$

Expandindo a definição de função, de *kernel*, de imagem e de $\Leftrightarrow$ na equação 4.1:

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash (f.R \subseteq S \Rightarrow R \subseteq f^o.S) \wedge (R \subseteq f^o.S \Rightarrow f.R \subseteq S)$$

A sequente acima é equivalente a demonstrar as duas sequentes abaixo:

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash f.R \subseteq S \Rightarrow R \subseteq f^o.S \tag{4.3}$$

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash R \subseteq f^o.S \Rightarrow f.R \subseteq S \tag{4.4}$$

A prova da sequente 4.3 usando o cálculo *pointfree*

$$\dfrac{id \subseteq f^o.f \quad \dfrac{\dfrac{}{R = R}\, id \subseteq (=)}{R \subseteq R}\, = e}{id.R \subseteq f^o.f.R}\, . \subseteq \quad \dfrac{\dfrac{\dfrac{}{f^o = f^o}\, id \subseteq (=)}{f^o \subseteq f^o}\, = e \qquad \dfrac{[f.R \subseteq S]^{\not m}}{f^o.f.R \subseteq f^o.S}\, . \subseteq}{id.R \subseteq f^o.S}\, \subseteq \quad \dfrac{}{id.R = R}\, .id =}{\dfrac{R \subseteq f^o.S}{f.R \subseteq S \Rightarrow R \subseteq f^o.S} \Rightarrow i, m}$$

E a prova da sequente 4.4:

$$\dfrac{\dfrac{\dfrac{}{f = f}\, id \subseteq (=)}{f \subseteq f}\, = e \qquad \dfrac{[R \subset f^o.S]^{\not n}}{f.R \subseteq f.f^o.S}\, . \subseteq \quad \dfrac{\dfrac{\dfrac{}{S = S}\, id \subseteq (=)}{S \subseteq S}\, = e \qquad f.f^o \subseteq id}{f.f^o.S \subseteq id.S}\, . \subseteq}{f.f^o.S \subseteq S}\, \subseteq \quad \dfrac{}{id.S = S}\, .id =}{\dfrac{f.R \subseteq S}{R \subseteq f^o.S \Rightarrow f.R \subseteq S} \Rightarrow i, n}$$

A lei de Shunting foi provada em PVS. Na versão *pointfree*, a lei de Shunting expressa na sequente 4.1 fica assim:

```
shunting_law1: LEMMA
(subset?((=),converse(f) o f) and
subset?(f o converse(f),(=))) IMPLIES
(subset?(f o R, S) <=> subset?(R,converse(f) o S))
```

Figura 4.1: Uma das leis de Shunting em *pointfree*

Usando quantificadores, a lei de Shunting fica assim:

```
shunting_law1: LEMMA
(FORALL(a,c):((EXISTS(b) : a = f(b) and R(b,c))=> S(a,c)))IFF
FORALL(b1,c1):(R(b1,c1) => S(f(b1),c1))
```

Figura 4.2: Uma das leis de Shunting usando quantificadores

A demonstração referente a especificação da figura 4.1 teve 43 linhas de prova. Já a demonstração da figura 4.2 foi feita com 1 linha de prova. Entretanto, especificar o que foi proposto na figura 4.2 foi mais complexo que na figura 4.1, pois há mais quantificadores o que possibilita o erro.

Também se expande as definições de função, *kernel*, $\Leftrightarrow$ e de imagem na sequente 4.4:

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash (R \subseteq S.f \Rightarrow R.f^o \subseteq S) \wedge (R.f^o \subseteq S \Rightarrow R \subseteq S.f)$$

A sequente acima pode ser divida em duas subsequentes:

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash R \subseteq S.f \Rightarrow R.f^o \subseteq S \tag{4.5}$$

$$(id \subseteq f^o.f), (f.f^o \subseteq id) \vdash R.f^o \subseteq S \Rightarrow R \subseteq S.f \tag{4.6}$$

Eis a prova de 4.5:

$$\dfrac{\dfrac{[R \subseteq S.f]^{\not m} \quad \dfrac{\dfrac{}{f^o = f^o}\, id \subseteq (=)}{f^o \subseteq f^o}\, = e}{R.f^o \subseteq S.f.f^o}\, \cdot \subseteq \quad \dfrac{\dfrac{}{S.id = S}\, .id \quad \dfrac{\dfrac{\dfrac{}{S \subseteq S}\, id \subseteq (=)}{S \subseteq S}\, = e \quad f.f^o \subseteq id}{S.f.f^o \subseteq S.id}\, \cdot \subseteq}{S.f.f^o \subseteq S}\, =}{\dfrac{R.f^o \subseteq S}{R \subseteq S.f \Rightarrow R.f^o \subseteq S}\, \Rightarrow i, m}\, \subseteq$$

Eis a prova de 4.6:

$$\dfrac{\dfrac{\dfrac{}{R.id = R}\, .id \quad \dfrac{\dfrac{\dfrac{}{R = R}\, id \subseteq (=)}{R \subseteq R}\, = e \quad id \subseteq f^o.f}{R.id \subseteq R.f^o.f}\, \cdot \subseteq}{R \subseteq R.f^o.f}\, = \quad \dfrac{[R.f^o \subseteq S]^{\not m} \quad \dfrac{\dfrac{}{f = f}\, id \subseteq (=)}{f \subseteq f}\, = e}{R.f^o.f \subseteq S.f}\, \cdot \subseteq}{\dfrac{R \subseteq S.f}{R.f^o \subseteq S \Rightarrow R \subseteq S.f}\, \Rightarrow i, m}\, \subseteq$$

A sequente 4.2 foi especificada e provada em PVS conforme as figuras abaixo.

```
shunting_law2: LEMMA
(subset?((=),converse(f) o f) and
subset?(f o converse(f), (=))) IMPLIES
(subset?(R1, S1 o f) <=> subset?(R1 o converse(f), S1))
```

Figura 4.3: Outra lei de Shunting em versão *pointfree*

```
shunting_law2: LEMMA
(FORALL(c,b): R1(c,b) => S1(c,f(b)))IFF
(FORALL(c1,a1): (EXISTS(b1) : a1 = f(b1) and R1(c1,b1)) => S1(c1,a1))
```

Figura 4.4: Outra lei de Shunting usando quantificadores

No momento de provar as leis acima, ocorreu o mesmo fato: a versão *pointfree* teve mais comandos de prova (38 linhas) que a versão com quantificadores (1 linha). Entretanto, especificar a versão *pointfree* foi menos complexo, pois há menos quantificadores e isso permite menos nos objetos a serem associados nas relações.

A partir da lei de Shunting podem-se derivar quatro regras:

$$\dfrac{id \subseteq f^o.f \quad f.f^o \subseteq id \quad f.R \subseteq S}{R \subseteq f^o.S}\, SL$$

$$\dfrac{id \subseteq f^o.f \quad f.f^o \subseteq id \quad R \subseteq f^o.S}{f.R \subseteq S}\, SL$$

$$\dfrac{id \subseteq f^o.f \quad f.f^o \subseteq id \quad R \subseteq S.f}{R.f^o \subseteq S}\, SL$$

$$\dfrac{id \subseteq f^o.f \quad f.f^o \subseteq id \quad R.f^o \subseteq S}{R \subseteq S.f}\, SL$$

4.2 Inclusão e igualdade de funções

Quando uma função está contida em outra, elas são iguais:

$$g\ function, f\ function \vdash g \subseteq f \Rightarrow g = f$$

Expandindo a definição de função, *kernel* e imagem:

$$g.g^o \subseteq id,\ id \subseteq g^o.g,\ f.f^o \subseteq id,\ id \subseteq f^o.f \vdash g \subseteq f \Rightarrow g = f$$

Para demonstrar, usa-se a lei de Shunting (prova dividida em 3 pedaços):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{id.f = f}\ .id \quad \dfrac{}{[g \subseteq f]^{pr}}}{g \subseteq id.f} =}{f.f^o \subseteq id \quad id \subseteq f^o.f}\ SL}{\{g.f^o \subseteq id\}^1} \quad \dfrac{\overbrace{g.f^o \subseteq id}^{1,pr}}{(g.f^o)^o = g.f^o}\ id}{(g.f^o)^o \subseteq id} \quad \dfrac{\overbrace{g.f^o \subseteq id}^{1,pr}}{(g.f^o)^o = f^{oo}.g^o}\ .^o}{f^{oo}.g^o \subseteq id} = \quad \dfrac{}{f^{oo} = f}\ oo}{\{f.g^o \subseteq id\}^2} =$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{id.g = g}\ .id \quad \dfrac{g.g^o \subseteq id \quad id \subseteq g^o.g}{f \subseteq id.g} \quad \dfrac{\overbrace{f.g^o \subseteq id}^{2,pr}}{}\ SL}{f \subseteq g} = \quad [g \subseteq f]^{pr}}{g = f} = i}{g \subseteq f \Rightarrow g = f} \Rightarrow i, m$$

Nas figuras abaixo há especificações de propriedade de inclusão de funções:

```
function_inclusion: LEMMA
(subset?((=), converse(f) o f) and
subset?(f o converse(f), (=)) and
subset?((=), converse(g) o g) and
subset?(g o converse(g), (=))) IMPLIES
(subset?(g,f) => g = f)
```

Figura 4.5: Inclusão de funções especificado por *pointfree*

```
function_inclusion: LEMMA
(FORALL(a,b): a=g(b) => a=f(b)) => (FORALL(b1) : g(b1) = f(b1))
```

Figura 4.6: Inclusão de funções especificado por quantificadores

Na demonstração relativa à figura 4.5 foram produzidas 27 linhas de prova. E na demonstração relativa à 4.6, 1 linha. Nesse caso as complexidades de especificação foram as mesmas.

4.3 Relação simétrica e antissimétrica é equivalente a ser correflexiva

Sabe-se pelo capítulo anterior que se uma relação é correflexiva ela é simétrica pela regra abaixo.

$$\frac{A \subseteq id}{A^o = A}\ id$$

E ela também será simétrica, ou seja, $A \subseteq id \vdash A \cap A^o \subseteq id$ conforme pode ser visto abaixo.

$$\dfrac{\dfrac{\dfrac{\cancel{[N \subseteq A \cap A]}}{N \subseteq A}\cap e \quad A \subseteq id}{N \subseteq id}\subseteq \quad \dfrac{A \subseteq id}{A^o = A}\ id}{A \cap A \subseteq id}\subseteq i, N \qquad A \cap A^o \subseteq id}=$$

Também, como foi escrita na figura contendo a taxinomia de relações no capítulo 2, foi dito que se uma relação é simétrica e antissimétrica ela é correflexiva. Isso pode ser especificado conforme a sequente:

$$A \cap A^o \subseteq id,\ A^o = A \vdash A \subseteq id$$

Eis a prova relativa à sequente anterior:

$$\dfrac{\dfrac{\cancel{[N \subseteq A]} \quad \cancel{[N \subseteq A]}}{N \subseteq A \cap A}\cap i \quad \dfrac{A \cap A^o \subseteq id \quad A^o = A}{A \cap A \subseteq id}=}{\dfrac{N \subseteq id}{A \subseteq id}\subseteq i, N}\subseteq$$

A partir daí, foram especificadas as propriedades em PVS na versão *pointfree* e usando quantificadores:

```
correflexive_eq: LEMMA
(converse(C) = C and subset?(intersection(C,converse(C)),(=)))<=>
correflexive?[A,A,B](C)
```

Figura 4.7: Caracterização de correflexivas em *pointfree*

```
correflexive_eq: LEMMA
((FORALL(a,a1): C(a,a1) <=> C(a1,a)) and
(FORALL(a,a1): (C(a,a1) and C(a1,a)) => a = a1)) <=>
(FORALL(a,a1): C(a,a1) => a = a1)
```

Figura 4.8: Caracterização de correflexivas por quantificadores

A prova da especificação da figura 4.7 contém 14 linhas e a da figura 4.8, 7 linhas. Especificar a segunda demorou, pois foi preciso pensar em como colocar os quantificadores.

4.4 Fecho das relações correflexivas

Para as relações correflexivas, há a propriedade do fecho, em que valem as seguintes propriedades:

$$A \subseteq id \vdash R.A \subseteq S \Leftrightarrow R.A \subseteq S.A$$

$$A \subseteq id \vdash A.R \subseteq S \Leftrightarrow A.R \subseteq A.S$$

Nas sequentes acima, pode-se expandir as definições de $\Leftrightarrow$, gerando as seguintes sequentes com suas respectivas provas:

$$A \subseteq id \vdash R.A \subseteq S \Rightarrow R.A \subseteq S.A$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cancel{[M \subseteq A]}\quad \cancel{[M \subseteq A]}}{M \subseteq A \cap A}\cap i}{A \subseteq A \cap A}\subseteq i, M \quad \dfrac{\dfrac{\cancel{[N \subseteq A \cap A]}}{N \subseteq A}\cap e}{A \cap A \subseteq A}\subseteq i, N}{A = A \cap A} = i \quad \dfrac{A \subseteq id \quad A \subseteq id}{A \cap A = A.A}\, id.\cap}{\{A = A.A\}^1}\, =}{\dfrac{\dfrac{[R.A \subseteq A]^{h} \quad \dfrac{\dfrac{}{A = A}\, id \subseteq (=)}{A \subseteq A} = e}{R.A.A \subseteq S.A}\cdot \subseteq \quad \overbrace{A = A.A}^{1}}{R.A \subseteq S.A} =}{R.A \subseteq S \Rightarrow R.A \subseteq S.A} \Rightarrow i, h$$

$$A \subseteq id \vdash R.A \subseteq S.A \Rightarrow R.A \subseteq S$$

$$\dfrac{\dfrac{[R.A \subseteq S.A]^{h} \quad \dfrac{\dfrac{\dfrac{}{S = S}\, id \subseteq (=)}{S \subseteq S} = e \quad A \subseteq id}{S.A \subseteq S.id}\cdot \subseteq \quad \dfrac{}{S.id = S}\, .id}{R.A \subseteq S.A} \subseteq}{R.A \subseteq S.A \Rightarrow R.A \subseteq S} \Rightarrow i, h$$

$$A \subseteq id \vdash A.R \subseteq S \Rightarrow A.R \subseteq A.S$$

$$\dfrac{\dfrac{[A.R \subseteq S]^{h}}{A.A.R \subseteq A.S}\cdot \subseteq \quad \overbrace{A = A.A}^{1}}{A.R \subseteq A.S} =}{A.R \subseteq S \Rightarrow A.R \subseteq A.S} \Rightarrow i, h$$

$$A \subseteq id \vdash A.R \subseteq A.S \Rightarrow A.R \subseteq S$$

$$\dfrac{\dfrac{[A.R \subseteq A.S]^{h} \quad \dfrac{A \subseteq id \quad \dfrac{\overline{S = S}\ id \subseteq (=)}{S \subseteq S}\ = e}{A.S \subseteq id.S}\ . \subseteq}{A.R \subseteq id.S}\ \subseteq \qquad \dfrac{}{id.S = S}\ .id =}{\dfrac{A.R \subseteq S}{A.R \subseteq A.S \Rightarrow A.R \subseteq S}\ \Rightarrow i, h}$$

As propriedades acima foram especificadas da seguinte forma no padrão *pointfree*:

```
correflexive_close: LEMMA
correflexive?[A,B,C](C) =>
(subset?(C o R, S) <=> subset?(C o R, C o S))
```

Figura 4.9: Fecho de correflexivas em *pointfree*

```
correflexive_close2: LEMMA
correflexive?[A,B,C](C) =>
(subset?(R1 o C,S1) <=> subset?(R1 o C, S1 o C))
```

Figura 4.10: Fecho de correflexivas em *pointfree*

Abaixo, as mesmas propriedades por quantificadores:

```
correflexive_close: LEMMA
(FORALL(a,a1) : C(a,a1) => a = a1) =>
((FORALL(a,c) : (EXISTS(a1): C(a,a1) and R(a1,c)) => S(a,c)) <=>
(FORALL(a,c) : (EXISTS(a1): C(a,a1) and R(a1,c)) => (EXISTS(a1) : C(a,a1) and S(a1,c))))
```

Figura 4.11: Fecho de correflexivas por quantificadores

```
correflexive_close2: LEMMA
(FORALL(a,a1) : C(a,a1) => a = a1) =>
((FORALL(c,a) : (EXISTS(a1): R1(c,a1) and C(a1,a)) => S1(c,a)) <=>
(FORALL(c,a) : (EXISTS(a1): R1(c,a1) and C(a1,a)) => (EXISTS(a1) : S1(c,a1) and C(a1,a))))
```

Figura 4.12: Fecho de correflexivas por quantificadores

Nas figuras acima, as provas com a versão *pointfree* possui 26 e 23 linhas de prova, respectivamente.

Na versão com quantificadores, cada prova custou 15 linhas.

4.5 Correflexivas como condições laterais

Para relações correflexivas, valem as seguintes propriedades:

$$A \subseteq id, B \subseteq id \vdash A \subseteq B \Leftrightarrow A \subseteq \top.B$$

$$A \subseteq id, B \subseteq id \vdash A \subseteq B \Leftrightarrow A \subseteq B.\top$$

As sequentes acima podem ser divididas em quatro se expandir a definição de $\Leftrightarrow$.

$$A \subseteq id, B \subseteq id \vdash A \subseteq \top.B \Rightarrow A \subseteq B$$

$$\dfrac{\dfrac{\dfrac{A \subseteq id \quad \cancel{[A \subseteq \top.B]^{m}}}{A \subseteq id \cap \top.B}\cap i \quad \dfrac{B \subseteq id}{id \cap \top.B = id.B}\cap\top.id}{A \subseteq id.B}= \quad \dfrac{}{id.B = B}\,.id}{\dfrac{A \subseteq B}{A \subseteq \top.B \Rightarrow A \subseteq B}\Rightarrow i, m}=$$

$$A \subseteq id, B \subseteq id \vdash A \subseteq B \Rightarrow A \subseteq \top.B$$

$$\dfrac{\dfrac{\cancel{[A \subseteq B]^{m}} \quad \dfrac{}{id \subseteq \top}\subseteq\top}{id.A \subseteq \top.B}\cdot\subseteq \quad \dfrac{}{id.A = A}\,.id}{\dfrac{A \subseteq \top.B}{A \subseteq B \Rightarrow A \subseteq \top.B}\Rightarrow i, m}=$$

$$A \subseteq id, B \subseteq id \vdash A \subseteq B.\top \Rightarrow A \subseteq B$$

$$\dfrac{\dfrac{\dfrac{A \subseteq id \quad \cancel{[A \subseteq B.\top]^{m}}}{A \subseteq id \cap B.\top}\cap i \quad \dfrac{B \subseteq id}{id \cap B.\top = B.id}\cap\top.id}{A \subseteq B.id}= \quad \dfrac{}{B.id = B}\,.id}{\dfrac{A \subseteq B}{A \subseteq B.\top \Rightarrow A \subseteq B}\Rightarrow i, m}=$$

$$A \subseteq id, B \subseteq id \vdash A \subseteq B \Rightarrow A \subseteq B.\top$$

$$\dfrac{\dfrac{\cancel{[A \subseteq B]^{m}} \quad \dfrac{}{id \subseteq \top}\subseteq\top}{A.id \subseteq B.\top}\cdot\subseteq \quad \dfrac{}{A.id = A}\,.id}{\dfrac{A \subseteq B.\top}{A \subseteq B \Rightarrow A \subseteq B.\top}\Rightarrow i, m}=$$

As figuras abaixo mostram lemas relativos às correflexivas como condições laterais.

```
lateral_condition1: LEMMA
(correflexive?[A,B,C](C) and correflexive?[A,B,C](I)) =>
(subset?(C,I) <=> subset?(C,I o Top[A,B,C]))

lateral_condition2: LEMMA
(correflexive?[A,B,C](C) and correflexive?[A,B,C](I)) =>
(subset?(C,I) <=> subset?(C,Top[A,B,C] o I))
```

Figura 4.13: Corrreflexiva como condição lateral por *pointfree*

Os números de linhas de prova para cada especificação *pointfree* foram 23 e 24 respectivamente e para versão com quantificadores, 10 linhas cada.

4.6 Leis de De Morgan

As leis de De Morgan para conjuntos são:

```
lateral_condition1: LEMMA
(FORALL(a,a1) : C(a,a1) => a = a1) and (FORALL(a,a1) : I(a,a1) => a = a1)=>
((FORALL(a,a1): C(a,a1) => I(a,a1)) <=>
(FORALL(a,a1): C(a,a1) => EXISTS(e):I(a,e)))

lateral_condition2: LEMMA
(FORALL(a,a1) : C(a,a1) => a = a1) and (FORALL(a,a1) : I(a,a1) => a = a1)=>
((FORALL(a,a1): C(a,a1) => I(a,a1)) <=>
(FORALL(a,a1): C(a,a1) => EXISTS(e):I(e,a1)))
```

Figura 4.14: Correflexiva como condição lateral por quantificador

$$\vdash \overline{M} \cup \overline{N} = \overline{M \cap N}$$

$$\vdash \overline{M} \cap \overline{N} = \overline{M \cup N}$$

A primeira das sequentes acima pode ser dividida e provadas, respectivamente, em:

$$\vdash \overline{M} \cup \overline{N} \subseteq \overline{M \cap N}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{M} \cup \overline{N} \subseteq H}}{\overline{M} \subseteq H}\cup e \quad \dfrac{}{\overline{\overline{()}} = ()}}{\overline{H} \subseteq \overline{\overline{M}}} \quad \dfrac{}{\overline{\overline{M}} = M}\,=}{\overline{H} \subseteq M} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\overline{M} \cup \overline{N} \subseteq H}}{\overline{N} \subseteq H}\cup e \quad \dfrac{}{\overline{\overline{()}} = ()}}{\overline{H} \subseteq \overline{\overline{N}}} \quad \dfrac{}{\overline{\overline{N}} = N}\,=}{\overline{H} \subseteq N}}{\overline{H} \subseteq M \cap N} \cap i \quad \dfrac{}{\overline{()} \subseteq \overline{()}}}{\overline{M \cap N} \subseteq \overline{\overline{H}}} \quad \dfrac{}{\overline{\overline{H}} = H}\,=}{\overline{M \cap N} \subseteq H} \quad \dfrac{}{\overline{\overline{()}} = ()}}{\overline{M \cap N} \subseteq \overline{M} \cup \overline{N}} \subseteq i, H$$

$$\vdash \overline{M \cap N} \subseteq \overline{M} \cup \overline{N}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{M \cap N} \subseteq K}}{\overline{K} \subseteq \overline{\overline{M \cap N}}} \quad \dfrac{}{\overline{()} \subseteq \overline{()}} \quad \dfrac{}{\overline{\overline{M \cap N}} = M \cap N}\,=}{\{\overline{K} \subseteq M \cup N\}^{1}}}{\overline{K} \subseteq M \cup N}^{1,K}\cup e \quad \dfrac{}{\overline{\overline{()}} = ()}}{\overline{K} \subseteq \overline{\overline{M}}} \quad \dfrac{}{\overline{\overline{K}} = K}\,=}{\overline{M} \subseteq K} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{M} \cup \overline{N} \subseteq K}}{\overline{K} \subseteq M \cup N}}{\overline{K} \subseteq N}^{1,K}\cap e \quad \dfrac{}{\overline{()} \subseteq \overline{()}}}{\overline{N} \subseteq \overline{\overline{K}}} \quad \dfrac{}{\overline{\overline{K}} = K}\,=}{\overline{N} \subseteq K}}{\overline{M} \cup \overline{N} \subseteq K} \cup i \quad \dfrac{}{\overline{\overline{()}} = ()}}{\overline{M} \cup \overline{N} \subseteq \overline{M \cap N}} \subseteq i, K$$

Disso pode se derivar a seguinte regra:

$$\dfrac{}{\overline{M} \cup \overline{N} = \overline{M \cap N}}\, DM$$

A partir dessa regra derivada pode-se provar:

$$\vdash \overline{M} \cap \overline{N} = \overline{M \cup N}$$

Dividindo-se a sequente anterior em duas subsequentes:

$$\vdash \overline{M} \cap \overline{N} \subseteq \overline{M \cup N}$$

$$\vdash \overline{M \cup N} \subseteq \overline{M} \cap \overline{N}$$

Que podem ser provadas respectivamente por:

$$\dfrac{\dfrac{\dfrac{}{\overline{\overline{M} \cup \overline{N}} = \overline{\overline{M} \cap \overline{N}}}\,DM}{\overline{M} \cup \overline{N} \subseteq \overline{M} \cap \overline{N}} = e \qquad \dfrac{}{\overline{\overline{N}} = N} \qquad \dfrac{}{\overline{()} = ()}}{\overline{M} \cup N \subseteq \overline{M} \cap \overline{N}} = \qquad \dfrac{}{\overline{\overline{M}} = M} \qquad \dfrac{}{\overline{()} = ()}$$

$$\dfrac{\dfrac{M \cup N \subseteq \overline{\overline{M} \cap \overline{N}}}{\overline{\overline{M} \cap \overline{N}} \subseteq \overline{M \cup N}} \qquad \dfrac{}{\overline{()} \subseteq \overline{()}}}{\overline{M} \cap \overline{N} \subseteq \overline{M \cup N}} = \qquad \dfrac{}{\overline{\overline{M} \cap \overline{N}} = \overline{M} \cap \overline{N}} \qquad \dfrac{}{\overline{()} = ()}$$

$$\dfrac{\dfrac{\dfrac{}{\overline{\overline{M} \cup \overline{N}} = \overline{\overline{M} \cap \overline{N}}}\,DM}{\overline{M} \cap \overline{N} \subseteq \overline{M} \cup \overline{N}} = e \qquad \dfrac{}{\overline{\overline{N}} = N} \qquad \dfrac{}{\overline{()} = ()}}{\overline{M} \cap \overline{N} \subseteq \overline{M} \cup N} = \qquad \dfrac{}{\overline{\overline{M}} = M} \qquad \dfrac{}{\overline{()} = ()}$$

$$\dfrac{\dfrac{\overline{M} \cap \overline{N} \subseteq M \cup N}{M \cup N \subseteq \overline{\overline{M} \cap \overline{N}}} \qquad \dfrac{}{\overline{()} \subseteq \overline{()}}}{\overline{M \cup N} \subseteq \overline{M} \cap \overline{N}} = \qquad \dfrac{}{\overline{\overline{M} \cap \overline{N}} = \overline{M} \cap \overline{N}} \qquad \dfrac{}{\overline{()} = ()}$$

A lei de De Morgan foi especificada em versão *pointfree* e com quantificadores:

```
de_morgan1 : LEMMA
union(complement(R),complement(S)) = complement(intersection(R,S))

de_morgan2 : LEMMA
intersection(complement(R),complement(S)) = complement(union(R,S))
```

Figura 4.15: Lei de De Morgan por *pointfree*

```
de_morgan1 : LEMMA
NOT R(a,c) OR NOT S(a,c) <=> NOT (R(a,c) AND S(a,c))

de_morgan2 : LEMMA
NOT R(a,c) AND NOT S(a,c) <=> NOT (R(a,c) OR S(a,c))
```

Figura 4.16: Lei de De Morgan por quantificador

A demonstração por versão *pointfree* custou 50 e 8 linhas de prova respectivamente. Mas a versão por quantificadores custou uma linha de prova cada.

4.7 Especificação de Correção de Funções

Em verificação formal de *software*, correção de funções é feita através de contratos. Os contratos de funções são dado um tipo na entrada ela deve fornecer um tipo na saída. Seja f uma função, ϕ_I uma correflexiva sobre a entrada de f e ϕ_O sobre a saída. Dizer que f está correta em versão *pointfree* é

$$Ker(f.\phi_I) \subseteq \phi_O$$

Ela é equivalente a $f.\phi_I \subseteq \phi_O.f$.
Logo, pode-se provar a seguinte sequente:

$$f\ function \vdash Ker(f.\phi_I) \subseteq \phi_O \Leftrightarrow f.\phi_I \subseteq \phi_O.f$$

Ela pode ser dividida nas seguintes sequentes:

$$id \subseteq f^o.f, f.f^o \subseteq id \vdash (f.\phi_I).(f.\phi_I)^o \subseteq \phi_O \Rightarrow f.\phi_I \subseteq \phi_O.f$$

$$id \subseteq f^o.f, f.f^o \subseteq id \vdash f.\phi_I \subseteq \phi_O.f \Rightarrow (f.\phi_I).(f.\phi_I)^o \subseteq \phi_O$$

Para prová-las será usada uma regra derivada. Ela provem da seguinte sequente:

$$\vdash \phi_P.\phi_P = \phi_P$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\cancel{[G \subseteq \phi_P \cap \phi_P]}}{G \subseteq \phi_P} \cap e}{\phi_P \cap \phi_P \subseteq \phi_P} \subseteq i, G \quad \dfrac{\dfrac{\cancel{[S \subseteq \phi_P]} \quad \cancel{[S \subseteq \phi_P]}}{S \subseteq \phi_P \cap \phi_P} \cap i}{\phi_P \subseteq \phi_P \cap \phi_P} \subseteq i, S}{\phi_P \cap \phi_P = \phi_P} = i \quad \dfrac{\dfrac{\overline{\phi_P \subseteq id} \subseteq id \quad \overline{\phi_P \subseteq id} \subseteq id}{\phi_P \cap \phi_P = \phi_P.\phi_P} id.\cap}{\phi_P.\phi_P = \phi_P} =$$

Com essa tem-se a sequinte regra derivada:
$\dfrac{}{\phi_P.\phi_P = \phi_P}\ RD$

Com isso pode-se provar:

$$id \subseteq f^o.f, f.f^o \subseteq id \vdash (f.\phi_I).(f.\phi_I)^o \subseteq \phi_O \Rightarrow f.\phi_I \subseteq \phi_O.f$$

$$\dfrac{\dfrac{\dfrac{[f.\phi_I.(f.\phi_I)^o \subseteq \phi_O]^{H} \quad \overline{\phi_I^o.f^o = (f.\phi_I)^o} \cdot^o =}{f.\phi_I.\phi_I^o.f^o \subseteq \phi_O} \quad \dfrac{\overline{\phi_I \subseteq id} \subseteq id}{\phi_I^o = \phi_I} id =}{\{f.\phi_I.\phi_I.f^o \subseteq \phi_O\}^1}$$

$$\dfrac{\dfrac{\overbrace{f.\phi_I.\phi_I.f^o \subseteq \phi_O}^{1,\cancel{H}} \quad \overline{\phi_I.\phi_I = \phi_I} RD =}{f.\phi_I.f^o \subseteq \phi_O} \quad \dfrac{id \subseteq f^o.f \quad f.f^o \subseteq id}{f.\phi_I \subseteq \phi_O.f} SL}{(f.\phi_I).(f.\phi_I)^o \cap id \subseteq \phi_O \Rightarrow f.\phi_I \subseteq \phi_O.f} \Rightarrow i, H$$

Pode-se provar também:

$$id \subseteq f^o.f, f.f^o \subseteq id \vdash f.\phi_I \subseteq \phi_O.f \Rightarrow (f.\phi_I).(f.\phi_I)^o \subseteq \phi_O$$

$$\dfrac{\dfrac{[f.\phi_I \subseteq \phi_O.f]^{\not 1} \quad id \subseteq f^o.f \quad f.f^o \subseteq id}{f.\phi_I.f^o \subseteq \phi_O}\, SL \quad \dfrac{}{\phi_I.\phi_I = \phi_I}\, RD}{\{f.\phi_I.\phi_I.f^o \subseteq \phi_O\}^2} =$$

$$\dfrac{\dfrac{\dfrac{}{\phi_I \subseteq id}\subseteq id}{\phi_I^o = \phi_I}\, id \quad \overbrace{f.\phi_I.\phi_I.f^o \subseteq \phi_O}^{2,\not 1}}{f.\phi_I.\phi_I^o.f^o \subseteq \phi_O} = \quad \dfrac{}{(f.\phi_I)^o = \phi_I^o.f^o}\,{}^{\cdot^o} =$$

$$\dfrac{(f.\phi_I).(f.\phi_I)^o \subseteq \phi_O}{f.\phi_I \subseteq \phi_O.f \Rightarrow (f.\phi_I).(f.\phi_I)^o \subseteq \phi_O} \Rightarrow i, L$$

Para as sequentes logo acima, tem-se as seguintes especificações na versão *pointfree* e com quantificadores respectivamente em 4.17 e 4.18.

```
specs : LEMMA
(subset?((=),converse(f) o f) AND subset?(f o converse(f), (=))
AND correflexive?[A,A,A](C) AND correflexive?[B,B,B](J)) =>
(subset?(f o J, C o f) <=> subset?( f o J o converse(f o J), C))
```

Figura 4.17: Propriedade de contratos funcionais por *pointfree*

```
specs : LEMMA
((FORALL(a,a1): C(a,a1) => a = a1) AND (FORALL(b,b1): J(b,b1) => b = b1)) =>
((FORALL(a,b): (EXISTS(k): a = f(k) AND J(k,b)) => (EXISTS(e): C(a,e) AND e = f(b)))<=>
(FORALL(a,a1): (EXISTS(k,m,n): a = f(k) AND J(k,m) AND J(n,m) AND a1 = f(n)) => C(a,a1)))
```

Figura 4.18: Propriedade de contratos funcionais por quantificador

A demonstração pela versão *pointfree* pussui 19 linhas de prova e a versão com quantificadores, 16.

4.8 Distribuição do converso pela união e intersecção

O converso pode se distribuir pela união:

$$\vdash (A \cup B)^o = A^o \cup B^o$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\cancel{A \cap B \subseteq K}]}{A \subseteq K}\cap e}{A \subseteq A \cup B}\subseteq i, K}{A^o \subseteq (A \cup B)^o}\, o\subseteq \quad \dfrac{\dfrac{\dfrac{[\cancel{A \cup B \subseteq L}]}{B \subseteq L}\cap e}{B \subseteq A \cup B}\cup i, L}{B^o \subseteq (A \cup B)^o}\, o\subseteq}{\{A^o \cup B^o \subseteq (A \cup B)^o\}^2}\cup i$$

$$\dfrac{\dfrac{\dfrac{[\cancel{A^o \cup B^o \subseteq H}]}{A^o \subseteq H}\cup e}{A^{oo} \subseteq H^o}\, o\subseteq \quad \dfrac{}{A^{oo} = A}\, oo}{A \subseteq H^o} = \quad \dfrac{\dfrac{[\cancel{A^o \cup B^o \subseteq H}]}{B^o \subseteq H}\cup e}{B^{oo} \subseteq H^o}\, o\subseteq \quad \dfrac{}{B^{oo} = B}\, oo}{B \subseteq H^o} =$$

$$\dfrac{\dfrac{A \cup B \subseteq H^o}{(A \cup B)^{oo} \subseteq H^o}\cup i \quad \dfrac{}{(A \cup B)^{oo} = A \cup B}\, oo}{(A \cup B)^o \subseteq H}\, o\subseteq = \dfrac{}{\{(A \cup B)^o \subseteq A^o \cup B^o\}^1}\subseteq i, H$$

$$\frac{\overbrace{(A \cup B)^o \subseteq A^o \cup B^o}^{1} \quad \overbrace{A^o \cup B^o \subseteq (A \cup B)^o}^{2}}{(A \cup B)^o = A^o \cup B^o} = i$$

De maneira semelhante, o converso pode se distribuir pela intersecção:

$$\vdash (A \cap B)^o = A^o \cap B^o$$

$$\frac{\dfrac{\dfrac{\dfrac{[M \subseteq A \cap B]}{M \subseteq A} \cap e}{A \cap B \subseteq A} \subseteq i, M}{(A \cap B)^o \subseteq A^o} o \subseteq \quad \dfrac{\dfrac{\dfrac{[N \subseteq A \cap B]}{N \subseteq B} \cap e}{A \cap B \subseteq B} \subseteq i, N}{(A \cap B)^o \subseteq B^o} o \subseteq}{\{(A \cap B)^o \subseteq A^o \cap B^o\}^1} \cap i$$

$$\frac{\dfrac{\dfrac{\dfrac{[L \subseteq A^o \cap B^o]}{L \subseteq A^o} \cap e}{L^o \subseteq A^{oo}} o \subseteq \quad \dfrac{A^{oo} = A}{L^o \subseteq A} oo = \qquad \dfrac{\dfrac{\dfrac{[L \subseteq A^o \cap B^o]}{L \subseteq B^o} \cap e}{L^o \subseteq B^{oo}} o \subseteq \quad \dfrac{B^{oo} = B}{L^o \subseteq B} oo =}{L^o \subseteq A \cap B} \cap i}{L^{oo} \subseteq (A \cap B)^o} o \subseteq \quad \dfrac{L^{oo} = L}{L \subseteq (A \cap B)^o} oo =}{\{A^o \cap B^o \subseteq (A \cap B)^o\}^2} \subseteq i, L$$

$$\frac{\overbrace{(A \cap B)^o \subseteq A^o \cap B^o}^{1} \quad \overbrace{A^o \cap B^o \subseteq (A \cap B)^o}^{2}}{(A \cap B)^o = A^o \cap B^o} = i$$

```
converse_ditribution1 : LEMMA
converse(union(R,S)) = union(converse(R),converse(S))

converse_ditribution2 : LEMMA
converse(intersection(R,S)) = intersection(converse(R),converse(S))
```

Figura 4.19: Distribuição de conversos via *pointfree*

```
converse_distribution1 : LEMMA
((FORALL(a,c): R1(c,a) = R(a,c)) AND (FORALL(a,c): S1(c,a) = S(a,c)) AND
(FORALL (a,c): U(c,a) = (R(a,c) OR S(a,c)))) =>
(FORALL (a,c) : U(c,a) <=> (R1(c,a) OR S1(c,a)))

converse_distribution2 : LEMMA
((FORALL(a,c): R1(c,a) = R(a,c)) AND (FORALL(a,c): S1(c,a) = S(a,c)) AND
(FORALL (a,c): U(c,a) = (R(a,c) AND S(a,c)))) =>
(FORALL (a,c) : U(c,a) <=> (R1(c,a) AND S1(c,a)))
```

Figura 4.20: Distribuição de conversos via quantificador

Nas figuras 4.19 e 4.20 mostra a especificação da distribuição de converso. Via *pointfree* cada demonstração gastou 60 e 54 linhas respectivamente. Por quantificadores, precisou-se de uma linha de prova cada.

A partir da propriedade distributiva do converso em relação à união e a intersecção pode-se ter as seguintes regras derivadas:

$$\frac{}{(A \cap B)^o = A^o \cap B^o}\ \cap^o$$
$$\frac{}{(A \cup B)^o = A^o \cup B^o}\ \cup^o$$

4.9 Demonstrações das regras derivadas

Por fim, as demonstrações das regras derivadas.

$$\vdash A = A$$

Demonstra-se por:

$$\frac{\dfrac{\cancel{[G \subseteq A]}}{\{A \subseteq A\}^1}\ \subseteq i, G \quad \overbrace{A \subseteq A}^{1}}{A = A} = i$$

E gera como regra derivada:

$$\frac{}{A = A}\ id \subseteq (=)$$

$$A \subseteq \bot \vdash A = \bot$$

Demonstra-se por:

$$\frac{\dfrac{}{\bot \subseteq A}\ \bot \subseteq \quad A \subseteq \bot}{A = \bot} = i$$

E gera como regra derivada:

$$\frac{A \subseteq \bot}{A = \bot} = \bot$$

$$\top \subseteq A \vdash A = \top$$

Demonstra-se por:

$$\frac{\dfrac{}{A \subseteq \top}\ \subseteq \top \quad \top \subseteq A}{A = \top} = i$$

E gera como regra derivada:

$$\frac{\top \subseteq A}{A = \top} = \bot$$

$$\vdash (C \cap A) \cup (C \cap B) = C \cap (A \cup B)$$

Demonstra-se por:

$$\frac{\dfrac{}{(\overline{C} \cup \overline{A}) \cap (\overline{C} \cup \overline{B}) = \overline{C} \cup (\overline{A} \cap \overline{B})}\ \cup\cap \quad \dfrac{}{\overline{C \cap B = C} \cup \overline{B}}\ DM}{\{(\overline{C} \cup \overline{A}) \cap \overline{C \cap B = C} \cup (\overline{A} \cap \overline{B})\}^3} =$$

$$\frac{\dfrac{\overbrace{(\overline{C} \cup \overline{A}) \cap \overline{C \cap B = C} \cup (\overline{A} \cap \overline{B})}^{3} \quad \dfrac{}{\overline{C \cap A = C} \cup \overline{A}}\ DM}{\{\overline{C \cap A} \cap \overline{C \cap B = C} \cup (\overline{A} \cap \overline{B})\}^5} =$$

$$\frac{\dfrac{\overbrace{\overline{C \cap A} \cap \overline{C \cap B = C} \cup (\overline{A} \cap \overline{B})}^{5} \quad \dfrac{}{\overline{(C \cap A) \cup (C \cap B)} = \overline{C \cap A} \cap \overline{C \cap B}}\ DM}{\{\overline{(C \cap A) \cup (C \cap B)} = \overline{C} \cup (\overline{A} \cap \overline{B})\}^4} =$$

$$\dfrac{\dfrac{\overbrace{\overline{(C\cap A)\cup(C\cap B)}=\overline{C}\cup(\overline{A}\cap\overline{B})}^{4}\quad\overline{\overline{A\cup B}=\overline{A}\cap\overline{B}}\,DM}{\overline{(C\cap A)\cup(C\cap B)}=\overline{C}\cup\overline{A\cup B}}=\qquad\dfrac{}{\overline{C\cap(A\cup B)}=\overline{C}\cup\overline{A\cup B}}\,DM}{\dfrac{\dfrac{\dfrac{\{\overline{(C\cap A)\cup(C\cap B)}=\overline{C\cap(A\cup B)}\}^{1}}{\overline{C\cap(A\cup B)}\subseteq\overline{(C\cap A)\cup(C\cap B)}}=e}{\{(C\cap A)\cup(C\cap B)\subseteq C\cap(A\cup B)\}^{2}}\,\overline{()}\subseteq\overline{()}}=$$

$$\dfrac{\dfrac{\dfrac{\overbrace{\overline{(C\cap A)\cup(C\cap B)}=\overline{C\cap(A\cup B)}}^{1}}{\overline{(C\cap A)\cup(C\cap B)}\subseteq\overline{C\cap(A\cup B)}}}{C\cap(A\cup B)\subseteq(C\cap A)\cup(C\cap B)}\,\overline{()}\subseteq\overline{()}\quad\overbrace{(C\cap A)\cup(C\cap B)\subseteq C\cap(A\cup B)}^{2}}{(C\cap A)\cup(C\cap B)=C\cap(A\cup B)}=i$$

De onde deriva-se:

$$\dfrac{}{(C\cap A)\cup(C\cap B)=C\cap(A\cup B)}\cap\cup$$

$$\vdash A\cup\overline{A}=\top$$

Demonstra-se por:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{\overline{\overline{\overline{A}}}}\subseteq\overline{A\cup\overline{A}}}\,\overline{\overline{\overline{\overline{A}}}}\subseteq\overline{()}}{\overline{\overline{\overline{\overline{A}}}}\subseteq\overline{\top}}\qquad\dfrac{}{\overline{\overline{\overline{\overline{A}}}}=A\cup\overline{A}}\,\overline{\overline{()}}=}{\{A\cup\overline{A}\subseteq\overline{\top}\}^{1}}$$

$$\dfrac{\dfrac{\dfrac{}{\overline{A\cup\overline{A}}=\top}=\top}{\overline{A}\cup A\subseteq\top}=e\qquad\dfrac{}{\overline{\overline{A}}=A}\,\overline{\overline{()}}=}{\overline{A}\cup\overline{\overline{A}}\subseteq\top}=\qquad\dfrac{}{\overline{A\cup\overline{A}}=\overline{A}\cup\overline{\overline{A}}}\,DM=$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{A\cup\overline{A}}\subseteq\top}\,\overline{()}\subseteq\overline{()}}{\overline{\top}\subseteq\overline{\overline{A\cup\overline{A}}}}\qquad\dfrac{}{\overline{\overline{A\cup\overline{A}}}=A\cup\overline{A}}\,\overline{\overline{()}}=}{\{\overline{\top}\subseteq A\cup\overline{A}\}^{2}}$$

$$\dfrac{\overbrace{\overline{A\cup\overline{A}}\subseteq\top}^{1}\quad\overbrace{\overline{\top}\subseteq A\cup\overline{A}}^{2}}{A\cup\overline{A}=\overline{\top}}=i\qquad\dfrac{\dfrac{\dfrac{\dfrac{}{\overline{\top}\subseteq\overline{\top}}\,\overline{()}\subseteq\overline{()}}{\overline{\overline{\top}}\subseteq\overline{\top}}\qquad\dfrac{}{\overline{\overline{\top}}=\top}\,\overline{\overline{()}}=}{\top\subseteq\overline{\top}}}{\overline{\top}=\overline{\top}}=\top}{A\cup\overline{A}=\top}=$$

Portanto, tem-se a seguinte regra derivada:

$$\dfrac{}{A\cup\overline{A}=\top}=\top$$

$$\vdash id.A=A$$

Demonstra-se por:

$$\dfrac{\dfrac{\dfrac{}{A^o.id = A^o}\,.id \quad \dfrac{\dfrac{\dfrac{}{id = id}\,id \subseteq (=)}{id \subseteq id = e}}{id^o = id}\,id}{A^o.id^o = A^o} = \quad \dfrac{}{(id.A)^o = A^o.id^o}\,.o}{\dfrac{\{(id.A)^o = A^o\}^1}{(id.A)^o \subseteq A^o} = e}{\{(id.A)^{oo} \subseteq A^{oo}\}^2}\,o \subseteq$$

$$\dfrac{\dfrac{\overbrace{(id.A)^o = A^o}^{1}}{A^o \subseteq (id.A)^o} = e}{A^{oo} \subseteq (id.A)^{oo}}\,o \subseteq \quad \dfrac{}{(id.A)^{oo} = id.A}\,oo}{A^{oo} \subseteq id.A} = \quad \dfrac{}{A^{oo} = A}\,oo}{\{A \subseteq id.A\}^3} =$$

$$\dfrac{\dfrac{}{(id.A)^{oo} = id.A}\,oo \quad \dfrac{\dfrac{}{A^{oo} = A}\,oo \quad \overbrace{(id.A)^{oo} \subseteq A^{oo}}^{2}}{(id.A)^{oo} \subseteq A} =}{id.A \subseteq A} = \quad \overbrace{A \subseteq id.A}^{3}}{id.A = A} = i$$

Logo. tem-se a regra:

$$\dfrac{}{id.A = A}\,.id$$

$$\vdash \bot.A = \bot$$

Deriva-se:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A^o.\bot = \bot}\,.\bot}{A^o.\bot \subseteq \bot} = e}{(A^o.\bot)^o \subseteq \bot^o}\,o \subseteq \quad \dfrac{}{(A^o.\bot)^o = A^{oo}.\bot^o}\,.o}{\bot^o.A^{oo} \subseteq \bot^o} = \quad \dfrac{}{A^{oo} = A}\,oo}{\bot^o.A \subseteq \bot^o} = \quad \dfrac{\dfrac{\dfrac{}{\bot \subseteq \bot^o}\,\bot \subseteq}{\bot^o \subseteq \bot^{oo}}\,o \subseteq \quad \dfrac{}{\bot^{oo} = \bot}\,oo}{\bot^o \subseteq \bot} = \dfrac{}{\bot = \bot^o} = \bot}{\bot.A \subseteq \bot} = \dfrac{}{\bot.A = \bot} = \bot$$

Portanto, vale a seguite regra:

$$\dfrac{}{\bot.A = \bot}\,.\bot$$

$$\vdash C.A \cup C.B = C.(A \cup B)$$

Demonstra-se por:

$$\dfrac{\dfrac{}{((A^o \cup B^o).C^o)^o = ((A^o \cup B^o).C^o)^o}\,id \subseteq (=) \quad \dfrac{}{A^o.C^o \cup B^o.C^o = (A^o \cup B^o).C^o}\,.\cup}{\{(A^o.C^o \cup B^o.C^o)^o = ((A^o \cup B^o).C^o)^o\}^2} =$$

$$\dfrac{\overbrace{(A^o.C^o \cup B^o.C^o)^o = ((A^o \cup B^o).C^o)^o}^{2} \quad \dfrac{}{(C.B)^o = B^o.C^o}\,.o}{(A^o.C^o \cup (C.B)^o)^o = ((A^o \cup B^o).C^o)^o} = \quad \dfrac{}{(C.A)^o = A^o.C^o}\,.o}{\{((C.A)^o \cup (C.B)^o)^o = ((A^o \cup B^o).C^o)^o\}^3} =$$

$$\dfrac{\overbrace{((C.A)^o \cup (C.B)^o)^o = ((A^o \cup B^o).C^o)^o}^{3} \quad \overline{(C.A \cup C.B)^o = (C.A)^o \cup (C.B)^o}\ \cup^o =}{\{(C.A \cup C.B)^{oo} = ((A^o \cup B^o).C^o)^o\}^4}$$

$$\dfrac{\overbrace{(C.A \cup C.B)^{oo} = ((A^o \cup B^o).C^o)^o}^{4} \quad \overline{(A \cup B)^o = A^o \cup B^o}\ \cup^o =}{\{(C.A \cup C.B)^{oo} = ((A \cup B)^o.C^o)^o\}^1}$$

$$\dfrac{\overbrace{(C.A \cup C.B)^{oo} = ((A \cup B)^o.C^o)^o}^{1} \quad \overline{(C.(A \cup B))^o = (A \cup B)^o.C^o}\ .^o =}{\{(C.A \cup C.B)^{oo} = (C.(A \cup B))^{oo}\}^6}$$

$$\dfrac{\overbrace{(C.A \cup C.B)^{oo} = (C.(A \cup B))^{oo}}^{6} \quad \overline{(C.(A \cup B))^{oo} = C.(A \cup B)}\ ^{oo} =}{\{(C.A \cup C.B)^{oo} = C.(A \cup B)\}^5}$$

$$\dfrac{\overbrace{(C.A \cup C.B)^{oo} = C.(A \cup B)}^{5} \quad \overline{(C.A \cup C.B)^{oo} = C.A \cup C.B}\ ^{oo} =}{C.A \cup C.B = C.(A \cup B)}$$

Logo, tem-se a regra:

$$\dfrac{}{C.A \cup C.B = C.(A \cup B)}\ .\cup$$

$$A \subseteq id \vdash A.B = A.\top \cap B$$

Demonstração:

$$\dfrac{\dfrac{\dfrac{}{\top^o \subseteq \top}\ \subseteq \top}{\top^{oo} \subseteq \top^o}\ ()^o \subseteq ()^o \quad \overline{\top^{oo} = \top}\ ^{oo} =}{\dfrac{\top \subseteq \top^o}{\{\top = \top^o\}^2} = \top}$$

$$\dfrac{\dfrac{}{B^o.A^o = \top.A^o \cap B^o} \subseteq id.\top \quad \overline{(B^o.A^o)^o = (B^o.A^o)^o}\ id \subseteq (=)}{(B^o.A^o)^o = (\top.A^o \cap B^o)^o} = \quad \overbrace{\top = \top^o}^{2} =}{\{(B^o.A^o)^o = (\top^o.A^o \cap B^o)^o\}^3}$$

$$\dfrac{\dfrac{\overbrace{(B^o.A^o)^o = (\top.A^o \cap B^o)^o}^{3}}{(B^o.A^o)^o = (\top^o.A^o \cap B^o)^o} = \quad \overline{(A.\top)^o = \top^o.A^o}\ .^o =}{\{(B^o.A^o)^o = ((A.\top)^o \cap B^o)^o\}^1}$$

$$\dfrac{\overbrace{(B^o.A^o)^o = ((A.\top)^o \cap B^o)^o}^{1} \quad \overline{(A.\top \cap B)^o = (A.\top)^o \cap B^o}\ \cap B =}{(B^o.A^o)^o = (A.\top \cap B)^{oo}} \quad \dfrac{}{(A.B)^o = B^o.A^o}\ .^o =}{\{(A.B)^{oo} = (A.\top \cap B)^{oo}\}^4}$$

$$\dfrac{\overbrace{(A.B)^{oo} = (A.\top \cap B)^{oo}}^{4} \quad \overline{(A.\top \cap B)^{oo} = A.\top \cap B}\ ^{oo} =}{(A.B)^{oo} = A.\top \cap B} \quad \dfrac{}{(A.B)^{oo} = A.B}\ ^{oo} =}{A.B = A.\top \cap B}$$

4.10 Resumo

A tabela a seguir resume o trabalho que cada tipo de prova produz em PVS.

Propriedade	Especificação	Nº de linhas *pointfree*	Nº de linhas sem *pointfree*
lei de Shunting	$f\ function \vdash (f.R \subseteq S \Leftrightarrow R \subseteq f^o.S)$	43	1
lei de Shunting	$f\ function \vdash (R \subseteq S.f \Leftrightarrow R.f^o \subseteq S)$	38	1
Continência de funções	$g\ function, f\ function \vdash g \subseteq f \Rightarrow g = f$	27	1
Simetria e Antissimetria é Correflexão	$\vdash A \cap A^o \subseteq id \wedge\ A^o = A \Leftrightarrow A \subseteq id$	14	7
Fecho de Correflexiva	$A \subseteq id \vdash R.A \subseteq S \Leftrightarrow R.A \subseteq S.A$	26	15
Fecho de Correflexiva	$A \subseteq id \vdash A.R \subseteq S \Leftrightarrow A.R \subseteq A.S$	23	15
Condição Lateral	$A \subseteq id, B \subseteq id \vdash A \subseteq B \Leftrightarrow A \subseteq \top.B$	23	10
Condição Lateral	$A \subseteq id, B \subseteq id \vdash A \subseteq B \Leftrightarrow A \subseteq B.\top$	24	10
Lei de De Morgan	$\vdash \overline{M} \cup \overline{N} = \overline{M \cap N}$	50	1
Lei de De Morgan	$\vdash \overline{M} \cap \overline{N} = \overline{M \cup N}$	8	1
Correção de Funções	$f\ function \vdash Ker(f.\phi_I) \subseteq \phi_O \Leftrightarrow f.\phi_I \subseteq \phi_O.f$	19	16
Distribuição de converso	$\vdash (A \cup B)^o = A^o \cup B^o$	60	1
Distribuição de converso	$\vdash (A \cap B)^o = A^o \cap B^o$	54	1

Tabela 4.1: Trabalho de prova

Na tabela nota-se que a versão por quantificadores produziu as menores provas

Capítulo 5

Conclusão

O cálculo relacional baseando na notação *pointfree* mostrou-se correto, pois provou-se em PVS que cada passo de prova não contém erros. Mas falta verificar que é possível demonstrar se todas as afirmações verdadeiras podem ser provadas com o cálculo, ou seja, se ela é completa. Outro ponto que falta é mostrar o grau de expressividade do cálculo em relação a notação por quantificadores.

Esperava-se que um cálculo baseado em teoria dos conjuntos produzisse demonstrações mais curtas que o cálculo do PVS, que é baseado em cálculo de Gentzen. Entretanto todas as provas produzidas no capítulo 4 contradizeram essa hipótese. Evidências de que esse cálculo provê árvores maiores é a divisão das provas feitas em subprovas para que elas coubessem na página. Um outro problema apresentado pelo cálculo foi a quantidade de regras existente para resolução. Como a quantidade de regras são maiores, demora-se mais para memorizar e, consequentemente, aprender.

No entanto, a notação *pointfree* deixa as fórmulas mais compactas, pois todas as especificações ficaram todas menos complexas. E quanto menor uma especificação mais difícil produzir erros com ela.

Como as especificações ficam mais compactas e as provas maiores é possível combinar as várias formas de se provar uma proposição. Pode-se especificar usando *pointfree*, mas expandir as definições em seus respectivos quantificadores e usar o cálculo de Gentzen. Assim, garante-se especificações e provas menos complexas. De fato, ferramentas como *Alloy* permite as especificações em estilo *pointfree* evitando erros em especificações por ser uma notação mais compacta. A ferramenta auxiliar para especificações é a taxinomia de relações, que definem padrões de tipos de relações.

Referências

[1] M. Ayala-Rincón and F.L.C. de Moura. *Applied Logic for Computer Scientists - computational deduction and formal proofs*. Universidade de Brasília, Campus Darcy Ribeiro, 2013. Em Elaboração. 7

[2] E. F. Codd. *A Relational Model of Data for Large Shared Data Banks*, volume 13. ACM, New York, NY, USA, June 1970. 4

[3] Secretaria de educação de Joinville. Matriz curricular. https://educacao.joinville.sc.gov.br/conteudo/31-Matriz+Curricular.html, 2011. 8

[4] K. Devlin. *The Math Gene*. Basic Books, Great Britain, Weidenfeld and Nicolson, 2000. 9, 15

[5] E. Bianchini e H. Paccolla. *Matemática volume 2*. Moderna, São Paulo, Brasil, 1990. 8

[6] S. Ferferman. *Tarski's influence on computer science*. Logic in Computer Science, 2005. LICS 2005. Proceedings. 20th Annual IEEE Symposium on, 2005, 2005. 4

[7] M. Huth and M. Ryan. *Logic In Computer Science*. Cambridge, 2004. 7, 26

[8] D. Jackson. *Software Abstractions Logic, Language, and Analysis*. The MIT Press, Cambridge, Massachussets London, England, 2007. 1

[9] R. C. Lyndon. *The Representation of Relational Algebras*. Ann. Math., 1950. 4, 22

[10] R. D. Maddux. *The Origin of Relation Algebras in the the Development and Axiomatization of Calculus of Relations*, volume 50. Studia Logica: An International Journal for Symbolic Logic, 1990. 4

[11] J. N. Oliveira. *Program Desing by Calculation*. High-Assurance Software Laboratory, Universidade do Minho,Campus de Gualtar – Braga – Portugal, 2000. Em Elaboração. 2, 9

[12] J. N. Oliveira. *Pointfree Foundations for (Generic) Lossless Decomposition*. High-Assurance Software Laboratory, Universidade do Minho,Campus de Gualtar – Braga – Portugal, 2007. Relatório de Pesquisa. 3, 4, 9

[13] J. N. Oliveira and M. A. Ferreira. *Alloy Meets the Algebra of Programming: A Case Study*, volume 39. IEEE Transactions on Software Engineering, 2013. 4

[14] J. N. Oliveira and V. C. Miraldo. Formulário de Álgebra Relacional. http://wiki.di.uminho.pt/twiki/pub/Education/MFES/Material/formulario.pdf, 2013. 4

[15] B. C. Pierce. *Types and Programming Languages*. MIT Press, Cambridge, MA, USA, 2002. 9

[16] Wiki Project. *The Haskell Programing Language*. http://www.haskell.org/haskellwiki, 2013. 2

[17] C. F. Brumfiel R. E. Eicholz, P G. O'Dafter and M. E. Shanks. *Elementary School Mathematics, Teachers' Edition to accompany*. Addison-Wesley Publishing Company, Inc, 1964. 8

[18] J. M. Rushby S. Owre, N. Shankar and D. W. J. Stringer-Calvert. *PVS Language Reference*. 2001. 3, 7

[19] J. M. Rushby S. Owre, N. Shankar and D. W. J. Stringer-Calvert. *PVS Prover Guide*. 2001. 3

[20] A. Tarski. *On The Calculus of Relations*, volume 6. The Journal of Symbolic Logic, 1941. 4, 21

[21] A. Tarski and F. B. Thompson. *Some general properties of cylindric algebras*. Bulletin of AMS, 1952. 4

[22] W. N. G. Thornton. *Binary Relations*. http://llama.freegeek.org/~wren/resources/cartography-of-math/BinaryRelations.pdf, 2013. 9

[23] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, New York, NY, USA, 1996. 9

[24] W.Wechler. *Universal Algebra for Computer Scientists*. Springer-Verglay, 1992. 9